# NetworkCanvas: Supporting Progressive Network Visualization Exploration via Adaptive Recommendations

Wenchao Li
The Hong Kong University of Science and Technology
Hong Kong SAR, China
wenc.li@outlook.com

Yuewen Gao
Nanjing University
Nanjing, China
yuewen_gao@smail.nju.edu.cn

Yu He
Nanjing University
Nanjing, China
yu_he@smail.nju.edu.cn

Cong Zhu
Huazhong University of Science and Technology
Wuhan, China
m202372013@hust.edu.cn

Ke Xu*
Nanjing University
Nanjing, China
xuke@nju.edu.cn

## Abstract

Network visualization has become essential for understanding complex relationships across domains, yet network complexity creates an overwhelming exploration space where users frequently miss critical patterns. Existing tools often require predetermined analysis goals and manual workflow construction, limiting accessibility for non-experts. We present NetworkCanvas, a progressive network visualization system that guides users through personalized exploration via adaptive recommendations. Our approach combines a learning mechanism that adapts to user feedback, an analytic state graph preserving exploration provenance with branching paths, and a context-aware feedback interpreter that suggests analytical continuations based on selection patterns. Controlled studies demonstrate that NetworkCanvas users identified more noteworthy observations, reported higher confidence, and exhibited more systematic exploration compared to a baseline without recommendations. These results demonstrate that recommendation-guided exploration improves outcomes over unguided manual analysis; however, because our baseline lacked recommendation functionality entirely, the specific contribution of adaptive personalization versus static guidance remains an open question. Qualitative findings suggest that recommendations reduce analysis paralysis and support systematic exploration.

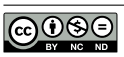
## CCS Concepts

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

## Keywords

Guided exploration, network visualization

*Corresponding author




*CHI '26, Barcelona, Spain*

ACM ISBN 979-8-4007-2278-3/2026/04
https://doi.org/10.1145/3772318.3791211

**ACM Reference Format:**
Wenchao Li, Yuewen Gao, Yu He, Cong Zhu, and Ke Xu. 2026. NetworkCanvas: Supporting Progressive Network Visualization Exploration via Adaptive Recommendations. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3772318.3791211

## 1 Introduction

Network visualization has become indispensable for understanding complex relationships across diverse domains, from social media interactions and biological pathways to cybersecurity threats and organizational structures [15, 23, 29]. The intricate nature of network topologies, coupled with diverse node and edge attributes, presents an overwhelming exploration space that often leads analysts to miss critical patterns or insights [5, 29, 61, 62].

Consider a cybersecurity analyst investigating threats in network traffic data containing thousands of IP addresses as nodes and communication patterns as weighted edges with types. They face extensive exploration choices: Should they start by examining node centrality to find command-and-control servers [18, 75]? Focus on flow patterns to detect anomalies [50]? Investigate community structures to identify botnet formations [64]? These challenges are particularly pronounced for non-experts who may lack the expertise to navigate such complexities effectively [5, 23, 61].

Current network visualization tools, such as Gephi [4] and Cytoscape [49], offer sophisticated analytical capabilities, including centrality algorithms, community detection, and temporal filtering. However, these tools require users to predetermine analysis goals and manually orchestrate operations into coherent workflows, limiting accessibility and user confidence [15, 38, 60, 62]. Moreover, the fundamental challenge extends beyond interface complexity to the absence of guidance that adapts to users' evolving understanding and exploration context [45, 55, 60]. While recommendation systems have shown promise in tabular data exploration [66–68, 76], they fall short addressing the unique challenges of network structures [15]. Provenance systems like CLUE [20] and VisTrails [7] capture analysis histories, and progressive visual analytics provide partial-result workflows [56, 60], yet these approaches rarely integrate adaptive recommendation and progressive guidance for network exploration [28, 55].

To address these challenges, we present *NetworkCanvas*, a progressive network visualization system supporting non-expert users through adaptive recommendations [39]. Our system actively learns users' analytical preferences and guides the user with personalized exploration workflows [8]. The key insight underlying NetworkCanvas is that effective network exploration can be modeled as an adaptive dialogue between analyst and system: each interaction refines the system's understanding of the user's goals and interests [32, 63]. Rather than requiring users to specify complete analytical workflows upfront, NetworkCanvas progressively suggests relevant analytical directions based on evolving exploration context [3, 42, 59]. Meanwhile, our unifies adaptive recommendation with provenance-aware branching and navigation, enabling access to exploration history and parallel exploration paths [9, 12].

NetworkCanvas introduces three synergistic technical modules addressing core challenges of adaptive recommendation. First, our *Question-Affinity Graph (QAG)* engine employs a learning mechanism to capture users' preferences across 10 analytical question categories and dynamically adjusts recommendation weights based on interaction feedback. Second, our *Analytic State Graph (ASG)* manager provides a provenance model that goes beyond simple history tracking [45]; it maintains exploration as a branching graph structure where users can pursue multiple analytical hypotheses in parallel, with each branch preserving its own context and recommendation state [20]. Third, our context-aware feedback interpreter bridges the semantic gap between user selections and analytical intent by analyzing both chosen recommendations and selected network entities to generate more relevant suggestions [19]. Validation through mixed-method evaluations demonstrates that participants using NetworkCanvas bookmarked 46% more observations they considered noteworthy and reported higher confidence than with a baseline tool without recommendations. These results establish the value of recommendation-based guidance for network exploration; however, our baseline design does not isolate whether benefits stem from adaptive personalization specifically or from providing structured guidance more generally. Qualitative analysis further suggests that recommendations supported systematic exploration, helped participants surface patterns they reported missing with manual exploration, and developed transferable analytical knowledge, providing suggestive evidence that quantitative differences reflect meaningful analytical progress rather than superficial enumeration [28, 56], though the specific contribution of adaptivity (vs. static guidance) awaits future factorial studies.

In summary, our contributions are as follows:

(1) A heuristic learning-based workflow-affinity model and context-aware multi-criteria recommendation framework for personalized network analysis.
(2) *NetworkCanvas*, a human-AI collaborative system that integrates adaptive guidance with provenance-aware branching to support non-expert users in network exploration.
(3) Empirical evidence from controlled studies that users with NetworkCanvas identify more noteworthy observations, report higher analytical confidence, and maintain systematic exploration behavior compared to baseline approaches, with qualitative findings providing suggestive evidence for adaptivity's role within the overall guidance framework.

## 2 Related Work

### 2.1 Progressive Visual Analytics

Progressive visual analytics (PVA) addresses computational delays in large-scale data analysis by enabling immediate user feedback and interaction with partial results [16, 56, 71]. Rather than following traditional analyze-then-visualize workflows, PVA transforms analysis into an iterative process where users inspect and steer computations as they unfold. Stolper et al. [56] established foundational PVA principles, demonstrating how analysts could interact with evolving patterns while maintaining computational efficiency. They highlighted two key requirements: algorithms must produce meaningful partial results, and visualizations must incorporate constantly refining results without overwhelming users.

Recent advances focus on computational steering and user guidance. Hogräfer et al. [24] introduced "steering-by-example," allowing users to prioritize data subspaces through relaxed queries from selected items. Fekete's ProgressiVis toolkit [14] provides comprehensive infrastructure for building PVA systems, while Ulmer et al. [60] systematically categorize progressive visualization properties, including uncertainty handling, steering capabilities, and visual stability. However, existing PVA approaches have significant limitations, particularly for network analysis, where exploration paths are less predictable than in tabular data. Most systems focus primarily on computational efficiency and algorithm steering [53, 59], providing limited attention to adaptive user guidance and learning from preferences [36]. Current progressive systems require users to manually direct computational resources or rely on predefined heuristics [43], creating barriers for non-expert users who need adaptive recommendation capabilities.

### 2.2 Visualization Recommendation and Exploration

Visualization recommendation systems aim to lower barriers to data exploration by automatically generating relevant visualizations [67]. Manual visualization specification requires substantial expertise and can overwhelm users with choices [35]. The field has evolved from rule-based approaches to sophisticated machine learning frameworks [13, 70]. Early systems established foundational approaches through rule-based engines. Voyager [67] introduced partial specification completion and faceted browsing, while Draco [37] formalized visualization design knowledge through constraint-based optimization. Recent extensions like Draco 2 [69] expanded support for multi-view visualizations.

Machine learning approaches have gained prominence [33, 65, 74]. Hu et al. [26] demonstrated that neural networks can predict visualization design choices, achieving 70–95% accuracy across five key design decisions. Recent work focuses on adaptive and context-aware systems: ShiftScope [47] introduced dual-agent reinforcement learning to adapt recommendations to users' evolving data focus, while AdaVis [72] provides adaptive and explainable recommendations considering user intent. Insight-centric approaches represent another promising direction [54]. Harris et al. [21] introduced SpotLight, which automatically discovers and ranks insights across 21 different types, bridging visualization design and analytical goals. Zhao et al. [73] developed techniques for exploring

implicit and explicit relations in faceted datasets, demonstrating how recommendation systems can reveal non-obvious data connections. Their approach highlights the potential for guided exploration to surface relationships that users might otherwise overlook, a principle we extend to network-specific analytical workflows.

Despite these advances, most recommendation systems operate independently of the exploration process and provide static suggestions without learning from ongoing user interactions [46]. While systems like Voyager [67] support iterative refinement, they lack continuous learning capabilities. Current systems cannot maintain exploration context or provide provenance-aware recommendations, preventing them from supporting complex analytical workflows where context and history are essential.

## 2.3 Interactive Network Exploration

Interactive network exploration faces unique challenges due to graph complexity, diverse attributes, and specialized analytical needs [6, 30, 62]. Research develops intuitive interfaces, guidance systems, and adaptive exploration support to help users navigate overwhelming analytical choices [44].

Foundational work established core principles for network visualization and interaction, including early efforts to balance systematic analysis with flexible exploration. Perer and Shneiderman [41] demonstrated this balance in their SocialAction system, which provided coordinated ranking views and attribute-based filtering to guide users through large social networks. Their work established that effective network exploration requires both structured analytical operations and flexible navigation accommodating diverse user strategies. While SocialAction employed expert-defined ranking criteria, NetworkCanvas extends this foundation by learning user-specific preferences to personalize guidance dynamically.

Shneiderman [52] later synthesized essential techniques, including filtering, clustering, and pathfinding for managing network complexity. Beck et al. [6] provided a comprehensive analysis of network visualization state-of-the-art. Complementing these foundational interaction techniques, recent work has examined how analysts compare structural differences across multiple networks. Fujiwara et al. [17] proposed interpretable contrastive network representations capturing both shared and distinctive structural features across networks. Their method enables analysts to identify meaningful variations in community organization, connectivity patterns, and motif-level structures, which are capabilities that could complement NetworkCanvas's single-network focus in future extensions. These works collectively highlight a persistent challenge: current tools require users to predetermine analysis goals and manually orchestrate operations into coherent workflows.

User-centered studies reveal significant barriers to effective exploration. AlKadi's investigation [2] identified eight distinct barriers, including technical challenges, conceptual understanding difficulties, and workflow management issues, revealing four distinct user types requiring different support strategies. Their work with the Vistorian platform revealed four distinct user types based on tool usage patterns. Each type requires different support strategies: demo users need onboarding, data strugglers need data preparation help, and explorers need advanced features.

Guidance and recommendation systems have emerged as critical research directions in interactive visual analytics. Crnovrsanin et al. [11] introduced collaborative filtering for network navigation, recommending potentially significant nodes based on user interaction history and similarity metrics. Zimmermann and Bruckner [77] developed multi-focus probe techniques for immersive environments, addressing context switching challenges. Data tours and semi-automatic exploration offer another promising direction. Systems like those developed by Li et al. [34] provide guided exploration paths through network data. These approaches combine automated analysis with user control. They offer structured navigation paths while preserving analytical freedom.

Yet existing network exploration tools have fundamental limitations. They predominantly focus on visualization and interaction techniques rather than adaptive guidance, require substantial domain expertise or analytical skills to navigate effectively [22], and do not learn from user behavior or adapt to evolving understanding. This creates significant barriers for non-expert users lacking specialized knowledge about network analysis methods. The gap between sophisticated visualization capabilities and user guidance remains a critical challenge in making network analysis accessible to broader audiences [2].

# 3 System Overview and Design Rationale

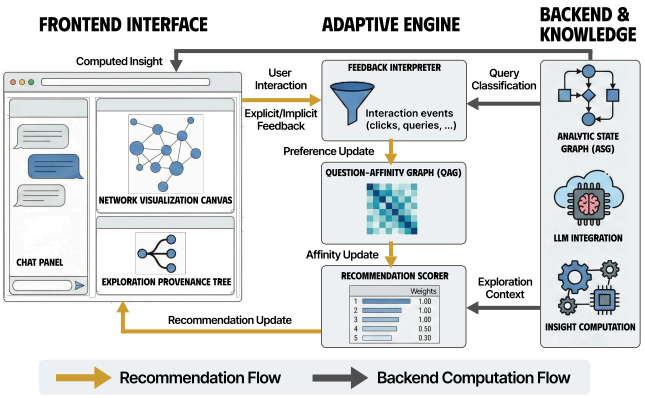



**Figure 1: Overview of the System Architecture. The pipeline is divided into three primary modules: (1) The Frontend Interface (left), arranged with the Chat Panel for queries on the left, the main Network Visualization Canvas in the center, and the Exploration Provenance Tree on the right; (2) The Adaptive Engine (middle), which processes implicit feedback via the Feedback Interpreter to update the Question-Affinity Graph (QAG) and Recommendation Scorer; and (3) The Backend & Knowledge module (right), housing the Analytic State Graph (ASG), LLM Integration and Insight Computation. Gold arrows denote the cyclical recommendation flow, while dark grey arrows indicate backend data transmission.**

Before detailing our technical approach, we provide a high-level overview of NetworkCanvas's architecture, the design rationale

that guided our decisions, and the user interface. This section establishes context for the subsequent usage scenario and technical details that motivate our implementation.

## 3.1 Design Rationale

Our design emerged from a synthesis of network exploration challenges identified in prior literature and formative observations with eight network analysis practitioners (four visualization researchers, two data scientists, two domain analysts) during approximately 12 hours of semi-structured interviews. These conversations revealed recurring themes that shaped our technical approach:

**DR1: Balancing Guidance and Control.** Prior work on mixed-initiative systems demonstrates that preserving user agency is critical for adoption [25], and AlKadi et al. [2] identified "feeling constrained" as a primary barrier to network tool adoption. Six of eight practitioners emphasized wanting to "stay in control" while appreciating guidance that surfaced overlooked options. This motivates our *adaptive dialogue* approach, where users can accept, modify, or ignore recommendations.

**DR2: Learning from Implicit Feedback.** Voyager [67] demonstrated that static recommendations fail to accommodate individual analytical styles, while ShiftScope [47] showed that user mental models evolve during exploration. Our interviews revealed substantial variation in exploration strategies (top-down vs. bottom-up), motivating our *Question-Affinity Graph* that learns preferences from which recommendations users accept versus ignore.

**DR3: Supporting Non-Linear Exploration.** CLUE [20] and GraphTrail [12] established that analysts require non-linear exploration beyond simple undo/redo. Our practitioners frequently pursued multiple hypotheses simultaneously, motivating our *Analytic State Graph* that maintains exploration as a branching structure.

**DR4: Context-Aware Recommendations.** Gotz and Zhou [19] showed that user interactions encode rich implicit signals about analytical intent. Selecting high-degree nodes should trigger different suggestions than selecting bridge nodes. This motivates our *Context-Aware Feedback Interpreter* that analyzes selection patterns to infer analytical goals.

**DR5: Managing Cognitive Load.** Cognitive load theory [57] and visualization complexity guidelines [51] support progressive revelation of functionality. Seven of eight practitioners reported feeling overwhelmed by initial choices, motivating our *progressive disclosure* mechanisms that reveal analytical possibilities based on demonstrated expertise.

**DR6: Cold-Start Problem.** The cold-start challenge is well-documented in recommender systems [48]. Our system addresses this through *heuristic initialization* based on common analytical workflows from network analysis literature.

## 3.2 Architecture Overview

NetworkCanvas implements a client-server architecture integrating adaptive recommendation with interactive network visualization (Figure 1). The backend hosts four synergistic modules forming a closed-loop adaptive system:

- **Question-Affinity Graph (QAG) Engine:** Models user preferences as a directed weighted graph across 10 analytical question categories, employing asymmetric heuristic updates to learn from implicit feedback while preventing filter bubbles.
- **Analytic State Graph (ASG) Manager:** Maintains exploration provenance as a branching directed acyclic graph, enabling parallel hypothesis investigation with full state traceability.
- **Multi-Criteria Recommendation Scorer:** Generates ranked suggestions by combining learned preferences, selection context, cross-branch synergy, and insight feasibility validation.
- **Context-Aware Feedback Interpreter:** Transforms low-level interactions (selections, timing, modifications) into structured learning signals that drive preference adaptation.

The ASG implements a state-centric model where each analytical step becomes a node with a defined lifecycle: recommendations first appear as PENDING, transition to ACCEPTED upon user selection, and reach COMPLETED after execution (with SKIPPED as an alternative terminal state for ignored suggestions). The system employs *stable materialization*: once recommendations surface as nodes, they persist in the graph structure rather than being destructively deleted, ensuring exploration decisions remain traceable and reversible. While the ASG supports parallel exploration branches, only one branch remains active at any time; the system maintains complete states for inactive branches, enabling users to resume any prior analytical thread.

A *Session Manager* coordinates workflow between user interactions, insight computation, and provenance updates. It maintains bidirectional synchronization between interface elements and ASG nodes, records timing data for each analytical step (informing the feedback interpreter's temporal analysis), and merges computational results with LLM-generated summaries into rich provenance records attached to completed nodes.

An LLM integration layer provides natural language query classification and result summarization through a two-layer generation structure: a factual layer restating computed metrics and an interpretation layer contextualizing results with appropriate hedging.

## 3.3 Interface Components

NetworkCanvas presents a unified workspace organized around five coordinated regions (Figure 2): **(A) Configuration Panel** controls visual encoding parameters and layout algorithms. **(B) Chat Panel** enables natural language interaction with adaptive recommendations displayed at the input area's bottom. **(C) Main Canvas** renders the network as an interactive node-link diagram supporting multiple selection modes (single, multi-select, lasso) with context-sensitive analytical overlays. **(D) Explanation Panel** displays computational results alongside LLM-generated summaries. **(E) Exploration Provenance Tree** visualizes the user's analytical journey as a branching tree structure, enabling navigation to previous states and parallel exploration branches.

The system surfaces adaptive recommendations through three coordinated touchpoints: context menus activated by right-clicking selected entities in the main visualization view, the persistent collapsible recommendation area within the Chat Panel or the Main Canvas displaying top-ranked suggestions, and inline prompts offering query completions as users type.

### 3.4 User Interaction Model

Users interact with NetworkCanvas through three complementary modalities: *Direct manipulation* on the canvas allows entity selection, view navigation, and contextual menu access. These interactions generate implicit feedback signals that inform recommendation adaptation. *Natural language queries* through the chat interface enable verbal expression of analytical goals, with the system classifying queries into analytical categories. *Recommendation acceptance* provides the primary mechanism for guided exploration: users can accept, modify, or ignore suggestions.

These components form a closed-loop system: user interactions generate feedback signals → feedback updates QAG preferences → updated preferences influence recommendations → recommendations shape subsequent interactions. This continuous adaptation transforms network exploration from static tool interaction into an evolving analytical dialogue.

## 4 Usage Scenario

Having introduced NetworkCanvas's architecture and design rationale (section 3), we illustrate how these components work together through a transportation analysis scenario.

Anna, a traffic engineer working for the Gold Coast city planning department, has been tasked with identifying bottlenecks and improvements to reduce congestion during peak hours. She loads the Gold Coast transportation network (4,807 intersections, 11,139 road segments with capacity and flow attributes).

*Initial Network Familiarization.* NetworkCanvas presents the road network with intersections as nodes and road segments as directed edges (Figure 2-C). The system generates starting recommendations: "Provide a global summary" and "Identify isolated road segments," reflecting the QAG's cold-start initialization that prioritizes Network Overview questions. Anna accepts the overview suggestion; the system computes that the network has 0.94% density with 2.3 average connections per intersection.

*Discovery of Capacity Patterns.* Intrigued by connectivity patterns, Anna uses lasso selection on high-degree nodes (8–12 connections) in the central district. Opening the chat panel (Figure 2-B), she sees the recommendations have shifted to node-focused analyses: "Find intersections with extreme capacity," "Find common routes," and "Analyze neighborhood accessibility." The Context-Aware Feedback Interpreter recognized her selection of hub nodes and adjusted QAG weights accordingly. She discovers the selected intersections handle 1,400–1,800 vehicles/hour, which far exceeds the 600 average.

*Focused Investigation of Problem Areas.* To understand how traffic flows through these critical intersections, Anna queries in the input box(Figure 2-B): "Find shortest paths between intersections," selecting high-capacity intersection pairs. The system computes and highlights the optimal routes in bright orange overlays on the network. The analysis (Figure 2-D) reveals that 73% of shortest paths pass through just 12% of intersections, indicating clear bottleneck patterns. The Exploration Provenance Tree (Figure 2-E) now shows her journey: from initial overview to identifying extreme nodes to path analysis, with each step preserved as an interactive node she can revisit.

*What-If Analysis and Route Optimization.* Motivated by these bottleneck findings, Anna notices the recommendation panel ((Figure 2-C)) in the bottom left now suggests "Simulate entity (intersection) removal impact," a what-if analysis that emerged as her exploration progressed toward optimization questions. This illustrates QAG engine's learning mechanism: her acceptance of bottleneck-related recommendations increased transition weights toward What-If Analysis. She accepts this recommendation, the system computes alternative routing scenarios, showing that removing the intersections would increase average travel time by 34% and create severe congestion in adjacent areas.

*Branching Analysis and Alternative Solutions.* Anna decides to explore a parallel analytical thread by creating a new branch in her exploration. Using the Exploration Provenance Tree (Figure 2-E), she forks from her initial capacity analysis to investigate toll road patterns. NetworkCanvas adapts recommendations for this link-focused branch: "Rank links by attribute (toll)" and "Visualize attribute (toll) histogram." The ASG maintains both analytical contexts, enabling her to pursue parallel hypotheses. This parallel exploration reveals that current toll rates create an imbalanced split between toll and free route usage, with potential for optimizing toll pricing to better distribute traffic load.

*Synthesis and Insights.* Navigating between branches, Anna discovers that combining bottleneck analysis with toll optimization suggests a coordinated intervention strategy. Through this guided exploration, she identified specific infrastructure improvements (five road expansions), policy interventions (dynamic toll pricing), and contingency planning scenarios (alternative routing during construction), transforming her initial broad investigation into actionable recommendations for urban traffic optimization.

## 5 Adaptive Recommendation Framework

This section details the technical mechanisms underlying NetworkCanvas's adaptive recommendation system. Effective recommendation requires learning what users find valuable. Instead of asking users to explicitly rate suggestions, our system observes natural interaction patterns, including which recommendations users accept, skip, or modify, and how quickly they respond, to infer analytical preferences from behavior alone. Learning from such implicit feedback is a well-studied problem in recommender systems [1, 10], where reinforcement learning (RL) offers one theoretically grounded formulation. However, full RL approaches introduce substantial complexity, including policy optimization, value-function estimation, and exploration-exploitation trade-offs, which can obscure system behavior and complicate debugging. Because user trust requires that recommendation logic remain transparent and predictable, we instead employ deterministic heuristic update rules that adjust transition probabilities based on observed behavior, prioritizing interpretability over theoretical optimality.

### 5.1 Question-Affinity Graph (QAG) Engine

The QAG engine models user analytical preferences as a directed weighted graph $G = (V, E, W)$ where vertices $V$ represent 10 analytical question categories, edges $E$ encode possible transitions between categories, and weights $W$ capture transition likelihoods.

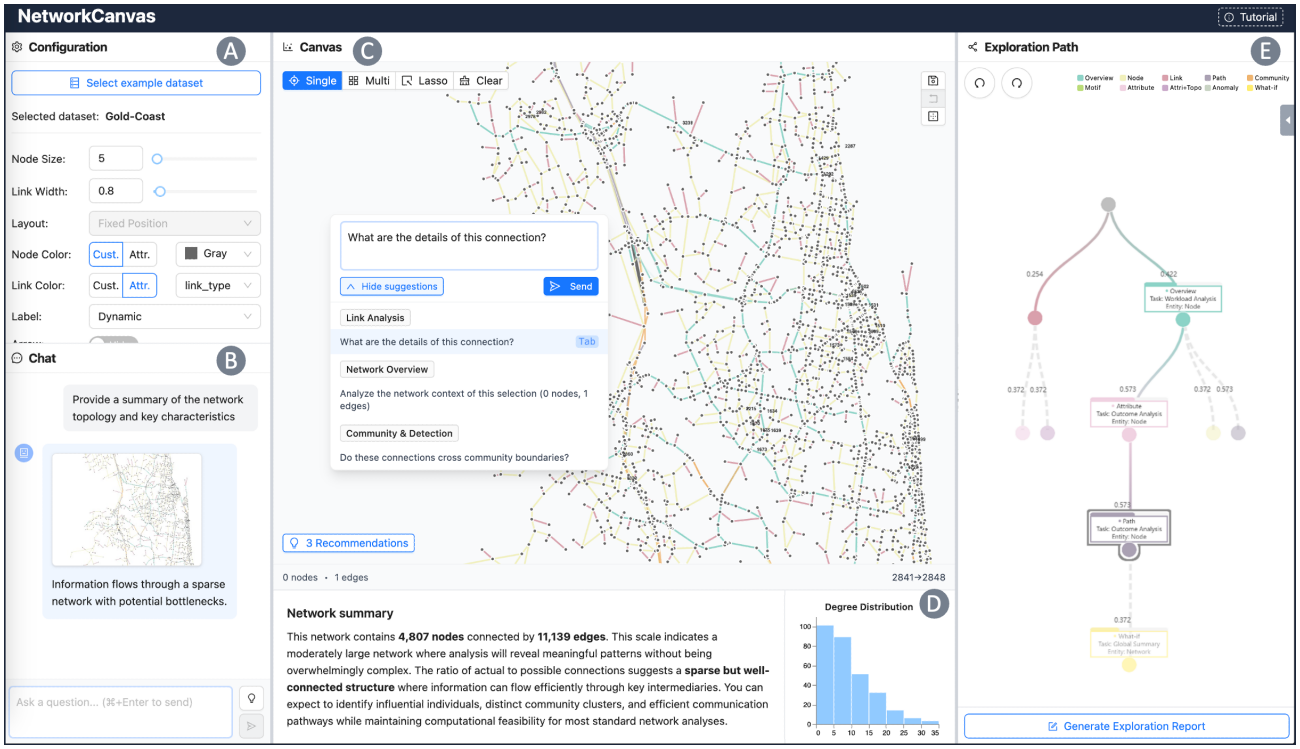



**Figure 2: NetworkCanvas Interface Overview. The system integrates five coordinated components: Configuration Panel (A) provides visual encoding controls to analytical insights; Chat Panell (B) enables natural language interaction with adaptive recommendations displayed at input bottom; Main Canvas (C) displays the network visualization with context-sensitive selection and recommendation overlays; Explanation Panel (D) shows LLM-generated explanations and computational results from completed insights; Exploration Provenance Tree (E) visualizes the branching analytical workflow history with interactive navigation between exploration states. The interface supports progressive disclosure, adapting complexity based on user expertise and exploration depth while maintaining complete analytical provenance.**

This graph structure enables learning-based preference adaptation (DR2) through minimal feedback while maintaining interpretable recommendation logic.

*5.1.1 Question Category Development.* We developed our 10 Question Categories through a literature-grounded synthesis of task taxonomies across information visualization, graph visualization, and network analysis. Our review incorporated canonical frameworks including Shneiderman's [51] task-by-data-type taxonomy and Lee et al.'s [31] task taxonomy for graph visualization, which together span fundamental perspectives on data types, graph structures, and analytic goals.

Building on this foundation, we extracted insight primitives from each taxonomy, identified recurring intent patterns, and reorganized them around core analytic entities in network science: nodes, links, groups, paths, motifs, communities, and global structure. The resulting draft categories underwent multiple rounds of internal refinement. Across the iterations, our research team (1) merged overlapping insight primitives (*e.g.*, "identify hubs" and "find influential nodes" → Node Analysis: Rank by Centrality); (2) split overly broad categories (*e.g.*, initial "Topology Analysis" subdivided into Path & Reachability, Community Detection, and Motif Discovery based on distinct algorithmic requirements); and (3) clarified boundary ambiguities through decision rules (*e.g.*, "analyzing flow along shortest paths" belongs to Path & Reachability rather than Link Analysis, as path computation precedes link examination). We additionally consulted with two domain practitioners (a visualization researcher and a network analyst) whose expertise helped validate the practical relevance of each category and reveal cases where boundaries or terminology required clarification. This iterative consolidation reduced an initial set of 12 candidate categories to the final 10 through card-sorting exercises. The process strictly adhered to the MECE (Mutually Exclusive, Collectively Exhaustive) principle, ensuring distinct boundaries while achieving comprehensive coverage.

The 10 analytical categories (Network Overview, Node Analysis, Link Analysis, Path & Reachability, Community Detection, Motif & Pattern Discovery, Attribute-based Exploration, Attribute-Topology Correlation, Outlier & Anomaly Detection, and What-If Analysis) provide comprehensive coverage of network analytical insights while remaining cognitively manageable. Each category maintains both self-loop weights (continuation probability) and transition weights to other categories, forming a $10 \times 10$ affinity matrix that

evolves through user interactions (see Appendix A and Appendix C for details).

*5.1.2 Heuristic Learning Mechanism.* The following update rules embody our heuristic adaptation approach. They are inspired by the RL principle of incrementally adjusting estimates based on the gap between expected and observed outcomes, but deliberately avoid the machinery of value functions and policy gradients in favor of intuitive multiplicative adjustments that are easy to reason about.

The learning mechanism employs asymmetric heuristic update formulas that differentiate between positive and negative feedback to prevent premature convergence while maintaining exploration diversity. When a user accepts a recommendation transitioning from category $A$ to category $B$, the system applies:

$$w_{A\to B}^{t+1} = w_{A\to B}^{t} + \eta \cdot (1 - w_{A\to B}^{t}) \tag{1}$$

where $\eta = 0.05$ provides gradual adaptation that resists noise while capturing genuine preference shifts. The multiplicative factor $(1 - w^t)$ ensures weights approach but never reach unity, maintaining recommendation diversity.

Conversely, when a user skips a suggested transition from $A$ to $C$, the system applies a soft penalty:

$$w_{A\to C}^{t+1} = w_{A\to C}^{t} - \lambda \cdot \eta \cdot w_{A\to C}^{t} \tag{2}$$

where $\lambda = 0.3$ moderates the penalty strength. This proportional reduction preserves non-zero weights, ensuring the system can still suggest previously skipped transitions if context strongly indicates their relevance, preventing the "filter bubble" effect common in recommendation systems.

*Design Rationale.* We employ asymmetric updates because negative signals in exploratory contexts are inherently ambiguous: skipping may indicate either disinterest or temporary pursuit of a different direction. The stronger positive/weaker negative design reflects this uncertainty asymmetry. The diminishing returns formula (Equation 1) prevents premature convergence: when $w^t = 0.9$, only 10% of the learning rate contributes to updates, compared to 50% when $w^t = 0.5$.

*Parameter Justification.* We determined $\eta = 0.05$ through pilot testing during system development, finding it provided gradual adaptation within a typical exploration session (10–15 interactions). Values below 0.03 caused insufficient personalization, resulting in recommendations that remained generic even after extended use. Values above 0.08 caused instability, with weights oscillating based on short-term interaction patterns rather than stable preferences. The penalty moderator $\lambda = 0.3$ balances responsiveness with stability, as lower values caused recommendations to persist too long after rejection, while higher values suppressed useful directions after only 2–3 skips.

*Cold-Start Initialization.* To address the cold-start problem (DR6), the QAG initializes with empirically derived heuristic weights based on common analytical workflows from network analysis literature. For instance, Network Overview → Node Analysis receives an initial weight of 0.6, reflecting a natural global-to-local progression, while Node Analysis → Path & Reachability starts at 0.75, capturing the common pattern of exploring connectivity after identifying important nodes. These initialization values provide immediate utility while the system personalizes through interaction feedback (see Appendix C).

## 5.2 Multi-Criteria Recommendation Scorer

The scoring engine generates contextually relevant recommendations (DR4) through a weighted combination of four factors:

$$\text{Score}(r) = \alpha\cdot\text{Affinity}(r) + \beta\cdot\text{DOI}(r) + \gamma\cdot\text{Synergy}(r) + \delta\cdot\text{Relevance}(r) \tag{3}$$

where the weights $\alpha = 0.4$, $\beta = 0.25$, $\gamma = 0.2$, and $\delta = 0.15$ were determined through iterative refinement during pilot testing across 2 development cycles. The dominant weight on QAG affinity ($\alpha = 0.4$) ensures that learned preferences drive recommendations while remaining below 0.5 to prevent the system from ignoring contextual signals. DOI receives the second-highest weight ($\beta = 0.25$) because immediate selection context proved highly predictive of relevant next steps in our observations. Synergy ($\gamma = 0.2$) and Insight Relevance ($\delta = 0.15$) serve as secondary factors that promote exploration breadth and prevent impossible suggestions, respectively. We acknowledge these weights represent heuristic choices; systematic sensitivity analysis across diverse user populations remains an opportunity for future work.

The **Affinity component** leverages QAG weights for personalization. For recommendation $r$ suggesting transition from $c_{\text{current}}$ to $c_{\text{target}}$, affinity equals $w_{c_{\text{current}}\to c_{\text{target}}}$.

The **Degree-of-Interest (DOI) component** measures alignment with current selection context through entity-type compatibility (nodes/edges/groups), cardinality appropriateness (single entity → detail analysis; sets → statistical approaches), and structural role relevance (hubs favor centrality analysis; bridges favor path exploration) [58].

The **Cross-Branch Synergy component** leverages the Analytic State Graph to identify complementary analyses across parallel exploration branches (DR3). The system maintains a insight complementarity matrix and provides novelty bonuses (+0.3) for unexplored insight types.

The **Insight Relevance component** ensures recommendations are executable given dataset characteristics, validating preconditions (two nodes for path analysis) and computational feasibility. Implementation details appear in Appendix B.

## 5.3 Context-Aware Feedback Interpreter

The feedback interpreter bridges the semantic gap between low-level interactions and high-level analytical intent, enabling learning from natural interactions without explicit feedback mechanisms (DR2).

The interpretation pipeline processes three dimensions in parallel. **Selection analysis** examines entity types (indicating analytical focus), cardinality (suggesting analytical approach), and structural properties (revealing interest patterns). Repeatedly selecting high-degree nodes signals interest in hub analysis, triggering increased weights for centrality-related categories.

**Temporal analysis** examines interaction timing: rapid acceptance indicates good preference alignment, while hesitation (hover $>$ 2 seconds) suggests marginal relevance. A sliding window of recent interactions detects preference shifts without abandoning learned preferences.

**Modification analysis** examines how users alter recommendations: accepting but immediately changing encodings suggests correct direction but suboptimal presentation; accepting and extending indicates high relevance.

The interpreter generates differentiated learning signals: direct acceptance produces standard positive updates ($\eta = 0.05$), enthusiastic acceptance triggers boosted learning ($\eta = 0.08$), and modified acceptance generates a reduced signal ($\eta = 0.03$). Similarly, quick skips produce standard penalties, while considered skips (hover > 2 seconds) generate reduced penalties ($\lambda_{\text{reduced}} = 0.15$).

## 5.4 Integration and Workflow

The three components operate in a synchronized pipeline balancing exploration (discovering preferences) and exploitation (leveraging learned preferences) to minimize cognitive load while preventing premature convergence (DR5).

When a user performs an action, the feedback interpreter analyzes interaction context and generates structured signals that simultaneously trigger QAG weight updates and provide context to the scoring engine. The scoring engine then generates candidates by identifying feasible analytical directions from the current state, computing multi-criteria scores for each candidate, and selecting the top-$k$ suggestions (typically $k = 3$), balancing diversity with relevance. Progressive disclosure (DR5) ensures initial interactions receive 3–4 fundamental suggestions, while demonstrated expertise triggers exposure to advanced options.

This adaptive framework enables NetworkCanvas to provide personalized guidance that evolves with user expertise, supports diverse analytical styles, and maintains a balance between helpful suggestions and user autonomy, thereby transforming network exploration from an overwhelming possibility space into a guided yet flexible analytical dialogue.

# 6 System Implementation

NetworkCanvas is implemented as a web-based system combining a React/TypeScript frontend with a provider-agnostic LLM backend, architected as a Python microservice. Building upon the architectural components introduced in section 3 and the adaptive recommendation framework detailed in section 5, this section describes the implementation specifics that realize these designs.

## 6.1 LLM Integration Module

The LLM integration layer provides natural language explanations and contextual analysis support through specialized services within the interactive exploration workflow. Our system employs large language models for two specific, well-scoped functions:

- **Query Classification**: Routing natural language questions to one of ten predefined analytical categories.
- **Result Summarization**: Generating textual explanations of computational outputs produced by deterministic algorithms.

Critically, LLMs do *not* perform graph computations, modify network data, or directly determine which entities are analytically significant. All quantitative analyses, including centrality calculations, community detection, and path finding, are executed by validated algorithmic implementations.

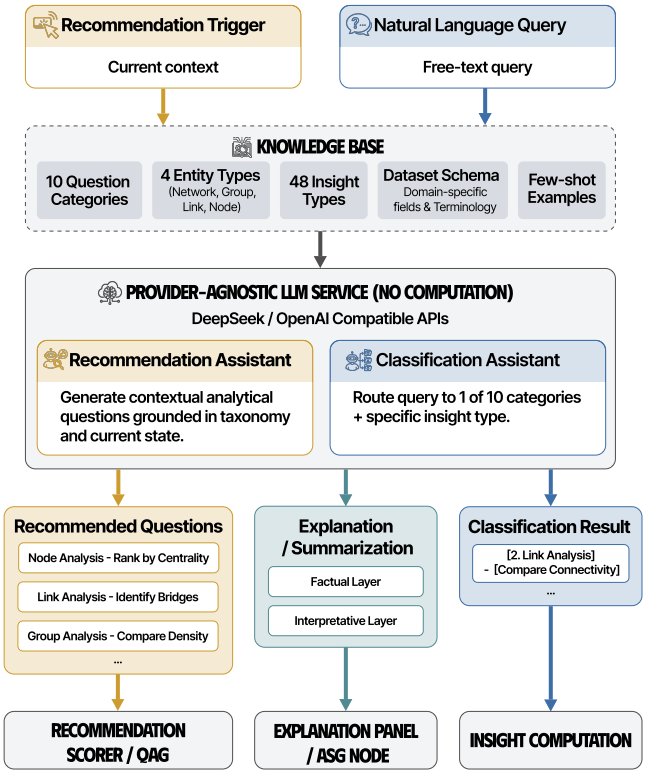



**Figure 3: LLM integration in NetworkCanvas. Two entry points, a recommendation trigger based on the current analytic context (left) and a free-text natural-language query (right), feed into a provider-agnostic LLM service that draws on a shared knowledge base comprising the question taxonomy, entity and insight types, the dataset schema, and few-shot examples. The LLM service uses this knowledge to (1) generate context-aware recommended analytical questions, (2) classify free-text queries into a question category and specific insight type for routing to computation, and (3) produce two-layer (factual + interpretative) summaries that accompany computed results in the interface. Critically, all quantitative analyses are performed by validated algorithms; LLMs provide only classification, natural language explanation, and summarization.**

The *LLM Service* implements a provider-agnostic design supporting multiple inference endpoints (DeepSeek, OpenAI-compatible APIs) through a unified interface with streaming response processing. The *Question Classification Service* routes natural language queries through a two-stage pipeline: a lightweight classifier identifies high-level intent, which then maps to specific analytical insights. Upon insight completion, the Session Manager constructs structured context objects and requests LLM summarization, which attaches to ASG nodes as persistent annotations.

*LLM Reliability Considerations.* We implement several mechanisms to ensure LLM outputs remain reliable and verifiable:

All query classifications are restricted to the 10 predefined analytical categories and their associated insight types (detailed in Appendix D). The system does not permit free-form LLM-generated analytical actions; every executable operation maps to a predefined insight type with validated implementation. Queries below the confidence threshold trigger clarifying prompts rather than arbitrary execution.

**Computation-Grounded Explanations.** Generated explanations are always grounded in numeric results produced by the Insight Computation layer. LLMs synthesize deterministic computational outputs with contextual interpretation, but cannot fabricate metrics or analysis results. Explanations explicitly distinguish computed facts ("The network contains 4,807 nodes with density 0.94%") from interpretive statements ("This suggests a well-connected urban grid") using consistent linguistic markers. Users can expand explanation panels to view underlying calculation methods and raw computation outputs.

**Limitations.** Our current implementation does not provide fine-grained uncertainty scores to end users, though confidence values are logged for system monitoring. LLM explanations may occasionally be incomplete; NetworkCanvas mitigates this by ensuring all analytic outputs can be verified against the visualization and underlying metrics. The reliability of specialized domains may require domain-specific prompt adaptation. Future work should explore uncertainty techniques [27] and expert-in-the-loop validation mechanisms for high-stakes analytical contexts.

**Cross-Domain Reliability Factors.** LLM reliability in NetworkCanvas varies across application contexts along several dimensions:

- **Terminology mapping:** Classification accuracy depends on whether domain terms map naturally to our 10 analytical categories. Transportation terminology ("intersection," "route") is well-represented in LLM training data, while specialized domains (*e.g.*, molecular biology: "protein complex," "metabolic pathway") may require explicit prompt-level mappings.
- **Category coverage:** Our question categories derive from general network analysis literature. Domains with specialized analytical patterns (*e.g.*, cascade analysis in social networks, pathway enrichment in biological networks) may require category extensions or domain-specific sub-taxonomies within the QAG structure.
- **Explanation grounding:** LLM-generated explanations contextualize computational results using domain knowledge. Quality depends on LLM familiarity with domain interpretation conventions; specialized domains may produce less reliable interpretations requiring expert verification.

## 6.2 Visualization Canvas

The Visualization Canvas implements insight-driven visualization templates that automatically adapt to analytical results. Nodes and edges are arranged according to user-selected layout methods (force-directed, circular, or fixed geographic layout when coordinates are available). Edge colors represent different attributes with custom encoding support.

The canvas provides rich guided interactions for progressive exploration. Users select from multiple modes (single-element, multi-select, lasso) in the upper-left corner. For selected entities (highlighted with a larger size), right-clicking opens a contextual chat box for data querying (shown in Figure 2-C). Based on selected data and query history, the system actively recommends further exploration suggestions categorized by analytical direction (*e.g.*, link analysis, network overview, community detection). Users can adopt suggestions directly or edit them to form new queries. For direct recommendations, users can also click the recommendation button at the lower left to facilitate exploration. Once submitted, analysis results are highlighted and centered in the node-link diagram to confirm outcomes.

## 6.3 Exploration Provenance Tree Interface

The Exploration Provenance Tree provides interactive visualization of analytical workflow history through a hierarchical node-link diagram reflecting ASG structure (Figure 4-A). Building on the ASG lifecycle model introduced in section 3, each branch encodes a detailed exploration trace where nodes represent specific graph questions transitioning through PENDING, ACCEPTED, SKIPPED, and COMPLETED states, while links capture transitions between question categories.

*Node Design.* The color of each node encodes the classified question categories. As shown in Figure 4-c, the current exploration node is visually highlighted, with a probability bar indicating generation likelihood and a compact label showing question category, insight type, and entity type. When hovering a node, a two-sided card pops out, offering exploration history information: the front provides the exploration snapshot, and the back reveals the corresponding analytical details.

*State Encodings.* Visual encodings convey ASG lifecycle states: PENDING nodes use dashed borders and probability-based opacity; ACCEPTED nodes feature solid borders with execution spinners; COMPLETED nodes appear as filled cards with metric summaries and LLM insights; SKIPPED nodes are de-emphasized through reduced opacity and dashed links. Recommendation probabilities shown at edge endpoints make inference processes transparent. Consistent color encoding distinguishes the 10 question categories.

*Hierarchical Folding Strategy.* As exploration deepens, we adopt a hierarchical folding strategy that progressively compresses irrelevant nodes while preserving essential context:

- **Level assignment**: Nodes on the main exploration path and within three hops of the current node retain full labels. Branches more than seven hops away fold into low-level nodes. All remaining nodes are retained as mid-level. Marked nodes are always preserved with full labels.
- **Visual encoding**: High-level nodes (Figure 4-c) show full labels; mid-level nodes (Figure 4-b) show condensed information with tooltip details; low-level nodes (Figure 4-a) become compact markers with reduced spacing.

*Design Alternatives.* As shown in Figure 5, we explored several alternatives for low-level node representations: Design (A) compressed tree branch preserving full edge structure, but is unsuitable

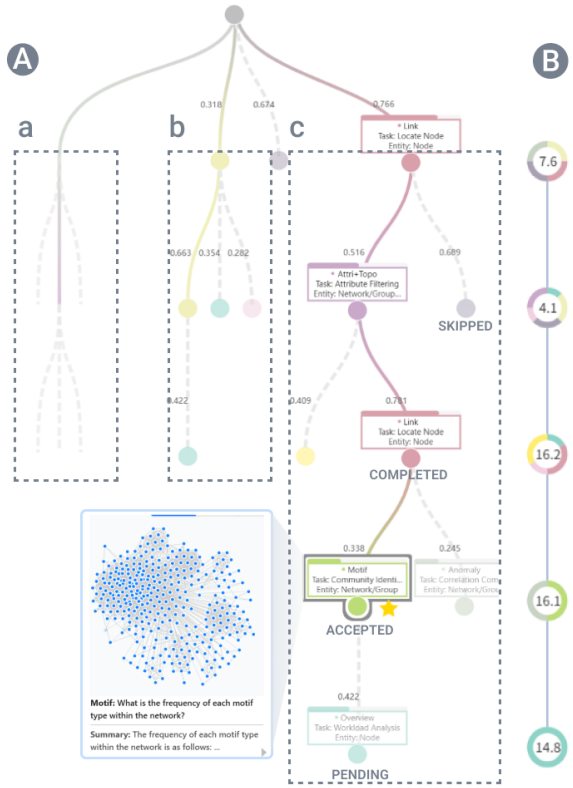



**Figure 4: Exploration Provenance Tree Overview. The main provenance tree (A) records the user's graph exploration trajectory. Each node represents an analytical insight (color-coded by question category), with edges showing transitions between insights. Node labels display the question category, insight type, and target entity. Probability values on edges and a probability bar beneath indicate recommendation confidence at generation time. Dashed edges denote skipped recommendations. It supports three levels of branch folding to address scalability: low-level (a), which preserves only the minimal branch topology with reduced spacing, mid-level (b), which shows unlabeled node markers with details accessible via tooltips, and full-detail (c). In the full-detail view, nodes encode four system states (*PENDING*, *COMPLETED*, *SKIPPED*, *ACCEPTED*). The currently active node is highlighted with a visual outline. Interaction logs (B) complement the tree view with temporal statistics. Donut-shaped markers encode time spent at each hierarchical depth level (inner ring) and the distribution of question types encountered (outer ring), enabling users to reflect on exploration patterns and identify analytical tendencies.**

for large graphs due to space requirements; Design (B) rectangular treemap adapting node width to root constraints, causing nodes to collapse into unreadable slivers as leaf count grows; Design (C) circular treemap variant, which is also space-inefficient at scale; Design (D) minimal spacing design preserving only branch structure with reduced inter-node spacing. We selected Design (D) as

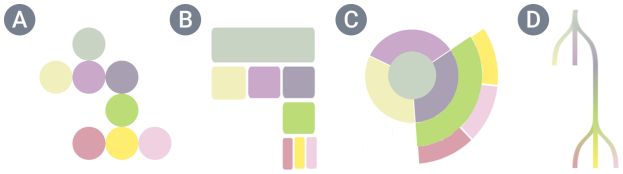



**Figure 5: Design alternatives for low-level node representations in the Exploration Provenance Tree. (A) Compressed tree branch preserving full edge structure. (B) Rectangular treemap-inspired design with adaptive node widths. (C) Circular treemap variant. (D) Minimal spacing design preserving branch structure with reduced inter-node spacing, which is the selected approach for its balance of scalability, space efficiency, and preservation of row height consistency.**

it maintains original node row heights after collapsing, providing stronger horizontal visual cues for tracking exploration depth.

*Interaction Log.* A collapsible side panel (Figure 4-B) complements the provenance tree with donut-shaped markers encoding time spent at each hierarchical level and distribution of question types encountered, helping users reflect on exploration strategies and uncover analytical tendencies.

## 6.4 Insight Computation Layer

NetworkCanvas operates in a closed loop of *insight recommendation* → *insight computation* → *insight interpretation*, recorded as a persistent exploration tree $T = (V, E, r)$. Each node $v \in V$ is a concrete insight instance with computed result, visualization, and narrative. The system maintains an active cursor $c_t \in V$; *backtracking* resets the cursor without altering $T$, while *branching* adds new children from any ancestor node.

*Insight Manager.* The Insight Manager bridges user goals with computational analysis, maintaining a registry mapping *RecommendedQuestion* types to executable insight classes with associated parameters and preconditions. Each insight class encapsulates domain-specific algorithms, such as PageRank for influence analysis, Louvain for community detection, and Dijkstra for shortest paths, along with validation logic ensuring computational feasibility. The manager aggregates heterogeneous outputs into standardized result schemas containing identified entities, computed metrics, and semantic annotations.

*Insight Getter.* The Insight Getter executes analytical tasks asynchronously, implementing streaming protocols for progressive result delivery where algorithmic structure permits. For iterative algorithms like PageRank, it streams intermediate convergence states; for community detection, it provides hierarchical clustering at multiple resolutions. The getter normalizes diverse outputs into a unified schema specifying affected entities, quantitative measures, and categorical classifications.

## 6.5 Data Flow and Runtime

As shown in Figure 6, the system implements an event-driven pipeline ensuring responsive interaction despite computational

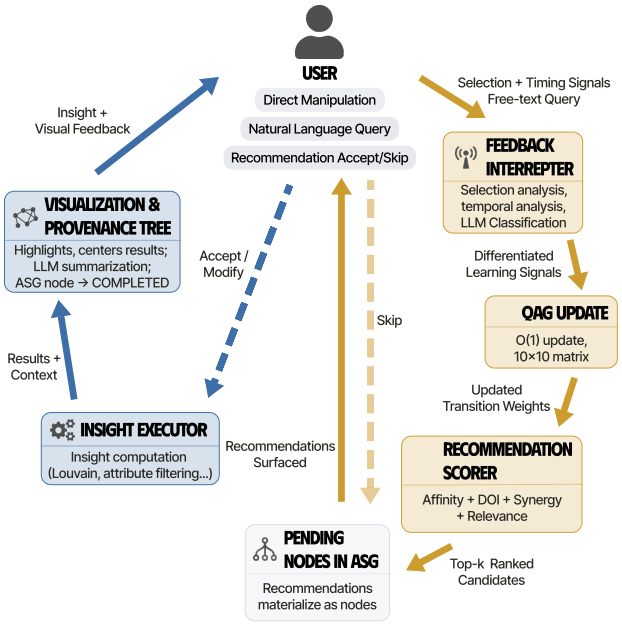



**Figure 6: Event-driven data flow of the proposed framework. User interactions (direct manipulation, natural language queries, recommendation responses) are processed by the Context-Aware Feedback Interpreter, which analyzes selection patterns, timing, and modification behavior to generate differentiated learning signals. These signals update the QAG in O(1) time, driving the Multi-Criteria Recommendation Scorer to produce top-k candidates that materialize as PENDING nodes in the ASG. Upon user acceptance, the Insight Executor computes results via deterministic algorithms, simultaneously updating the visualization canvas and triggering asynchronous LLM summarization. Completed insights persist in the ASG and appear in the Exploration Provenance Tree, closing the adaptive loop. Gold arrows trace the user-initiated path; blue arrows trace the computation-and-display path; dashed arrows indicate decision-point flows.**

complexity. User actions, including direct manipulation, natural language queries, and recommendation responses, generate selection, timing, and free-text signals that the Feedback Interpreter analyzes alongside LLM-based query classification. The interpreter emits differentiated learning signals that update the QAG preference weights and propagate to the Recommendation Scorer. Scored candidates materialize as PENDING nodes in the ASG and are surfaced to the user; skipped recommendations loop back while accepted ones trigger the Insight Executor. Computed results update the Visualization Canvas and Exploration Provenance Tree, with LLM summarization attached to the completed ASG node, closing the adaptive loop. The complete exploration cycle: User action → Feedback Interpreter → Differentiated Learning Signals → QAG update → Recommendation Scorer → Ranked candidates (PENDING nodes) → User accept/modify or skip → Insight Executor → Visualization Canvas refresh + LLM summary → Exploration Provenance Tree (ASG COMPLETED + Visual Feedback).

## 6.6 System Performance

We evaluated system responsiveness across network scales using a standard development workstation (Apple M1 Max, 32GB RAM, Chrome 141) to characterize performance boundaries relevant to our target user population. Table 1 summarizes key latency measurements across representative network sizes.

**Table 1: System performance metrics across network scales**

| | Network Size (nodes) | | | |
|---|---|---|---|---|
| **Operation** | **500** | **2k** | **5k** | **10k** |
| Initial rendering | 180ms | 420ms | 1.1s | 2.8s |
| QAG weight update | 2ms | 2ms | 2ms | 2ms |
| Recommendation generation | 45ms | 85ms | 180ms | 340ms |
| Selection feedback processing | 12ms | 18ms | 35ms | 72ms |
| Shortest path computation | 8ms | 45ms | 210ms | 890ms |
| Community detection (Louvain) | 120ms | 380ms | 1.4s | 4.2s |
| LLM explanation generation | 2.1s | 2.1s | 2.2s | 2.2s |

*Recommendation Latency.* The recommendation generation pipeline (comprising context analysis, QAG lookup, multi-criteria scoring, and candidate ranking) remains under the 200ms threshold for perceived instantaneity [40] up to approximately 3,000 nodes. For larger networks, recommendations are computed asynchronously with subtle loading indicators, maintaining interface responsiveness while completing background computation.

*Learning Updates.* The QAG update operation exhibits O(1) time complexity (fixed $10 \times 10$ matrix operations), contributing negligible latency regardless of network size. This design ensures that the adaptive learning mechanism never becomes a performance bottleneck.

*Insight Computation.* Analytical operations vary substantially in computational complexity. Lightweight operations (attribute filtering, node lookup) complete within 50ms for all tested network sizes. Graph algorithms (shortest path, centrality computation) scale with network structure, with community detection representing the most expensive operation. For networks exceeding 5,000 nodes, computationally intensive operations employ progressive result streaming, displaying partial results while computation continues.

*LLM Integration.* Natural language explanation generation averages 2.1 seconds latency using the DeepSeek API, with minimal variation across network sizes since prompt complexity depends on result summary length rather than network scale. This operation executes asynchronously and does not block user interaction; explanations appear in the interface upon completion without interrupting ongoing exploration.

*Scalability Boundaries.* Current implementation supports responsive interaction for networks up to approximately 10,000 nodes. Beyond this threshold, initial rendering exceeds 3 seconds, and some graph algorithms become prohibitively slow for interactive use. Supporting larger networks would require WebGL-based rendering, view-dependent level-of-detail computation, and server-side algorithm execution, which are engineering investments beyond our current research prototype scope but technically feasible for production deployment.

# 7 Evaluation

We conducted two complementary controlled studies to evaluate NetworkCanvas's effectiveness in supporting network exploration and analysis. Our evaluation addresses three primary research questions: (1) Does NetworkCanvas improve insight discovery compared to traditional network visualization approaches? (2) How effectively does the adaptive recommendation system support task completion in domain-specific contexts? (3) What are users' subjective experiences and preferences when using the system?

## 7.1 Study 1: Comparative Analysis of Network Exploration

*7.1.1 Methodology.* We designed a within-subject controlled study to compare NetworkCanvas against a baseline network visualization tool for exploratory analysis of data lineage networks. We recruited 14 participants (7 female, 7 male) from graduate programs and industry roles in data science and computer science (age: M = 28.4, SD = 3.2). Participants possessed basic network analysis understanding while having no prior NetworkCanvas experience.

We employed a within-subject factorial design with two factors: Visualization System (NetworkCanvas vs. Baseline) and Dataset Complexity (Simple vs. Medium), counterbalanced using a Latin Square design. Participants explored two real-world data lineage graphs: a simple network (464 nodes, 467 links) and a medium-complexity network (2,024 nodes, 2,032 links) containing three node types (Data Field, Data Job, Data Table) with parent-child relationships and data flow volume attributes.

**Baseline Selection Rationale.** The baseline condition replicated standard network visualization functionality similar to Gephi [4], including node-link visualization, manual exploration tools (pan, zoom, selection), layout algorithms (force-directed, circular), and basic network metrics (degree distribution, density statistics), but without recommendation features. Our baseline selection reflects two methodological considerations. *Confound control:* We isolated the recommendation paradigm by holding visualization affordances constant. Comparing against external tools such as Gephi [4] or Cytoscape [49] would introduce confounds from divergent interaction models, layout algorithms, and learning curves, making it difficult to attribute observed differences to recommendation functionality specifically. *Ecological validity:* Our target population comprises non-experts who typically explore networks through manual pan-zoom-select operations without structured analytical guidance. The baseline represents current practice for this population, establishing whether recommendation-based workflows provide meaningful benefits over unguided exploration. We acknowledge an important limitation: this design does not distinguish whether benefits stem from *adaptive* personalization or from providing *any* structured guidance. subsubsection 8.2.3 discusses this boundary and outlines planned ablation studies.

After standardized 15-minute training sessions for each tool, participants completed 15-minute exploration sessions per condition, documenting observations using bookmarking features with descriptive annotations. Post-session activities included insight presentations, questionnaires, and semi-structured interviews.

*7.1.2 Results.* **Observation Discovery Performance.** NetworkCanvas users bookmarked significantly more observations they considered noteworthy compared to the baseline condition: an average of 12.3 observations (SD = 3.1) versus 8.4 (SD = 2.7), representing a 46% increase ($t(13) = 4.82$, $p < 0.001$, $d = 1.35$). This effect was particularly pronounced in the medium complexity condition (NetworkCanvas: M = 14.1, SD = 3.4; Baseline: M = 8.9, SD = 3.0; $t(13) = 5.23$, $p < 0.001$). We emphasize that observation quantity does not directly equate to analysis quality; users might bookmark fewer but more meaningful findings, or more findings of lower individual significance.

**Observation Categorization.** Two authors independently coded each bookmark into structural (network topology), functional (data flow patterns), and anomaly categories (Cohen's $\kappa = 0.78$). NetworkCanvas users identified 43% more structural observations (M = 4.7 vs. 3.3, $t(13) = 3.21$, $p = 0.007$) and 52% more functional patterns (M = 5.1 vs. 3.4, $t(13) = 4.15$, $p = 0.001$). A post-hoc analysis categorizing bookmarks by analytical depth showed: Surface-level (descriptive statistics): NetworkCanvas 58%, Baseline 41%; Intermediate (pattern recognition): NetworkCanvas 49%, Baseline 32%; Deep (causal hypotheses): NetworkCanvas 33%, Baseline 17%. Notably, 8 participants (57%) identified critical dependency chains using NetworkCanvas that they failed to recognize with the baseline tool.

**Subjective Experience Ratings.** Table 2 summarizes subjective rating comparisons. NetworkCanvas received significantly higher ratings across all measured dimensions, with large effect sizes ($d > 1.3$). All 14 participants expressed preference for NetworkCanvas for future data lineage analysis.

**Table 2: Study 1 subjective ratings (7-point Likert scales)**

| Measure | NetworkCanvas | Baseline | t(13) | d |
|---|---|---|---|---|
| Ease of use | 6.1 (0.8) | 4.7 (1.1) | 4.92*** | 1.54 |
| Confidence in findings | 5.8 (0.9) | 4.3 (1.2) | 4.33** | 1.36 |
| Willingness to explore | 6.2 (0.7) | 4.1 (1.3) | 6.18*** | 1.94 |

*Note.* Values are Mean (SD). $^{**}p < .01$, $^{***}p < .001$. N=14, within-subject.

*7.1.3 Qualitative Findings.* Interview analysis using thematic coding revealed four key themes explaining NetworkCanvas's advantages and one theme highlighting concerns (inter-rater reliability: $\kappa = 0.78$).

**Guided Discovery Process (12/14 participants).** Participants described how recommendations provided structured pathways through complex analytical spaces. P7: *"The suggestions helped me see connections I wouldn't have thought to look for. It's like having an expert guide pointing out what's interesting."* P12: *"Every time I selected something, it gave me ideas for what to do next."*

**Reduced Analysis Paralysis (9/14 participants).** P3: *"With [baseline], I felt overwhelmed by all the nodes and didn't know where to start. NetworkCanvas gave me starting points and next steps."* P9: *"The first few minutes with baseline were just wandering. With NetworkCanvas, I felt like I had a plan from the beginning."*

**Enhanced Pattern Recognition (8/14 participants).** P11: *"The system would highlight clusters I didn't even notice, then suggest ways to examine them more closely."* P6: *"There was this one suggestion*

*about looking at nodes with high betweenness. When I tried it, it showed me exactly the bottleneck tables I was looking for."*

**Concerns About Over-Reliance and Bias (5/14 participants).** Despite positive experiences, some participants expressed concerns. P5: *"I wonder if I'm being led down a particular path and missing other important discoveries."* P8: *"After a while, I realized I was just clicking whatever it suggested without really thinking."* Three participants reported deliberately ignoring recommendations to pursue their own hypotheses, suggesting they maintained analytical agency despite the guidance.

## 7.2 Study 2: Task-Based Evaluation of Recommendation System

*7.2.1 Methodology.* Our second study evaluated NetworkCanvas's recommendation system effectiveness for goal-directed analysis tasks using urban transportation networks. We recruited 16 participants (9 female, 7 male) from undergraduate programs and entry-level professional roles, excluding those with network analysis or transportation planning expertise (age: M = 23.1, SD = 2.4). Participants were randomly assigned to NetworkCanvas with Recommendations (n = 8) or Baseline Exploration without recommendations (n = 8). Both conditions provided identical visualization capabilities, differing only in recommendation system availability.

**Participants.** We recruited 16 participants (9 female, 7 male) from undergraduate programs and entry-level professional roles, explicitly excluding those with network analysis or transportation planning expertise. Ages ranged from 20 to 28 years (M = 23.1, SD = 2.4). All participants demonstrated basic data literacy but lacked domain knowledge in transportation systems.

Participants used the Anaheim urban road network (416 intersections, 914 road segments) to complete two transportation planning tasks: (1) identifying critical infrastructure for emergency response (success: finding ≥4 of 5 high-betweenness centrality intersections and ≥3 critical segments), and (2) evaluating toll policy impacts on traffic patterns (success: identifying ≥3 affected origin-destination pairs with specific toll recommendations). After 10-minute training, main tasks lasted 15 minutes each under the think-aloud protocol.

*7.2.2 Results.* Given our sample size (n = 16 total, n = 8 per condition), the study was powered to detect large effects (Cohen's d > 0.8) with approximately 80% power at $\alpha = 0.05$. We supplement parametric tests with non-parametric alternatives as robustness checks.

**Task Completion Performance.** NetworkCanvas with recommendations significantly outperformed the baseline condition. Task 1: 87.5% success vs. 50% ($\chi^2(1) = 4.27$, p = 0.039, $\phi = 0.52$; Fisher's exact: p = 0.049). Task 2: 75% success vs. 37.5% ($\chi^2(1) = 4.57$, p = 0.032, $\phi = 0.53$; Fisher's exact: p = 0.039).

**Completion Time Analysis.** Task 1 completion times favored the recommendation condition (M = 12.3 min, SD = 2.1, 95% CI [10.5, 14.1]) compared to baseline (M = 14.8 min, SD = 2.9, 95% CI [12.4, 17.2]); t(14) = 2.18, p = 0.046, d = 0.99; Mann-Whitney U = 14, p = 0.038. Task 2 showed similar patterns but did not reach conventional significance (Recommendation: M = 13.1 min; Baseline: M = 15.6 min; t(14) = 1.89, p = 0.080, d = 0.88).

**Recommendation System Usage.** Participants accepted an average of 73% of system suggestions (SD = 12%), with higher utilization for Task 1 (M = 78%) than Task 2 (M = 68%), t(7) = 2.45, p = 0.044, d = 0.82.

**Subjective Ratings.** Table 3 summarizes subjective rating comparisons. All measures showed large effect sizes favoring the recommendation condition.

**Table 3: Study 2 subjective ratings (7-point Likert scales)**

| Measure | Rec. | Base. | t(14) | d |
|---|---|---|---|---|
| Analytical confidence | 5.9 (0.8) | 4.2 (1.1) | 3.82** | 1.77 |
| Analysis completeness | 5.7 (0.9) | 3.8 (1.2) | 3.94** | 1.79 |
| Ease of use | 5.8 (0.7) | 4.4 (1.0) | 3.53** | 1.62 |

*Note:* Values are Mean (SD). $^{**}p < .01$. N=16 (n=8 per condition), between-subjects.

Users also rated recommendation quality highly: relevance (M = 5.8, SD = 0.7), timing appropriateness (M = 5.4, SD = 0.9), and trustworthiness (M = 5.6, SD = 0.8). Mann-Whitney U tests confirmed all subjective rating differences (all p < 0.01).

*7.2.3 Qualitative Analysis.* Interview analysis revealed distinct analytical strategies between conditions. **Systematic Exploration:** Recommendation system users demonstrated more systematic exploration patterns. P2: "The recommendations helped me build up understanding step by step, rather than jumping around randomly." **Domain Knowledge Compensation:** Participants noted how recommendations compensated for their lack of expertise. P6: "I don't know much about traffic analysis, but the system suggested looking at things that made sense once I explored them." **Confidence and Over-Reliance:** While recommendation users expressed greater confidence (P4: "Having the system suggest what to look at next made me feel like I was doing the analysis correctly"), some worried about over-reliance. P7: "I found myself depending on the suggestions. I'm not sure I'd know what to do without them."

*7.2.4 Interpretation Caveat.* Because our baseline lacked any recommendation functionality, the observed improvements reflect the combined effect of guidance availability and adaptive personalization. Our design establishes the value of recommendation-equipped exploration at the system level but does not isolate whether adaptivity (learning user preferences over time) provides benefits beyond static recommendations. The high recommendation acceptance rate (73%) and qualitative reports of the system anticipating analytical needs suggest personalization contributed to outcomes, but future ablation studies comparing adaptive versus fixed-weight recommendations are needed to disentangle these effects (subsubsection 8.2.3).

# 8 Discussion

NetworkCanvas demonstrates that recommendation-based guidance can meaningfully support network exploration while preserving user analytical agency. Our evaluation revealed that participants using the complete system identified more observations they considered noteworthy and reported higher confidence compared to baseline approaches without recommendation capabilities. While we cannot definitively claim that more bookmarked observations equates to better analysis, our qualitative findings suggest that

recommendations helped users overcome analysis paralysis, explore more diverse analytical directions, and develop confidence in their conclusions. We note that our experimental design establishes the value of the recommendation paradigm but does not isolate whether benefits stem specifically from adaptive personalization or from providing any form of structured analytical guidance. subsubsection 8.2.3 examines this boundary and identifies the ablation studies needed for definitive attribution.

## 8.1 Technical Contributions and Design Trade-offs

*Synergistic Architecture.* The three core technical innovations work synergistically to create an adaptive analytical environment. The QAG's directed graph structure captures transition preferences between analytical categories, with learned weights converging to meaningful patterns within 3–4 recommendation cycles. The ASG's state-centric provenance model supports non-linear exploration patterns that better reflect real analytical workflows than traditional linear history approaches. The context-aware feedback interpreter bridges the semantic gap between low-level user interactions and high-level analytical intent by analyzing selection patterns such as entity types, cardinality, and structural roles to infer analytical goals without requiring explicit user specification.

*Learning Mechanism Design.* A central design challenge was learning user preferences from implicit feedback signals, such as accept, skip, and modification actions, without requiring explicit ratings or surveys. Our learning mechanism employs asymmetric heuristic updates: stronger positive feedback for accepted recommendations ($\eta = 0.05$) combined with gentler penalties for skipped suggestions ($\lambda = 0.3$). This design prevents the system from becoming overly directive while still adapting to user preferences. Importantly, while this approach draws conceptual inspiration from reinforcement learning (particularly the notion of adjusting behavior based on feedback signals), it is implemented as deterministic update rules rather than a full reinforcement learning formulation with policy optimization. This design trade-off prioritizes interpretability and predictable behavior over theoretical optimality, which we found important for building user trust during iterative development.

*Automation-Agency Balance.* The balance between automation and agency introduces trade-offs. Users who strongly preferred bottom-up exploration occasionally found recommendations distracting, while some participants became overly reliant on suggestions. The optimal balance depends on user expertise level: novices benefited from more guidance, while experts appreciated the system's ability to surface overlooked analytical directions without constraining their investigation strategies.

*Cross-Domain Applicability.* The 10 question categories were designed as domain-agnostic analytical primitives that map to specific domain concepts through terminology rather than functionality. Table 4 illustrates how the same categories translate consistently across three representative domains.

**Table 4: Question category mappings across network domains**

| Category | Data Lineage | Transport Networks | Biological Networks |
|---|---|---|---|
| Community Detection | Data pipeline clusters | Traffic zones | Protein complexes |
| Node Centrality | Critical data assets | Key intersections | Essential genes |
| Path Analysis | Data flow tracing | Route optimization | Metabolic pathways |
| Anomaly Detection | Data quality issues | Traffic anomalies | Disease markers |
| What-If Analysis | Impact of schema changes | Infrastructure removal | Gene knockout simulation |

## 8.2 Limitations and Scaling Challenges

Despite promising results, NetworkCanvas faces several limitations that constrain its immediate applicability. We organize these limitations into evaluation scope, technical constraints, and methodological boundaries.

*8.2.1 Evaluation Scope and Domain Generalizability.* Our evaluation focused on two distinct network types: data lineage networks representing directed acyclic graphs with hierarchical structures, and urban transportation networks representing geographic/spatial networks with physical routing constraints. While these domains demonstrate applicability across fundamentally different structural types, they share characteristics that may not generalize:

- **Network density:** Both datasets exhibited relatively sparse connectivity (density < 0.01). Dense social networks with clustering coefficients exceeding 0.5 may present different exploration challenges.
- **Degree distribution:** Our test networks showed approximately normal degree distributions. Scale-free biological networks with power-law distributions may require different recommendation strategies.
- **Temporal dynamics:** Both datasets were static snapshots. Dynamic networks with evolving structure would require extensions to capture time-varying patterns.
- **Network scale:** Our largest evaluation network contained approximately 14,000 nodes. Scalability beyond 20,000 nodes requires additional engineering for progressive loading and view-dependent computation.

To provide preliminary evidence of cross-domain transferability, we conducted informal pilot testing with three bioinformatics researchers analyzing protein-protein interaction networks (N = 1,846 proteins, 4,171 interactions). Participants reported similar recommendation acceptance rates (71% vs. 73% in formal studies) and characterized the system as "surprisingly applicable" to their domain, though they noted that QAG category labels required mental translation.

*8.2.2 Technical and Algorithmic Limitations.* Our heuristic-based learning approach, while interpretable and stable, lacks the theoretical guarantees of formal reinforcement learning methods. The fixed learning rates ($\eta = 0.05$, $\lambda = 0.3$) were empirically tuned for typical exploration sessions but may require adjustment for different user populations. Future work could explore adaptive learning rates that respond to session length or user expertise indicators.

Our participant samples, while adequate for detecting large effects, were relatively homogeneous (primarily graduate students and early-career professionals in technical fields). Broader validation with domain experts, novice users from non-technical backgrounds, and users with different cultural approaches to data exploration would strengthen external validity claims.

*8.2.3 Baseline Design and Interpretation Boundaries.* Our evaluation establishes that recommendation-equipped exploration outperforms manual exploration for non-experts, but the baseline design constrains interpretive scope. Specifically, we cannot disentangle the contributions of *adaptive personalization* from *structured guidance in general*. Alternative baselines would address this limitation:

- **Static guidance baseline:** Providing the same recommendation interface but with fixed QAG weights (no learning) would isolate the contribution of adaptive personalization.
- **Randomized-but-feasible suggestions:** Offering valid but randomly selected recommendations would test whether benefits derive from "having options to click" rather than intelligent suggestion ranking.
- **Expert-authored workflow templates:** Pre-defined analytical playbooks (*e.g.*, "start with overview, then examine hubs, then check paths") would test whether curated guidance suffices or whether learning is necessary.
- **Component ablation:** Systematically removing scoring components (DOI, Synergy, Relevance) would quantify each factor's contribution to recommendation quality.

We did not implement these alternatives due to scope constraints, but they represent essential next steps for understanding NetworkCanvas's mechanisms. subsection 8.3 details our planned studies and expected evidence patterns.

## 8.3 Future Directions

NetworkCanvas opens promising research directions spanning evaluation methodology, technical extensions, and application domains.

**Isolating Adaptive Personalization.** Building on the baselines identified in subsubsection 8.2.3, our primary methodological priority is a factorial study comparing adaptive NetworkCanvas against a static-weight variant and a randomized-suggestion control. We hypothesize that adaptivity becomes increasingly valuable over longer sessions as the system learns individual analytical styles; evidence would include improving recommendation acceptance rates within sessions, stronger alignment with self-reported strategies after personalization, and higher analytical depth in later exploration phases. An additional baseline could use modern LLMs, prompted with current selection and exploration history, to generate contextual recommendations. This would test whether our structured QAG approach offers benefits over end-to-end LLM reasoning for recommendation generation.

**Technical Extensions.** Extensions to temporal networks could leverage the ASG structure to maintain exploration history across time-varying graphs. Integration with domain-specific analysis workflows, incorporating domain ontologies into the QAG initialization, could improve initial recommendation quality for specialized applications. Cross-user learning models could bootstrap new users based on community analytical patterns while preserving individual customization. Collaborative extensions leveraging the ASG branching model could support multi-analyst scenarios with parallel investigation threads and shared QAG learning.

**Cross-Domain Validation.** Building on preliminary evidence from our bioinformatics pilot (subsubsection 8.2.1), systematic cross-domain validation requires addressing the reliability factors identified in subsection 6.1. We plan to develop domain-specific prompt templates that establish terminology mappings (*e.g.*, "protein" → "node") and validate classification accuracy on held-out, manually-labeled query sets for each target domain. Expert-in-the-loop validation, where domain specialists periodically rate explanation factuality, would enable iterative prompt refinement. Future versions should surface confidence scores to users, enabling appropriate trust calibration in high-stakes analytical contexts.

## 9 Conclusion

This paper presented NetworkCanvas, a progressive network visualization system that guides non-expert users through recommendation-driven, mixed-initiative guidance for network exploration. Our primary contribution lies in reconceptualizing network exploration as an adaptive dialogue between analyst and system. The Question-Affinity Graph engine learns user preferences through heuristic-based updates, the Analytic State Graph manager preserves exploration provenance as a branching structure, and the context-aware feedback interpreter bridges the gap between low-level interactions and high-level analytical intent. Our controlled studies demonstrated that participants using NetworkCanvas bookmarked more observations and reported higher confidence compared to a baseline tool without recommendations, with qualitative findings highlighting reduced analysis paralysis and enhanced pattern recognition. These results support the value of recommendation-based guidance for network exploration. Our evaluation does not disentangle the specific contribution of adaptive personalization from the general benefit of structured suggestions, but qualitative evidence that participants perceived the system as learning their preferences indicates that adaptivity likely played a role in sustained engagement and analytical depth. Isolating this contribution through comparative studies against static and randomized recommendation baselines remains a critical next step.

NetworkCanvas opens promising research directions, including comparative baseline studies (adaptive vs. static guidance), extensions to temporal networks leveraging the ASG structure, domain-specific QAG initialization for improved cold-start performance, cross-user learning models to bootstrap new users, and collaborative extensions supporting multi-analyst scenarios.

## Acknowledgments

The authors would like to thank the experts and participants for their help in the project, as well as the anonymous reviewers for their valuable comments. This work was conducted outside the scope of the authors' employment at Huawei Technologies Co., Ltd., and did not constitute a company work product. This work was supported by the Natural Science Foundation of Jiangsu Province (Grant No. BK20251232) and the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM902).

## References


[1] Mohammad Mehdi Afsar, Trafford Crump, and Behrouz H. Far. 2023. Reinforcement Learning based Recommender Systems: A Survey. *ACM Comput. Surv.* 55, 7 (2023), 145:1–145:38.

[2] Mashael AlKadi, Vanessa Serrano, James Scott-Brown, Catherine Plaisant, Jean-Daniel Fekete, Uta Hinrichs, and Benjamin Bach. 2023. Understanding Barriers to Network Exploration with Visualization: A Report from the Trenches. *IEEE Trans. Vis. Comput. Graph.* 29, 1 (2023), 907–917.

[3] Marco Angelini, Giuseppe Santucci, Heidrun Schumann, and Hans-Jörg Schulz. 2018. A Review and Characterization of Progressive Visual Analytics. *Informatics* 5, 3 (2018), 31.

[4] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. 2009. Gephi: An Open Source Software for Exploring and Manipulating Networks. In *Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA, May 17-20, 2009*, Eytan Adar, Matthew Hurst, Tim Finin, Natalie S. Glance, Nicolas Nicolov, and Belle L. Tseng (Eds.). The AAAI Press.

[5] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. 2014. The State of the Art in Visualizing Dynamic Graphs. In *16th Eurographics Conference on Visualization, EuroVis 2014 - State of the Art Reports, Swansea, UK, June 9-13, 2014*, Rita Borgo, Ross Maciejewski, and Ivan Viola (Eds.). Eurographics Association.

[6] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. 2017. A Taxonomy and Survey of Dynamic Graph Visualization. *Comput. Graph. Forum* 36, 1 (2017), 133–159.

[7] Steven P. Callahan, Juliana Freire, Emanuele Santos, Carlos Eduardo Scheidegger, Cláudio T. Silva, and Huy T. Vo. 2006. VisTrails: visualization meets data management. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, USA, June 27-29, 2006*, Surajit Chaudhuri, Vagelis Hristidis, and Neoklis Polyzotis (Eds.). ACM, 745–747.

[8] Davide Ceneda, Theresia Gschwandtner, Thorsten May, Silvia Miksch, Hans-Jörg Schulz, Marc Streit, and Christian Tominski. 2017. Characterizing Guidance in Visual Analytics. *IEEE Trans. Vis. Comput. Graph.* 23, 1 (2017), 111–120.

[9] Peng Chen, Beth Plale, You-Wei Cheah, Devarshi Ghoshal, Scott Jensen, and Yuan Luo. 2012. Visualization of network data provenance. In *19th International Conference on High Performance Computing, HiPC 2012, Pune, India, December 18-22, 2012*. IEEE Computer Society, 1–9.

[10] Xiaocong Chen, Lina Yao, Julian J. McAuley, Guanglin Zhou, and Xianzhi Wang. 2023. Deep reinforcement learning in recommender systems: A survey and new perspectives. *Knowl. Based Syst.* 264 (2023), 110335.

[11] Tarik Crnovrsanin, Isaac Liao, Yingcai Wu, and Kwan-Liu Ma. 2011. Visual Recommendations for Network Navigation. *Comput. Graph. Forum* 30, 3 (2011), 1081–1090.

[12] Cody Dunne, Nathalie Henry Riche, Bongshin Lee, Ronald A. Metoyer, and George G. Robertson. 2012. GraphTrail: analyzing large multivariate, heterogeneous networks while supporting exploration history. In *CHI Conference on Human Factors in Computing Systems, CHI '12, Austin, TX, USA - May 05 - 10, 2012*, Joseph A. Konstan, Ed H. Chi, and Kristina Höök (Eds.). ACM, 1663–1672.

[13] Will Epperson, Doris Jung Lin Lee, Leijie Wang, Kunal Agarwal, Aditya G. Parameswaran, Dominik Moritz, and Adam Perer. 2022. Leveraging Analysis History for Improved In Situ Visualization Recommendation. *Comput. Graph. Forum* 41, 3, 145–155.

[14] Jean-Daniel Fekete. 2015. Progressivis: A toolkit for steerable progressive analytics and visualization. In *1st Workshop on Data Systems for Interactive Analysis*. 5.

[15] Velitchko Andreev Filipov, Alessio Arleo, and Silvia Miksch. 2023. Are We There Yet? A Roadmap of Network Visualization from Surveys to Task Taxonomies. *Comput. Graph. Forum* 42, 6 (2023).

[16] Danyel Fisher, Igor O. Popov, Steven Mark Drucker, and m. c. schraefel. 2012. Trust me, i'm partially right: incremental visualization lets analysts explore large datasets faster. In *CHI Conference on Human Factors in Computing Systems, CHI '12, Austin, TX, USA - May 05 - 10, 2012*, Joseph A. Konstan, Ed H. Chi, and Kristina Höök (Eds.). ACM, 1673–1682.

[17] Takanori Fujiwara, Jian Zhao, Francine Chen, Yaoliang Yu, and Kwan-Liu Ma. 2022. Network Comparison with Interpretable Contrastive Network Representation Learning. *J. Data Sci. Stat. Vis.* 2, 5 (2022).

[18] Joseph Gardiner, Marco Cova, and Shishir Nagaraja. 2014. Command & Control: Understanding, Denying and Detecting. *CoRR* abs/1408.1136 (2014). arXiv:1408.1136

[19] David Gotz and Michelle X. Zhou. 2009. Characterizing users' visual analytic activity for insight provenance. *Inf. Vis.* 8, 1 (2009), 42–55.

[20] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Nicola Cosgrove, and Marc Streit. 2016. From Visual Exploration to Storytelling and Back Again. *Comput. Graph. Forum* 35, 3 (2016), 491–500.

[21] Camille Harris, Ryan A. Rossi, Sana Malik, Jane Hoffswell, Fan Du, Tak Yeon Lee, Eunyee Koh, and Handong Zhao. 2023. SpotLight: Visual Insight Recommendation. In *Companion Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 19–23.

[22] Jeffrey Heer and Ben Shneiderman. 2012. Interactive dynamics for visual analysis. *Commun. ACM* 55, 4 (2012), 45–54.

[23] Ivan Herman, Guy Melançon, and M. Scott Marshall. 2000. Graph Visualization and Navigation in Information Visualization: A Survey. *IEEE Trans. Vis. Comput. Graph.* 6, 1 (2000), 24–43.

[24] Marius Hogräfer, Marco Angelini, Giuseppe Santucci, and Hans-Jörg Schulz. 2022. Steering-by-example for Progressive Visual Analytics. *ACM Trans. Intell. Syst. Technol.* 13, 6 (2022), 96:1–96:26.

[25] Eric Horvitz. 1999. Principles of Mixed-Initiative User Interfaces. In *Proceeding of the CHI '99 Conference on Human Factors in Computing Systems: The CHI is the Limit, Pittsburgh, PA, USA, May 15-20, 1999*, Marian G. Williams and Mark W. Altom (Eds.). ACM, 159–166.

[26] Kevin Zeng Hu, Michiel A. Bakker, Stephen Li, Tim Kraska, and César A. Hidalgo. 2019. VizML: A Machine Learning Approach to Visualization Recommendation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, Stephen A. Brewster, Geraldine Fitzpatrick, Anna L. Cox, and Vassilis Kostakos (Eds.). ACM, 128.

[27] Jessica Hullman. 2020. Why Authors Don't Visualize Uncertainty. *IEEE Trans. Vis. Comput. Graph.* 26, 1 (2020), 130–139.

[28] Jaemin Jo, Sehi L'Yi, Bongshin Lee, and Jinwook Seo. 2021. ProReveal: Progressive Visual Analytics With Safeguards. *IEEE Trans. Vis. Comput. Graph.* 27, 7 (2021), 3109–3122.

[29] Bharat Kale, Maoyuan Sun, and Michael E. Papka. 2023. The State of the Art in Visualizing Dynamic Multivariate Networks. *Comput. Graph. Forum* 42, 3 (2023), 471–490.

[30] Andreas Kerren, Helen C. Purchase, and Matthew O. Ward. 2013. Introduction to Multivariate Network Visualization. In *Multivariate Network Visualization - Dagstuhl Seminar #13201, Dagstuhl Castle, Germany, May 12-17, 2013, Revised Discussions (Lecture Notes in Computer Science, Vol. 8380)*, Andreas Kerren, Helen C. Purchase, and Matthew O. Ward (Eds.). Springer, 1–9.

[31] Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. 2006. Task taxonomy for graph visualization. In *BELIV*. ACM Press, 1–5.

[32] Haotian Li, Yun Wang, and Huamin Qu. 2024. Where Are We So Far? Understanding Data Storytelling Tools from the Perspective of Human-AI Collaboration. In *Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI 2024, Honolulu, HI, USA, May 11-16, 2024*, Florian 'Floyd' Mueller, Penny Kyburz, Julie R. Williamson, Corina Sas, Max L. Wilson, Phoebe O. Toups Dugas, and Irina Shklovski (Eds.). ACM, 845:1–845:19.

[33] Haotian Li, Yong Wang, Songheng Zhang, Yangqiu Song, and Huamin Qu. 2022. KG4Vis: A Knowledge Graph-Based Approach for Visualization Recommendation. *IEEE Trans. Vis. Comput. Graph.* 28, 1 (2022), 195–205.

[34] Wenchao Li, Sarah Schöttler, James Scott-Brown, Yun Wang, Siming Chen, Huamin Qu, and Benjamin Bach. 2023. NetworkNarratives: Data Tours for Visual Network Exploration and Analysis. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI 2023, Hamburg, Germany, April 23-28, 2023*, Albrecht Schmidt, Kaisa Väänänen, Tesh Goyal, Per Ola Kristensson, Anicia Peters, Stefanie Mueller, Julie R. Williamson, and Max L. Wilson (Eds.). ACM, 172:1–172:15.

[35] Jock D. Mackinlay. 1986. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.* 5, 2 (1986), 110–141.

[36] Luana Micallef, Hans-Jörg Schulz, Marco Angelini, Michaël Aupetit, Remco Chang, Jörn Kohlhammer, Adam Perer, and Giuseppe Santucci. 2019. The Human User in Progressive Visual Analytics. In *21st Eurographics Conference on Visualization, EuroVis 2019 - Short Papers, Porto, Portugal, June 3-7, 2019*, Jimmy Johansson, Filip Sadlo, and G. Elisabeta Marai (Eds.). Eurographics Association, 19–23.

[37] Dominik Moritz, Chenglong Wang, Greg L. Nelson, Halden Lin, Adam M. Smith, Bill Howe, and Jeffrey Heer. 2019. Draco: Bringing Database Optimization to Visualization Systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 1–12.

[38] Tamara Munzner. 2009. A Nested Process Model for Visualization Design and Validation. *IEEE Trans. Vis. Comput. Graph.* 15, 6 (2009), 921–928.

[39] Belgin Mutlu, Eduardo E. Veas, and Christoph Trattner. 2016. VizRec: Recommending Personalized Visualizations. *ACM Trans. Interact. Intell. Syst.* 6, 4 (2016), 31:1–31:39.

[40] Jakob Nielsen. 1994. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[41] Adam Perer and Ben Shneiderman. 2006. Balancing Systematic and Flexible Exploration of Social Networks. *IEEE Trans. Vis. Comput. Graph.* 12, 5 (2006), 693–700.

[42] Ignacio Pérez-Messina, Marco Angelini, Davide Ceneda, Christian Tominski, and Silvia Miksch. 2025. Coupling Guidance and Progressiveness in Visual Analytics. *Comput. Graph. Forum* 44, 3 (2025).

[43] Nicola Pezzotti, Boudewijn P. F. Lelieveldt, Laurens van der Maaten, Thomas Höllt, Elmar Eisemann, and Anna Vilanova. 2017. Approximated and User Steerable tSNE for Progressive Visual Analytics. *IEEE Trans. Vis. Comput. Graph.* 23, 7 (2017), 1739–1752.

[44] Robert S. Pienta, Minsuk Kahng, Zhiyuan Lin, Jilles Vreeken, Partha P. Talukdar, James Abello, Ganesh Parameswaran, and Duen Horng Chau. 2017. FACETS: Adaptive Local Exploration of Large Graphs. In *Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, April 27-29, 2017*, Nitesh V. Chawla and Wei Wang (Eds.). SIAM, 597–605.

[45] Eric D. Ragan, Alex Endert, Jibonananda Sanyal, and Jian Chen. 2016. Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE Trans. Vis. Comput. Graph.* 22, 1 (2016), 31–40.

[46] Neha Rani, Srikar Kantamani, and Sharon Lynn Chu. 2025. Exploring the Impact of User Feedback for Trust in Context-Aware Recommender Systems in Search-as-Learning. In *Human-Computer Interaction - Thematic Area, HCI 2025, Held as Part of the 27th HCI International Conference, HCII 2025, Gothenburg, Sweden, June 22-27, 2025, Proceedings, Part VI (Lecture Notes in Computer Science, Vol. 15771)*, Masaaki Kurosu and Ayako Hashizume (Eds.). Springer, 83–99.

[47] Sanad Saha, Nischal Aryal, Leilani Battle, and Arash Termehchy. 2024. ShiftScope: Adapting Visualization Recommendations to Users' Dynamic Data Focus. In *Companion of the 2024 International Conference on Management of Data, SIGMOD/PODS 2024, Santiago, Chile, June 9-15, 2024*, Pablo Barceló, Nayat Sánchez-Pi, Alexandra Meliou, and S. Sudarshan (Eds.). ACM, 536–539.

[48] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland*, Kalervo Järvelin, Micheline Beaulieu, Ricardo A. Baeza-Yates, and Sung-Hyon Myaeng (Eds.). ACM, 253–260.

[49] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* 13, 11 (2003), 2498–2504.

[50] Rohini Sharma, Ajay Guleria, and R. K. Singla. 2018. An overview of flow-based anomaly detection. *Int. J. Commun. Networks Distributed Syst.* 21, 2 (2018), 220–240.

[51] Ben Shneiderman. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *VL*. IEEE Computer Society, 336–343.

[52] Ben Shneiderman and Cody Dunne. 2012. *Interactive Network Exploration to Derive Insights: Filtering, Clustering, Grouping, and Simplification.* Technical Report. 2–18 pages.

[53] Fabian Sperrle, Astrik Jeitler, Jürgen Bernard, Daniel A. Keim, and Mennatallah El-Assady. 2021. Co-adaptive visual data analysis and guidance processes. *Comput. Graph.* 100 (2021), 93–105.

[54] Arjun Srinivasan, Steven Mark Drucker, Alex Endert, and John T. Stasko. 2019. Augmenting Visualizations with Interactive Data Facts to Facilitate Interpretation and Communication. *IEEE Trans. Vis. Comput. Graph.* 25, 1 (2019), 672–681.

[55] Holger Stitz, Samuel Gratzl, Harald Piringer, Thomas Zichner, and Marc Streit. 2019. KnowledgePearls: Provenance-Based Visualization Retrieval. *IEEE Trans. Vis. Comput. Graph.* 25, 1 (2019), 120–130.

[56] Charles D. Stolper, Adam Perer, and David Gotz. 2014. Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics. *IEEE Trans. Vis. Comput. Graph.* 20, 12 (2014), 1653–1662.

[57] John Sweller. 1988. Cognitive Load During Problem Solving: Effects on Learning. *Cogn. Sci.* 12, 2 (1988), 257–285.

[58] Michael Tsang, Dehua Cheng, Hanpeng Liu, Xue Feng, Eric Zhou, and Yan Liu. 2020. Feature Interaction Interpretability: A Case for Explaining Ad-Recommendation Systems via Neural Interaction Detection. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[59] Cagatay Turkay, Erdem Kaya, Selim Balcisoy, and Helwig Hauser. 2017. Designing Progressive and Interactive Analytics Processes for High-Dimensional Data Analysis. *IEEE Trans. Vis. Comput. Graph.* 23, 1 (2017), 131–140.

[60] Alex Ulmer, Marco Angelini, Jean-Daniel Fekete, Jörn Kohlhammer, and Thorsten May. 2024. A Survey on Progressive Visualization. *IEEE Trans. Vis. Comput. Graph.* 30, 9 (2024), 6447–6467.

[61] Corinna Vehlow, Fabian Beck, and Daniel Weiskopf. 2017. Visualizing Group Structures in Graphs: A Survey. *Comput. Graph. Forum* 36, 6 (2017), 201–225.

[62] Tatiana von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J. van Wijk, Jean-Daniel Fekete, and Dieter W. Fellner. 2011. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Comput. Graph. Forum* 30, 6 (2011), 1719–1749.

[63] Dakuo Wang, Elizabeth F. Churchill, Pattie Maes, Xiangmin Fan, Ben Shneiderman, Yuanchun Shi, and Qianying Wang. 2020. From Human-Human Collaboration to Human-AI Collaboration: Designing AI Systems That Can Work Together with People. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, CHI 2020, Honolulu, HI, USA, April 25-30, 2020*, Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik (Eds.). ACM, 1–6.

[64] Jing Wang and Ioannis Ch. Paschalidis. 2017. Botnet Detection Based on Anomaly and Community Detection. *IEEE Trans. Control. Netw. Syst.* 4, 2 (2017), 392–404.

[65] Qianwen Wang, Chen Zhu-Tian, Yong Wang, and Huamin Qu. 2022. A Survey on ML4VIS: Applying Machine Learning Advances to Data Visualization. *IEEE Trans. Vis. Comput. Graph.* 28, 12 (2022), 5134–5153.

[66] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock D. Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Towards a general-purpose query language for visualization recommendation. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics, HILDA@SIGMOD 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, Carsten Binnig, Alan D. Fekete, and Arnab Nandi (Eds.). ACM, 4.

[67] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock D. Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Trans. Vis. Comput. Graph.* 22, 1 (2016), 649–658.

[68] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock D. Mackinlay, Bill Howe, and Jeffrey Heer. 2017. Voyager 2: Augmenting Visual Analysis with Partial View Specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017*, Gloria Mark, Susan R. Fussell, Cliff Lampe, m. c. schraefel, Juan Pablo Hourcade, Caroline Appert, and Daniel Wigdor (Eds.). ACM, 2648–2659.

[69] Junran Yang, Péter Ferenc Gyarmati, Zehua Zeng, and Dominik Moritz. 2023. Draco 2: An Extensible Platform to Model Visualization Design. 166–170 pages.

[70] Zehua Zeng and Leilani Battle. 2024. A Systematic Review of Visualization Recommendation Systems: Goals, Strategies, Interfaces, and Evaluations. *Found. Trends Databases* 14, 1 (2024), 1–71.

[71] Emanuel Zgraggen, Alex Galakatos, Andrew Crotty, Jean-Daniel Fekete, and Tim Kraska. 2017. How Progressive Visualizations Affect Exploratory Analysis. *IEEE Trans. Vis. Comput. Graph.* 23, 8 (2017), 1977–1987.

[72] Songheng Zhang, Haotian Li, Huamin Qu, and Yong Wang. 2024. AdaVis: Adaptive and Explainable Visualization Recommendation for Tabular Data. 5923–5938 pages.

[73] Jian Zhao, Christopher Collins, Fanny Chevalier, and Ravin Balakrishnan. 2013. Interactive Exploration of Implicit and Explicit Relations in Faceted Datasets. *IEEE Trans. Vis. Comput. Graph.* 19, 12 (2013), 2080–2089.

[74] Jian Zhao, Mingming Fan, and Mi Feng. 2022. ChartSeer: Interactive Steering Exploratory Visual Analysis With Machine Intelligence. *IEEE Trans. Vis. Comput. Graph.* 28, 3 (2022), 1500–1513.

[75] Xin Zhao, Shuowen Fu, Rui Yang, Lei Yang, Yunpeng Chen, Jiang Zhang, Jiang Long, Fangfang Zhou, and Ying Zhao. 2025. Investigating Visual Perception of Degree Centrality in Graph Visualization. *IEEE Trans. Vis. Comput. Graph.* 31, 6 (2025), 3679–3692.

[76] Sujia Zhu, Guodao Sun, Qi Jiang, Meng Zha, and Ronghua Liang. 2020. A survey on automatic infographics and visualization recommendations. *Vis. Informatics* 4, 3 (2020), 24–40.

[77] Eric Zimmermann and Stefan Bruckner. 2025. Multi-Focus Probes for Context-Preserving Network Exploration and Interaction in Immersive Analytics.

# A Question Categories

**Table 5: Question Categories with Associated Entities and Insight Types**

| Question Category | Entity | Related Insight Type | Insight Description |
|---|---|---|---|
| **1. Network Overview** Questions about summarizing and measuring global structural properties and metrics of the entire network. | Network | Global Summary | Provide a concise synopsis of the entire network's topology and key metrics (size, density, connectivity). |
| | | Count Entities | Compute total node and link counts within the network. |
| | | Density & Sparsity | Calculate the ratio of actual to possible links to assess overall compactness. |
| | | Isolation Detection | Identify disconnected or isolated sub-networks within the overall network. |
| | | Global Metric Computation | Derive network-level statistics such as diameter, average path length, and clustering coefficient. |
| **2. Node Analysis** Questions about locating, evaluating, and exploring properties of specific nodes within the network. | Node | Locate Node | Find a specific node by identifier or via attribute-based search. |
| | | Rank by Centrality | Order nodes based on degree, betweenness, closeness, or eigenvector scores. |
| | | Filter Nodes | Restrict the view to nodes satisfying degree centrality or metric thresholds. |
| | | k-Hop Neighborhood | Extract all nodes reachable within k hops from a source node. |
| | | Common Connection | Find nodes that are connected to all of a given set of starting nodes. |
| | | Details-on-Demand | Display all metadata and computed metrics for a selected node. |
| | | Articulation Point Detection | Identify nodes whose removal increases the number of connected components in the network. |
| **3. Link Analysis** Questions about identifying, filtering, ranking, and exploring the structural significance of links in the network. | Link | Identify Link | Select or highlight the edge connecting two specified nodes. |
| | | Filter Links | Show links that meet direction criteria, properties associated with the nodes they connect, or multiple criteria combination. |
| | | Bridge Detection | Find links whose removal increases the number of connected components. |
| | | Rank Links | Sort links by attributes such as weight, capacity, or frequency. |
| | | Link Set Comparison | Compare inner and outer links of a group for structural insights. |
| **4. Path & Reachability Analysis** Questions about finding, verifying, and analyzing connectivity paths between nodes or groups. | Group/Node | Shortest Path | Compute the minimal-hop or minimal-cost path between two nodes. |
| | | Reachability Check | Determine whether a valid route exists under given constraints. |
| | | All Paths Enumeration | List multiple disjoint or alternative routes up to a specified length. |
| | | Path Containment | Verify if a given path lies entirely within a selected group. |
| **5. Community & Detection** Questions about discovering, counting, comparing, and analyzing network communities or clusters. | Network/Group | Community Identification | Discover clusters via algorithms such as Louvain. |
| | | Count Communities | Return the total number of detected groups. |
| | | Group Membership | Query which community contains a specified node. |
| | | Inter-Group Connectivity | Measure links connecting distinct communities. |
| | | Group Metric Comparison | Compare size, density, or cohesion metrics across communities. |

**Table 6: (continued) Question Categories with Associated Entities and Insight Types**

| Question Category | Entity | Related Insight Type | Insight Description |
|---|---|---|---|
| **6. Motif & Local Pattern Discovery** Questions about detecting, counting, comparing, and evaluating statistically significant subgraph patterns or motifs. | Network/Group | Motif Detection | Locate occurrences of canonical subgraphs (e.g., triangles, feedforward loops). |
| | | Motif Counting | Compute the frequency of each motif type within the network. |
| | | Statistical Significance | Assess motif overrepresentation against randomized graph ensembles. |
| | | Motif Enumeration | List explicit node sets forming each instance of a given motif. |
| | | Pattern Comparison | Contrast motif distributions across subgraphs. |
| **7. Attribute-based Exploration** Questions about filtering, identifying extremes, visualizing distributions, and analyzing statistical summaries based on entity attributes. | Network/Group/ Node/Link | Attribute Filtering | Select entities meeting attribute predicates (e.g., weight > X). |
| | | Identify Extremes | Find nodes or links with maximum or minimum attribute values. |
| | | Distribution Analysis | Visualize attribute histograms or heatmaps to assess value dispersion. |
| | | Aggregate Statistics | Calculate mean, median, variance, or quartiles for selected attributes. |
| | | Comparative Filtering | Compare attribute distributions between subsets (e.g., groups). |
| **8. Attribute-Topology Correlation** Questions about assessing relationships and correlations between network topology and entity attributes. | Network/Group | Correlation Computation | Measure statistical correlation between structural metrics and attributes (e.g., degree vs. tenure). |
| | | Assortativity Analysis | Quantify homophily by assessing attribute similarity along edges. |
| | | Topology-Conditioned Aggregation | Summarize attribute statistics within structural subsets (e.g., communities). |
| | | Conditional Filtering | Filter entities by combined attribute and topology criteria (e.g., high-degree nodes with attribute X). |
| **9. Outlier & Anomaly Detection** Questions about detecting, identifying, and ranking unusual or anomalous nodes, links, groups, or subnetworks. | Network/Group/ Node/Link | Node Outlier Identification | Detect nodes with attribute or metric values that deviate significantly from the norm. |
| | | Link Anomaly Detection | Flag links exhibiting unexpected weights or interaction patterns. |
| | | Subgraph Anomalies | Identify groups or subnetworks with anomalous density or motif counts. |
| | | Anomaly Ranking | Score and rank entities by their anomaly magnitude to prioritize investigation. |
| **10. What-If Analysis** Questions about simulating hypothetical changes to network elements or parameters and evaluating resulting structural impacts. | Network/Group/ Node/Link | Element Removal Simulation | Delete nodes or links hypothetically and recompute metrics to assess impact. |
| | | Element Addition Scenario | Add hypothetical links or nodes and evaluate resulting structural changes. |
| | | Parameter Sensitivity | Vary weights or attribute parameters to observe metric fluctuations. |
| | | Optimization Queries | Identify minimal edits needed to achieve a topological goal (e.g., restore connectivity). |

# B Implementation Details of Recommendation Scoring

This appendix provides implementation details for the multi-criteria recommendation scoring introduced in section 5. While the main text describes the four scoring components at a conceptual level, this appendix specifies the concrete scoring rules, threshold values, and matrices that govern recommendation generation. We detail each component's computation, explain the rationale behind specific parameter choices, and present the formal scoring algorithm.

## B.1 QAG Affinity Component ($\alpha = 0.4$)

The affinity component directly leverages Question-Affinity Graph (QAG) weights learned through user interaction feedback. For a recommendation $r$ suggesting a transition from the current category $c_{\text{current}}$ to target category $c_{\text{target}}$, the affinity score equals the learned transition weight:

$$\text{Affinity}(r) = w_{c_{\text{current}} \to c_{\text{target}}} \tag{4}$$

This component receives the highest weight ($\alpha = 0.4$) to ensure learned user preferences drive recommendations while remaining below 0.5 so that contextual factors still influence final rankings.

## B.2 Degree-of-Interest (DOI) Component ($\beta = 0.25$)

The DOI component measures how well a recommendation aligns with the user's current selection context. Identical analytical operations may have vastly different relevance depending on what entities are selected; selecting a single hub node calls for different recommendations than selecting a cluster of peripheral nodes. The DOI computation operates across three dimensions:

**Entity Type Relevance.** We establish compatibility matrices between user selection types (nodes, edges, groups) and recommendation target entities. Direct matches receive maximum scores (1.0), while compatible combinations receive graduated scores based on analytical coherence. For instance, node selections strongly favor node-focused analyses (1.0) but maintain moderate compatibility with network-level analyses (0.7) and lower compatibility with link-specific operations (0.3).

**Cardinality-Based Scoring.** Selection cardinality fundamentally influences analytical appropriateness. Single entity selections ($n = 1$) favor detail-oriented analyses such as "details-on-demand" (1.0) and "k-hop neighborhood exploration" (0.9), reflecting the natural progression from selection to detailed investigation. Small sets ($2 \leq n \leq 10$) emphasize comparative analyses like "rank-by-centrality" (1.0) and "shortest-path computation" (0.9), supporting hypothesis testing between entities. Large selections ($n > 10$) prioritize statistical operations, including "aggregate-statistics" (1.0) and "distribution-analysis" (1.0), acknowledging that statistical inference becomes meaningful with sufficient samples.

**Structural Role Detection.** We employ lightweight centrality metrics to infer structural roles of selected entities, then boost recommendations aligned with these roles. Hub nodes (degree above 75th percentile) receive elevated scores for centrality analyses (0.9), neighborhood exploration (0.8), and impact simulation (0.8). Bridge nodes (high betweenness centrality) favor bridge detection (1.0), shortest path analyses (0.9), and removal simulation (0.9). Outlier nodes (degree below 25th percentile or above 95th percentile) emphasize anomaly detection (1.0) and detailed investigation (0.8).

The final DOI score combines these three dimensions through weighted averaging, with entity type contributing 40%, cardinality 35%, and structural role 25%.

## B.3 Cross-Branch Synergy Component ($\gamma = 0.2$)

The synergy component leverages the Analytic State Graph (ASG) to promote recommendations that complement ongoing analyses in parallel exploration branches. Users frequently pursue multiple analytical hypotheses simultaneously; this component ensures recommendations account for cross-branch coherence. The computation operates through three mechanisms:

**Insight Complementarity Analysis.** We maintain a complementarity matrix encoding analytical relationships between insight types. For instance, community detection strongly complements motif analysis (0.8) because local patterns and global clustering reveal structure at different scales. Centrality ranking synergizes with k-hop neighborhood exploration (0.8), as users naturally investigate the neighborhoods of highly central nodes. The complementarity matrix was constructed through analysis of network analysis literature and expert consultation, identifying common analytical progressions.

**Novelty Incentives.** Beyond complementarity, we incentivize exploration of analytical territories not yet visited in the current branch. Novel insights, which are neither completed nor currently pending in the active branch, receive novelty boosts (+0.3), encouraging broader exploration coverage. Partially explored insights (pending but not completed) receive modest encouragement (+0.1), while previously completed analyses receive no novelty bonus, preventing redundant re-exploration.

**Branch-Aware Context.** The synergy computation considers the analytical landscape across all active branches, identifying gaps where complementary analyses could provide cross-validation or alternative perspectives. This enables NetworkCanvas to suggest analyses that build bridges between parallel investigation threads.

## B.4 Insight Relevance Component ($\delta = 0.15$)

The insight relevance component ensures recommendations are executable given current dataset characteristics and selection preconditions, preventing user frustration from impossible or meaningless suggestions. The relevance assessment operates across three dimensions:

**Dataset Capability Matching.** Each analytical insight requires specific dataset features for meaningful execution. Correlation analyses require numeric node attributes, motif detection benefits from directed graphs, and geospatial analyses depend on coordinate information. We maintain capability requirement specifications for each insight type and evaluate dataset fitness accordingly. Insights receive full relevance scores (1.0) when all requirements are met, proportional scores when partially satisfied, and minimal scores (0.0–0.3) when critical capabilities are absent.

**Precondition Validation.** Beyond dataset capabilities, many insights have contextual preconditions. Shortest path computation requires at least two selected nodes, comparative filtering needs

multiple entities for meaningful comparison, and statistical significance testing demands sufficient network size for valid inference. Precondition checking validates these requirements against the current selection context and dataset characteristics, ensuring suggestions are actionable.

**Computational Complexity Appropriateness.** Network size significantly impacts analytical feasibility and user experience. We categorize insights by computational complexity (low, medium, high, very high) and scale appropriateness scores based on network size. Small networks ($n < 100$) favor low and medium complexity analyses while penalizing computationally expensive operations. Large networks ($n > 1000$) can support high-complexity analyses but penalize exhaustive algorithms.

### B.5 Scoring Integration

The final recommendation score combines all components through weighted summation (Equation 3). The weights ($\alpha = 0.4$, $\beta = 0.25$, $\gamma = 0.2$, $\delta = 0.15$) were calibrated through pilot studies to balance personalization with contextual relevance. All component scores are normalized to $[0, 1]$ before combination to ensure each contributes meaningfully to final rankings.

Algorithm 1 formalizes the scoring process. The algorithm returns both the final score and individual component scores, enabling transparency about why specific recommendations appear.

**Algorithm 1** Multi-Criteria Recommendation Scoring

**Require:** recommendation $r$, selection context $c$, QAG weight $w_{c_{\text{curr}} \to c_{\text{target}}}$, ASG manager $A$, dataset info $D$
**Ensure:** final score and component breakdown

```
Affinity(r) ← w_{c_curr → c_target}
DOI(r) ← computeDOI(r, c) {Entity + Cardinality + Role}
Synergy(r) ← computeSynergy(r, A) {Complementarity + Novelty}
Relevance(r) ← computeRelevance(r, D, c) {Capability + Precondition + Complexity}
Score(r) ← 0.4 · Affinity(r) + 0.25 · DOI(r) + 0.2 · Synergy(r) + 0.15 · Relevance(r)
return {Score(r), Affinity(r), DOI(r), Synergy(r), Relevance(r)}
```

## C Initial heuristic Question-Affinity Graph (QAG) Weights

The initial QAG affinity matrix is used to address the *cold-start* problem discussed in subsection 5.1. Before accumulating user interaction data, the system requires reasonable default transition probabilities to generate meaningful recommendations from the first interaction. Each entry $w_{A \to B} \in [0, 1]$ is a heuristic prior estimating how natural it is to transition from category $A$ to category $B$ during network exploration.

Table 7 presents the complete $10 \times 10$ matrix. Rows indicate the current category (FROM); columns indicate the next category (TO). Diagonal entries $w_{A \to A}$ represent within-category continuation, with elevated values (0.70–0.80) reflecting depth-first exploration patterns. These values are *affinities* (relative strengths) rather than probabilities, so rows need not sum to 1. At runtime, they serve as an interpretable prior that adapts online via the update rules in Equation 1 and Equation 2.

**Table 7: Initial heuristic Question-Affinity Graph weights. Rows indicate source categories; columns indicate target categories. Higher values indicate more likely transitions. Diagonal entries (bold) represent within-category continuation.**

| FROM \ TO | 1. Overview | 2. Node | 3. Link | 4. Path | 5. Community |
|---|---|---|---|---|---|
| **1. Overview** | **0.70** | 0.60 | 0.55 | 0.50 | 0.65 |
| **2. Node** | 0.30 | **0.75** | 0.50 | 0.75 | 0.55 |
| **3. Link** | 0.25 | 0.45 | **0.70** | 0.65 | 0.45 |
| **4. Path** | 0.35 | 0.60 | 0.50 | **0.70** | 0.45 |
| **5. Community** | 0.45 | 0.60 | 0.40 | 0.45 | **0.75** |
| **6. Motif** | 0.30 | 0.50 | 0.40 | 0.40 | 0.60 |
| **7. Attribute** | 0.30 | 0.50 | 0.35 | 0.35 | 0.55 |
| **8. Attr+Topo** | 0.30 | 0.55 | 0.40 | 0.35 | 0.65 |
| **9. Anomaly** | 0.40 | 0.55 | 0.45 | 0.40 | 0.50 |
| **10. What-if** | 0.35 | 0.50 | 0.45 | 0.40 | 0.45 |

| FROM \ TO | 6. Motif | 7. Attribute | 8. Attr + Topo | 9. Anomaly | 10. What-if |
|---|---|---|---|---|---|
| **1. Overview** | 0.40 | 0.45 | 0.35 | 0.30 | 0.30 |
| **2. Node** | 0.45 | 0.50 | 0.65 | 0.40 | 0.70 |
| **3. Link** | 0.40 | 0.40 | 0.55 | 0.35 | 0.75 |
| **4. Path** | 0.40 | 0.35 | 0.40 | 0.30 | 0.65 |
| **5. Community** | 0.65 | 0.60 | 0.70 | 0.50 | 0.50 |
| **6. Motif** | **0.70** | 0.35 | 0.45 | 0.40 | 0.60 |
| **7. Attribute** | 0.35 | **0.80** | 0.75 | 0.45 | 0.40 |
| **8. Attr+Topo** | 0.40 | 0.55 | **0.75** | 0.50 | 0.60 |
| **9. Anomaly** | 0.45 | 0.40 | 0.45 | **0.80** | 0.70 |
| **10. What-if** | 0.40 | 0.35 | 0.40 | 0.35 | **0.75** |

# D System Prompts

To support reproducibility, we present the core prompt templates used in NetworkCanvas's LLM integration module (subsection 6.1). These prompts were iteratively refined to balance instruction adherence, domain accuracy, and output consistency. Our prompt engineering strategy incorporates four key mechanisms:

- **Context Preservation**: Leveraging the *RunnableWithMessageHistory* wrapper from LangChain to maintain conversational state across multi-turn analysis sessions.
- **Domain Knowledge Injection**: Explicitly embedding the complete 10-category taxonomy (including 4 entity types and 48 sub-insight types from Appendix A) into the system prompt to ground LLM outputs in established network analysis concepts.
- **Few-Shot Learning**: Providing concrete examples (query → classification) to guide the model towards precise intent recognition and structural compliance.
- **Output Constraints**: Enforcing strict formatting requirements to ensure reliable parsing by downstream components.

The following sections detail the prompt templates for classification, recommendation, and domain-specific context injection.

## D.1 Classification Assistant

The classification assistant routes natural language queries to predefined analytical categories. The system prompt establishes the expert role and constrains outputs to the taxonomy schema:

```
ROLE: Expert classifier for network analysis questions.

TASK: Map the user's question to our predefined taxonomy.

DECISION STEPS (internal):
1. Infer the primary entity: Network | Node | Link | Group
2. Select the single best matching category (1--10)
3. Select the best-matching insight type within that category

PRINCIPLES:
- Output exactly one classification line; no rationale, no
  extra text.
- If the question is ambiguous, choose the closest match by
  primary intent.
- Do not invent new categories or insight types beyond the
  provided taxonomy.
```

## D.2 Classification Template Structure

The classification prompt routes user queries to the appropriate analytical category. It includes the full taxonomy definition to minimize hallucination.

```
IDENTITY: {identity}

TAXONOMY: QUESTION CATEGORIES AND INSIGHT TYPES (abbreviated)
1. Network Overview (Entity: Network)
- Description: Questions about global structural properties.
- Insight Types (examples):
  - Global Summary: Provide a synopsis of topology and key metrics.
    Example question: "Present the email network: size, density,
    connectivity."
    Expected label: 1. Network Overview - Global Summary
  - Count Entities: Compute total node and link counts.
  - ...

2. Node Analysis (Entity: Node)
- Description: Questions about specific node properties.
- Insight Types: ...

[Categories 3-10 follow the same structure]

INPUT: One natural-language question.

OUTPUT (single line only):
"[Number]. [Category] - [Insight Type]"
Example: "2. Node Analysis - Rank by Centrality"

INSTRUCTION:
Classify this question (single-line output only): {question}
```

## D.3 Recommendation Assistant

The recommendation assistant generates contextually relevant analytical questions based on the user's current exploration state:

```
ROLE: Network analysis recommendation expert.

EXPERTISE: Graph theory, social network analysis, complex
systems, data analytics.

TASK: Generate high-quality, actionable analysis questions that
deepen the user's exploration.

PRINCIPLES:
- Be specific and operational (what to compute/compare/filter,
  on which entity).
- Prefer measurable attributes/metrics; avoid vague or rhetorical
  questions.
- Avoid hallucinating dataset fields: only use attribute present
  in the provided context.
- Keep phrasing professional, concise, and user-facing.
```

## D.4 Recommendation Template Structure

This prompt guides the generation of contextually relevant analytical questions (recommendations) to deepen the user's analysis.

```
IDENTITY: {identity}

TASK: Generate exactly 3 professional network analysis questions,
grounded in the current exploration state.

INPUTS:
- Category: {category}
- Insight options: {option}
- Current selection: {selection}
- Full taxonomy (categories + insight types): {taxonomy}
- Domain context (dataset schema + terminology): {context}

RULES:
- Each question MUST map to one taxonomy category and one insight
  type (use exact names).
- Output 3 distinct recommendations (do not repeat the same insight
  type verbatim).
- Questions MUST be actionable (compute, compare, rank, filter,
  detect, explain).
- Use only attributes/fields present in {context}. If a threshold
  is appropriate, make it realistic.
- Prefer domain terms from {context} (e.g., "intersection",
  "road segment") over abstract "node/link".

OUTPUT FORMAT (repeat 3 times, and nothing else):
[Number]. [Category] - [Insight Type]
Description: [One sentence describing the analytical operation]
Question: [One sentence phrased as a user question, ending with "?"]
```

## D.5 Domain Context Injection

To adapt to different datasets, we inject schema information and domain-specific descriptions. This example provides context for a transportation network dataset.

```
DATASET: Anaheim Transportation Network

SCHEMA:
- Nodes (Entity: Node): intersections with fields
  (id, longitude, latitude)
- Links (Entity: Link): road segments with fields:
  - source, target: connected intersections
  - capacity: maximum vehicles/hour
  - length, free_flow_time: distance and time metrics
  - toll: cost for traversal
  - link_type: road classification

SCALE (reference):
| City    | Zones | Nodes | Links |
|---------|-------|-------|-------|
| Anaheim | 38    | 180   | 914   |

EXAMPLES:
- Example node: {id: 1, lon: -117.88, lat: 33.87}
- Example link: {source: 1, target: 117,
  capacity: 9000, toll: 0, speed: 4842}

GUIDELINES:
- Use transportation terminology (intersections, road segments,
  routes, congestion).
- Keep questions clear and concrete; reference plausible traffic
  scenarios.
- When using thresholds, pick realistic values consistent with
  the schema.
```