Visual Analytics for Temporal Hypergraph Model Exploration

Maximilian T. Fischer, Devanshu Arya, Dirk Streeb, Daniel Seebacher,
Daniel A. Keim, and Marcel Worring

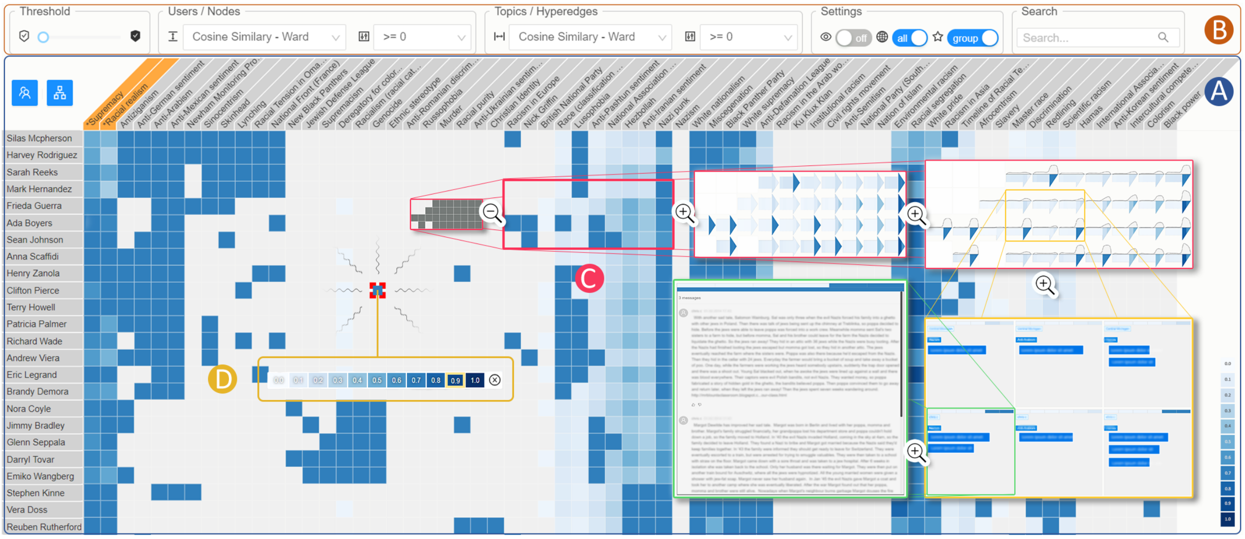


Fig. 1. HYPER-MATRIX, a novel approach to explore and refine temporal hypergraph models using visual analytics. The interactive multi-level matrix-based visualization Ⓐ enables the inspection of the model, together with the upper interface Ⓑ. The main area shows the second semantic zoom level applied to an obfuscated real-world dataset in criminal investigations, while the five insets Ⓒ show the other drill-down levels for exploration. The technique allows to interactively Ⓓ contribute domain knowledge, the resulting implications have ripple effects on the whole machine learning model, thereby refining it.

Abstract— Many processes, from gene interaction in biology to computer networks to social media, can be modeled more precisely as temporal hypergraphs than by regular graphs. This is because hypergraphs generalize graphs by extending edges to connect any number of vertices, allowing complex relationships to be described more accurately and predict their behavior over time. However, the interactive exploration and seamless refinement of such hypergraph-based prediction models still pose a major challenge. We contribute HYPER-MATRIX, a novel visual analytics technique that addresses this challenge through a tight coupling between machine-learning and interactive visualizations. In particular, the technique incorporates a geometric deep learning model as a blueprint for problem-specific models while integrating visualizations for graph-based and category-based data with a novel combination of interactions for an effective user-driven exploration of hypergraph models. To eliminate demanding context switches and ensure scalability, our matrix-based visualization provides drill-down capabilities across multiple levels of semantic zoom, from an overview of model predictions down to the content. We facilitate a focused analysis of relevant connections and groups based on interactive user-steering for filtering and search tasks, a dynamically modifiable partition hierarchy, various matrix reordering techniques, and interactive model feedback. We evaluate our technique in a case study and through formative evaluation with law enforcement experts using real-world internet forum communication data. The results show that our approach surpasses existing solutions in terms of scalability and applicability, enables the incorporation of domain knowledge, and allows for fast search-space traversal. With the proposed technique, we pave the way for the visual analytics of temporal hypergraphs in a wide variety of domains.

Index Terms—Hypergraph, communication analysis, geometric deep learning, semantic zoom, matrix ordering, visual analytics.

1 INTRODUCTION

A significant volume of real-world data consists of entities and their relationships and can accordingly be modeled mathematically using graph-based approaches. Such approaches are widely applied in many domains, ranging from natural and social sciences to engineering and business. Examples include modeling biological and chemical processes like protein-protein interactions [34], relationships in computer [43] as well as human communication networks [33], or knowledge network exploration in business processes [21]. Whereas static graphs can represent the fixed relationships between entities, using an undirected or directed graph as a model, many of the examples presented above are more accurately described as processes with complex interrelations that may change or evolve. Here, geometric deep learning methods together with interactive visualization can help to more accurately model, predict, and explore the model evolution. Considering, for example, conversations, a topic is a time-dependent grouping encompassing users, which cannot be described using a static graph. This evolution of relations should be modeled by dynamic networks. Compared to regular graphs, using edges or separate node types, such modeling

- *Maximilian T. Fischer, Dirk Streeb, Daniel Seebacher, and Daniel A. Keim are with the University of Konstanz, Germany. E-mail: {max.fischer, dirk.streeb, daniel.seebacher, keim}@uni-konstanz.de.*
- *Devanshu Arya, and Marcel Worring are with the University of Amsterdam, The Netherlands. E-mail: {d.arya, m.worring}@uva.nl.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

often reflects the actual process more accurately. Dynamic networks are, however, more challenging to model and have traditionally been modeled as regular, undirected graphs, mainly due to computational and visualization limitations. In recent years, modeling has extended to dynamic networks [26], but some limitations remain.

Consequently, one can take a step further and use temporal hypergraphs. Hypergraphs generalize graphs by extending edges to connect any number of vertices, allowing complex relationships to be described more accurately [36] while reducing ambiguity and network inflation. Utilizing temporal hypergraph prediction models, however, introduces its own set of challenges.

First, as the model structure is more complex, it is relevant how the information is communicated to the analyst through visualization (cf. [20]) and how domain knowledge feedback is incorporated. Static hypergraphs can be considered as standard sets, with different visualizations available [2]. Temporal hypergraphs, meanwhile, add a time-dependent evolution, making it harder to convey the relevant information meaningfully.

Secondly, many traditional graph-based concepts cannot directly be applied to hypergraphs. Hyperedges, as arbitrary sized sets of connected nodes, add another order of complexity. In previous works [6,7], we presented how geometric deep learning can be applied to hypergraphs and showed how this method could be leveraged to predict behavioral patterns in social media hypergraph models.

Consequently, the incorporation of machine learning techniques into an interactive model to more accurately predict changes in the hypergraph due to changes in the data introduces new problems. While deep learning avoids assiduous manual feature engineering and algorithm design, it reduces explainability and accountability of the results. Domain experts usually have some domain-specific intuition—a mental model and structure—about inherent and implicit relations and groupings not available in the data, enabling them to judge the plausibility of hypotheses and to steer the exploration. Yet, they face difficulties articulating their domain knowledge through machine learning into the predictions and tracing its influence. This holds especially for very complex models, like temporal hypergraphs. The knowledge formalization requires a very detailed a priori understanding of the problem by domain experts, which is not always available. For the same reason, it is challenging to capture the knowledge independently of the model without rapid, iterative feedback. Hence, the machine learning outcome often correlates strongly with the adequacy of the initial problem modeling and the quality of the training data, while domain expertise and domain knowledge are frequently not leveraged to their full potential.

To address these issues, we present HYPER-MATRIX, making the following contributions:

- A novel, interactive framework for temporal hypergraph exploration through the use of semantic zooming relying on a multi-level matrix-based approach and various exploration concepts.
- The extension of a geometric machine learning architecture [6,7] with a relevance feedback model.
- A tight coupling between the visualization and the machine learning relevance feedback model for evaluation and seamless refinement, offering the integration of domain knowledge and making the corresponding model changes visually transparent.
- One case study describing an application of the technique to the law enforcement field.
- A formative evaluation with law enforcement experts using real-world communication data, demonstrating that our technique surpasses existing solutions, enabling the effective analysis of large amounts of information in a targeted way.

Our approach bridges the gap between visual exploration and separate model training, allowing domain experts to enhance the machine learning predictions with implicit domain knowledge in the same step as evaluating and exploring the temporal hypergraph model predictions.

2 RELATED WORK

This research is an entry into the interactive temporal hypergraph model exploration in the context of explainable support by machine learning. In the literature hypergraphs are studied from both a visualization as well as a machine learning perspective. In the following discussion, we adhere to the same distinction and relate our work to the visualization of temporal hypergraphs as well as their application in machine learning.

2.1 Visualization of Hypergraphs

We first shortly discuss the situation for (static) hypergraphs as well as dynamic graphs, before looking at temporal hypergraphs. Hypergraphs can be considered as a set of sets. The survey on set visualizations by Alsallakh et al. [2] shows that several visualizations are applicable to hypergraphs. Hypergraphs are often drawn as regular graph networks or bipartite networks. When making their dimensionality explicit, they can be drawn as subsets—like Venn diagrams or radial sets—or in node-link form [42], using colored hulls or other, specifically adapted approaches [31]. A third possibility is to use a matrix-based approach, which improves scalability [25]. Subsets and node-link diagrams suffer from limited scalability, quickly leading to occlusion and clutter. Bound in the number of visual attributes they can employ, these techniques typically reach their constraints in the order of one or two dozens of hyperedges [40]. Further, they are difficult to extend with a temporal component, having already used up most visual attributes.

In comparison to set based approaches, dynamic graphs change over time, leaving the choice [12] between employing animation or an additional timeline component. The former puts significant strain on the mental map when many connections change, while the latter is limited by the available screen space in the number of discrete timesteps it can show. The survey [12] also points out that node-link diagrams remain the most commonly used type of visualization. However, these approaches mostly lack the extendability to hypergraphs.

When studying temporal hypergraphs, the issues arising from the dimensionality and the temporal nature all build up. Indeed, there is almost no prior work on the visualization of temporal hypergraphs specifically. Two notable exceptions exist, which allow visualizing—but not modifying or refining—temporal hypergraphs: First, the recent works by Valdivia et al. [38–40]. Their visualization approach is also shown later in Figure 6c as part of the case study. Second, the previous work by Streeb et al. [37] introduces an in-line visualization of the temporal evolution. Valdivia et al. begin to tackle the research gap by proposing PAOHvis, thereby claiming to provide the "first [...] highly readable representation of dynamic hypergraphs". While this is a strong claim to make, the literature review showed a broad diversity between the approaches, but none—except the two mentioned above—is directly suitable for temporal hypergraph visualization, supporting this conclusion. Utilizing the previously discussed approaches as substitutes for a tailored visualization often does not adequately leverage the additional information available with temporal hypergraphs and does not address the tasks that come with hypergraph topology and evolution. For those, we refer to Section 2.3. Shortcomings in existing approaches include, for example, Streeb et al. providing only the prediction abstraction level in their visual interface (cf. Level 3 in Section 4.1). Similarly, this is true for Valdivia, although they support coloring by a group. This can lead to information overload, as filtering using thresholds is the only way to reduce the information. In contrast, usage of semantic zoom enables an exploration of the complete hypergraph (cf. Section 4.1) without the need to preliminary apply filters while enabling tailored visualizations showing detailed information when focusing on different abstraction levels. Prominent examples of matrix-based visualizations are the Zoomable Adjacency Matrix Explorer [17] that enables users to zoom and pan with interactive performance from an overview to the most detailed views and the visual analysis system of Behrisch et al. [14]. It features a flexible semantic zoom to navigate through sets of matrices at different levels of detail. Further, both Streeb and Valdivia, only support sorting by weights and average (cf. size ordering in Section 4.2), compared to our default matrix-based sorting, improving cluster identification. Significantly, all existing approaches aim at analyzing a fixed hypergraph model. None focus on interactively working *with* the model and iteratively improving it (cf. Sections 3.2 and 4.3).

At last, while not strictly related to the research on temporal hypergraphs *per se*, we want to mention approaches that are, at least partly, similar to ours, and also conventional tools so far applied in practice. Here we concentrate on how hypergraph-like data is handled in the law enforcement field, relevant for the case study and the evaluation through domain experts (see also Sections 5 and 6). The visual analysis of communication data—but without any hypergraph visualization or a tunable model—is not novel and has been researched both from the analytical side [30] as well as the visualization side [44]. Also, the idea of semantic zooming for matrix-like visualizations has been described previously [41], however, in a different way and in the area of software management. Further, it was also described how an overlay magic lens [19] can be used instead of zooming, to keep the context and allow for faster search space traversal from locations far apart, which we partly employ for the partition hierarchy (Section 4.2). In practice, for the law enforcement field, we found that data which benefits from a hypergraph modeling, like communication patterns or process analysis, is prevalent, but not supported by any system. Gephi [9] is sometimes used, but analysts often prefer Pajek [10, 11], as it supports larger networks. The most popular tool is IBM i2 Analyst's Notebook's [23] graph component due to the prevalence and familiarity in this domain.

2.2 Machine Learning for Hypergraph Models

Learning with hypergraphs was introduced by Zhou et al. [46] to model high-order correlations for semi-supervised classification and clustering. It generalizes the efficient methodology of spectral clustering to hypergraphs by proposing a label propagation method to minimize the differences in labels of vertices sharing the same hyperedge. The correlation among hyperedges was further explored by Hwang et al. [22], assuming that highly correlated hyperedges have similar weights. More recent works [15] concentrate on parametric learning of weights using propagation of node features across hyperedges [18, 45].

Understanding communication patterns of users on social networking sites has created opportunities for richer studies of social interactions and better prediction of behavioral patterns. In multimedia, link prediction on hypergraphs has been a popular topic of research in social network analysis. This includes predicting metadata information such as tags and groups for entities in social networks, e.g., images from Flickr [7], music recommendation by exploiting network proximity information of users in Last.fm [16] and predicting higher-order links (such as tweets with a specific hashtag) in Twitter [28]. Besides, hypergraph learning models are being used in multimodal data analysis to integrate complementary information from multiple modalities effectively. Liu et al. [29] proposed a multi-hypergraph learning method to handle incomplete multimodal data for disease diagnosis in neuroimaging and Arya et al. [5] proposed a framework to learn a compact representation for each modality in a multimodal hypergraph using a tensor-based representation. These works have shown the importance of hypergraph based learning for predicting implicit links within a network. However, none of these approaches pose an interactive learning formulation that can assimilate user feedback as an external source of information to either improve the predictive capability of a model or to even change the intrinsic properties such as learnable parameters of a model. In this work, we extend our previous work [6] on link prediction in communication networks capable of fine-tuning the trained model by incorporating external relevance feedbacks.

2.3 Tasks for Evaluation of Temporal Hypergraph Models

Tasks in temporal hypergraph analysis relate to dynamic networks and set comparisons. A task taxonomy of the former is provided in the survey by Beck et al. [12], and for the latter in the survey by Alsallakh et al. [3]. For temporal hypergraphs, in particular, the tasks sometimes substantially differ; for example, one being the analysis of changes of both connections and attributes over time. The proposed technique does not directly fit with any existing task taxonomy, positioning itself between disciplines [4]. For a discussion on existing taxonomies and their applicability to temporal hypergraphs, we refer to the existing work by Valdivia et al. [40] and summarize only the main aspects here. Our technique supports not all traditional tasks in set analysis [3], and in dynamic network analysis [1, 8, 24, 27], summarized in [12]. However, it provides support for several additional tasks relevant to our driving application. These include the clustering of related groups independently of their temporal connection, the inspection of shared attributes of connections, the following of temporal evolutions, while both retaining an overview and simultaneously being able to explore details. In short, the experts are interested in connectivity information involving both graph topology as well as attribute values, which can be separated between time ranges. One main requirement is the need to include external (domain) knowledge that is not directly available as raw data and includes conceptualized topics in line with their mental categorization. These tasks are not sufficiently described or supported by existing taxonomies, as they neglect the additional complexity incorporated by hypergraphs and the domain knowledge integration.

Given the sparse research in hypergraph visualization, it is unsurprising that there is no prior work on bridging both fields; this is the gap we aim to fill: offering a technique that addresses the shortcomings discussed above, enabling the exploration and refinement of hypergraph models using interactive visualization, closing the visual analytics loop.

3 Extension of Machine Learning to Hypergraphs

In the following two sections, we describe the overall workflow of our approach, shown in Figure 2. We begin with an exemplary description of one geometric deep learning model, adapted to a task relevant for our law enforcement domain experts: the temporal prediction and analysis of patterns in communication data. It acts as a blueprint for problem-specific temporal hypergraph models. In Section 4, we then discuss the interactive exploration using visual analytic principles.

3.1 Notation and Formulation of a Temporal Hypergraph

In set theory, an undirected hypergraph $H = (V, E)$ is defined as an ordered pair, where $V = \{v_1, .., v_n\}$ represents the n vertices (hypernodes) and subsets of these vertices $E = \{e_1, .., e_m\}$ constitute the m distinct hyperedges. H is represented by the incidence matrix $\mathbb{I}{=}|V|{\times}|E|$, with entries $i(v_j,e_k)$ 1 if $v_j{\in}e_k$ and 0 otherwise. We define the neighborhood of v_j as the set $\mathcal{N}(v_j)$ of nodes within the same hyperedges as v_j.

In adapting a generic temporal hypergraph model to our use case, we follow our previous work [6], representing the relationship between internet forum users and their behavioral characteristics (both "explicit" and "implicit"). The available metadata (in particular forum category) forms the explicit characteristic of a user, while their topics of discussion outline the implicit communication characteristic. Thereby, we construct two separate hypergraphs depicting the connection of users with these explicit and implicit behavioral characteristics. To model the temporal component, let us define a temporal hypergraph by $H_{[t]}$, at a given time t, where each user is represented as a node, and each type of explicit/implicit characteristic is represented as a separate hyperedge. We denote the explicit and implicit hypergraphs, at any given time t, by $H^0_{[t]}$ and $H'_{[t]}$, respectively. Consequently, in $H'_{[t]}$, each topic is depicted as a separate hyperedge and users (nodes) who adhere to a common topic of interest are connected by it. Thus, forecasting the evolution of users' topics of interest for time $t+1$ becomes equivalent to the task of finding new relations over the existing relations in hypergraph $H'_{[t]}$.

3.2 Relevance Feedback to Deep Learning Model

As indicated, the underlying model for forecasting future interests of internet forum users is based on predicting links in temporal hypergraphs. The task of link prediction on a hypergraph $H_{[t]}$ with a fixed set of edges E aims at updating the set e_k. This link prediction can be formulated as missing value imputation or a matrix completion task on $\mathbb{I}$. In the following, we extend our previous work [6, 7], to allow for the incorporation of feedback in Hyper-Matrix. Therefore, we first reconsider the module for training a geometric deep-learning model. Then, we formulate how feedback from the user can be employed to update the model.

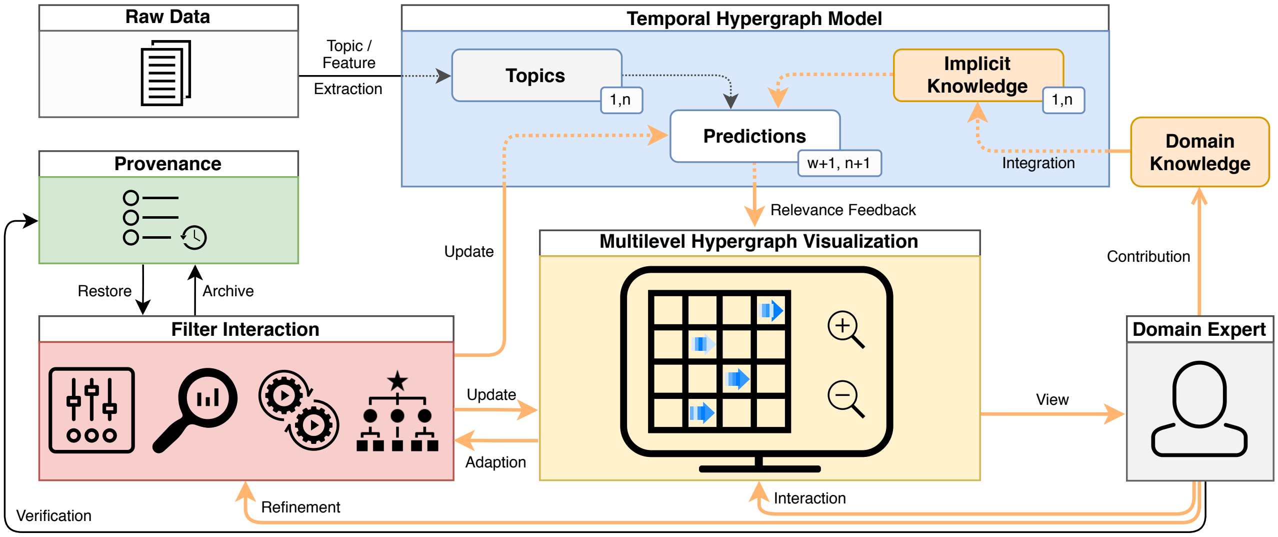


Fig. 2. High-level workflow of our technique, showcasing the main components and the interaction flow for the exploration and refinement of temporal hypergraph models, adapted to use case A in Section 5. The workflow begins with **raw data** 🗎 extraction and the generation of a **temporal hypergraph model**. The model state is visualized using a matrix-based **multilevel hypergraph visualization** ▦ , allowing for various exploration and **filter schemes**, including search 🔎 and filters ▦ , a dynamically modifiable partition hierarchy ⛼ , and matrix-reordering techniques ⚙ . The **domain expert** 👤 can interact with the model by either refining the filter schemes or by contributing domain knowledge, which both update the model. The model feedback can then be explored and accepted, closing the **visual analytics loop** →. The chronology of interactions and contributions are available for recovery or verification as a **provenance history** ☰ , facilitating accountability.

Training Module Let $\mathbb{I}_{[t]}$ denote the incidence matrix of $H'_{[t]}$ at time t which we can factorize as $\mathbb{I}_{[t]}=X_tY_t^T$ with X_t and Y_t the row and column matrices, respectively. Hypergraph $H^0_{[t]}$ will be utilized as an auxiliary set of explicit information between users for predicting links in the implicit hypergraph $H'_{[t]}$. The information in $H^0_{[t]}$ is encoded by extracting its Laplacian denoted by Δ_0. The Laplacian Δ_0 gives a measure for the relatedness between any pair of users [16]. Using such a similarity measure can significantly enhance the user-topic link prediction outcomes by reducing extraneous noise and thus smoothing the model output.

To train the model, we employ a semi-supervised learning setup, hence the predictive loss is backpropagated by using a small set (around 5–8 %) of known links in $H'_{[t+1]}$. These known links create an upper bound for the number of timesteps the model can predict in $\hat{\mathbb{I}}_{[t+1]}$. Details can be found in [6, 7]. For training, we take the incidence matrix $\mathbb{I}_{[t]}$ at time t and use the hypergraph link prediction model $\mathbb{H}_{GDL}$ to learn the best parameter set $\Phi[t]$ for predicting the incidence matrix $\mathbb{I}_{[t+1]}$ at time $t+1$:

$$\hat{\mathbb{I}}_{[t+1]}, \Phi_{[t]} = \mathbb{H}_{\text{GDL}}(\mathbb{I}_{[t]}, \Delta_0) \tag{1}$$

Feedback Module In order to integrate domain knowledge into the underlying model, we propose a novel interactive learning formulation to incorporate feedback from the domain expert. These feedbacks are assumed to contain definitive implicit information about the topic of interest for certain users in the dataset. Instead of just updating the information by directly changing the topic (hyperedge) of the respective users (nodes), these feedbacks should also create a “ripple effect” on the overall connections in the hypergraph $H'_{[t]}$. That is, if the feedback $f_{[t]}$ at time t involving the single user (u_j) denoted by node v_j in the hypergraph $H'_{[t]}$, then incorporating $f[t]$ will entail a twofold operation: *1. Update*: Topics for user u_j are updated, i.e., add/remove v_j to/from the respective hyperedges $E=\{e_1,..,e_m\}$ corresponding to $f_{[t]}$. *2. Predict*: Change topics for users in close communication with u_j based on their relatedness to u_j, i.e., re-calculate the connection strength for vertices in $\mathcal{N}(v_j)$ with the hyperedges $E=\{e_1,..,e_m\}$. The first operation is a straightforward updating of the matrix $\mathbb{I}_{[t+1]}$ by updating new values corresponding to nodes and edges suggested in the feedbacks $f_{[t]}$. The change in the neighborhood connections are calculated by using the updated matrix $\mathbb{I}_{[t+1]}+f_{[t]}$ as input to our link prediction model $\mathbb{H}_{\text{GDL}}$. However, in the feedback module, instead of learning parameters through an iterative process, the learned parameters $\Phi_{[t]}$ are used as initialization of the already trained model $\mathbb{H}_{\text{GDL}}$. This ensures the model converges in far less time after incorporating the feedbacks $f_{[t]}$ than when learned from scratch. The following equation shows the representation of the feedback module in symbolic form:

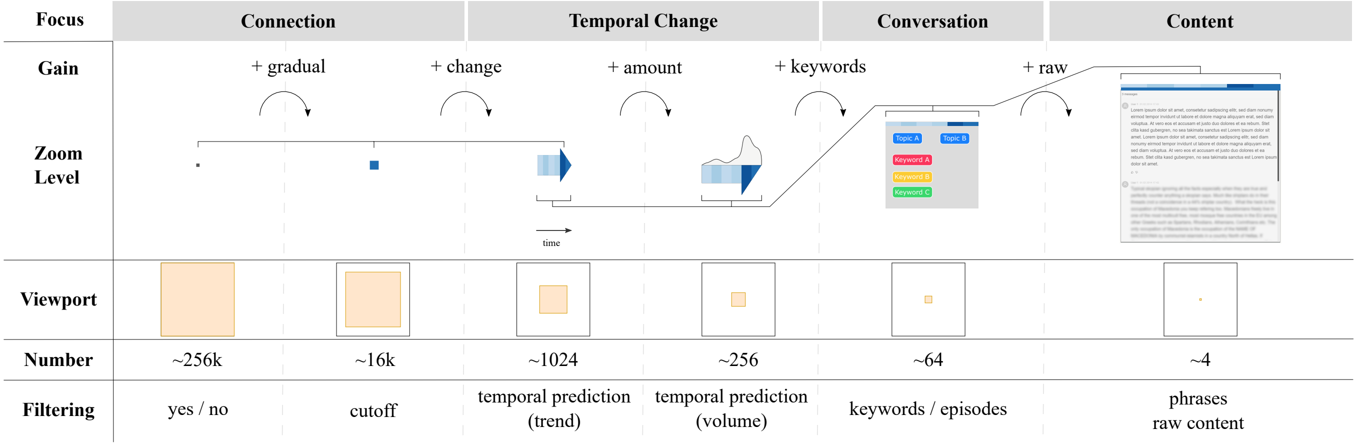
$$\hat{\mathbb{I}}_{[t+1]} = \mathbb{H}_{\text{GDL}}(\mathbb{I}_{[t+1]} + \mathsf{f}_{[t]}, \Delta_0, \Phi_{[t]}) \tag{2}$$

4 Interactive Hypergraph Model Exploration

In this section, we focus on the visualization and interaction with the temporal hypergraph model, providing a tight coupling between the data manipulation and display (see Figure 2). We begin by describing how the model state can be depicted using a matrix-based visualization that provides drill-down capabilities across multiple levels via semantic zoom. Drill-down is thereby defined as the seamless zooming through the different levels during exploratory analysis, starting from a general overview to increasingly more focused and detailed information, as highlighted in Figure 3. To facilitate the interactive exploration, we present user-steering based on classical filters for standard search tasks, a dynamically modifiable partition hierarchy to include user-based structuring, and various matrix reordering techniques for the focused analysis of connections and groups. We then specify the interactions that allow domain knowledge to be incorporated into the machine learning model via relevance feedback and highlight how the updated predictions can be reflected in the existing visualization. This workflow facilitates the explainability of the underlying model, thus enabling the domain experts to provide more meaningful feedback. Finally, we describe how all interactions, domain knowledge input, and model output are stored in a provenance history, providing accountability and making the decision-making processes more transparent.

4.1 Model Visualization ▦

As discussed above, the complexity of temporal hypergraphs makes them difficult to visualize. Hence, we propose a multi-level matrix-based approach, specifically tailored to the hyper-dimensionality as well as the temporal component. The visualization (see Figure 1) consists of a menu bar on top, controlling the interaction concepts discussed later, and, for the main part, a matrix-like viewport, showing nodes as rows and hyperedges as columns, with corresponding row



Focus	Connection		Temporal Change		Conversation	Content
Number	~256k	~16k	~1024	~256	~64	~4
Filtering	yes / no	cutoff	temporal prediction (trend)	temporal prediction (volume)	keywords / episodes	phrases raw content

Fig. 3. Semantic zoom levels and the different filtering levels (cf. Section 4.1). At each zoom step, the analyst gains another type of information about the model, filtering a different layer of complexity. As the focus becomes more detailed, the visualization takes up more space (zoom level and viewport as shown not to scale), while the number of visible entities decreases accordingly. The temporal predictions are shown in different forms throughout all levels (see fine grey line), with the detailed temporal evolution first shown in Level 3 and continuing down to Level 6.

and column headers. This viewport provides freely pan-able and zoomable drill-down capabilities across six levels of semantic zoom, shown in Figure 3, increasing or decreasing the information detail: from an overview of model predictions down to contents. For this purpose, we use three different level types: cells, arrows, and content boxes. Colored cell visualizations are used in Levels 1 and 2. An arrow-like representation reflecting a timeline is used in Levels 3 and 4. The base of the arrow represents the past, while the head reflects predictions. As the predictions become more uncertain with time, the arrowhead becomes smaller, reflecting the increased uncertainty and thus the decreased relevancy of the prediction. Levels 5 and 6 add text-based elements like keywords or raw content. Level 3 and beyond all contain the temporal aspect.

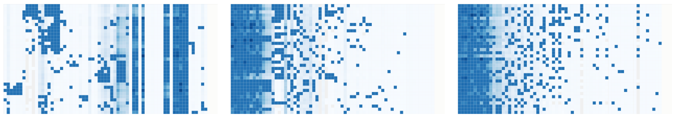
The visualization depends on the zoom state of the viewport. During drill-down, the focus shifts from a general structure overview over the temporal evolution to the raw content, providing the expert with more and more detailed information. Before we start with the description of this process, we define some necessary terms. As the feedback model outputs probabilities for the connections (see Section 3.2), gradual differences can be analyzed. When setting a minimum threshold for a connection to be meaningful, this allows for a *binary choice*. Showing a color encoding of the connection strength allows for a more expressive representation of the *gradual differences*. Setting a *cutoff* threshold can still be used to avoid cluttering with low-probability entries. The drill-down shifts the focus of the analysis. It starts at the (binary) connectivity information, extends to gradual connection strength (Level 2), to the temporal change represented as an arrow (Level 3), to the temporal change encoded using position instead of only color (Level 4), then to information summarizing the underlying content for the predictions, in this case, keywords (Level 5), and, at last, to the raw data (Level 6). The design choice for an arrow glyph representation in Levels 3 and 4 is based on five reasons: (1) The principal idea of an arrow glyph was previously published [37] and found to be beneficial. Then, (2) given the target audience, a representation as an *arrow of time* is closely related to everyday experience. Further, (3) the separation into arrow base and head allows a clear distinction between past data and model predictions, which is very important for the target audience. The arrowhead also allows to visually reflect the decreasing prediction accuracy by becoming smaller. In terms of (4) visual advantages, an arrow provides a distinct shape, while, e.g., a cell is easily perceived to merge with neighboring cells, which is undesired. The choice also comes with disadvantages, introducing white space and can sometimes lead to distracting patterns. Finally, (5) a design study on combining timeline and graph visualization by Saraiya et al. [35] shows that our approach—simultaneously overlaying the timeline—is best suited for detecting outliers. This is one of the main tasks for these levels, given the focus on change. The study also supports the design choice of showing only a single timestep in Levels 1 and 2, as the focus is on the topological structure. However, different visual representations like horizon graphs might be better suited when focusing on a continuous analysis. The seamless changes between levels speed up navigating through large models while eliminating demanding context switches. Moreover, at each step, the information becomes more complex, requiring more screen space to visualize. For a regular HD screen, we give rough guidance ($\mathcal{O}(n)$) on the number of elements that can be usefully shown on-screen, amounting to around 256k grid cells of connectivity information and around four for the raw content.

4.2 Interactive Exploration and Drill-Down

To facilitate the interactive exploration, we contribute a user-steering based on classical concepts and filters for standard search tasks, a dynamically modifiable partition hierarchy to include user-controlled structuring and various matrix reordering techniques for the focused analysis of connections and groups. All these interactions concepts are reactive, and the visualization can smoothly and instantly update (< 100 ms), except for the domain knowledge integration in Section 4.3.

Interaction and Filter Concepts Standard methods available in an interactive visualization are included, like (1) highlighting selected rows or columns, (2) highlighting hovered cells, (3) tooltip-based menus, (4) marking (i.e., starring) individual entries to highlight them for tracking and follow-up, (5) adding textual notes, and (6) showing additional meta-information. Modal views allow to (7) control the partition hierarchy (see details below), while setting an (8) overall cutoff threshold allows controlling the confidence threshold of the underlying model. A (9) global search function provides the ability to search for node- and edge information as well as content and highlights the matching components. At last, the menu bar allows (10) controlling the matrix reordering (see detail below).

Dynamically Modifiable Partition Hierarchy To allow domain experts to articulate their mental categorization to the model, the experts can create (nested) groups of different nodes or hyperedges, creating hierarchies. The nodes or hyperedges hereby relate to the leaves of the dendrogram. The groups can be expanded or contracted either directly from the node or hyperedge headers, visually indicated by color, or by editing them inside the partition hierarchy viewer in a modal overlay. The viewer shows a dendrogram-based representation with freely reorderable entries. Each branch of this dendrogram can be independently collapsed or expanded, i.e., the abstraction level is local to each branch and not globally set. For example, it is possible to



(a) Multi-step hierarchy (b) Size (c) First occurrence

Fig. 4. Comparison of different matrix reordering techniques to facilitate the detection of similar groups and connections. Compared to the unordered state and the slightly improved ordering by size, the adoption of a default multi-step, dendrogram-based reordering, modified and adapted from [13], enhances the clustering by similarity.

collapse a large, uninteresting sub-branch, including the nested nodes it contains, while simultaneously having one branch fully expanded and another only up to the penultimate level. This is also independent of the overall visualization level, similar in concept to multiple fixed magic lenses, visually supporting different analysis paths. The hierarchy allows, for example, to group complementing entities together, to build meta-entities, and even hierarchies of entities.

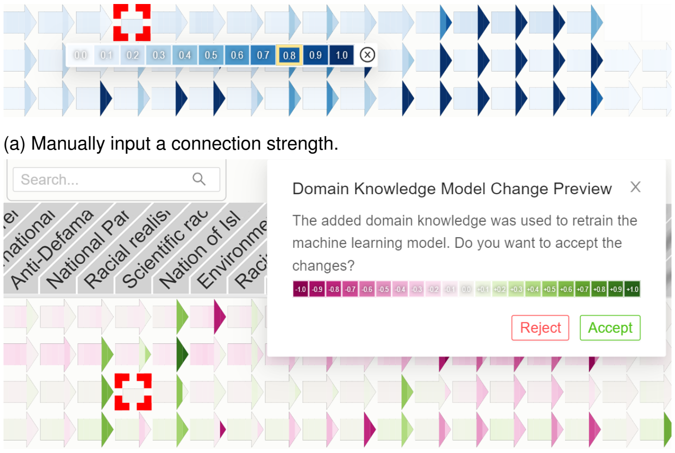
Matrix Reordering and Sorting To support the tasks relevant for our driving application (see Section 2.3), a matrix reordering is desirable such that related users and topics appear close to each other. Due to the independent and often conflicting interpretations of both axes and the sparseness of the underlying matrix, the direct application of standard 2D numeric sorting algorithms (e.g., Multi-scale-, Chen-, or Travelling salesman problem ordering) [13] often leads to unsatisfactory results, as they are mainly applicable to pairwise comparison matrices.

As part of the visualization, we offer three main different reordering strategies, as shown in Figure 4: (a) matrix-reordering (default), (b) sorted by size (connectivity), (c) first occurrence (original). The reordering is applied individually for each axis, as the requirement may differentiate between search tasks, not always favoring a block-like clustering. It also provides more flexibility for adopting other sorting methods in domain adaptions of our technique. The underlying sorting principles build upon a dendrogram-based serial matrix reordering discussed by Behrisch et al. [13]. It forms a multi-step process, combining the sorting of node and edge similarity vectors. Supported dendrogram methods are ward-, single-, average-, and complete linkage, combined with any pairwise distance function like Euclidean, cosine, or Jaccard. We refrain from discussing individual choices, which can vary strongly on domain adaption. For our case study, the Jaccard and cosine distance provide consistent results.

4.3 Visual Analytics for Model Updates

To increase the traceability of domain knowledge integration and explainability of the resulting model changes, we propose an interactive change feedback visualization, that seamlessly integrates with our visualization. The two-step process is shown in Figure 5. An expert can integrate domain knowledge by selecting a cell and setting a new connection strength (Figure 5a), thereby complement missing or override model *input* data. This input is used to partly retrain the model and refine its predictions as described in Section 3.2, leading to a ripple effect. Thereby, the model has prediction authority, i.e., the user cannot manually fix the ultimate output to guarantee model authenticity. A spinner indicates the few seconds long operation. The resulting *changes* are displayed inside the same view (Figure 5b). A diverging color scale is used, showing changes instead of predictions. Through two visually distinct scales, it is immediately apparent if predictions or changes in the predictions are shown. The view integration allows for consistency, reducing the mental workload, and improving mental mapping.

Changes can be inspected on all levels of the visualization. The exploration is *not* restricted to just the current viewport, finding even weak connections. Change detection is facilitated, allowing rejection if deemed implausible or acceptance if convincing, enabling the followup of multiple analysis paths. By iteratively and interactively queering the model and see how it responds to domain knowledge integration, experts can discern better how connections and processes in the model are related, improving understanding and increasing explainability.



(a) Manually input a connection strength.

Search...

Domain Knowledge Model Change Preview

The added domain knowledge was used to retrain the machine learning model. Do you want to accept the changes?

(b) Resulting changes in the model predictions.

Fig. 5. Resulting changes in the model prediction (ripple effect) from an input are visualized by a diverging color scale (from negative to no to positive change). They can be explored and rejected or accepted. This allows for model verification and multiple, different analysis paths.

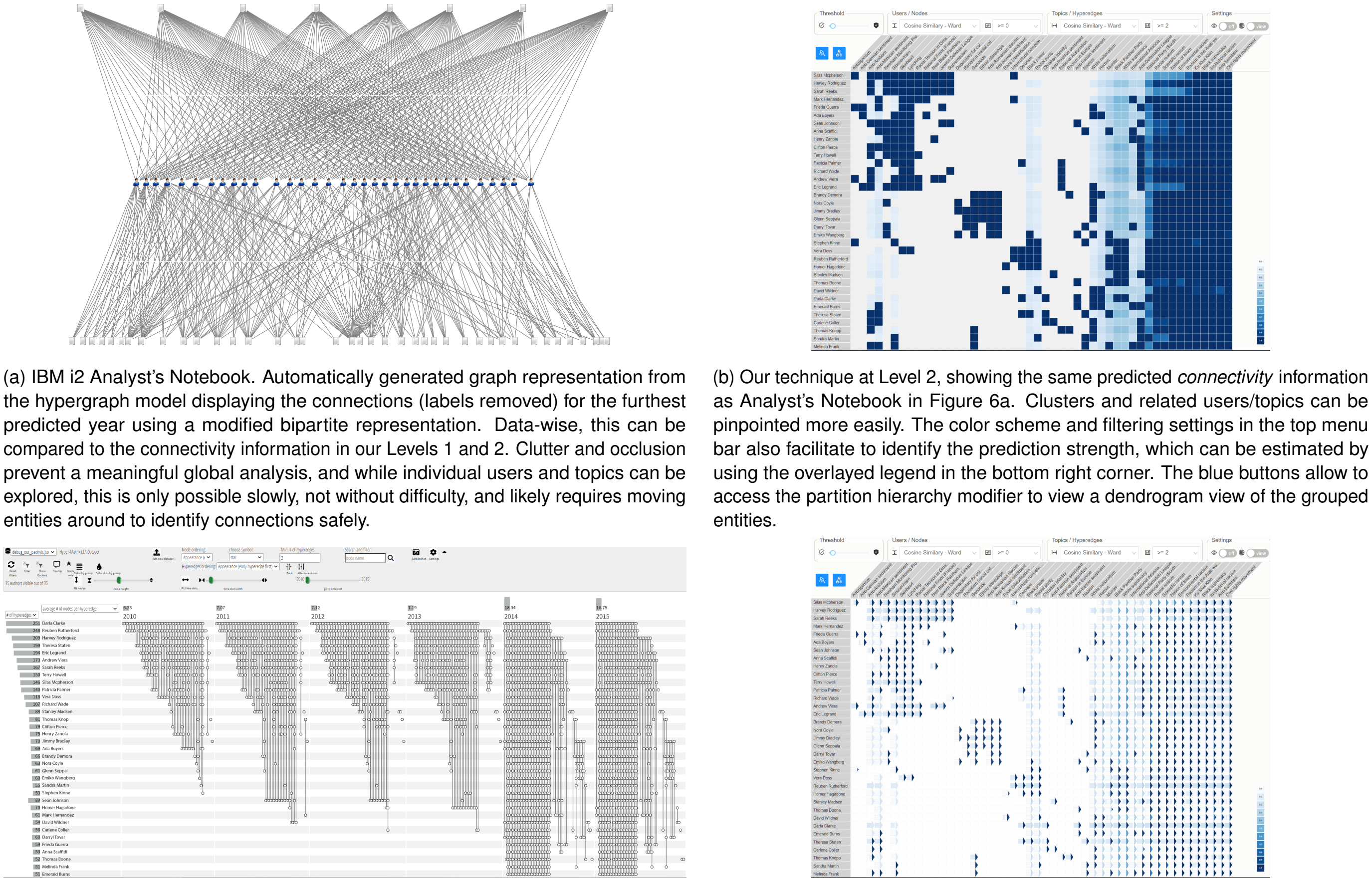
Experts in many applications are interested in their analytical progress and must reproducibly document the steps. We address this by a re-loadable provenance, storing the interaction sequence, domain knowledge input, model output, and fixed RNG seeds. This allows for inspection, verification, and traceability while providing accountability and making decision processes transparent. The provenance history allows undoing analysis steps, preventing dead-ends, revisiting and explaining past steps, but also bridging off to diverging analysis trails.

5 Case Study: Internet Forum Communication Data

To demonstrate the visual exploration of temporal hypergraph models in HYPER-MATRIX, we conduct a case study, showing the applicability of our technique and improvements compared to existing approaches.

The communication data was collected from an internet forum well-known to law enforcement. It contains 335 188 text posts from 4904 users. We pre-processed the data using standard NLP methods to extract 158 topics, based on a domain-specific ontology. As described in Section 3, users are associated with nodes and topics become dynamic hyperedges. To allow for a reasonable side-by-side comparison with the existing approaches, shown in Figure 6, we had to restrict to a subset, consisting of 35 users, 65 topics, and six timesteps. This is around four times more than conventional approaches are designed for. We confirmed that our prototype works for significantly larger networks (cf. Section 7). Our prediction model is fed with four years (timesteps) of historical data and then predicts the evolution of the next two years as two timesteps. Almost any real-world data is noisy and may miss some relationships. Consequently, some of the conclusions drawn here may be inaccurate. However, we focus on demonstrating the concepts and benefits of the visual analysis process HYPER-MATRIX provides.

The task we want to focus on in this case study is the identification of related groups and missing links, common in criminal investigations. To identify users discussing the same topics and topics discussed by the same group, the matrix reordering and connectivity information in Level 2 can be used to see structures, as shown in Figure 6b. Their spatial closeness acts as primary identification criterion, as similar row/column vectors are grouped closely. From this, their spatial closeness, describing the multi-step alignment, supports discovering related users or topics discussed simultaneously, but also latent connections. Distinct orderings can be applied separately to nodes and hyperedges, for example, to either favor overall similarity (cosine) or matching parts (Jaccard). For other requirements, it is also possible to include different metrics. To reduce noise and exclude weak connections, the top menu allows to set a threshold for the connection strength for historical and



(a) IBM i2 Analyst's Notebook. Automatically generated graph representation from the hypergraph model displaying the connections (labels removed) for the furthest predicted year using a modified bipartite representation. Data-wise, this can be compared to the connectivity information in our Levels 1 and 2. Clutter and occlusion prevent a meaningful global analysis, and while individual users and topics can be explored, this is only possible slowly, not without difficulty, and likely requires moving entities around to identify connections safely.

(b) Our technique at Level 2, showing the same predicted *connectivity* information as Analyst's Notebook in Figure 6a. Clusters and related users/topics can be pinpointed more easily. The color scheme and filtering settings in the top menu bar also facilitate to identify the prediction strength, which can be estimated by using the overlayed legend in the bottom right corner. The blue buttons allow to access the partition hierarchy modifier to view a dendrogram view of the grouped entities.

(c) PAOHvis [39]. The temporal hypergraph evolution shows the individual hyperedges, allowing to find connected users and topics. However, the hypergraph size is at the upper limit for a feasible visualization, already leading to some cluttering. Also, due to the temporal splitting, the comparability between years is hindered for such complex, non-sparse hyperedges compared to our technique, but better suited for comparing topics in the same year.

(d) Our technique at Level 3, showing the same *temporal evolution* information as PAOHvis in Figure 6c. The scalability is increased, showing no occlusion and the comparability of trends (important for the case study) is improved. This is due to retained cell ordering and short comparison distance. The downside is a reduced comparability between topics in the same year. The nature of the predictions depend on the model.

Fig. 6. Case study comparison of different approaches using the same internet forum hypergraph model dataset and exactly the same data view (connection strength > 0.1, min. 2 hyperedges). Compared are the state-of-the-art industry solution IBM i2 Analyst's Notebook (Figure 6a), PAOHvis (Figure 6c) against our technique, showing the information at two different levels of abstraction (Figures 6b and 6d). Further, both external approaches only support a fixed network while our technique allows for an interactive refinement and domain knowledge integration.

predicted data. A flag controls the ordering mode to either respect the filtered or the full dataset (including filtered elements). To further structure the view, the experts can manually click and select to group users and topics to reflect their mental categorization of users and topics. This allows to reflect domain-specific ontologies (e.g., similar concepts) or represent known formations of users.

Zooming to the lower visualization levels shows the temporal development. Compared to existing approaches (see Figure 6c) our technique (Figure 6d) increases the scalability and comparability for dense temporal evolution. Compared to the industry standard Figure 6a, presenting the temporal evolution as a timeline-like arrow within each cell reduces comparison distances. Levels 5 and 6 allow an expert to understand the actual data on which a predicted connection is based: The main keywords of the relevant text fragments and, respectively, the actual raw text fragments (cf. Figure 3). This ability allows the expert to verify predictions and detect shortcomings as, for example, irony and coded synonyms are still difficult to be detected automatically. If the expert has identified shortcomings on any level, e.g., missing connections or wrong attribution of an ambiguous term, the technique allows for the inclusion of this additional domain knowledge. To externalize knowledge, the expert selects the corresponding connection and specifies the proposed strength on a scale between 0 and 1. This translates to definite knowledge about no and guaranteed connection, respectively. More nuanced values like .7 allow the expert to reflect his own uncertainty. This allows them to try out hunches while simultaneously preserving some model flexibility. For this reason, the change preview (cf. Figure 5b) is extremely relevant for the domain experts, as it allows them to see directly how their knowledge transforms the model prior to accepting the changes. They can explore the consequences by zooming and panning through all levels and correlate their findings with their intuition or other facts. If unsatisfied, they can go back. Otherwise, they can continue and repeat this visual analytics loop multiple times. This rapid feedback supports the expert in refining the model without being blind to the resulting consequences, but being able to control and explore the latest model state at all times. As the domain experts focus is on exploratory analysis the iterative refinement supports finding connections and missing links faster. With domain knowledge that is difficult to be integrated a priori, step-by-step changes are more understandable.

6 Formative Evaluation

We performed formative evaluation sessions involving three domain experts (P1-P3). P1 is a criminal investigator working for a European law enforcement agency, having more than 30 years of experience, 20 years spent in digital and criminal investigations. His expertise includes communication and network analysis, familiarity with commercial systems like IBM i2 Analyst's Notebook [23], the graph visualization tool Gephi [9], as well as the large network analyzer Pajek [10, 11]. P2 works at the same agency in a different division, and has more than 20 years of experience in criminal investigations, specialized in group structure and content analysis. P3 is a senior project lead at a governmental research institute, studying analytical raw data analysis for more than ten years.

6.1 Study Procedure

The formative evaluation was conducted individually via remote screen sharing, taking about 60 minutes. For later review of these remote screen sharing sessions, they were recorded after receiving the formal consent of the experts. In the first 10 minutes a demo presented how to perform the visual analysis, explore and refine data and processes, and integrate domain knowledge in the search process and in the machine learning model. The next 30 minutes were spent between the experts using the system and providing feedback, as well as additional on-demand demonstrations. The tasks the experts performed include overview, the identification of the most promising leads, and the drill-down through the different zoom-levels down to the actual raw content, in this case, communication data. Further, we demonstrated and debated the different interaction techniques, like cutoff values and thresholds, matrix sorting and reordering strategies, and the dynamically modifiable partition hierarchy, as well as the machine learning feedback process.

In the last 20 minutes, the authors interviewed the experts asking 32 prepared questions (see annexes). During each of the formative evaluation sessions, the experts engaged actively, trying out concepts, asking questions, commenting on the features, and pointing out issues. If an expert already partially gave comments during the 30 minutes session, they were offered to extend their answer. For example, when an aversion or surprising idea was mentioned, we additionally focused on these aspects. The interview was designed to elicit aspects of our technique that the experts find relevant for their work or confusing or misinterpretable, as well as opinions on the individual approaches.

6.2 Findings and Lessons Learned

The **main observations** during the study are that our approach can effectively support most analytical requirements of the experts and that the experts favor both the **rapid exploration** of large datasets at different levels as well as the ability to integrate and contribute with their **domain knowledge**. This matches with their need to identify general trends in single combinations of users and topics and simultaneously identify co-occurrences. For this, the general prediction is more important than being able to identify differences between entities in the same year (cf. Figure 6). The underlying model we built upon [6] has proven to perform sufficiently well in this prediction task with an AUC (area under curve) of the ROC (receiver operating characteristic) of .88 and a recall value of .81. Excluded from the requirements are concepts outside the design scope, like purely mathematical capabilities as, for example, general centrality calculations, for which algorithms exist and could be included. In the following, we structure and summarize the main findings based on the expert's interactions and comments.

The domain experts agree that our approach of structuring information in **multiple levels** of details, using a **matrix-based approach**, is novel and therefore is not used in practice in their domain. For example, so far P1 has worked with either text-based or graph-based tools, and thinks our approach can "perfectly complement" existing workflows. The experts highlight the ability to effortlessly explore so much information (cf. P3), thereby "saving time" (P1), enabling a "quick analysis" (P3), while providing a "great overview … with much details, … but without overloading" (P2) the analyst, with an ease that is unexpected, given previous experience with this amount of data (cf. P2). We observed, that the experts often switch between the levels for targeting (upper levels) and then exploration and confirmation (lower levels). As P2 notes, this increases the size limit of the visually analyzable graph models, enhancing upon existing systems. "Together with the search capability" (P1), this allows for a very flexible workflow, enabling a good overview even for larger datasets.

The initial overview visualizations (Levels 1 and 2) are welcomed for providing a fast overview (cf. P1). The **color scheme** in Level 2 is regarded as comprehensible without explanation and aligning with expectations (cf. P1). It helps to provide guidance "where to start" (P1), and supports analysts in "planing their actions" (P3). To make the color scheme absolutely comparable, P3 requested the addition of a color legend. The **glyphs** are appreciated for providing details on the temporal distribution and future predictions (P2, P3). The glyph-based arrow representation in Levels 3 and 4 is appreciated for providing details on the temporal distribution (cf. P2, P3) interesting to the experts, and, most importantly, "the future predictions" (P2) in context of the historical data. Depicting future predictions in the arrowhead and the past data in the shaft, and seeing both together was described as "helpful" (P3). The alignment by fixed timesteps, like years, is regarded as precise and practicable (cf. P1) by the experts. In comparison, the distribution as line chart in Level 4 received mixed responses, with P1 and P3 finding it beneficial for their understanding to get a better, absolute reading, while P2 feels "it does not add much". The **keyword visualization** (Level 5) is regarded as fine for an abstract summary of the content visualization but could be extended (cf. P3). This layer, representing the "main connection" (P1) to the actual raw data, is important (cf. P1), and only shown when relevant in high zoom levels, "where the text content is relevant" (P1).

The ability to **search** through all underlying textual data and highlight matches in the views was received enthusiastically by all experts, as they can also transfer and fulfill some of their existing workflow, e.g., content- and text-based workflows, with our technique. It allows to explore global tendencies while enabling to query locally (cf. P2), not being distracted by other matches "not relevant at the moment" (P2).

While the visualization alone helps them already some ways, providing them "with improved degree of detail … unknown so far" (P1), all the experts also agree that the **interaction concepts** constitute an essential and relevant part of the approach, "helping them with strategical and operational decision" (P1). The **matrix reordering strategies** significantly improving the visual clarity of the overview, are regarded as "very interesting" (P2), and enable the experts to detect "groups" (P1) as well as connections easily, allowing them to "quickly identify hotspots" (P2), while putting less emphasis on weak connections. This is regarded as very supportive, being rarely supported in analysis systems (cf. P1), "saving costs and time" (P1). We observed that the experts use this as system guidance. The **partition hierarchy** is regarded by all experts as "essential" (P2), with P3 describing it as a "core functionality". It allows grouping different model parts into physical concepts, applying structure comparable to existing mental models (cf. P2), improving the mental mapping. It "makes decision easier" (P3) and allows to "connect things" (P3).

The experts further describe that with existing tools, one major problem is that their **mental concepts** and models can "not [be integrated] enough" (P2) in the exploration, making it harder and less comprehensible. They notice that our approach supports them in three ways not present in existing tools: (1) the **interactive exploration** allowing to follow their instinct, (2) the modifiable **partition hierarchy** to express and capture their mental concepts, and, "most importantly" (P1), (3) the ability to integrate their **domain and external knowledge** directly in the model. While the experts wished that they could already "generate a report [… and] export single entries" (P2) as commercial systems do, they note the enormous conceptional benefits of our technique. They regard them as "optimal" (P1), as there "are concepts and knowledge that cannot be modeled with machine learning [alone]" (P1) and are not "available" (P1) in the data. This knowledge then "cannot be integrated so far" (P1), is often documented in the head of the domain expert or "on a post-it note on the desk" (P1), leading to a high risk of the knowledge being "lost" (P1) or not leveraged. According to P1, the **knowledge integration** is performed iteratively during exploration, which we also observed as the experts adding knowledge intermittently, beginning with their main suspects and then expanding, adding knowledge when necessary either from post-its or when reading a name triggers a memory. The experts think that our **feedback loop** contributes to their analysis (cf. P1), replacing and "perfectly complementing" (P1) existing workflows. They regard the ability to *interactively* insert their knowledge as versatile. P1 noted that inserting all knowledge beforehand would be error-prone and "practically impossible" for larger datasets. To see "**validation** [possibilities] on changes" (P3) is especially important for vetting, and the change view is regarded as "very clear" (P1), allowing them a first glance, beneficial for prefiltering, steering and follow-up search guidance (cf. P1) to better divide their time for exploration. For improved usability P1 suggested to enable clicking to jump directly to the raw data in the change preview mode for validation. P1 regards the ability for a **global accept/reject** as sufficient for now, conceding that a partial accept could be explored in the future, although he does not see an immediate benefit. They state that the **0–1 scale** is "understandable and usable" (P1), but note that using the "5x5x5 system" (P1)—a commonly used police system based on letters A–E and 1–5 for source and intelligence evaluation [32]—would be immediately understood and universally accepted in the target domain. The approach allows them to integrate their domain knowledge on multiple levels, together with the ability to perform a "quick analysis" (P3) of "large amounts of information" (P2) "in a targeted" (P1), non-overloading manner. From the observations of the experts, we derived a set of **tentative tasks**, relevant in law enforcement: (1) finding linked users/topics, (2) connecting users which share related topics to identify co-conspirators, (3) using classical text-based search in the raw data to identify users, (4) finding and judging an in/decrease of user activity for a topic, (5) finding a temporal co-occurrence between topics and users, (6) adding domain knowledge to a specific user and specific topic and judging the implications, (7) transfer raw data patterns and identify related users, and (8) confirming the model predictions by cross-validation plausibility with the raw data texts.

7 Discussion and Future Work

During the evaluation, we received multiple proposals on how our approach could be extended further, including by mathematical analysis methods and industry-grade interfaces. In the following, we discuss the limitations and broader applicability of our approach, also in the context of future work. For our prototype approach, we adapted the generic blueprint of a machine learning model to the case study. This use case has its own limitation, requiring structured data with time and author information, and dependent on advanced topic extraction models. We tested our prototype successfully with 1 000 users, 800 topics, and 15 timesteps on an HD screen, typically the upper size for large investigations. In terms of data type, the technique can cope both with sparse and none-sparse matrix structures. For the former, the matrix reordering allows to prioritize more relevant connections and order them further on the top left, reducing the required screen usage for the main parts. Of course, a *homogeneous* sparse matrix does not benefit from that. In this case, and for none-sparse matrices, the different zoom levels shift the size limitations. Nevertheless, they do not scale infinitely. Scrolling would be needed when scaling further, even for the overview level. According to domain expert P1, there the primary concern would be the number of users (y-axis), but using the partition hierarchy and matrix reordering could partially mitigate the issue. When increasing the number of time steps, the arrow becomes more detailed, shifting from blocks to a more continuous stream, becoming less distinguishable. For our use case, this fine-grained time is not primarily relevant because the experts aim at seeing who has recently been interested in a topic. However, it might become an issue when the task requires to extract detailed timestamps. Therefore one could use hovering, magnification on demand, or a more specialized visualization. Also, the visualization presented is better at analyzing trends and connectivity tasks on an overview level. Comparing the same time step in Levels 3 and beyond between two non-aligned nodes, however, becomes harder. For further work, we envision an adaptable overview layer showing a specific time point, allowing cross-cell comparability. When adapting to different use cases, some of the filtering methodology likely has to be changed. For example, when supporting biochemical process analysis, the raw attributes are not texts anymore, which (1) would need a different visualization for the content in the two lowest display level, but would also impact (2) the search functionality, which would need to be adapted to search and filter for biological and chemical properties instead of text. The discussed visualization components serve only as examples for the visual analytics workflow presented. When adapting to a different field, there exist manifold possibilities for extensions, by integrating domain-specific visualization components. We provision this by a modular view architecture, supporting independent layer modules. Further enhancements are multiple magic lenses to allow for simultaneous drill-down to different levels.

In the future, we envision improvements to the feedback system, for example, showing how domain knowledge propagates not only between two model states, i.e., before and after adding knowledge but also explaining the effects of previously introduced knowledge, for example, by interactively highlighting the individual influences on hover. This is supported by our architecture, but the computation time scales linearly with the number of domain knowledge inputs, which leads to computation times of several minutes and more, making it infeasible in an interactive environment for fast iterations. We hope to improve this by enhanced engineering, reducing the model setup and reloading times by advanced ways of updating the hypergraph model.

8 Conclusion

Many processes are difficult to describe using traditional graph-based concepts and benefit from more precise yet more complex modeling as temporal hypergraphs. We address this challenge by using a geometric deep learning approach and extend it to hypergraphs. However, such deep learning models typically do not incorporate domain knowledge, usually unavailable in the data. This is not least because domain experts struggle to articulate their knowledge without rapid, iterative feedback and intuitive representations matching their mental models, alternatively requiring a detailed a priori understanding of the problem. Hence, domain expertise is often not leveraged to its full potential.

We contribute a technique, named Hyper-Matrix, to make temporal hypergraph model exploration more accessible for domain experts by enabling the integration of domain knowledge into the process and support their mental models through a multi-level matrix-based visualization architecture. The technique enables the interactive evaluation and seamless refinement of such models while providing a tight coupling and rapid, iterative feedback cycles to the underlying machine learning model. Model changes in response to the integration of domain knowledge are visualized transparently by a change preview, allowing experts to foster a more detailed understanding of how the underlying model works while externalizing their knowledge to teach the machine.

The approach allows to swiftly explore vast search spaces while maintaining focus and eliminating demanding context switches. Drill-down capabilities across multiple levels allow studying details and model contents on demand while retaining the overview. This architecture facilitates a focused analysis of relevant model aspects, allowing experts to detect patterns more rapidly and accurately. It is complemented by interactive filtering and search, various matrix reordering techniques, and a dynamically modifiable partition hierarchy, allowing the integration of domain knowledge in the visualization layers.

We evaluate our approach in one case study and through formative evaluation with law enforcement experts using real-world communication data. The results show that our approach surpasses existing solutions in terms of scalability and applicability, enabling the incorporation of domain knowledge and allowing fast and targeted search-space traversal. While we focused on topic prediction for law enforcement as driving application, the interactions and concepts work with any temporal hypergraph, being model agnostic and applicable more generically to a wider variety of domains. With our technique, we hope to pave the way for domain experts to a more interactive exploration and refinement of temporal hypergraph models, enabling them to use their knowledge not only for steering but also to articulate it into the machine learning model.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 700381. This material reflects only the authors' views, and the Commission is not liable for any use that may be made of the information contained therein.

References

[1] J. Ahn, C. Plaisant, and B. Shneiderman. A Task Taxonomy for Network Evolution Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):365–376, 2014. doi: 10.1109/TVCG.2013.238

[2] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. J. Rodgers. Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges. In *Proceedings of the Eurographics Conference on Visualization (EuroVis)*, 2014. doi: 10.2312/eurovisstar.20141170

[3] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. J. Rodgers. The State-of-the-Art of Set Visualization. *Computer Graphics Forum*, 35(1):234–260, 2016. doi: 10.1111/cgf.12722

[4] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data*. Springer-Verlag, Berlin/Heidelberg, 2006. doi: 10.1007/3-540-31190-4

[5] D. Arya, S. Rudinac, and M. Worring. HyperLearn: A Distributed Approach for Representation Learning in Datasets With Many Modalities. In *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2245–2253, 2019.

[6] D. Arya, S. Rudinac, and M. Worring. Predicting Behavioural Patterns in Discussion Forums Using Deep Learning on Hypergraphs. In *International Conference on Content-Based Multimedia Indexing (CBMI)*, 2019.

[7] D. Arya and M. Worring. Exploiting Relational Information in Social Networks Using Geometric Deep Learning on Hypergraphs. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pp. 117–125, 2018.

[8] B. Bach, E. Pietriga, and J.-D. Fekete. GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):740–754, 2014. doi: 10.1109/TVCG.2013.254

[9] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An Open Source Software for Exploring and Manipulating Networks. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, ICWSM, pp. 361–362. AAAI, 2009.

[10] V. Batagelj and A. Mrvar. Pajek - Program for Large Network Analysis. *Connections*, 21(2):47–57, 1998.

[11] V. Batagelj and A. Mrvar. Pajek - Analysis and Visualization of Large Networks. In *Graph Drawing*, Lecture Notes in Computer Science, pp. 77–103. Springer, Berlin, Heidelberg, 2002.

[12] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. A Taxonomy and Survey of Dynamic Graph Visualization. *Computer Graphics Forum*, 36(1):133–159, 2017. doi: 10.1111/cgf.12791

[13] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete. Matrix Reordering Methods for Table and Network Visualization. *Computer Graphics Forum*, 35(3):693–716, 2016. doi: 10.1111/cgf.12935

[14] M. Behrisch, J. Davey, F. Fischer, O. Thonnard, T. Schreck, D. A. Keim, and J. Kohlhammer. Visual Analysis of Sets of Heterogeneous Matrices Using Projection-Based Distance Functions and Semantic Zoom. In *Computer Graphics Forum*, vol. 33, pp. 411–420, 2014.

[15] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric Deep Learning: Going Beyond Euclidean Data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[16] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X. He. Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music content. In *Proceedings of the 18th ACM International Conference on Multimedia*, pp. 391–400, 2010.

[17] N. Elmqvist, T.-N. Do, H. Goodell, N. Henry, and J.-D. Fekete. ZAME: Interactive Large-Scale Graph Visualization. In *2008 IEEE Pacific Visualization Symposium*, pp. 215–222, 2008.

[18] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao. Hypergraph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3558–3565, 2019.

[19] M. Ghoniem, G. Shurkhovetskyy, A. Bahey, and B. Otjacques. VAFLE: Visual Analytics of Firewall Log Events. In P. C. Wong, D. L. Kao, M. C. Hao, and C. Chen, eds., *Visualization and Data Analysis 2014*, SPIE Proceedings, p. 901704. SPIE, 2014. doi: 10.1117/12.2037790

[20] B. Heintz and A. Chandra. Beyond Graphs. *ACM SIGMETRICS Performance Evaluation Review*, 41(4):94–97, 2014. doi: 10.1145/2627534.2627563

[21] R. Helms and K. Buijsrogge. Knowledge Network Analysis: A Technique to Analyze Knowledge Management Bottlenecks in Organizations. In *16th International Workshop on Database and Expert Systems Applications (DEXA'05)*, pp. 410–414, 2005.

[22] T. Hwang, Z. Tian, R. Kuangy, and J.-P. Kocher. Learning on Weighted Hypergraphs to Integrate Protein Interactions and Gene Expressions for Cancer Outcome Prediction. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 293–302, 2008.

[23] IBM. i2 Analyst's Notebook, 2020.

[24] N. Kerracher, J. Kennedy, and K. Chalmers. A Task Taxonomy for Temporal Graph Visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 2015. doi: 10.1109/TVCG.2015.2424889

[25] B. Kim, B. Lee, and J. Seo. Visualizing Set Concordance with Permutation Matrices and Fan Diagrams. *Interacting with computers*, 19(5):630–643, 2007. doi: 10.1016/j.intcom.2007.05.004

[26] F. Kuhn and R. Oshman. Dynamic Networks: Models and Algorithms. *ACM SIGACT News*, 42(1):82–96, 2011.

[27] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task Taxonomy for Graph Visualization. In *Proceedings of the 2006 AVI workshop on Beyond time and errors novel evaluation methods for information visualization - BELIV '06*, p. 1, 2006. doi: 10.1145/1168149.1168168

[28] D. Li, Z. Xu, S. Li, and X. Sun. Link Prediction in Social Networks Based on Hypergraph. In *Proceedings of the 22nd International Conference on World Wide Web*, pp. 41–42, 2013.

[29] M. Liu, Y. Gao, P.-T. Yap, and D. Shen. Multi-Hypergraph Learning for Incomplete Multimodality Data. *IEEE Journal of Biomedical and Health Informatics*, 22(4):1197–1208, 2017.

[30] Q. Luo and D. Zhong. Using Social Network Analysis to Explain Communication Characteristics of Travel-related Electronic Word-of-Mouth on Social Networking Sites. *Tourism Management*, 46:274–282, 2015. doi: 10.1016/j.tourman.2014.07.007

[31] W. Meulemans, N. H. Riche, B. Speckmann, B. Alper, and T. Dwyer. KelpFusion: A Hybrid Set Visualization Technique. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1846–1858, 2013. doi: 10.1109/TVCG.2013.76

[32] National Policing Improvement Agency, ed. *Guidance on the Management of Police Information - Appendix 2*. 2nd ed., 2010.

[33] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, M. A. de Menezes, K. Kaski, A.-L. Barabási, and J. Kertész. Analysis of a Large-Scale Weighted Network of One-to-One Human Communication. *New Journal of Physics*, 9(6):179, 2007.

[34] N. Pržulj. Protein-Protein Interactions: Making Sense of Networks Via Graph-Theoretic Modeling. *Bioessays*, 33(2):115–123, 2011.

[35] P. Saraiya, P. Lee, and C. North. Visualization of Graphs with Associated Timeseries Data. In J. Stasko and M. O. Ward, eds., *Proceedings of the IEEE Symposium on Information Visualization*, InfoVis, pp. 225–232. IEEE, 2005. doi: 10.1109/INFVIS.2005.1532151

[36] F. Shi, J. G. Foster, and J. A. Evans. Weaving the Fabric of Science: Dynamic Network Models of Science's Unfolding Structure. *Social Networks*, 43:73–85, 2015. doi: 10.1016/j.socnet.2015.02.006

[37] D. Streeb, D. Arya, D. A. Keim, and M. Worring. Visual Analytics Framework for the Assessment of Temporal Hypergraph Prediction Models. In *Proceeedings of the Set Visual Analytics Workshop at IEEE VIS 2019*, 2019.

[38] P. Valdivia, P. Buono, and J.-D. Fekete. Hypenet: Visualizing Dynamic Hypergraphs. doi: 10.2312/eurp.20171162

[39] P. Valdivia, P. Buono, C. Plaisant, N. Dufournaud, and J.-D. Fekete. Using Dynamic Hypergraphs to Reveal the Evolution of the Business Network of a 17th Century French Woman Merchant. In *VIS 2018 - 3rd Workshop on Visualization for the Digital Humanities*. Berlin, Germany, 2018.

[40] P. Valdivia, P. Buono, C. Plaisant, N. Dufournaud, and J.-D. Fekete. Analyzing Dynamic Hypergraphs with Parallel Aggregated Ordered Hypergraph Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2019. doi: 10.1109/TVCG.2019.2933196

[41] F. van Ham. Using Multilevel Call Matrices in Large Software Projects. In *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714)*, pp. 227–232. IEEE, 19-21 Oct. 2003. doi: 10.1109/INFVIS.2003.1249030

[42] C. Vehlow, F. Beck, and D. Weiskopf. Visualizing Group Structures in Graphs: A Survey. *Computer Graphics Forum*, 36(6):201–225, 2017. doi: 10.1111/cgf.12872

[43] Y.-J. Wang, M. Xian, J. Liu, and G.-y. Wang. Study of Network Security Evaluation Based on Attack Graph Model. *Journal of China Institute of Communications*, 28(3):29, 2007.

[44] Y. Wu, S. Liu, K. Yan, M. Liu, and F. Wu. OpinionFlow: Visual Analysis of Opinion Diffusion on Social Media. *IEEE Transactions on Visualization and Computer Graphics*, 2014. doi: 10.1109/TVCG.2014.2346920

[45] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar. HyperGCN: A New Method For Training Graph Convolutional Networks on Hypergraphs. In *Advances in Neural Information Processing Systems*, pp. 1509–1520, 2019.

[46] D. Zhou, J. Huang, and B. Schölkopf. Learning With Hypergraphs: Clustering, Classification, and Embedding. In *Advances in Neural Information Processing Systems*, pp. 1601–1608, 2007.