UAB
Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

This is the **published version** of the bachelor thesis:

Gascón Marzo, Andreu. *Social Networks Modelling*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026

This version is available at https://ddd.uab.cat/record/336333

under the terms of the BY-NC-ND license.

Social Networks Modelling

Andreu Gascón Marzo

February 4, 2026

Abstract— Predicting user interactions in decentralized social networks requires capturing both complex community structures and rapidly evolving temporal dynamics. This study presents a scalable Spatio-Temporal Graph Attention framework for multi-task link prediction, utilizing a massive dataset from the Bluesky social protocol. We introduce a Snowball-Tiered Sampling methodology that condenses 11 billion records into a representative million-scale environment while preserving the network's Power Law characteristics. Our architecture integrates 384-dimensional MiniLM semantic embeddings directly into a heterogeneous graph flow, allowing the model to prioritize topical alignment over simple proximity. By employing a dual-attention mechanism—spatial GAT layers for neighborhood context and temporal self-attention for behavioral "velocity"—the model achieves high predictive precision for the next day interactions.

Keywords: Graph Neural Networks, Social Network Modelling, Attention Network, Bluesky

1 Introduction

In the digital age, online social networks have become a **global and fundamental infrastructure for communication**, information dissemination, and social interaction. Platforms like X (formerly Twitter), Bluesky, Facebook, and Instagram generate unprecedented volumes of data daily, capturing the complex, dynamic, and interconnected nature of human behaviour. Understanding the underlying mechanisms that drive user engagement and social connectivity is crucial for a wide range of applications, from personalised content recommendation and targeted advertising to the detection of information campaigns and the modelling of societal trends.

To computationally model these platforms, they are **naturally represented as graph networks**, where individual users and content are **nodes**, and their interactions—such as following, posting, liking, and reposting—form the **edges**. This graph structure encapsulates the rich social context that influences an individual's behaviour. The field of Graph Machine Learning, particularly Graph Neural Networks (GNNs), has emerged as the premier paradigm for learning from this relational data. For instance, recent work has demonstrated the effectiveness of GNNs in predicting customer engagement on social media [15], and studies have successfully leveraged models like GraphSAGE [6] and Graph Attention Networks (GATs) [12] to predict **social relations by learning powerful node embeddings.**

A critical limitation of these foundational approaches, even those handling heterogeneity, is that they often **treat networks as static entities**, analysing a single snapshot in time. Social networks are **inherently dynamic** and temporal; user interests evolve, new connections are forged, and the strength of influence fluctuates. A user who was highly influential on a topic last month may not be relevant today, and a new friendship can alter the content a user is exposed to.

This reality has spurred research into **dynamic graph models**. The field of dynamic graph representation learning is built on the premise of using evolving network patterns to create node embeddings that can **forecast future links**. Models like GCN_MA [16] and TemporalGAT [9] exemplify this by integrating GNNs with sequential models like LSTMs or Temporal Convolutional Networks to learn node representations that capture both **spatial structure and temporal dynamics** for the explicit purpose of dynamic link prediction.

Therefore, the core challenge of this work is to synthesise and extend these advances into a **model capable of not only understanding the spatial structure of a social graph but also its continuous evolution over time**. While existing models often learn representations primarily as an intermediate step for link prediction, this project aims to focus also on generating high-quality future-state embeddings for both users and posts as a versatile output. This requires a dynamic spatio-temporal model that can **natively handle the heterogeneity** of social graphs (with multiple node and relation types) and their expanding nature. By building upon the foundations of GATs and integrating a sophisticated temporal attention mechanism to overcome the compression limitations of sequential models, this project will create a model that **learns how a user's and posts char-**

• Contact email: Andreu.Gascon@autonoma.cat
• Supervised by: Ania Sikora (Departament d'Arquitectura de Computadors i Sistemes Operatius)
• Academic Year 2025/26

acteristics, importance, and connections co-evolve with their neighbourhood.

This paper is structured as follows: **Section 2** provides a critical review of the current state of the art and related literature. **Section 3** outlines the research objectives of this study. **Section 4** presents the project timeline and planning phases. **Section 5** details the comprehensive methodology, spanning from our custom sampling strategies to the spatio-temporal attention architecture. **Section 6** presents the experimental results and performance metrics across the multi-task link prediction environment. Finally, **Section 7** offers concluding remarks and suggests directions for future research.

2 Related Work

This project extends research conducted in collaboration with Haoyuan Li from the *Department d'Arquitectura de Computadors i Sistemes Operatius (CAOS)* [5] of the *Universitat Autònoma de Barcelona (UAB)* [11]. Their previous work, "Applying Machine Learning for Characterising Social Networks in Agent-Based Models" [7], leveraged **Variational Autoencoders (VAEs)** to learn low-dimensional, interpretable representations of user behaviour to generate synthetic populations.

Building upon this foundation, the current project aims to develop a model capable of predicting interactions between users within the **Bluesky dataset**. The ultimate objective is to create a dynamic environment where synthetic users generated via VAEs can be integrated into the model. This will allow the system to predict how these generated populations form connections and co-evolve, shifting the focus from static user characterisation to the simulation of active, evolving social systems.

Our research is significantly influenced by the **Dynamic Self-Attention Network (DySAT)** [10], which pioneered the use of self-attention to capture the evolution of graph structures over time on graph email and movie review datasets (Enron, Yelp). By partitioning the network into discrete temporal snapshots, DySAT employs a dual-attention architecture: a *Structural Attention* layer to capture local neighbourhood features and a *Temporal Self-Attention* layer to track how a specific node's behaviour shifts across snapshots. However, a critical limitation of the original DySAT model is its **homogeneous nature**; it treats all edges as a single relation. Furthermore, DySAT primarily focuses on structural link prediction between users.

To address the multi-relational nature of Social Network data, researchers developed **DyHAN**, which effectively functions as a **heterogeneous evolution** of the DySAT architecture [14] encompassing four distinct types of interactions: retweets, replies, mentions, and follower links in **Twitter** (An almost identical Social Network to **Bluesky**, the one we will be using for our project). However, while DyHAN successfully navigates **edge heterogeneity**, it—like its predecessors—typically operates on structural IDs or **simplified attributes**. Our work diverges here by integrating **deep semantic post embeddings** directly into the heterogeneous flow. Furthermore, whereas typical DyHAN implementations utilize subsets in the range of 10,000 to 50,000 nodes, our approach scales the hierarchical attention logic to **500,000 users and 20 million posts**.

3 Objectives

We can distinguish 5 main objectives for our project:

- Conduct a systematic review of the **current landscape in Graph Machine Learning**, focusing on the evolution from static GNNs to dynamic spatio-temporal architectures such as **DySAT and DyHAN**.
- Build a **dense, representative sample of the original Bluesky** dataset that maintains both **social structures and behavioural dynamics**. By applying a two-phase sampling method—comprising *Recursive Snowball Sampling* for users and *Tiered Capping* for posts—the objective is to preserve the network's inherent **Power Law characteristics** [3]. Grounded in the principle of participation inequality and the 90-9-1 rule [8], this approach ensures the model captures the disproportionate influence of elite creators while accounting for the high-volume engagement of the majority [4].
- Include a **novel multimodal approach** that integrates **dense semantic embeddings** into the graph flow to bridge interaction history with content intent.
- Develop a hierarchical attention model that actualises embeddings in three stages: **(1)** *Self-Representation* via parametric projection, **(2)** *Spatial Attention* to rank neighbour relevance across two-hop neighborhoods, and **(3)** *Temporal Self-Attention* to calculate "temporal patterns." This stage specifically aims to capture how a user's interests co-evolve with their community without the memory loss associated with recurrent units.
- Execute **dynamic link prediction** across three distinct interaction types (*Follows, Likes, Reposts*). while solving the "Expanding Graph" problem through *Historical Interaction Anchoring*, allowing users to interact with a 7-day post buffer while predicting 24-hour future interactions.
- Implementing a **batching strategy** to construct localised subgraphs for training using **Negative Sampling**. This methodology enables the processing of million-scale networks on limited hardware.

4 Planning

The project execution is organized into five phases following the main objectives. It lasts aproximately 14 weeks of work.

Phase 1: State-of-the-Art Review & Theoretical Framework (2 Weeks)

- **1.1 Literature Revision:** Conduct a systematic review of the GML landscape, focusing on the evolution from static GNNs to dynamic spatio-temporal architectures like DySAT and DyHAN.
- **1.2 Previous Work Integration:** Revise previous doctoral work on Variational Autoencoders (VAEs) to possibly integrate agent-based modeling with modern attention mechanisms.

- **1.3 Environment Setup:** Configure the development stack, including Python, PyTorch Geometric, DuckDB, and the GPU virtual machine environment.

Phase 2: Dataset Engineering & Scalable Sampling (3 Weeks)

- **2.1 Power Law Analysis:** Perform an analysis of the degree distribution in Social Networks to establish downsampling constraints.
- **2.2 Snowball-Tiered Implementation:** Develop and execute the two-phase sampling method (Recursive Snowball and Tiered Capping) to condense 11 billion records while preserving Power Law characteristics.
- **2.3 Expanding Graph Solution:** Implement the 7-day rolling window for posts and 24-hour interaction snapshots to maintain temporal consistency.

Phase 3: Attribute Engineering (2 Weeks)

- **3.1 NLP Model Selection:** Validate and integrate the MiniLM-L12 transformer to generate 384-dimensional semantic embeddings for posts.
- **3.2 Statistical Attributes Selection:** Choose and precompute the aggregation of daily activity metrics from the raw 11-billion-row dataset to maintain real-world social weight.

Phase 4: Spatio-Temporal Model Development & Evaluation (5 Weeks)

- **4.1 Spatial Attention Layer:** Engineer the two-hop GAT layers for neighborhood context aggregation.
- **4.2 Temporal Self-Attention Mechanism:** Add the temporal attention logic across the 3-day sliding window to track behavioral "velocity".
- **4.3 Negative Sampling Strategy:** Set up the 1:5 contrastive training batching logic for hardware efficiency.

Phase 5: Results, Conclusions, and Future Work (2 Weeks)

- **5.1 Performance Metric Analysis:** Conduct a comprehensive evaluation of the model's predictive power.
- **5.2 Synthesis of Future Trajectories:** Propose refinements for the model, including Edge-Type Aware Attention and the shift toward long-term lifecycle modeling.

5 Methodology

The project will follow the steps on Figure 1:

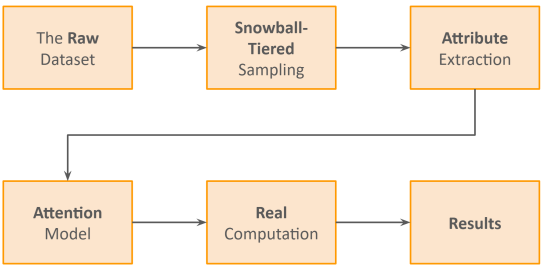

Fig. 1: Project Methodology Overview

5.1 The Raw Dataset

We bypassed the latency and rate-limiting constraints of direct Bluesky API scraping by sourcing data from the **Bluesky Data Repository** [1] [2]. This repository, maintained by Leo Balduf at the Technical University of Darmstadt (TU Darmstadt), provides comprehensive, pre-aggregated dumps of network activity collected by aggregating real-time updates from the Bluesky Firehose and full-network mirrors from February 2024 to 14 April 2025.

5.1.1 Parquet files & DuckDB

The repository organises the Bluesky social graph into a series of **Apache Parquet files** depicted in Table 1, a column-oriented storage format that enables efficient data compression and high-speed retrieval.

File	Total Rows	Size on Disk
profiles	32,170,299	1.77 GB
posts	1,290,686,604	153.28 GB
follows	2,156,778,700	23.17 GB
likes	6,938,743,344	206.59 GB
reposts	888,094,540	15.71 GB
key	34,251,851	0.55 GB

Table 1: Dataset Statistics Overview

While Python is the primary language for our model logic and training, using raw Python to handle Parquet data at this scale would create a **massive computational bottleneck**. Instead, we utilise **DuckDB**, an in-process **SQL-based** database, for all the data sampling and data-preprocessing steps.

5.1.2 Nodes & Edge Formation

The project defines the network structure using the `profiles.parquet` and `posts.parquet` files as the **source of nodes**, while the remaining files function as relational tables to map the connections between them.

User Nodes: Each user is identified by their unique `id`.

Post Nodes: Posts are defined by the combination of `author_id` and `post_id`.

Edges: Derived from relational tables to map the interactions between nodes:

- **Post Edges:** (User → Post)
- **Follow Edges:** (User → User)
- **Like Edges:** (User → Post)
- **Repost Edges:** (User → Post)

A detailed breakdown of the internal schema for these nodes and edges, including data types and primary key mappings, is provided in [annex:schemas]Annex 1.

A hand made visualization of a graph is depicted in Figure 2

A Real Visualization of the graph can be seen in 20

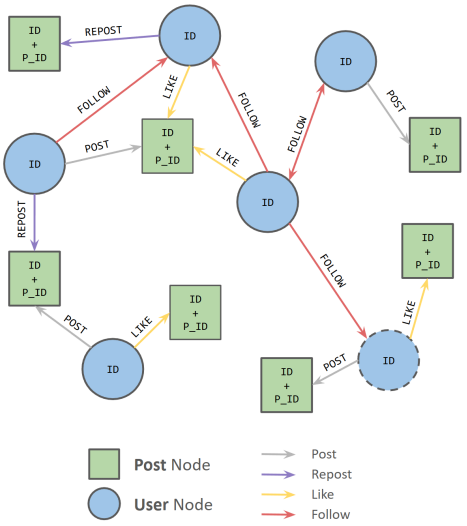

Fig. 2: Heterogeneous Graph Structure and Interaction Mapping

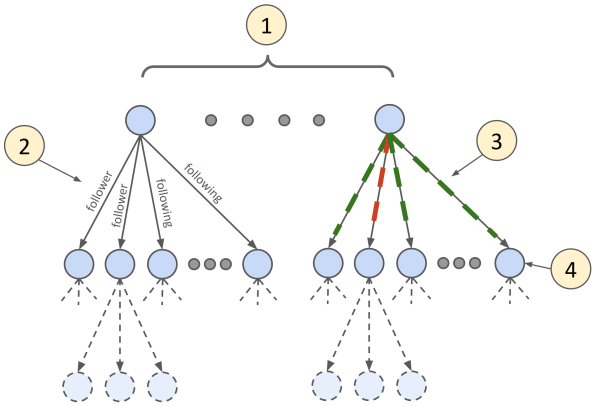

Fig. 3: Phase 1: Recursive Snowball Sampling Process illustrating the transition from seed selection to network expansion.

5.2 Snowball-Tiered Sampling

Downsampling is a computational necessity. Training a GNN on 11 billion edges would require petabytes of VRAM and months of processing time. We strategically reduced the dataset to 500,000 User nodes and 20,000,000 Post nodes using a Snowball Tiered Sampling approach.

The primary objective of this method is to optimize information density while preserving the underlying social structure. This ensures the model can distinguish distinct clusters across the heterogeneous graph while maintaining the influence and frequency of power users alongside the dominant interaction volume from "Lurkers".

5.2.1 Phase 1: Snowball User Sampling

The User Sampling process follows a four-stage recursive strategy inspired by Snowball Sampling:

1. **Seed Selection (Initial Snowflakes):** The process originates from the Top 1,000 Elite Users, identified by their "follow" counts.
2. **Bi-Directional Social Traversal:** The algorithm performs a bi-directional search that identifies both the followers and followees of the Elite Users.
3. **Stochastic Selection:** Random selection from the newly discovered pool of neighbors ensures diversity and reach to niche communities.
4. **User Cap:** To prevent neighbor explosion, a strict 150,000-user cap is enforced per iteration.

5.2.2 Phase 2: Post Tiered Sampling

To determine the sampled 500,000 user's **structural importance**, the model calculates a Custom Impact Score:

$$\begin{aligned} Impact\ Score = &(Followers \cdot 1.0) \\ &+ (Likes\ Received \cdot 2.0) \\ &+ (Reposts\ Received \cdot 3.0) \end{aligned} \tag{1}$$

To maintain the network's Power Law characteristics [3], we implement a sampling strategy grounded in the principle of **participation inequality** [8]. By randomly selecting posts based on percentile rank (Table 2), we account for the distribution where a **minority of "Elite Users"** generates the **majority of content** [4].

Tier	Classification	Percentile	Post Cap
Tier 1	Elite Users	Top 1%	800
Tier 2	Core Users	Next 19%	150
Tier 3	Peripheral Users	Bottom 80%	30

Table 2: TIER DISTRIBUTION AND STOCHASTIC POST CAPPING BASED ON USER IMPACT SCORES.

5.2.3 Phase 3: Snapshot Creation

We utilize the `created_at` timestamp present in every row of the dataset to partition the sampled data into discrete temporal snapshots.

- **24-Hour Interaction Windows:** All interaction edges (posting, follows, likes, and reposts) are sliced into **24-hour** increments.
- **7-Day Post Buffering:** We maintain a **7-day** look-back window for post nodes to account for users interacting with older content.

5.2.4 Sampling Results

Our Snowball Tiered Sampling successfully condensed an 11-billion-row dataset into a dense, million-scale graph with high structural integrity. The LWCC of 0.71 reflects that while all nodes are **globally connected**, roughly 30% appear as "isolated islands" within specific 24-hour snapshots due to **temporal inactivity.** A Modularity of 0.49 confirms the preservation of **natural social "bubbles**," while an Average Degree of 8.25 provides **sufficient neighborhood context for GNN triangulation** without over-smoothing node identities.

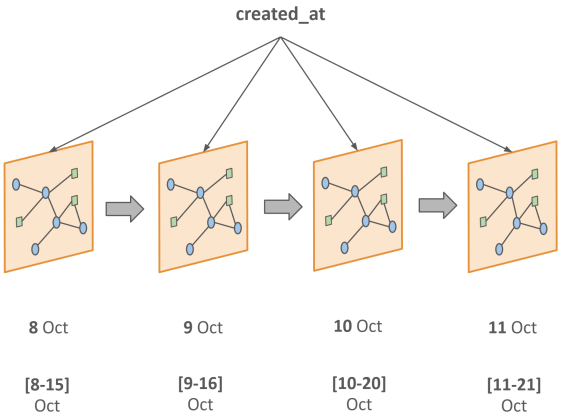

Fig. 4: Temporal Snapshot Generation showing the 24h Interaction and 7d Post Windows

Metric	Mean Value
LWCC	0.7138 (± 0.0221)
Modularity	0.4921 (± 0.0147)
Avg. Degree	8.2521 (± 1.1554)
Total Nodes	∼1.21 Million

Table 3: GRAPH STRUCTURAL METRICS AVERAGED OVER 90 DAILY SNAPSHOTS.

5.3 Attribute Extraction

Once we have correctly subsampled our raw data, we need **features** to provide the nodes with a **"social scent"**: a multi-dimensional vector that describes the user's and posts' identity, behaviour, and content so the GNN learns that two users are similar, not just because they follow the same person, but because they post in the same language, at the same time of day, or about the same topic. We distinguish between three types of features: **Structural**, **Statistical**, and **Semantic**.

A detailed list of these attributes and their specific processing methods is provided in [annex:attributes]Annex 2.

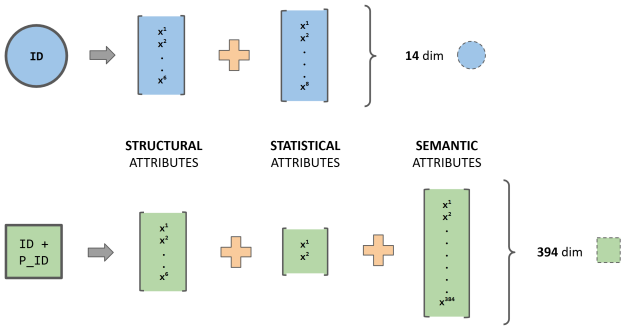

Fig. 5: Node Attribute Dimensionality: Breakdown of structural, statistical, and semantic features for User nodes (14 dimensions) and Post nodes (394 dimensions).

5.3.1 Structural Attributes

Structural attributes serve as the intrinsic "static identity" of each node, derived directly from the raw columns in `profiles.parquet` and `posts.parquet`. These features capture the fixed properties of an entity—such as whether a user has a **profile banner** or if a post contains **embedded media**. We transform the raw categorical data into numerical tensors using distinct processing methods (see [annex:attributes]Annex 2 for the full feature list).

This process yields **six** distinct structural dimensions for both User and Post entities.

5.3.2 Statistical Attributes

Statistical Attributes provide the **"momentum"** from day to day of the nodes. These features quantify a node's **influence and activity** levels within the network, aggregating statistics every snapshot.

A critical distinction in our methodology is that we use the **Raw Dataset** (11 billion records) for these counts, not the subsampled 500k world. By aggregating from the raw dataset, our 500,000 users and 20 milion posts carry their real-world "weight" into the GNN, and since we will apply normalisation techniques, the relative distance between user statistics will be maintained.

This process yields **8** distinct statistical dimensions for Users and **2** Statistical Dimensions for Posts. (see [annex:attributes]Annex 2 for the full feature list)

5.3.3 Semantic Attributes (Post Embeddings)

A core novelty of our proposed framework, specifically distinguishing it from the DyHAN architecture [14], is the inclusion of dense **Semantic Post Embeddings**. While existing heterogeneous models typically rely on structural IDs or shallow statistical features, we utilize the `MiniLM-L12` transformer [13] to generate 384-dimensional vectors for 20 million sampled posts.

This semantic integration will allow both users and posts to share a **common latent space** defined also by content rather than just interaction frequency. Consequently, the GNN can learn **semantic relationships** where Post nodes with similar embeddings (pointing in the same direction in the vector space) allow the model to cluster users based on shared topics, even across linguistic boundaries. This shifts the predictive focus from purely structural link prediction to a **content-aware interaction modeling** approach.

This specific NLP pipeline provides **three strategic advantages**: multilingual support, computational efficiency at scale, and a fixed output dimensionality.

Fig. 6: Semantic Feature Extraction Pipeline: Utilizing `MiniLM-L12-v2` to transform multilingual post text into a unified 384-dimensional vector space.

As illustrated in Figure 6, the MiniLM-L12-v2 model does not simply group posts by keyword; it captures the nuanced intent of the author. For example, a post expressing a

negative sentiment toward AI is projected in a different vector direction than one expressing a positive sentiment, even though both share the same topical keywords.

5.4 Attention Model

The core objective of the model is to solve the **Link Prediction problem**. In a network that changes constantly, we want to answer one question: **What is the probability that node A will interact with node B in the next 24 hours?** Because Bluesky is a Heterogeneous Graph, we aren't just predicting one type of link. We are performing **Multi-Task Learning**, aiming to predict three distinct edges at a time: follow, like, and repost. To achieve this, our architecture processes the network's complexity through three distinct sequential phases (three embedding actualizations)

1. **Feature Encoding (Self-Representation):** Establishes a baseline embedding for every node.
2. **Spatial Embedding:** Utilizes a multi-head attention mechanism to evaluate the influence of local neighbors within a discrete temporal snapshot actualizing the initial embedding.
3. **Temporal Embedding:** Analyses these social snapshots across a 3-day sliding window, identifying shifts in user intent and long-term trends to using multi/head self attention mechanism to produce the final spatio-temporal embedding used for prediction.

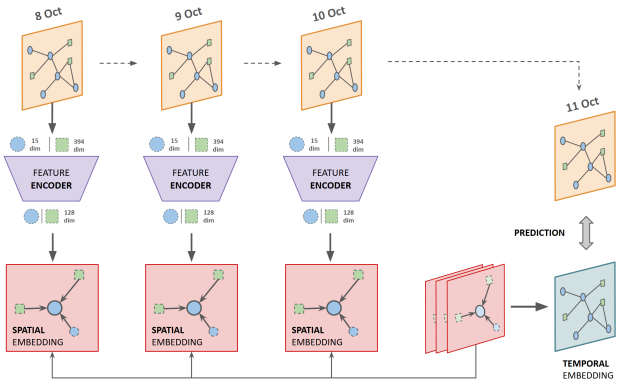

Fig. 7: Architecture Overview: The sequential pipeline for link prediction. Discrete temporal snapshots undergo Feature Encoding to align dimensions, followed by Spatial Embedding through neighborhood attention, and concluding with Temporal Embedding via a 3-day sliding window to generate final predictions.

5.4.1 Feature Encoding (Self-Representation)

The Feature Encoding phase establishes a node's intrinsic identity by constructing a **Self-Representation** based solely on its **internal attributes**. To bridge the gap between User nodes (14 attributes) and Post nodes (394 attributes), the model employs a parametric transformation with learnable weights to **project both entities into a shared 128-dimensional hidden space**. As illustrated in Figure 8, by "stretching" user features and "compressing" post features, the model creates a Static Representation where users and posts are represented in a vacuum as homogeneous vectors. This stage ensures that disparate node types are **mathematically comparable** and ready for social influence analysis within the same vector space.

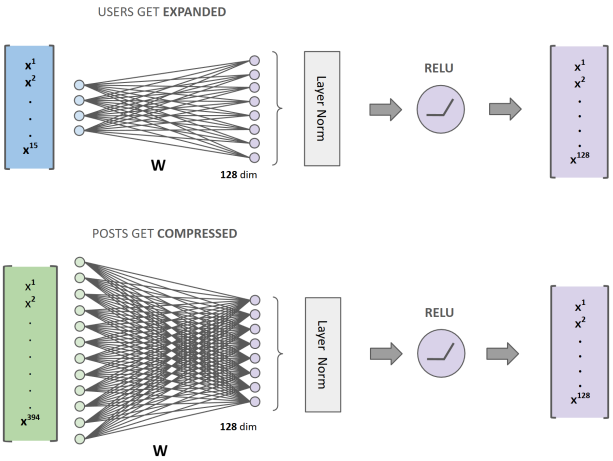

Fig. 8: Feature Encoding and Projection: Parametric transformation showing the "stretching" of 14-dimensional user features and the "compression" of 394-dimensional post features into a shared 128-dimensional hidden space

5.4.2 Spatial Embedding

The primary goal of the Spatial Attention Phase is to generate a **new, context-aware embedding** for every node by intelligently aggregating information from its local neighborhood within a specific temporal snapshot. This process transforms a node's "Self-Representation" into a "**Spatial Embedding"** that reflects its immediate social environment.

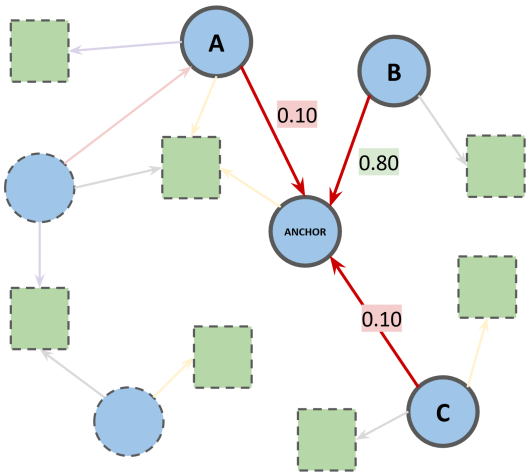

Fig. 9: Ranked Message Passing

Instead of treating all neighbours equally—which leads to **"feature dilution"** where important signals are lost in social noise—the model employs a Multi-Head Attention mechanism to dynamically **rank neighbour relevance** and aggregate information based on the **compatibility score** 9. This Score is obtained through 5 main phases that can be understood as a sequence of **questions** (Process can be followed in 10 and 11):

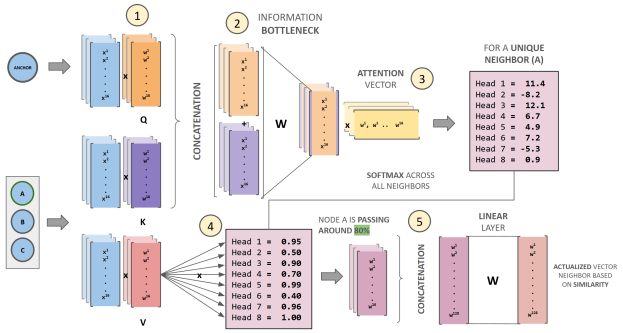

Fig. 10: Attention Pipeline

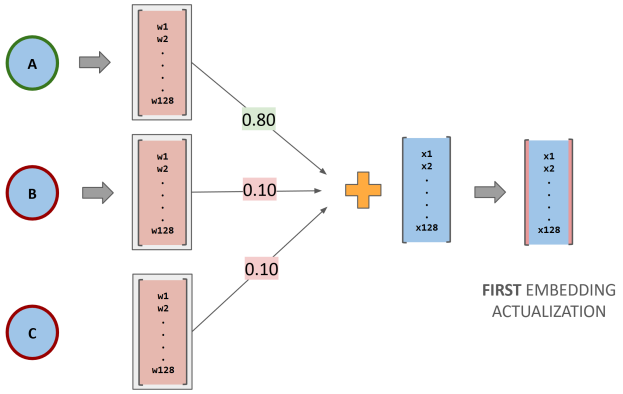

Fig. 11: Embedding Actualization

1. **Identity Projection (Q, K, V):** This stage acts as the "Social Search" initialisation, where the target node asks, *"What am I currently looking for?"* (**Query**) while its neighbours answer with, *"What do I have to offer?"* (**Key**) and *"What is my actual content?"* (**Value**) . Specifically, since we are using 8 attention heads, the input 128-D embedding (h) is projected into three 16-D subspaces (128-dim / 8) —q, k, and v—using learnable weight matrices W^Q, W^K, and W^V . These projections serve as the primary "Search Logic," allowing the model to focus on specific dimensions of social relevance, such as **topical alignment** or **network influence**, before any interaction is evaluated.

$$q_i = W^Q h_i, \quad k_j = W^K h_j, \quad v_j = W^V h_j \qquad (2)$$

2. **Compatibility Analysis (The Match):** Once the roles are established, the model evaluates the mathematical alignment between the matching heads by asking, *"How well does the neighbour's signal match the target's request?"* to determine if the **relationship is meaningful**. This is implemented by concatenating the 16-D Query (q_i) and 16-D Key (k_j) into a 32-D vector, which is then compressed back to 16-D through a dedicated **weight matrix**. This *Information Bottleneck* is critical; it forces the model to ignore irrelevant noise and retain only the most **essential features that define a connection** between those two specific nodes. This new 16-dime vector represents a **summary** for that specific head connection.

3. **The Attention Vector (The Decision):** This phase serves as the final "grade," where a decision-maker assesses the match and asks, *"On a scale of 0 to 10, how much should the target listen to this neighbour?"* . Technically, the compressed 16-D summary is multiplied by a learnable Attention Vector (a^T) to produce a raw relational score, e_{ij}. If certain dimensions of the summary represent "Social Noise," the corresponding weights in a^T shrink toward zero, effectively muting the signal before it reaches the final aggregation:

$$e_{ij} = \text{LeakyReLU}\left(a^T \cdot [W(q_i \parallel k_j)]\right) \qquad (3)$$

4. **Score Normalisation:** Score Normalisation: All raw scores (e_{ij}) are passed through a Softmax function, which transforms them into a **probability distribution** (α_{ij}) that sums to exactly 1.0. By multiplying each neighbour's 16-D Vector Value Embedding by its corresponding score, the model produces a final **Relational Summary** where high-relevance signals are amplified, and background social noise is effectively erased.

5. **Final Fusion:** Once each of the 8 specialized heads has finished evaluating the neighborhood and has outputted its own 16-dimensional weighted vector, the model acts as a final coordinator, asking, "How do these different perspectives combine into one cohesive social identity?". Technically, these eight 16-dimensional vectors are "stapled" back together (concatenated) to reconstruct a full 128-dimensional representation. This combined report is then passed through a **final learnable weight matrix** (W^O) that identifies correlations between the different experts—for example, merging a high "technical interest" score from one head with a high "social authority" score from another—to produce the final, high-fidelity spatial embedding.

After this process, the node changes from a solitary entity into a smart, 128-dimensional vector that understands its surroundings (Figure 11). This happens because the model "listens" to the most important neighbors while ignoring social noise (Figure 9). As shown in the Attention Pipeline (Figure 10), 8 specialized heads assign specific weights to neighbors to decide how much information they should pass. These weighted signals are then combined and projected into a final 128-dimensional embedding that captures the node's full social context.

5.4.3 Temporal Embedding

The Temporal Embedding evolves the node's representation by shifting focus from external social context to **internal behavioral history over a 3-day window.** Instead of searching for neighbors, the node performs a search across its **own past spatial embeddings** (temporal Self-Attention) to understand "velocity"—asking, "Is this interest a growing trend or a passing fad?". Technically, the model utilizes 4 temporal heads to track multiple "behavioral clocks," such as long-term stability versus high-volatility bursts. By attending to its own history (h_{t-2}, h_{t-1}, h_t), the current state (Query) is compared against past states (Keys) to determine

how much historical behavior should influence the final prediction. These weighted historical values are then aggregated to produce a final 128-dimensional Spatio-Temporal Embedding that captures the evolution of user intent.

$$Z_t = \sum_{k \in \{t-2, t-1, t\}} \text{Softmax}\left(\frac{(W^Q h_t) \cdot (W^K h_k)^T}{\sqrt{d}}\right)(W^V h_l \tag{4}$$

The computation is indeed **identical to the spatial attention mechanism,** but instead of ranking external neighbors in space, it ranks the i**mportance of a node's own historical states** $(t-1, t-2, t-3)$ and aggregates the spatial embedding according to the ranking.

5.4.4 Link Prediction

At this phase, every node is represented by a **128-dimensional Spatio-Temporal Embedding** (Z). This single vector is the culmination of the entire pipeline: it contains the node's **intrinsic identity** (Feature Encoding), the **context** of its neighborhood (Spatial Attention), and the **"velocity"** of its behavior over the 3-day window (Temporal Attention).

To determine the likelihood of interaction between two nodes, the model evaluates their mathematical alignment within the 128-dimensional latent space:

- **Interaction Probability:** The model performs a **dot product** between the spatiotemporal embedding of a source node (Z_u) and a target node (Z_{target}) to measure their alignment. A high dot-product indicates that the nodes' "social scents" are **closely aligned** in the latent space.

$$\text{score} = Z_u \cdot Z_{target}^T \tag{5}$$

- **The Sigmoid "Squash":** This raw similarity score is passed through a Sigmoid function (σ), which translates the value into a **probability** $\hat{y}$ **between 0.0 and 1.0**. For instance, a score of 0.83 suggests a highly probable interaction, while 0.27 suggests random noise.

$$\hat{y} = \sigma(Z_u \cdot Z_{target}^T) \tag{6}$$

- **Ground Truth Comparison:** The model compares this prediction $(\hat{y})$ against the *Ground Truth* (y) extracted from the **target snapshot** t_0. If a user actually performed an interaction (like, repost, or follow) in t_0, the label is $y = 1$; otherwise, it is treated as a negative sample where $y = 0$.

As illustrated in Figure 12, the model performs a **"Similarity Check"** by calculating the dot product between two spatio-temporal embeddings to determine their interaction likelihood.

If the model assigns a **high probability** (e.g., $\approx 73\%$) to a random pair that has **no actual interaction** in the ground truth $(y = 0)$, it triggers a **Powerful Correction** via Binary Cross-Entropy (BCE) loss. This corrective feedback loop forces the architecture to reorganise its weights, physically **"pushing"** the mismatched embeddings apart in the vector space to minimise their similarity while simultaneously **"pulling"** real interacting pairs closer together.

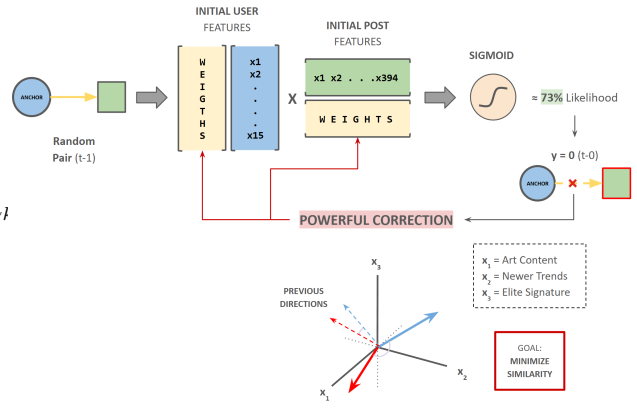

Fig. 12: The Link Prediction Head and Node Similarity

5.5 Real Computation

Because calculating similarity for every possible pair in the graph is computationally impossible, the model utilizes a specialized batch training approach.

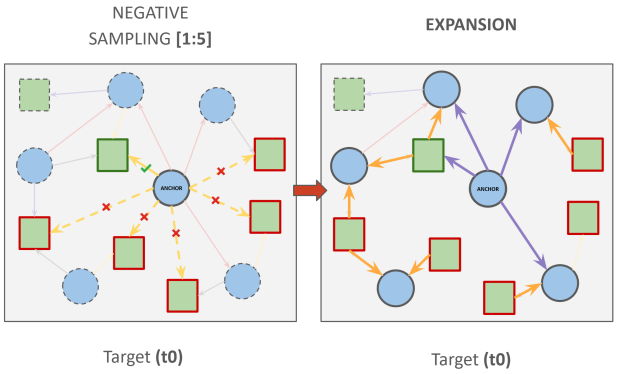

Fig. 13: The Recursive Expansion Process With Negative Sampling (For simplicity, we only show here 1 Hop expnasion)

- **Negative Sampling (1:5 Ratio):** For every real interaction identified in the target snapshot (t_0)—the *Positive Edge*—the model randomly selects five "decoy" nodes that the anchor did not interact with. This process, widely utilized in **Graph Contrastive Learning**, where we force the model to **not learn in a vacuum what a true edge is.**

- **Two-Hop Recursive Expansion:** To provide the necessary "Social Scent" for a decision, the model expands the neighbourhood of these six node candidates. This involves identifying **primary neighbours** (*First Hop*) and then expanding one step further to capture **community context** (*Second Hop*), resulting in a "frozen" structural tree of maximum **900 nodes** since we expand 15 nodes for the first hop and 10 for the second hop (6x15x10).

- **Temporal Nodes Extraction(t_{-1} to t_{-3}):** The model retrieves the attributes and connections for these specific 900 nodes across the previous three snapshots **if they exist**.

Once the 900 nodes identified in the t_0 target snapshot and retrieved from the previous snapshots(t_{-1} to t_{-3}), they undergo the **Attention Model** pipeline to generate the final prediction explained in Section 4.4.

As illustrated in Figure 14, this stage transforms raw historical data into refined Spatio-Temporal Embeddings (Z):

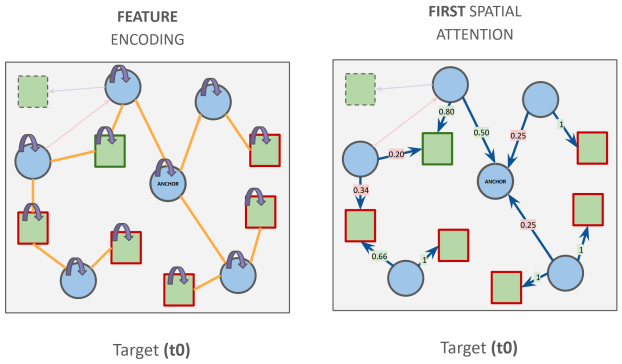

Fig. 14: The Hierarchical Training Pipeline: From Feature Encoding to Spatial Attention (For simplicity, we only show here Spatial Attention Layer)

1. **Feature Encoding:** Before social signals are exchanged, all nodes in the sampled tree undergo feature encoding (5.4.1).

2. **Round 1: Secondary Context ($H_2 \rightarrow H_1$):** The first spatial attention layer operates on the edges between second-hop and first-hop nodes. Target neighbours absorb a community "summary" from their own surroundings (5.4.2).

3. **Round 2: Primary Context ($H_1 \rightarrow$ Target Nodes):** The second spatial attention layer propagates these refined signals to the Anchor Node and The Negative Sampled Nodes (5.4.2).

4. **Spatio-Temporal Fusion:** This hierarchical aggregation is performed independently for each snapshot in the 3-day window. The resulting sequence of spatial summaries $(h_{t-3}, h_{t-2}, h_{t-1})$ is then passed through the Temporal Multi-Head Self-Attention layer(5.4.3).

5. **Link Prediction and Contrastive Evaluation:** Finally, the actualised 128-dimensional spatio-temporal embeddings for the Anchor Node (Z_u) and the target nodes (Z_{target})—including the single real interaction and the five decoys—are compared as explained in Section 4.4.4. The model calculates the interaction probability $\hat{y} = \sigma(Z_u \cdot Z_{target}^T)$ for each pair. In this contrastive setup, we expect the model to assign a significantly higher probability to the real edge ($y = 1$) than to the random decoys ($y = 0$).

While the recursive expansion logic focuses on a single anchor node for clarity, in each training step, the model processes a batch of **128 real edges** ($y = 1$) alongside their corresponding 5×128 fake decoys ($y = 0$). This creates a batch sample of **768 node pairs** that we can compare within the same gradient update.

This batch-driven approach follows a **Sequential Multi-Task Learning protocol**, where the model iterates through the distinct interaction types in a synchronized chain. Rather than mixing different signals into a single gradient step, the pipeline processes iteratively a dedicated batch of Follows, followed by a batch of Likes, and concludes with Reposts.

While the training batches are organized sequentially to specialize in predicting a **unique interaction type**, the Recursive Expansion (Figure 11) and the Hierarchical Aggregation (Figure 12) remain entirely **interaction-blind**. That means it evaluates the importance of a neighbor based solely on its attributes and hidden representation, regardless of whether that neighbor is connected by a Follow, Like, or Repost.

This represents a potential **limitation for future refinement**. By treating all connections as structurally identical during aggregation, the model might **overlook** the nuance that a Repost often signals a much **stronger** topical alignment than a **passive** Follow.

6 Results

The final spatiotemporal embeddings are evaluated across three distinct link prediction tasks: *Follows*, *Likes*, and *Reposts*.

The performance metrics presented in this section are derived from the average evaluation across a 7-day snapshot training. Following the 7th day of training, the model reached a performance plateau, where both the Binary Cross-Entropy (BCE) loss and the corresponding ranking metrics stabilized, indicating that the architecture had successfully captured the latent behavioral patterns within the sampled network.

To evaluate the model's ability to distinguish real interactions from decoys, we utilize a combination of classification and ranking metrics. These are split into two categories: *Classification Strength* (identifying a match) and *Ranking Precision* (positioning the truth within a set of decoys).

1. **Classification Metrics (The Threshold of Intent)**

   - **AUC (0.9376 for Follows):** This is the most critical metric for our $1:5$ contrastive setup. It indicates that given a choice between a **real interaction and a random decoy**, the model correctly identifies the ground truth (real interaction) over **93%** of the time for Follows and around 86-88% for Likes and Reposts.
   - **F1-Score:** Despite having **elite ranking precision**, the F1-Score for Follows is the lowest. This suggests that while the model is very confident about which specific users someone will follow (high precision), it is conservative and misses a larger portion of the total "follow" activity occurring across the network (lower recall). For Likes (0.557) and reposts (0.6616) the model is more effective at capturing the overall volume and general patterns of engagement (Likes and Reposts) even if it cannot rank the specific targets as accurately in a large pool of candidates.

2. **Ranking Metrics (The Batch Competition)** In each training batch, the model generates interaction probabilities for 128 anchor nodes against a combined pool

of 768 candidates (128 real + 640 decoys). The ranking metrics measure where the ground truth sits within this competitive field:

- **Hit Rate (HR@50):** This demonstrates that in nearly 93% of test cases, the real targets are successfully identified within the top 50 candidates suggested by the model in case of the Follow batches. For Likes and Reposts the metric drops below 66%.

  If we compute the node similarity for a node in the batch against all the other sea of 767 nodes and rank the results, the model is **able to find his true interaction above the top 50 ranking**.

- **Hit Rate (HR@5)**the model identifies the correct target within the top 5 most likely candidates in 20.67% of cases for Likes and 22.31% of cases for Reposts. This indicates that while the model is highly precise at predicting follows (82% HR@5 for Follows), it struggles significantly more to rank ground-truth interactions in Reposts and Likes.

- **MRR):** It is calculated as $1/rank$). As the "Gold Standard" for search quality, an MRR of approximately 0.58 suggests that, on average, the real interaction is ranked between the 1st and 2nd position out of the 768 total candidates in case of Follows.

  That means that in the pool of 768 edges, a node is able to find his real interaction if we compute node similarity at an **average of top 2 for Follows bathces and top 6-7 for Likes and Reposts.**

Table 4: MULTI-TASK MODEL PERFORMANCE: EVALUATION OF BINARY CROSS-ENTROPY (BCE) LOSS, AUC, F1-SCORE, AND RANKING METRICS (HIT RATE AND MRR) ACROSS DIFFERENT INTERACTION TYPES.

Metric	Follows	Likes	Reposts
Loss (BCE)	0.6096	0.2981	0.2796
AUC	0.9376	0.8651	0.8871
F1-Score	0.3919	0.5577	0.6116
HR@5	**0.8206**	0.2067	0.2231
HR@50	**0.9264**	0.6107	0.6597
HR@500	0.9504	0.9600	0.9688
MRR	**0.5779**	0.1612	0.1429

7 CONCLUSIONS & FUTURE IMPROVEMENTS

This study successfully demonstrates a robust, scalable framework for link prediction in dynamic heterogeneous social networks. By synthesizing **Snowball-Tiered Sampling** with a two-stage **Spatio-Temporal Attention** mechanism, we effectively condensed 11 billion records into a high-fidelity million-scale environment that preserves the underlying social architecture.

The core achievement of this work is the generation of 128-dimensional spatio-temporal embeddings that represent a node's intrinsic identity, immediate community context, and historical behavioral velocity. Our results provide several key insights:

- **Structural Predictability:** The model excels at identifying short-term structural changes, achieving an elite **AUC of 0.9376** and an **MRR of 0.5779** for new follows. This confirms that the two-hop expansion provides a sufficiently deep "social scent" to accurately capture the structural bridges between clusters.

- **Semantic Integration:** The core novelty of the integration of 384-dimensional `MiniLM` post embeddings directly influenced the **spatio-temporal message-passing flow**. By moving beyond structural IDs and shallow attributes, the model establishes a unified latent space where posts with similar semantic intent are **clustered** alongside users with matching behavioral histories. This allowed the "social scent" to transcend simple graph proximity, enabeling the model to predict interactions based on **topical alignment** rather than just historical frequency.

- **Expanding Graph Resilience:** Our methodology successfuly solved the expanding graph problem allowing for "Historical Interaction" where users can interact with content created up to 7 days in the past while maintaining real-**time behavioral interactions** from the last 24 hours.

While the current architecture provides a powerful foundation for interaction modeling, we can refine the quality of the latent space and the precision of the predictions:

- **Hard Negative Sampling:** Current training utilizes a standard 1:5 random negative sampling ratio. Future iterations could benefit from *Hard Negative Sampling*, where "decoys" are selected based on high semantic similarity to the anchor but no actual interaction. This would force the model to learn even finer distinctions within similar communities.

- **Deletion-Aware Approach:** Currently, the dataset is "addition-only." Incorporating unfollows and unlikes as negative signals would allow the model to learn "negative social scents" identifying not just where a user is going, but what content or communities they are **actively rejecting**.

- **Data Reduction:** Our results suggest that we may not require 20 million posts to create **robust embeddings**. A "smarter" semantic filter that selects posts based on diversity of topics rather than volume could reduce computational overhead while maintaining embedding quality.

- **Long-Term Approach:** While this approach effectively captures immediate interaction spikes, we observed that the model's performance metrics **plateau after seven days of training**. This suggests that the current architecture has reached its maximum capacity for learning **short-term behavioral patterns** but

lacks the temporal depth to interpret more **complex, long-term trends.** Shifting to monthly snapshots with a more selective node-set would allow the model to track "social identity" and personality evolution rather than just transient bursts.

- **Edge-Type Aware Aggregation:** A current limitation lies in the "interaction-blind" nature of the spatial attention phase, where all neighbors are treated as s**tructurally identical regardless of the link type.** Since a *Repost* often signals a much stronger topical alignment than a *Follow*, future refinements could implement **edge-weighted attention**. This would allow the model to explicitly factor in the "strength" of the connection during the message-passing phase.

This project prioritized building a working, large-scale system over fine-tuning every small setting. While we chose "attention" mechanisms to help the model remember complex patterns, we haven't yet compared this directly against simpler methods, like **standard recurrent memory** (GRU or LSTM architectures), to see which is more efficient. Additionally, future tests on settings like learning rates and neighborhood branching factors will be needed to find the perfect balance.

The ultimate power of this methodology lies in its future integration with **synthetic populations.** By enriching these spatio-temporal embeddings with behavioral archetypes, we can transform this predictive tool into a **generative simulation engine,** capable of modeling how different users might co-evolve and interact within a social ecosystemm.

REFERENCES

[1] Leo Balduf. Bluesky Data Repository. https://bsky.leobalduf.com/datasets.html, 2025. Accessed: 2026-01-30.

[2] Leonhard Balduf, Saidu Sokoto, Onur Ascigil, Gareth Tyson, Björn Scheuermann, Maciej Korczyński, Ignacio Castro, and Michał Król. Looking at the blue skies of bluesky. In *Proceedings of the 2024 ACM on Internet Measurement Conference*, 2024.

[3] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[4] Bradley Carron-Arthur, John A. Cunningham, and Kathleen M. Griffiths. Describing the distribution of engagement in an internet support group by post frequency: A comparison of the 90–9–1 principle and zipf's law. *Internet Interventions*, 1(4):165–168, 2014.

[5] Department d'Arquitectura de Computadors i Sistemes Operatius (CAOS). Universitat autònoma de barcelona, 2025.

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[7] Haoyuan Li, Lidia Conde Matos, Eduardo César Galobardes, and Anna Sikora. Applying machine learning for characterizing social networks agent-based models. *arXiv preprint arXiv:2504.21609*, 2025.

[8] Jakob Nielsen. The 90-9-1 rule for participation inequality in social media and online communities. *Nielsen Norman Group*, 2006.

[9] Emanuele Rossi and et al. Temporal graph attention networks for dynamic graphs. *arXiv preprint arXiv:2006.14631*, 2020.

[10] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Wei, and Hao Yang. DySAT: Deep iterative self-attention network for dynamic graph representation learning. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, pages 519–527, 2020.

[11] Universitat Autònoma de Barcelona (UAB), 2025.

[12] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *International Conference on Learning Representations (ICLR)*, 2018.

[13] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 5776–5788, 2020.

[14] Xueqiang Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Peng, Philip S Yu, and Yanfang Ye. Dynamic heterogeneous graph embedding using hierarchical attentions. *Knowledge and Information Systems*, 62:1111–1136, 2020.

[15] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[16] Pan Zhiqiang, Zhao Liang, and et al. Gcn_ma: A graph convolutional network with moving average for dynamic link prediction. *IEEE Access*, 2020.

A RAW DATASET SCHEMA

DETAILED DATASET SCHEMAS

This section provides a comprehensive breakdown of the structural attributes and metadata contained within the processed Parquet files.

profiles.parquet (User Nodes)

Column	Data Type	Description
did_id	Integer	Anonymised unique identifier used as the Primary Key for the User node.
created_at	DateTime	Timestamp representing the account creation date.
description	String	The user's profile biography.
labels	String	Community or safety-related labels applied to the profile.
avatar	Dictionary	Metadata regarding the user's profile picture.
banner	Dictionary	Metadata regarding the user's profile banner image.
joined_via_starter_pack	Boolean	Flag indicating if the account was created via a Bluesky starter pack.
additional_fields	Dictionary	Catch-all for supplementary metadata provided by the API.

Table 5: DETAILED SCHEMA FOR USER NODE DEFINITIONS IN PROFILES.PARQUET.

posts.parquet (Post Nodes)

Column	Data Type	Description
did	String	Original DID of the author (mapped to did_id).
rkey	String	Record key: unique identifier of the Post node.
created_at	DateTime	Timestamp of post publication.
text	String	Raw textual content (used for BERT embeddings).
languages	List	Identified languages in the post content.
is_reply	Boolean	Binary flag indicating if the post is a response.
embed	Dictionary	Contains metadata for external media, images, or website previews.
facets	Dictionary	Contains rich-text features such as mentions (mentions) and hyperlinks (links).
labels	Dictionary	Safety or community-driven labels applied to the post content.
reply	Dictionary	References to the root and parent post IDs for thread reconstruction.
tags	Dictionary	User-defined hashtags or categorizations included in the record.

Table 6: DETAILED SCHEMA FOR POST NODE DEFINITIONS AND ASSOCIATED METADATA DICTIONARIES.

follows.parquet (Follow Edge)

Column	Data Type	Description
did_id	Integer	The did_id of the follower. Used as the source node ID in the graph.
subject_id	Integer	The did_id of the followee. Used as the target node ID in the graph.
created_at	DateTime	Precise timestamp for temporal graph analysis and snapshot filtering.
rkey	String	Unique record key for the follow event on the Bluesky network.

Table 7: DETAILED SCHEMA FOR FOLLOW EDGES IN FOLLOWS.PARQUET.

likes.parquet (Like Edge)

Column	Data Type	Description
did_id	Integer	The ID of the user performing the like.
subject	Dictionary	A structured field containing the target post's author ID and unique rkey.
created_at	DateTime	The temporal marker used to attribute the like to a specific daily snapshot.
rkey	String	The unique identifier for the like record itself on the network.

Table 8: DETAILED SCHEMA FOR LIKE EDGES IN LIKES.PARQUET.

reposts.parquet (Repost Edge)

Column	Data Type	Description
did_id	Integer	The ID of the user performing the repost.
subject	Dictionary	Structured field identifying the target post's author and rkey.
created_at	DateTime	Temporal marker used to assign the interaction to a specific 24-hour snapshot.
rkey	String	The unique identifier for the repost record on the Bluesky network.

Table 9: DETAILED SCHEMA FOR REPOST EDGES IN REPOSTS.PARQUET.

keys.parquet (Anonymization Key)

Column	Data Type	Description
did	String	The raw Decentralized Identifier (DID) from the Bluesky PLC (Place of Lineage and Control).
did_id	String	The internal, anonymized numerical ID used to represent the node in the HGT model and interaction files.

Table 10: SCHEMA FOR ID MAPPING AND ANONYMIZATION IN KEYS.PARQUET.

B STRUCTURAL AND STATISTICAL ATTRIBUTES

NODE ATTRIBUTE SPECIFICATIONS

This section details the specific feature engineering applied to User and Post nodes, distinguishing between static structural properties and dynamic statistical momentum.

Structural Attributes

Structural attributes represent the intrinsic characteristics of a node. These are processed into numerical tensors using various encoding techniques to ensure mathematical compatibility with the GNN layers.

Entity	Attribute	Preprocessing Method	Example / Range
User	has_avatar	Binarization (Existence check)	1 (Yes) / 0 (No)
User	has_banner	Binarization (Existence check)	1 (Yes) / 0 (No)
User	has_description	Binarization (Length ¿ 0)	1 (Yes)
User	description_len	Logarithmic Scaling	2.45 (Normalised)
User	from_starter_pack	Binarization	0 (No)
User	creation_fraction	Temporal Fractioning	0.85 (8:30 PM)
Post	language_cat	Label Encoding (Top 20 + Other)	0 (en), 5 (es)
Post	has_labels	Binarization	0 (Clean)
Post	has_tags	Binarization	1 (Has Hashtags)
Post	has_embed	Binarization (Media/Link)	1 (Image attached)
Post	is_reply	Binarization	0 (Original Post)
Post	creation_fraction	Temporal Fractioning	0.12 (2:50 AM)

Table 11: STRUCTURAL ATTRIBUTE EXTRACTION AND ENCODING LOGIC FOR USERS AND POSTS.

Statistical Attributes (Snapshot Momentum)

Statistical attributes capture the daily activity levels and social influence of nodes. These are aggregated from the raw 11-billion-row dataset to maintain real-world weights.

Node Type	Statistical Metric (Daily Counts)	Aggregation Level
User	posts_count	Total daily publications
User	follows_given	Out-degree (Social activity)
User	follows_received	In-degree (Social authority)
User	likes_given	Interaction volume
User	likes_received	Content resonance
User	reposts_given	Curation frequency
User	reposts_received	Information viralness
User	avg_post_length	Character mean per day
Post	likes_received	Direct engagement count
Post	reposts_received	Direct propagation count

Table 12: DYNAMIC STATISTICAL ATTRIBUTES CALCULATED PER 24-HOUR TEMPORAL SNAPSHOT.

C FIGURES FOR BETTER VISUALIZATION

ANNEX: ARCHITECTURAL VISUALIZATIONS

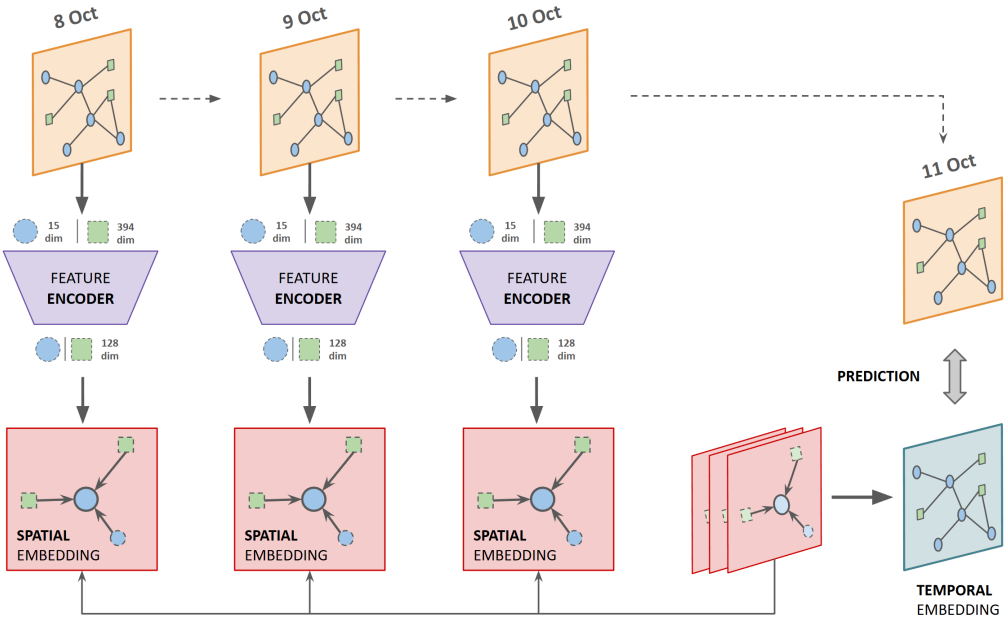

Fig. 15: Full Model Overview: Illustrating the hierarchical flow from Spatial Attention to Temporal Fusion.

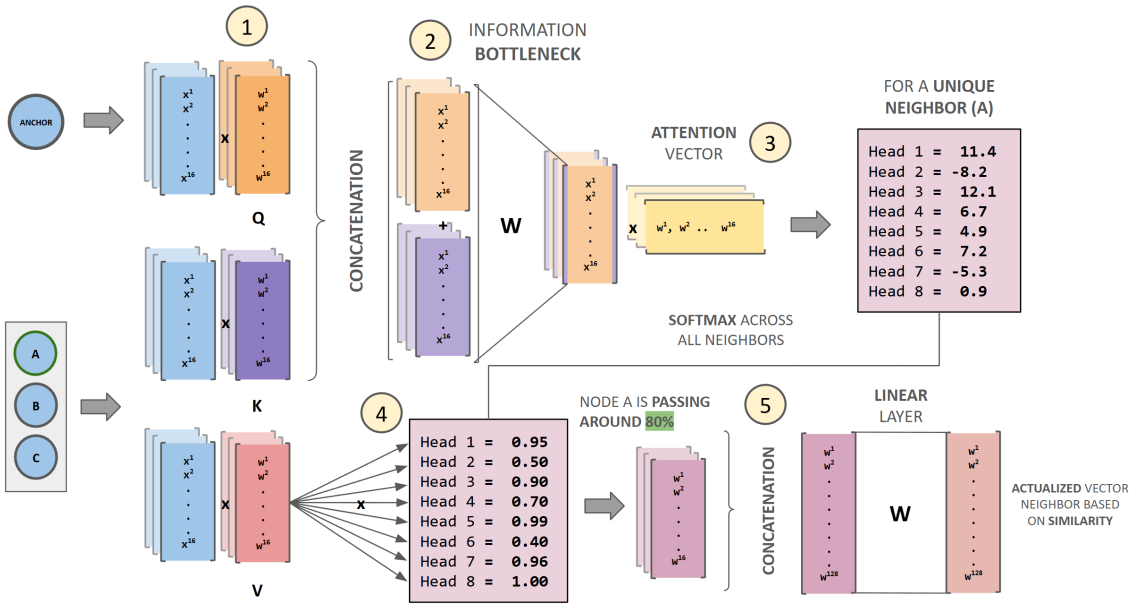

Fig. 16: Multi-Head Attention Pipeline: Detailed breakdown of the Q, K, and V transformations and Softmax weighting.

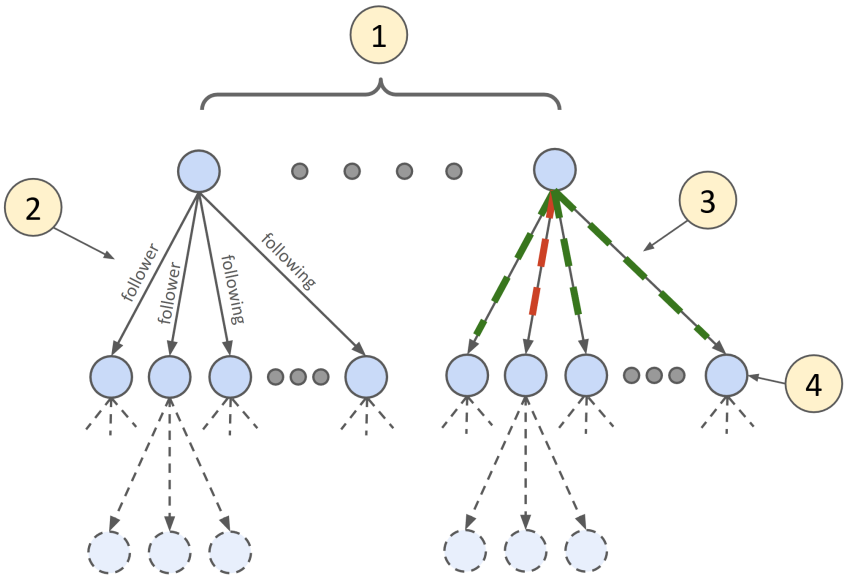

Fig. 17: Snowball Sampling Strategy: The recursive 2-hop neighborhood expansion process.

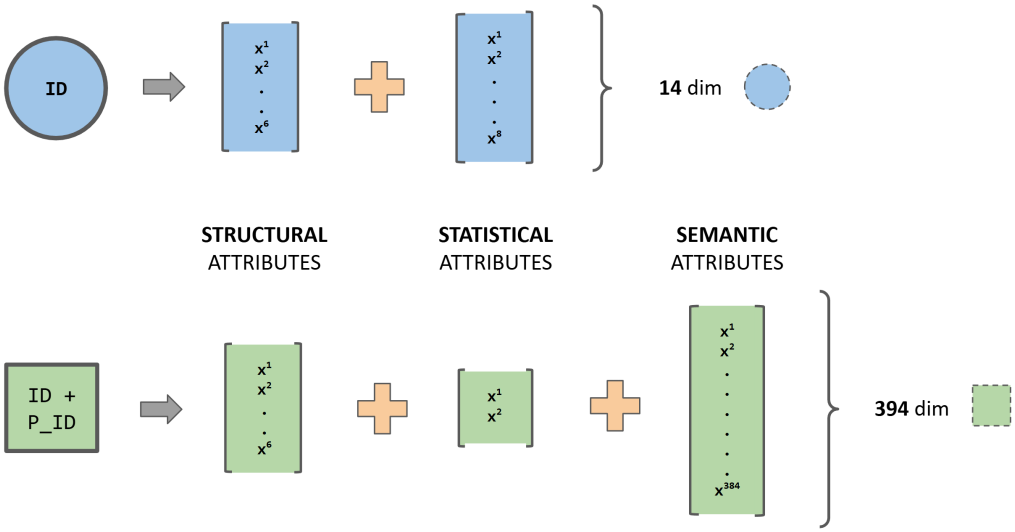

Fig. 18: Dimensionality Flow: Visualizing the transition from 128-dimensional input to 16-dimensional head subspaces.

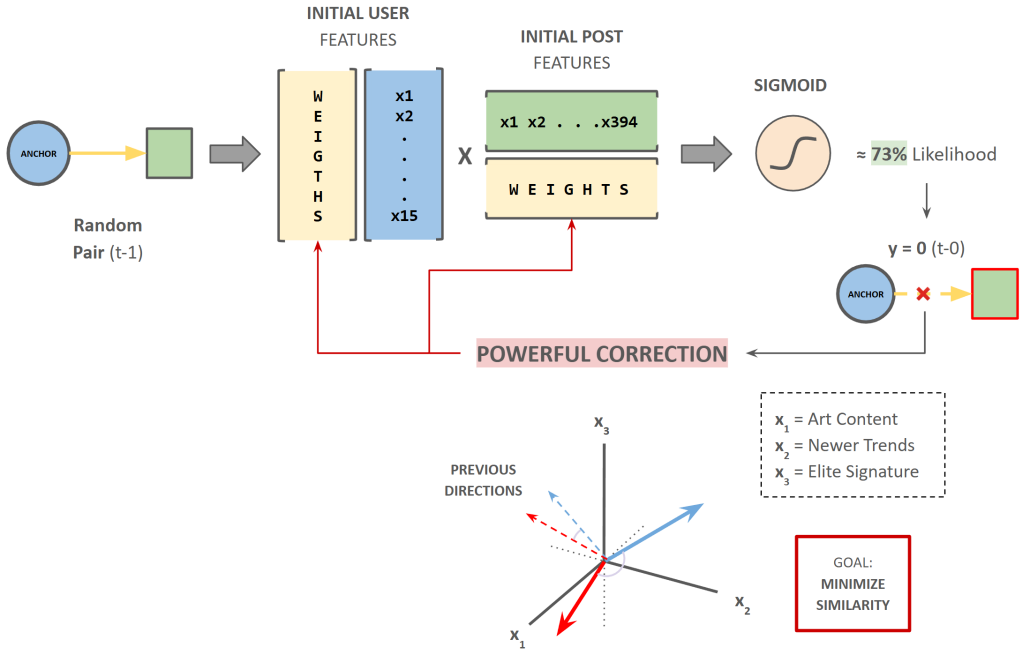

Fig. 19: Link Prediction Head: The final Sigmoid-based probability calculation for the interaction target.

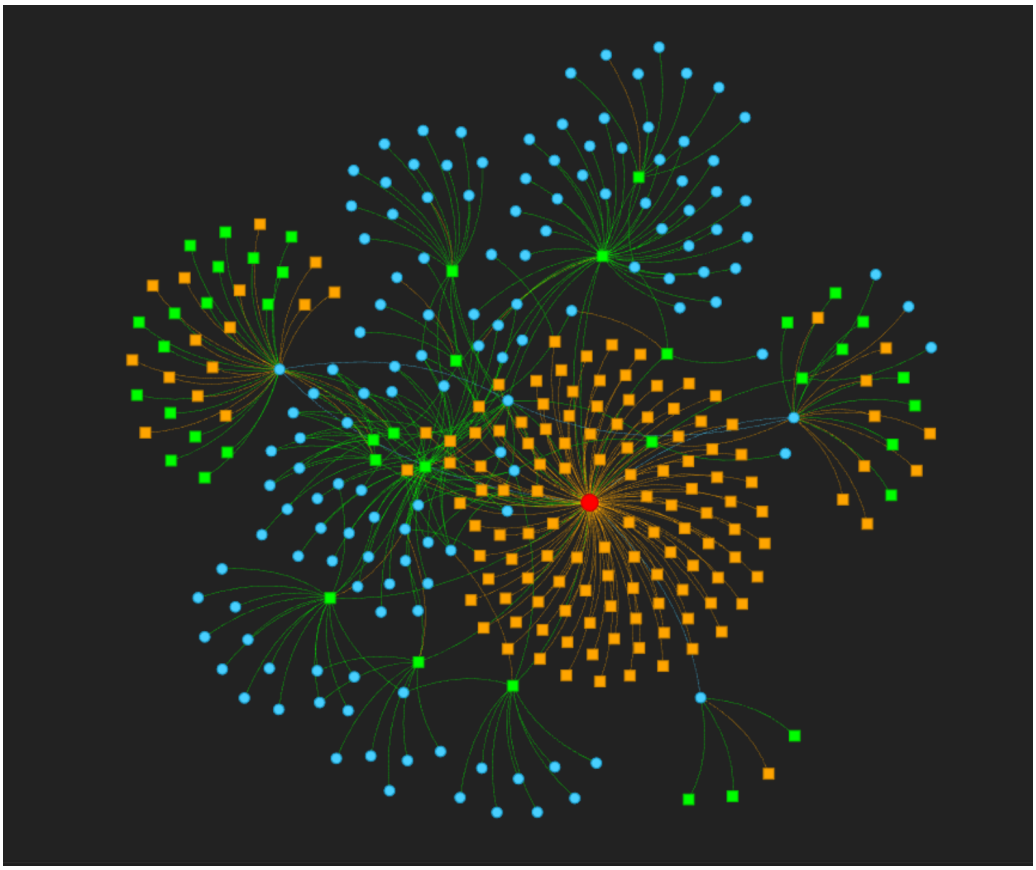

Fig. 20: Real Visualisation of the Graph. Recursive Sampling up to 1000 nodes in a snapshot over a user's (Red) connections. Green Arrows stand for likes and orange stand for user posts