Temporal NBFNet: Path-Based Reasoning over Dynamic Graphs

In21-S7-CS4681 - Advanced Machine Learning - Research Assignment

GNN005

PROGRESS REPORT

Parameswaran Sajeenthiram (210553J)

B.Sc. Engineering (Hons)

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

August 2025

Contents

Abstract

Dynamic graphs capture the evolving nature of real-world relational data, such as knowledge graphs, social networks, and communication logs. Neural Bellman–Ford Networks (NBFNet) have emerged as a powerful path-based model for link prediction, leveraging the recursive structure of the Bellman–Ford algorithm to perform efficient message passing. However, the original NBFNet is limited to static graphs and cannot handle temporal dependencies or evolving structures. This project aims to extend NBFNet into the dynamic graph domain by incorporating temporal embeddings, time-aware decay functions, and query-time masking. Within the constraints of a 10-week project, the focus is on implementing a temporal variant of NBFNet and benchmarking it on standard temporal knowledge graph datasets (ICEWS14, ICEWS18, GDELT). This report presents a literature review of NBFNet and temporal graph learning, outlines the project planning and methodology, and provides a timeline for implementation and evaluation.

1 Introduction

Graphs are powerful abstractions that represent entities (nodes) and their relationships (edges). They form the foundation for a wide variety of real-world domains, such as biological interaction networks, recommendation systems, transportation systems, and knowledge graphs. Traditionally, many graph learning models assume static structures. However, in practice, graphs evolve continuously: new interactions form, entities emerge, and relationships change over time. Capturing these dynamics is crucial for accurate prediction, reasoning, and decision-making.

Neural Bellman–Ford Networks (NBFNet) [1] represent a class of graph neural networks designed to reason over paths. By embedding the recursive computation of shortest paths into a neural framework, NBFNet has achieved state-of-the-art results in static link prediction tasks. Its interpretability and efficiency have made it popular for reasoning over heterogeneous and knowledge graphs.

Despite its success, NBFNet remains limited to static graphs. In contrast, dynamic or temporal graphs pose additional challenges:

- Edges are time-stamped, and reasoning must respect temporal ordering.
- Node states evolve with new interactions.
- Long-term dependencies across time horizons must be modeled.

These challenges have motivated the development of temporal graph neural networks (e.g., TGN, TGAT, DyRep). However, none of these models inherit the path-based reasoning and interpretability advantages of NBFNet.

Research Problem: *How can NBFNet's path-based recurrence be extended to dynamic graphs, while preserving efficiency and interpretability?*

The core idea is to integrate temporal embeddings, time decay mechanisms, and query-time masking into the Bellman–Ford recurrence. This allows path reasoning to remain time-aware without a complete architectural redesign.

2 Implementation Plan

2.1 Objectives

- Extend NBFNet with temporal reasoning by integrating time into its message-passing operations.
- Implement a Temporal NBFNet prototype within 8 weeks, focusing on temporal embeddings, decay, and query-time masking.
- Evaluate the model on benchmark dynamic graph datasets.
- Compare against static NBFNet and selected temporal baselines (TGAT, TGN).

2.2 Scope Clarification

Possible extensions (memory modules, attention mechanisms, online learning) are acknowledged. Due to time constraints, the project will focus only on:

- Sinusoidal or learned temporal embeddings.
- Temporal decay weighting.
- Query-time masking to prevent "future leakage."

2.3 Expected Contributions

- A novel Temporal NBFNet (TNBFNet) framework.
- Open-source code with reproducibility.
- Empirical results showing improvements over static NBFNet on temporal datasets.
- A conference-ready research paper.

2.4 Detailed Project Timeline

Weeks 1–2 (Aug 13 – Aug 26): Literature Review

Focus on reviewing:

- NBFNet and path-based reasoning models.
- Temporal/dynamic GNNs (TGN, TGAT, DyRep, Jodie).
- Temporal knowledge graph reasoning models (TLogic, TNCN, TeMP).

Organize references into a structured hierarchy and prepare comparative notes. **Outcome:** research gaps identified to inform methodology.

Weeks 3–6 (Aug 27 – Sep 16): Model Design & Prototype Implementation

Design Temporal NBFNet with temporal Bellman–Ford recurrence. Implement prototype in PyTorch/TorchDrug, starting with temporal encoder and modified message passing.

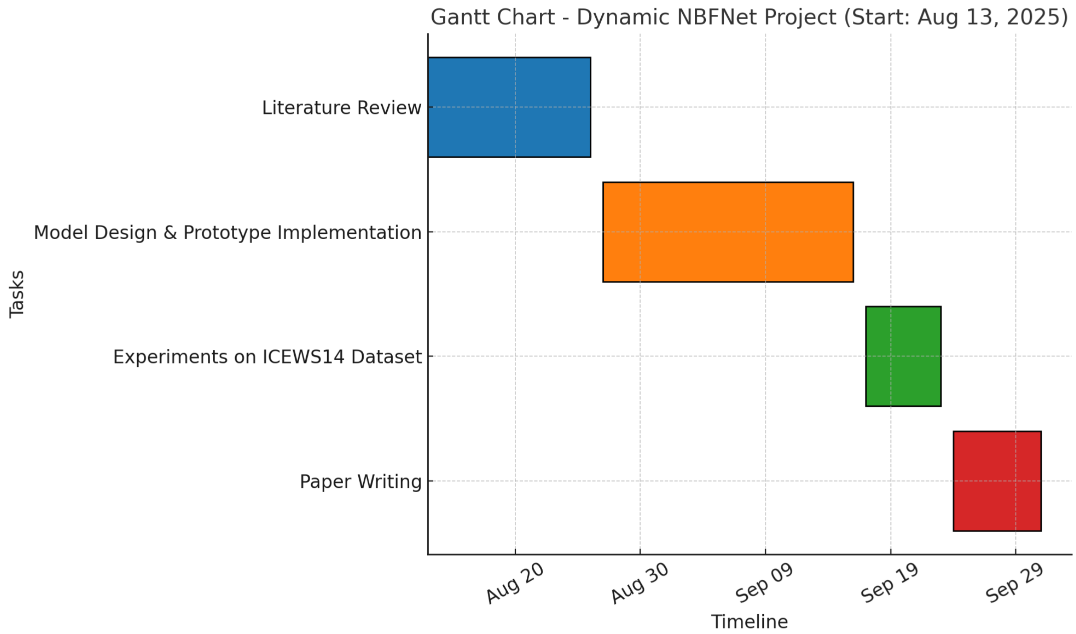

Figure 1: Timeline

Weeks 7 (Sep 17 – Sep 23): Experiments on ICEWS14 Dataset

Train and validate Temporal NBFNet. Compare against NBFNet, TGAT, TGN. **Deliverable:** preliminary results with reproducible scripts.

Week 8 (Sep 24 – Oct 1): Paper Writing

Finalize Paper with tables, figures, methodology, experiments, discussion, and future work. Ensure reproducibility and clean code documentation.

3 Literature Review

Graphs are a fundamental representation for relational data, capturing entities as nodes and their interactions as edges. They are widely used in domains such as social networks, biology, knowledge graphs, and recommender systems. Traditional machine learning models, which rely on fixed-dimensional vector representations, fail to capture the intricate dependencies embedded in graph structures. Consequently, Graph Neural Networks (GNNs) have emerged as a powerful framework for learning expressive node and edge representations by iteratively aggregating information from local neighborhoods [2, 3]. Early graph learning research focused primarily on static graphs, where the topology is assumed to remain unchanged over time. In this context, link prediction—the task of inferring missing or unobserved edges—has been a benchmark for evaluating the representational capacity of graph models. Among recent advances, Neural Bellman-Ford Networks (NBFNet) [1] combine path-based reasoning with differentiable message passing to provide interpretable multi-hop link predictions. However, real-world graphs are often dynamic: nodes and edges can appear, disappear, or evolve over time, introducing additional complexity. Addressing temporal dynamics requires models capable of capturing both structural and temporal dependencies while maintaining scalability and interpretability. This literature review explores GNN-based approaches for static and dynamic graphs, emphasizing link prediction, path-based reasoning, and the research gap that motivates the extension of NBFNet to dynamic settings.

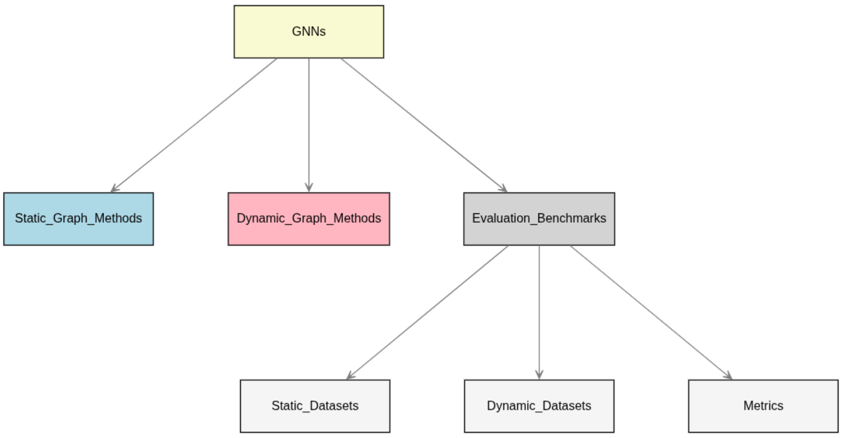

Figure 2: Literature review hierarchy.

3.1 Foundations of Graph Neural Networks

Graph Neural Networks generalize deep learning to graph-structured data by aggregating information from node neighborhoods to update node embeddings. In the message-passing framework, each node combines its current feature representation with messages received from adjacent nodes, effectively capturing local graph structure [2]. The Graph Convolutional Network (GCN) [4] was one of the first widely adopted architectures, applying spectral convolutions to propagate information across nodes. The Graph Attention Network (GAT) [5] introduced an attention mechanism to weigh contributions from different neighbors, while GraphSAGE [6] employs neighborhood sampling to enable inductive learning on large graphs. From a theoretical perspective, GNNs can be analyzed through the lens of the Weisfeiler–Lehman (WL) graph isomorphism test; Xu et al. [7] connect GNN expressivity with the WL test and analyze the limits of message passing. Despite their effectiveness, static GNNs face limitations. Deep networks may suffer from oversmoothing, where node embeddings converge to similar values, reducing discriminative power. Moreover, they do not account for temporal evolution, making them inadequate for dynamic graphs where relationships change over time.

3.2 Static Link Prediction

Static link prediction aims to infer missing edges in a graph with a fixed structure. Early approaches relied on topological heuristics, including common neighbors, Adamic–Adar, and preferential attachment (e.g., [8]). While computationally efficient, these heuristics are limited in capturing multi-hop relational dependencies or complex semantic relationships. Embedding-based approaches advanced link prediction by representing nodes and relations in a latent space. Translational models such as TransE [9] learn embeddings such that the vector of the head entity plus the relation vector approximates the tail entity. Bilinear models like DistMult [10] and ComplEx [11] factorize adjacency tensors to model interactions between entities and relations.

These methods achieve strong performance on benchmarks such as FB15k and WN18 [9] and WN18RR [12], but are limited in handling multi-hop reasoning or interpretability. GNN-based static link prediction leverages neighborhood aggregation to learn richer node representations. Relational GCNs (R-GCNs) [13] extend GCNs to multi-relational graphs, aggregating information across heterogeneous edge types, and have demonstrated strong performance on knowledge graph completion. However, traditional GNNs struggle with long-range reasoning due to shallow receptive fields and oversmoothing. Path-based reasoning models explicitly model relational paths rather than relying solely on embedding similarity. Neural LP [14] learns differentiable logical rules for reasoning over knowledge graphs, while MINERVA [15] applies reinforcement learning to navigate paths between entities. NBFNet [1] introduces a differentiable version of the Bellman–Ford algorithm, propagating relational information along paths to enable interpretable multi-hop reasoning. NBFNet achieves state-of-the-art results on static link prediction benchmarks, combining accuracy with human-readable reasoning chains. Yet, NBFNet's static assumption limits its applicability in dynamic or temporal graphs.

3.3 Dynamic Graph Representation Learning

Dynamic link prediction extends static link prediction to evolving graphs, forecasting future edges based on historical interactions. Snapshot-based dynamic link prediction applies static link predictors sequentially across temporal snapshots, but suffers from coarse temporal resolution. Continuous-time dynamic link prediction leverages temporal embeddings and memory updates. TGAT [16] uses time-aware attention to incorporate edge timestamps, while TGN [17] updates node memories dynamically. DyRep [18] estimates interaction likelihoods using temporal point processes. Temporal knowledge graph completion (TKGC) extends these ideas to knowledge graphs with time-stamped facts. Models such as Know-Evolve [19], HyTE [20], RE-Net [21], and CyGNet [22] predict not only which interactions occur but also their temporal order. Despite these advances, dynamic models typically lack explicit, interpretable path-based reasoning, leaving a critical gap for temporal extensions of NBFNet.

3.4 Dynamic Link Prediction and TKGC

Dynamic link prediction aims to forecast future edges. Snapshot-based predictors are coarse; continuous-time ones use temporal embeddings and memory. Temporal KGC models include Know-Evolve [19], HyTE [20], RE-Net [21], and CyGNet [22]. However, these do not preserve path-based interpretability.

3.5 Path-based Reasoning and NBFNet

Path-based reasoning offers interpretability in link prediction by tracing multi-hop relational chains. NBFNet [1] propagates relational signals along paths using a differentiable Bellman–Ford formulation. On static benchmarks such as FB15k-237 and WN18RR, NBFNet achieves high accuracy while generating human-readable reasoning chains. However, NBFNet cannot handle temporal graphs, limiting its ability to predict future interactions or capture evolving relationships.

Bridging this gap requires integrating temporal embeddings or memory mechanisms into path-based reasoning.

3.6 Bridging Path Reasoning with Temporal Graphs

Recent research attempts to incorporate temporal reasoning into path-based frameworks. RE-Net [21] encodes event paths using recurrent architectures, CyGNet [22] models cyclical temporal patterns. While these approaches improve temporal interpretability, they lack the structured dynamic-programming foundation of NBFNet. Conversely, embedding-based temporal models like TGN [17] and TGAT [16] effectively capture temporal dependencies but do not provide explicit reasoning chains. The combination of path-based reasoning and temporal modeling motivates the development of a Temporal NBFNet, integrating Bellman–Ford-style propagation with time-aware embeddings and memory updates. This approach promises interpretable multi-hop reasoning for dynamic link prediction.

3.7 Evaluation Datasets

Static link prediction is typically evaluated on FB15k and WN18 [9], FB15k-237 and WN18RR [12], using metrics such as Hits@K and Mean Reciprocal Rank (MRR). These datasets provide standardized benchmarks for evaluating multi-relational link prediction models, including heuristic, embedding-based, GNN-based, and path-based reasoning methods such as NBFNet [1]. Dynamic link prediction and temporal knowledge graph completion rely on ICEWS14/ICEWS05-15 [23], GDELT [24], and YAGO-T [25]. ICEWS contains timestamped political event records, while GDELT tracks global events with fine-grained temporal resolution. YAGO-T is a temporal extension of the YAGO knowledge graph that encodes time-stamped facts for evolving relationships. Evaluation on these datasets extends beyond static accuracy to temporal forecasting measures, such as predicting the occurrence and timing of future interactions. Scalability is also assessed based on node and edge counts, temporal resolution, and computation time, enabling comparison across models and highlighting computational challenges inherent to dynamic graph reasoning.

3.8 Applications and Implications

Graph learning models are applied across multiple domains. In knowledge graph completion, path-based reasoning enhances interpretability. Temporal recommendation systems benefit from dynamic embeddings to predict user–item interactions. Social network analysis leverages dynamic GNNs to forecast evolving interactions and communities. Finally, GraphRAG pipelines integrate interpretable temporal reasoning with retrieval-augmented generation [26], improving the relevance and recency of answers in question answering systems.

3.9 Research Gap

Static link prediction models, including NBFNet, excel at multi-hop reasoning but do not accommodate temporal evolution. Dynamic models capture temporal dependencies but lack explicit, interpretable reasoning chains. Bridging these paradigms motivates the development of Temporal NBFNet, which integrates structured path reasoning with dynamic embeddings and memory updates. Such a model addresses both accuracy and interpretability challenges in temporal link prediction, offering a new benchmark for dynamic graph reasoning.

4 Methodology

Knowledge graphs (KGs) represent structured facts as triples $\langle h, r, t \rangle$, where h and t are entities and r is a relation. For example, $\langle \text{Einstein}, \text{bornIn}, \text{Ulm} \rangle$ indicates that Einstein was born in Ulm. The task of knowledge graph completion (KGC) is to predict missing links; e.g., given $\langle \text{Einstein}, \text{bornIn}, ? \rangle$, infer the correct tail entity.

4.1 Early Approaches: Embedding-based Models

TransE and family: Entities and relations are represented in a latent space, and triples are scored by algebraic constraints such as $h + r \approx t$. **DistMult / ComplEx:** Bilinear forms enable relation-specific interactions.

These approaches are computationally efficient and effective for local patterns but struggle with structured *multi-hop* reasoning. For example, the chain reasoning: "X is a citizen of Y, Y is in continent Z $\Rightarrow$ X lives in continent Z" cannot be modeled naturally.

4.2 GNN Approaches: Message Passing

Graph Neural Networks (GCNs, GATs, R-GCNs) propagate messages along edges in a neighborhood. However:

- They operate in fixed-radius neighborhoods. Multi-hop reasoning requires stacking many layers.
- Deep stacks risk *over-smoothing*, making node embeddings indistinguishable.
- They blur *path semantics*, which are critical for relational reasoning.

4.3 Neural Bellman–Ford Networks (NBFNet)

NBFNet [1] proposes a path-based alternative inspired by the classical Bellman–Ford shortest path algorithm.

4.3.1 Bellman–Ford Recap

The Bellman–Ford algorithm computes shortest paths by iteratively updating node distances:

$$d_v^{(k)} = \min_{(u,v) \in E} d_u^{(k-1)} + w(u, v). \tag{1}$$

At each iteration k, the distance to node v is updated from neighbors' distances plus edge weights.

4.3.2 NBFNet Reformulation

NBFNet generalizes this to a neural recurrence:

$$h_v^{(k)} = \bigoplus_{(u,r,v)\in E} f_\theta(h_u^{(k-1)}, r), \tag{2}$$

where:

- $h_v^{(k)}$: representation of node v after k-hop reasoning.
- f_θ: relation-specific transformation function.
- $\oplus$: learnable aggregation (e.g., sum, mean, or attention-based).

Instead of computing distances, NBFNet aggregates *relational paths*, combining multi-hop evidence.

4.3.3 Boundary Condition (Query Injection)

For predicting $\langle h, r, ?\rangle$, a query embedding $q = \mathrm{Emb}(r)$ is injected at the head entity h. The Bellman–Ford recurrence propagates this query through the graph, enabling path-sensitive reasoning toward candidate tails t.

4.3.4 Scoring

After K_{hop} iterations (capturing paths up to length K_{hop}), the candidate tail t is scored as:

$$s(h, r, t) = \mathrm{MLP}\Big([h_t^{(K_{\text{hop}})}, q]\Big). \tag{3}$$

Strengths: Multi-hop reasoning, avoidance of over-smoothing, and explainable path attribution (by backtracking recurrences).

Limitation: NBFNet assumes a static KG; it does not enforce temporal causality in dynamic settings.

4.4 Extending to Temporal Knowledge Graphs

A temporal KG (TKG) encodes quadruples $\langle h, r, t, \tau\rangle$, where τ is the timestamp. A static KG is a special case where τ is absent.

The temporal KGC task is: given $\langle h, r, ?, \tau_q\rangle$, predict the correct tail valid at query time τ_q.

Key challenges:

- *Causality*: Only past edges ($\tau_e \leq \tau_q$) can be used.
- *Recency*: Recent facts should weigh more heavily than older ones.
- *Temporal patterns*: Relations may be periodic or evolve over time.

4.4.1 Temporal Neural Bellman–Ford Networks (T-NBFNet)

We extend NBFNet with three mechanisms: (i) time encoding, (ii) temporal decay, and (iii) causal masking.

4.4.2 Temporal Edge Encoding

Each timestamp τ_e is mapped to a sinusoidal encoding $\phi(\tau_e)$ and is concatenated to the relation embedding when forming edge messages.:

$$\phi(\tau_e) = [\sin(\tau_e \cdot \omega_1), \cos(\tau_e \cdot \omega_1), \ldots, \sin(\tau_e \cdot \omega_{d/2}), \cos(\tau_e \cdot \omega_{d/2})], \tag{4}$$

where $\{\omega_i\}$ are logarithmically spaced frequencies, and d is the dimensionality of the encoding (set equal to the hidden dimension of messages). This captures multi-scale periodicity, following TGAT [16].

4.4.3 Temporal Decay and Causal Masking

Define decay weight:

$$w(e, \tau_q) = \begin{cases} \exp\Big(-\ln 2 \cdot \frac{\tau_q - \tau_e}{H}\Big), & \tau_e \leq \tau_q,\ (\tau_q - \tau_e) \leq W, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where:

- H: half-life hyperparameter (relation-independent, tuned on validation).
- W: window size (optional; $W = \infty$ reduces to full history).

Rules:

- If $\tau_e > \tau_q$, the edge is masked (causality).
- If $\tau_q - \tau_e > W$, the edge is excluded regardless of decay.
- Otherwise, older events decay smoothly with rate controlled by H.

4.4.4 Temporal Relational Convolution

The message at hop k becomes:

$$m_{u\to v}^{(k)} = f_\theta(h_u^{(k-1)}, r_{u,v}, \phi(\tau_{u,v})) \cdot w((u, v), \tau_q). \tag{6}$$

Only temporally valid edges contribute.

4.4.5 Temporal Bellman–Ford Recurrence

$$h_v^{(k)} = \text{Aggregate}\{m_{u\to v}^{(k)} : \tau_{u,v} \leq \tau_q\}. \tag{7}$$

We use Principal Neighborhood Aggregation (PNA), which combines mean, max, min, and standard deviation with degree scaling. Alternative aggregators are possible, but PNA is our default choice.

4.4.6 Scoring

After K_{hop} hops:

$$s(h, r, t, \tau_q) = \mathrm{MLP}\Big([h_t^{(K_{\text{hop}})}, q, \phi(\tau_q)]\Big)\,, \tag{8}$$

where $q = \mathrm{Emb}(r)$ and $\phi(\tau_q)$ injects query-time information.

4.4.7 Training with Time-Aware Negatives

We use binary cross-entropy (BCE). For a positive $\langle h, r, t, \tau\rangle$, negatives are corrupted quadruples $\langle h, r, t', \tau\rangle$ such that no fact (h, r, t', τ') exists with $\tau' \leq \tau$. Negatives are always sampled at the same query time τ to avoid false negatives from past valid facts.

The loss is:

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B}\Big[\log\sigma(s_i^+) + \tfrac{1}{K_{\text{neg}}}\sum_{j=1}^{K_{\text{neg}}}\log\sigma(-s_{ij}^-)\Big], \tag{9}$$

where:

- B: batch size (number of positive quadruples in a batch).
- i: index over positive samples in the batch.
- K_{hop}: number of message-passing hops.
- K_{neg}: number of negative samples per positive.
- s_i^+: score of the positive quadruple $\langle h, r, t, \tau\rangle$.
- s_{ij}^-: score of the j-th negative sample for the i-th positive.
- $\sigma(\cdot)$: sigmoid function.

4.4.8 Evaluation Protocol

We follow filtered temporal ranking [16, 21]. For each query, candidate entities that formed valid facts $\tau \leq \tau_q$ are filtered out. Metrics:

- Mean Rank (MR),
- Mean Reciprocal Rank (MRR),
- Hits@K (e.g., 1, 3, 10).

To validate the effectiveness of TNBFNet, we compare against TGN [17], Know-Evolve [19], HyTE [20], RE-Net [21], and CyGNet [22].

Datasets with Reported Results for TGN

- ICEWS14 (political event prediction)
- ICEWS18 (extended version with more events)
- GDELT (global event dataset with fine temporal granularity)

These are the same datasets used for T-NBFNet, enabling a direct comparison of metrics such as MRR, Hits@K, and mean rank. This evaluation highlights the improvements offered by path-based, time-aware reasoning over standard temporal GNNs like TGN.

5 Conclusion

In this progress report, we presented the motivation, literature review, and methodology for extending Neural Bellman–Ford Networks (NBFNet) to dynamic graph settings. The proposed Temporal NBFNet (TNBFNet) introduces temporal embeddings, decay mechanisms, and query-time masking to capture evolving relational patterns while maintaining path-based interpretability.

So far, the project has clarified the problem scope, identified relevant datasets (ICEWS14, GDELT), and outlined the design of the temporal recurrence mechanism. The next stage will involve implementing the prototype, running initial experiments, and benchmarking TNBFNet against established temporal graph models such as TGAT and TGN.

Overall, the work is on track to deliver a reproducible framework, comparative evaluation, and insights into the trade-offs between interpretability and temporal modeling. The final phase will focus on experiments, ablation studies, and preparing the research output.

References

[1] Zhaocheng Zhu, Keyulu Xu, Tommi Yu, Stefanie Jegelka, Stefanie Jegelka, Shing-Hon Wong, Jie Chen, Rong Zhang, Qiaozhu Li, and Jie Zhou. Neural bellman-ford networks: A general graph neural network framework for link prediction. In *Advances in Neural Information Processing Systems*, volume 34, pages 29476–29490, 2021. URL `https://papers.nips.cc/paper/2021/hash/7d6044e64d97bea6a756dee07de8d3eb-Abstract.html`.

[2] Jie Zhou, Ganqu Cui, Shengding Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 2020.

[3] 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*, 2020.

[4] Haiqi Zhang, Haibo Zhang, Yefeng Zheng, Xudong Gao, Qingfeng Wu, and Naixue Xiong. Semi-supervised classification of graph convolutional networks with laplacian

rank constraints. Springer, January 2021. doi: 10.1007/s11063-020-10404-7. URL `https://doi.org/10.1007/s11063-020-10404-7`.

[5] Petar Veličković, Guillem Cucurull, Adriana Romero Casanova, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[6] William Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[7] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[8] Mark Newman. Clustering and preferential attachment in growing networks. In *Physical Review E*, 2001.

[9] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NeurIPS*, 2013.

[10] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, 2015.

[11] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *ICML*, 2016.

[12] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *AAAI*, 2018.

[13] Michael Schlichtkrull, Thomas Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *ESWC*, 2018.

[14] Fan Yang, Zhilin Yang, and William Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *NeurIPS*, 2017.

[15] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alexander Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *ICLR*, 2018.

[16] Da Xu, Chuan Ruan, Ismail Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *ICLR*, 2020.

[17] Emanuele Rossi, Benjamin Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *ICLR*, 2020.

[18] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *ICLR*, 2019.

[19] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *ICML*, 2017.

[20] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. Hyte: Hyperplane-based temporally aware knowledge graph embeddings. In *EMNLP*, 2018.

[21] Woojeong Jin, Meng Qu, Xiang Ren Pan, and Xiang Ren. Recurrent event network: Global structure inference over temporal knowledge graph. In *NeurIPS*, 2020.

[22] Cunchao Zhu, Guoqing Cheng, Meng Xu, Zhen Zhang, and Shenghua Gao. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. December 2020. doi: 10.48550/arXiv.2012.08492. URL `https://doi.org/10.48550/arxiv.2012.08492`. Accessed: 2024-08-29.

[23] Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, and Steve Shellman. Integrated crisis early warning system (icews) event data, 2015. Available at https://dataverse.harvard.edu/dataverse/icews.

[24] Kalev Leetaru and Philip Schrodt. Gdelt: Global database of events, language, and tone. *International Studies Association*, 2013.

[25] Farzaneh Mahdisoltani, Asia Biega, and Fabian Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*, 2013.

[26] Chen Xiao, Qian Chen, Wenjie Li, and Yongfeng Zhang. Graphrag: Retrieval-augmented graph reasoning. *arXiv preprint arXiv:2312.13722*, 2023.