# Temporal NBFNet: A Temporal Extension of Neural Bellman–Ford Networks for Dynamic Graph Reasoning

Parameswaran Sajeenthiran[1] and Uthayasanker Thayasivam[2]

1- Department of Computer Science and Engineering, University of Moratuwa
Moratuwa, Sri Lanka

2- Department of Computer Science and Engineering, University of Moratuwa
Moratuwa, Sri Lanka

**Abstract**. Dynamic graphs capture evolving relationships in domains such as knowledge graphs, social networks, and event streams. While Neural Bellman–Ford Networks (NBFNet) provide interpretable and efficient path-based reasoning for static link prediction, they cannot handle temporal dependencies. To bridge this gap, we propose a Temporal Neural Bellman–Ford Network (T-NBFNet) that integrates time encodings, decay weighting, query-time masking to enforce causality, and memory modules inspired by Temporal Graph Networks for capturing long-term node states. Preliminary results suggest that T-NBFNet maintains interpretability of path reasoning while performing well on dynamic graphs.

## 1 Introduction

Graphs naturally represent relational data, where nodes denote entities and edges capture relationships. In dynamic environments such as social networks, recommendation systems, and temporal knowledge graphs (TKGs), these relations evolve continuously, requiring models that reason over both structure and time. Temporal Graph Neural Networks (GNNs) address this challenge by integrating temporal awareness into graph learning. However, most existing approaches rely on localized message passing and fail to capture multi-hop temporal dependencies that are essential for interpretability and long-term reasoning. To address this limitation, we propose the **Temporal Neural Bellman–Ford Network (T-NBFNet)**—a temporal extension of NBFNet [1] that performs interpretable multi-hop reasoning across time using a differentiable Bellman–Ford recurrence. The implementation is available at `https://github.com/aaivu/In21-S7-CS4681-AML-Research-Projects/tree/main/projects/210553J-GNN_Knowledge-Graphs`.

## 2 Related Work

Graph Neural Networks (GNNs) extend deep learning to non-Euclidean domains by propagating and aggregating information between connected nodes. Early models such as Graph Convolutional Networks (GCN) [2], Graph Attention Networks (GAT) [3], and GraphSAGE [4] demonstrated strong performance on various graph tasks but often suffer from oversmoothing and limited receptive fields. Dynamic graph learning extends these ideas to evolving networks where nodes and edges appear or disappear over time. Snapshot-based approaches sequentially apply static predictors, while continuous-time methods introduce temporal embeddings or recurrent memory updates. Models such as TGAT [5], TGN [6], and DyRep [7] capture fine-grained temporal dependencies using attention mechanisms and memory modules. Although RE-Net [8] and CyGNet [1] model event sequences, they lack structured dynamic programming capabilities. T-NBFNet bridges this gap by unifying temporal modeling and path-based reasoning.

## 3 Methodology

This section first summarizes the Neural Bellman–Ford Network (NBFNet) as the foundation of our model, and then introduces its temporal extensions, leading to the complete Temporal Neural Bellman–Ford Network (T-NBFNet).

### 3.1 Background: Neural Bellman–Ford Networks

The Neural Bellman–Ford Network (NBFNet) [1] reformulates the classic Bellman–Ford shortest-path algorithm into a differentiable message-passing framework for link prediction. Given a query triple $(u, r, v)$, the model recursively aggregates relational messages along possible paths from $u$ to $v$.

Each node $v$ maintains a hidden representation $h_v^{(t)}$ at layer $t$, initialized as:

$$h_v^{(0)} = \mathrm{INDICATOR}(u, v, q), \tag{1}$$

where the indicator encodes whether $v$ is the source or target node of the query.

At each layer $t$, the node representation is updated by aggregating messages from its neighbors:

$$h_v^{(t)} = \mathrm{AGGREGATE}\big(\{\mathrm{MESSAGE}(h_x^{(t-1)}, w_q(x, r, v))\} \cup \{h_v^{(0)}\}\big), \tag{2}$$

where $w_q(x, r, v)$ applies a relation-specific transformation. This recurrence simulates multi-hop path reasoning but remains limited to static graphs.

### 3.2 Temporal Extensions

To handle dynamic graphs, T-NBFNet introduces three key temporal mechanisms—temporal encoding, temporal decay with causal masking, and memory-based updates—while preserving NBFNet's recursive path structure.

*(1) Temporal Encoding.* Each timestamp $\tau$ is represented using a sinusoidal encoding that captures periodic and continuous temporal patterns:

$$\mathrm{TimeEnc}(\tau) = [\sin(\tau\omega_i), \cos(\tau\omega_i)]_{i=1}^{d/2}, \tag{3}$$

where $\omega_i$ are fixed frequencies. This encoding allows the network to perceive time as a smooth, multi-scale signal.

*(2) Temporal Decay and Masking.* Let $\tau_q$ denote the query timestamp (i.e., the time at which the model makes a prediction), and $\tau_e$ denote the timestamp of an observed event or edge in the temporal knowledge graph. To ensure that the model adheres to causal reasoning, we enforce two temporal rules. **Causality:** Any edge with a timestamp $\tau > \tau_q$ (i.e., occurring after the query time) is ignored during message passing. **Recency weighting:** Each valid past edge is scaled by a time-dependent decay weight $w(\Delta\tau)$:

$$w(\Delta\tau) = \begin{cases} \exp\left(-\frac{\log 2}{\lambda} \cdot \Delta\tau\right) & \text{(exponential decay)} \\ \max\left(0, 1 - \frac{\Delta\tau}{W}\right) & \text{(linear decay with window } W\text{)} \\ 1 & \text{(no decay)} \end{cases} \tag{4}$$

Here, $\Delta\tau = \tau_q - \tau_e$ measures how long ago the edge occurred. The parameters $\lambda$ and $W$ control the decay behavior. $\lambda$ (temporal half-life) determines the rate of exponential decay: an edge that occurred $\lambda$ time units ago contributes roughly half as much as a recent edge, with larger $\lambda$ giving more weight to older events. $W$ (linear decay window) defines the temporal horizon for linear decay: edges older than $W$ are ignored ($w = 0$), while edges within the window decay linearly from 1 to 0 as they approach age $W$.

*(3) Memory-Augmented Updates.* Each node maintains a memory vector $M_v$ that stores its temporal state. When a new event occurs, the memory is updated via a gated recurrent unit (GRU):

$$M'_v = \text{GRU}([m, \text{TimeEnc}(\tau - \tau_{\text{last}})], M_v), \tag{5}$$

where $\tau_{\text{last}}$ denotes the last update time. This mechanism allows nodes to retain long-term temporal context beyond local messages.

### 3.3 Temporal Bellman–Ford Recurrence

Combining the above, the temporal message-passing rule becomes:

$$h_v^{(t)} = \text{AGGREGATE}\big(\{m_{u \to v}^{(t-1)} \mid (u, r, v, \tau) \in E, \tau \le \tau_q\} \cup \{h_v^{(0)}\}\big), \tag{6}$$

$$m_{u \to v} = f_{\text{msg}}(h_u, r, \tau) \cdot w(\tau, \tau_q), \tag{7}$$

where:

- $m_{u \to v}$ is the **message** sent from node $u$ to node $v$ along the edge $(u, r, v)$, incorporating temporal information,
- $f_{\text{msg}}$ is the **message function** that encodes node $u$'s current representation $h_u$, the relation $r$, and the timestamp $\tau$ into a relation- and time-aware message vector,
- $w(\tau, \tau_q)$ applies a temporal decay weight to enforce recency and causality.

This formulation enables interpretable, time-aware path reasoning while respecting causality.

### 3.4 Algorithmic View

Algorithm 1 summarizes the temporal reasoning process. Lines marked [**T-NBFNet**] highlight the additional components—causal masking, decay computation, time encoding, and memory updates—that differentiate T-NBFNet from the original NBFNet.

**Algorithm 1** Temporal Neural Bellman–Ford Networks (T-NBFNet)

**Require:** source node $u$, query $q$, number of layers $T$
**Ensure:** pair representations $h_q(u, v)$ for all $v$

Initialize $h_v^{(0)} \leftarrow \text{INDICATOR}(u, v, q)$ for all $v$
**for** $t \leftarrow 1$ to $T$ **do**
  **for** each $v \in V$ **do**
    $M \leftarrow \{h_v^{(0)}\}$
    **for** each incoming edge $(x, r, v, \tau)$ in $E(v)$ **do**
      [**T-NBFNet**] Mask edges with $\tau > \tau_q$
      [**T-NBFNet**] Compute temporal decay $w(\tau, \tau_q)$
      [**T-NBFNet**] Encode timestamp $\text{TimeEnc}(\tau)$
      $m \leftarrow \text{MESSAGE}(h_x^{(t-1)}, w_q(x, r, v, \tau))$
      [**T-NBFNet**] Apply temporal weighting $m \leftarrow m \cdot w(\tau, \tau_q)$
      $M \leftarrow M \cup \{m\}$
    **end for**
    $h_v^{(t)} \leftarrow \text{AGGREGATE}(M)$
    [**T-NBFNet**] Update node memory $M_v$ via GRU with time difference
  **end for**
**end for**
**return** $h_v^{(T)}$

### 3.5 Training Objective

T-NBFNet is trained for temporal link prediction using binary cross-entropy loss with negative sampling:

$$\mathcal{L} = -\sum_{(h,r,t,\tau)} \Big[ \log \sigma(s(h, r, t, \tau)) + \sum_{(h',r,t',\tau) \in N} \log(1 - \sigma(s(h', r, t', \tau))) \Big], \quad (8)$$

where $s(h, r, t, \tau)$ denotes the predicted score, and $N$ is the set of negative samples generated under temporal constraints.

## 4 Experiments and Results

We evaluate T-NBFNet on four widely used temporal knowledge graph benchmarks: ICEWS14 [9], ICEWS18 [7], WIKI [10], and YAGO [11]. We report Mean Reciprocal Rank (MRR) and Hits@10 with temporal filtering.

For all experiments, T-NBFNet uses a 6-layer architecture with hidden dimension 32, sinusoidal time encodings (dimension 32), GRU-based node memory, and exponential decay with half-life $\lambda$=300. The model is trained using Adam with a learning rate of $5\times10^{-4}$, and regularized with dropout and layer normalization.

| **Model** | **ICEWS18** | **ICEWS14** | **WIKI** | **YAGO** |
|---|---|---|---|---|
| **Static Models** | | | | |
| DistMult | 0.22 / 0.42 | 0.19 / 0.36 | 0.46 / 0.51 | 0.59 / 0.65 |
| R-GCN | 0.23 / 0.36 | 0.26 / 0.45 | 0.38 / 0.42 | 0.41 / 0.53 |
| ConvE | 0.37 / 0.51 | 0.40 / 0.55 | 0.48 / 0.51 | 0.63 / 0.66 |
| RotatE | 0.23 / 0.39 | 0.30 / 0.43 | 0.51 / 0.51 | 0.64 / 0.66 |
| **Temporal / Dynamic Models** | | | | |
| TA-DistMult | 0.29 / 0.45 | 0.21 / 0.35 | 0.48 / 0.52 | 0.62 / 0.65 |
| EvolveRGCN | 0.17 / 0.34 | 0.17 / 0.33 | 0.46 / 0.49 | 0.60 / 0.62 |
| R-GCRN+MLP | 0.35 / 0.50 | 0.37 / 0.52 | 0.48 / 0.50 | 0.54 / 0.61 |
| RE-NET | 0.43 / 0.56 | 0.46 / 0.59 | **0.52** / 0.54 | 0.65 / 0.68 |
| CyGNet | **0.47 / 0.57** | **0.49 / 0.60** | 0.46 / 0.53 | **0.63 / 0.69** |
| **T-NBFNet (ours)** | 0.26 / 0.36 | 0.26 / 0.47 | 0.48 / **0.54** | 0.63 / 0.64 |

Table 1: **Temporal link prediction results (MRR / Hits@10)** across four temporal knowledge graph benchmarks. Results for baselines are taken from the RE-NET paper [8], using the same dataset splits and evaluation protocol. T-NBFNet results are produced using our own implementation.

T-NBFNet improves upon the static baselines (e.g., DistMult, R-GCN, ConvE, and RotatE) across all datasets, demonstrating the benefit of introducing temporal reasoning into the Bellman–Ford framework. While temporal models such as RE-NET and CyGNet achieve higher overall performance, T-NBFNet offers a strong balance between temporal modeling and interpretability. Notably, on WIKI and YAGO datasets with coarser temporal granularity—T-NBFNet achieves performance comparable to state-of-the-art temporal models, highlighting its robustness to irregular time intervals.

## 5 Conclusion and Future Work

We proposed **T-NBFNet**, a temporal extension of the Neural Bellman–Ford Network that integrates sinusoidal encodings, decay weighting, causal masking, and memory modules. T-NBFNet achieves competitive performance across temporal benchmarks while preserving interpretability.Future work includes enabling continuous-time reasoning and improving the memory module for better long-term temporal modeling.

## References


[1] Zhu, Z., Zhang, Y., Cui, P., Zhu, W. (2021). Neural Bellman–Ford Networks: A General Graph Neural Network Framework for Link Prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[2] Kipf, T. N., Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

[3] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y. (2018). Graph Attention Networks. In *International Conference on Learning Representations (ICLR)*.

[4] Hamilton, W. L., Ying, R., Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[5] Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K. (2020). Inductive Representation Learning on Temporal Graphs. In *International Conference on Learning Representations (ICLR)*.

[6] Rossi, E., Kenlay, H., Gorinova, M., Monti, F., Bronstein, M., Chamberlain, B. (2020). Temporal Graph Networks for Deep Learning on Dynamic Graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+)*.

[7] Trivedi, R., Farajtabar, M., Biswal, P., Zha, H. (2019). DyRep: Learning Representations over Dynamic Graphs. In *International Conference on Learning Representations (ICLR)*.

[8] Jin, W., Qu, M., Jin, X., Ren, X. (2019). Recurrent Event Network: Autoregressive Structure Inference over Temporal Knowledge Graphs. *arXiv preprint arXiv:1904.05530*. `https://arxiv.org/abs/1904.05530`

[9] Boschee, E., Lautenschlager, J., O'Brien, S., Shellman, S., Starz, J., Ward, M. (2015). ICEWS Coded Event Data. In *Harvard Dataverse*, doi:10.7910/DVN/28075.

[10] Leblay, J., Chekol, M. W. (2018). Deriving Validity Time in Knowledge Graph. In *Companion Proceedings of the The Web Conference (WWW)*, pp. 1771–1776.

[11] Mahdisoltani, F., Biega, J., Suchanek, F. M. (2013). YAGO3: A Knowledge Base from Multilingual Wikipedias. In *CIDR Workshop on Innovative Data Systems Research (CIDR)*.