Approximating Temporal Betweenness with Temporal GNNs: The TSBM and TSFMBM Models*

1 Related Work and Motivation

The *TATKC* model [4] introduced a temporal graph neural network (GNN) for approximating Temporal Katz Centrality (TKC), leveraging a time-aware attention mechanism followed by a downstream MLP. While *TATKC* performs effectively in predicting TKC, it is less suited for capturing Temporal Betweenness Centrality due to its inherently path-centric nature, which demands richer structural information. Specifically, *TATKC*'s reliance on raw node embeddings and temporal attention neglects crucial mid-path signals that are essential for identifying nodes with high temporal betweenness.

Motivated by these limitations, we propose two new models that extend the learning-based GNN paradigm to approximate path-based temporal centrality measures: the *TSBM* model for Temporal Shortest Betweenness Centrality and the *TSFMBM* model for Temporal Shortest-Foremost Betweenness Centrality. Our models incorporate features specifically designed to capture path relevance and intermediate-node influence, addressing the representational gaps observed in *TATKC*.

Formal definitions of temporal graphs and temporal betweenness centrality are provided in Appendices A.

2 Integration of the new node feature: `ptd_feat`

In the multi-layer perceptron (MLP) architecture, an additional scalar feature `ptd_feat` (representing the pass-through degree [1]), which basically it counts how many temporal paths of length 2 pass through a node, is incorporated alongside the primary node feature `src_feat`. The `src_feat` is of dimension $[B, 128]$, where B denotes the batch size, while the `ptd_feat` is a scalar for each node ($[B, 1]$).

2.1 MLP Architecture

In highly imbalanced temporal graphs where only a small fraction of nodes have non-zero betweenness centrality, models must be able to sharply distinguish important from unimportant nodes. We compare two strategies for incorporating **pass-through degree (ptd)** as an additional feature: *Concatenation* and *Feature-wise Linear Modulation (FiLM)* [2].

- Concatenation (CONCAT): Here, the ptd feature is concatenated with the original node feature, treating it as an additional input. This straightforward approach allows the model to learn the importance of ptd during training but may not provide fine-grained control over its influence.
- FiLM: FiLM applies feature-wise affine transformations to the node features, conditioned on the ptd feature. This allows for more nuanced modulation, enabling the model to adaptively scale and shift features based on ptd, potentially capturing more complex relationships.

For the FiLM modulation pathway, `ptd_feat` is passed through `ptd_scale_proj` and `ptd_bias_proj`, each projecting $[B, 1] \to [B, 128]$. These projected vectors serve as the `scale` and `bias` parameters in the FiLM operation: the modulation is computed as `modulated_feat = src_feat * scale + bias`, preserving the original shape $[B, 128]$.

*The source code is available on `https://github.com/filipposchr/TBM`

In parallel, the `ptd_feat` is also processed through `ptd_proj`, a nonlinear projection with a ReLU activation and dropout, to produce another $[B, 128]$ vector. This output is then concatenated with the `modulated_feat`, resulting in a combined feature of shape $[B, 256]$.

The concatenated feature is subsequently passed through `input_proj`, which maps it back down to a `final_dim` of 128, maintaining dimensional compatibility for the downstream MLP. Finally, the output of shape $[B, 128]$ is processed by a deep MLP and projected to a single scalar per node via a final linear layer, resulting in an output of shape $[B, 1]$, which is then squeezed to $[B]$ for compatibility with standard loss functions and evaluation.

3 Loss function

The default model, **TATKC**, utilizes a pairwise ranking loss function, implemented in `loss_cal` to train the model for approximating TKC ranking scores. This function relies on margin ranking loss, which randomly samples node pairs and encourages the model to preserve correct ordering by scoring higher ground-truth nodes above lower ones.

4 Experiments

Hyper-parameters. The model is conducted with a learning rate of 0.01 for 15 epochs, with the node and time embedding dimension as 128 and number of neighbor samples to 20 (same as TATKC model).

4.1 Experimental Setup

Datasets. We utilize 13 real datasets from Konect[1] and Network Repository[2] for evaluation. The statistics are summarized in Table 1. The training dataset consists of 48 real-world datasets from Konect, each ranging from 1000 to 12000 nodes (see Table 1).

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert T\rvert$
mathoverflow	24,759	390,414	389,952
facebook-wall	35,817	198,028	194,904
topology	16,564	198,038	32,823
mlwikiquote	43,889	142,340	137,389
plwikiquote	581,646	1,472,273	1,452,278
digg-reply	30,398	87,627	83,943
SMS	44,090	544,607	467,838
retweet-pol	18,469	61,157	60,501
slashdot-reply	51,083	139,789	89,862
wang-amazon	26,911	29,062	2,175
mgwikipedia	220,064	750,811	736,680
tgwiktionary	33,968	81,516	67,065
ltwiktionary	689,678	1,693,277	1,633,334

Table 1: Statistics of the testing datasets.

Evaluation Metrics. We evaluate the model's effectiveness in terms of top-N% accuracy, Kendall's tau correlation and Jaccard index.

1. **Top-K Accuracy** is defined as:

$$\text{Top-}N\% = \frac{|\text{Top-}N\% \text{ predicted nodes} \cap \text{Top-}N\% \text{ true nodes}|}{\lceil |V| \times N\% \rceil}$$

[1] Konect is available at `http://konect.cc/`
[2] Network Repository is available at `https://networkrepository.com/networks.php`

where $|V|$ is the number of nodes and $\lceil \cdot \rceil$ denotes the ceiling function. This metric measures the fraction of correctly predicted top-N% nodes compared to the ground truth, and is commonly used to evaluate effectiveness in identifying highly central nodes.

Evaluation Code Behavior:

The function `compute_topk_metrics_all` implements this definition precisely:

- `N = len(preds)` corresponds to $|V|$.
- `topk = max(1, math.ceil(N * (k / 100)))` implements $\lceil |V| \cdot N\% \rceil$.
- `torch.topk(preds, topk)` selects the top-K predicted nodes.
- `torch.topk(labels, topk)` selects the top-K true (ground-truth) nodes.
- The intersection of these sets is computed, and the final metric is:

$$\frac{|\text{predicted top-}K \cap \text{true top-}K|}{K}$$

2. **Weighted Kendall Tau (WKT)** is a rank-based correlation metric that measures the ordinal association between two sequences. In this evaluation, it is used to assess how well the predicted ranking of nodes aligns with the ground truth ranking based on temporal betweenness.

   (a) **Tiebreaking Strategy:** The ranking procedure uses the function `rank_with_id_tiebreak(values)`, with the `"ordinal"` method, which assigns unique integer ranks based on the order of appearance (row index) when ties occur. This avoids ambiguity from tied ranks, ensuring that every element receives a unique rank, which is required for computing Kendall tau consistently.

   (b) **Computation of WKT** (function `compute_kendall_tau`):

      i. **Standard WKT over all nodes** is computed as:

```
label_ranked_full = rank_with_id_tiebreak(labels)
pred_ranked_full = rank_with_id_tiebreak(preds)
kt_full, _ = weightedtau(pred_ranked_full, label_ranked_full)
```

      This version compares the full ranked list of predictions and ground truth across all nodes, regardless of whether the ground truth is zero or not.

      ii. **Filtered WKT on non-zero ground truth values only** is computed as:

```
nonzero_mask = labels > 0
label_filtered = labels[nonzero_mask]
pred_filtered = preds[nonzero_mask]
label_ranked_filt = rank_with_id_tiebreak(label_filtered)
pred_ranked_filt = rank_with_id_tiebreak(pred_filtered)
kt_filt, _ = weightedtau(pred_ranked_filt, label_ranked_filt)
```

      This filtered variant restricts the correlation calculation to the subset of nodes with non-zero ground truth centrality. This is important in sparse graphs where most nodes have zero temporal betweenness, which would otherwise dominate the ranking and obscure differences among important nodes.

   (c) The WKT values reported in Table 3 and Table 5 correspond to the *filtered* WKT computed only on nodes with non-zero ground truth (defined in 2.a.ii). The standard WKT computed over all nodes is consistently higher.

   (d) **Interpretation:**

      - A higher `kt_full` indicates better alignment in global ranking across all nodes.
      - A higher `kt_filt` reflects better ranking precision among the truly important (non-zero) nodes.

- In datasets where many nodes have a ground truth value of zero, WKT computed over all nodes can be lower if the model assigns different (noisy) prediction scores to those zero-value nodes (e.g., x, y, z). This introduces noise into the global ranking and reduces Kendall tau. In contrast, WKT computed only over the non-zero ground truth nodes ignores these irrelevant variations and can yield a higher score. This filtered WKT better reflects the model's ranking quality on the meaningful part of the distribution. When the model assigns the same predictions to all zero-labeled nodes, the all-node and non-zero-only WKT scores tend to be closer.

3. **Jaccard Index** is defined as:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

where A and B are two sets. In the context of this model evaluation, A represents the set of predicted important nodes and B the ground truth important nodes.

(a) **Computation of Jaccard**

The following lines compute the Jaccard index in the `compute_topk_metrics_all` function :

```
pred_full_set = set(torch.topk(preds, len(true_full_set)).indices.tolist())
full_jacc = len(pred_full_set & true_full_set) / len(pred_full_set | true_full_set)
```

This corresponds to the following steps:

- `true_full_set` contains all node indices for which the ground truth value is greater than zero (i.e., nodes deemed important by the centrality ground truth).
- `pred_full_set` is constructed by selecting the same number of top predicted nodes as there are in the true set, ensuring fair comparison.
- The numerator computes the size of the intersection between the predicted and true sets.
- The denominator computes the size of the union of the two sets.

(b) **Interpretation:**

The Jaccard Index here captures the global agreement between the predicted and true sets of important nodes—regardless of ranking—by comparing the overlap of non-zero centrality predictions with ground truth. A higher Jaccard value indicates better alignment between model predictions and actual influential nodes.

4.2 Experimental Result

We evaluate two models: **TSBM** (Temporal Shortest Betweenness Model), which focuses on temporal shortest-path betweenness (sh), and **TSFMBM** (Temporal Shortest-Foremost Betweenness Model), designed for temporal shortest-foremost betweenness (sfm). Specifically, we evaluate the performance of four model variants: $\mathbf{TSBM_C}$, $\mathbf{TSBM_{CF}}$ (see Table 2 and Table 3), and their shortest-foremost equivalents, $\mathbf{TSFMBM_C}$ and $\mathbf{TSFMBM_{CF}}$ (see Table 4 and Table 5). Evaluation is conducted using Top-N accuracy, Jaccard index, and Weighted Kendall's Tau (WKT) metrics across a diverse set of real-world temporal graphs. The $_\mathbf{C}$ variants rely solely on feature concatenation, while the $_\mathbf{CF}$ variants extend this setup with FiLM-based conditioning over the concatenated features.

For comparison, we include WKT scores for the MANTRA [3] model, a recent approximation-based approach for temporal betweenness estimation. The models are trained using the standard `loss_cal` function.

Some scores for the MANTRA model are omitted due to the extremely low connectivity of certain temporal graphs (e.g., 0.0000096 in `tgwiktionary`), where the vast majority of nodes have zero betweenness (only 235 out of 33,968 nodes have non-zero temporal betweenness). In such sparse graphs, achieving a reliable approximation would require an extremely small ε (i.e., the approximation accuracy parameter), for example $\varepsilon < 0.000009$, which makes MANTRA either impractical to compute or too inaccurate to be meaningful.

Dataset	Top-1%			Top-5%			Top-10%			Top-20%			Jaccard		
	TSBM$_{\mathtt{C}}$	TSBM$_{\mathtt{CF}}$	MANTRA	TSBM$_{\mathtt{C}}$	TSBM$_{\mathtt{CF}}$	MANTRA	TSBM$_{\mathtt{C}}$	TSBM$_{\mathtt{CF}}$	MANTRA	TSBM$_{\mathtt{C}}$	TSBM$_{\mathtt{CF}}$	MANTRA	TSBM$_{\mathtt{C}}$	TSBM$_{\mathtt{CF}}$	MANTRA
mathoverflow	**0.86**	0.5	0.85	0.82	0.82	0.7	0.76	0.76	0.57	0.8	0.8	0.39	0.95	0.94	0.54
facebook	0.56	0.56	0.85	0.65	0.65	0.83	0.7	0.7	0.79	0.77	0.78	0.67	0.9	0.9	0.69
topology	**0.7**	0.66	0.68	0.66	0.66	0.65	0.68	0.67	0.55	0.7	0.7	0.39	0.78	0.79	0.44
mlwikiquote	**0.76**	0.67	0.9	0.84	0.84	0.56	0.8	**0.82**	0.34	0..63	**0.85**	0.49	0.77	**0.83**	0.49
pliwkiquote	0.68	0.68	0.49	0.83	0.83	0.04	0.82	**0.94**	0.08	0.49	**0.97**	0.48	0.93	0.93	0.47
digg_reply	0.65	0.65	0.88	0.7	0.7	0.79	0.75	0.75	0.6	0.93	0.92	0.35	0.99	0.99	0.67
SMS	0.64	0.64	0.56	0.63	0.64	0.22	0.67	0.67	0.15	0.85	**0.99**	0.15	0.83	**0.86**	0.42
rt-pol	0.75	0.75		0.77	0.77		0.91	0.91		0.65	**0.87**		0.96	0.96	
slashdot	0.81	0.81	0.75	0.82	0.82	0.67	0.84	0.84	0.41	0.85	**0.91**	0.39	0.9	0.91	0.59
wamazon	0.95	0.95	0.61	0.72	**1**	0.74	0.87	**1**	0.87	0.92	**1**	0.93	1	1	0.64
mgwikipedia	0.84	**1**		0.18	**0.99**		0.11	**0.99**		0.11	**0.99**		0.95	0.95	
tgwiktionary	0.84	**1**		0.19	**0.99**		0.19	**0.99**		0.23	**0.99**		0.99	0.99	
ltwiktionary	0.31	**1**		0.11	**0.99**		0.13	**0.99**		0.2	**0.99**		0.99	0.98	

Table 2: Top-N accuracy and Jaccard similarity for the **TSBM** (Temporal Shortest Betweenness Model) on a diverse set of real-world temporal graphs. TSBM$_{\mathtt{C}}$ refers to the model variant using feature concatenation only, while TSBM$_{\mathtt{CF}}$ applies FiLM-based conditioning over concatenated features. Bold values highlight the best performance across the GNN methods.

Dataset	WKT (all nodes)			WKT (non zero)		
	TSBM$_{\mathtt{C}}$	TSBM$_{\mathtt{CF}}$	MANTRA	TSBM$_{\mathtt{C}}$	TSBM$_{\mathtt{CF}}$	MANTRA
mathoverflow	0.92	**0.94**	0.75	0.42	**0.87**	0.86
facebook	0.91	0.92	0.83	0.86	0.86	0.9
topology	0.91	0.91	0.73	0.84	0.83	0.82
mlwikiquote	0.8	**0.95**	0.76	0.8	0.79	0.86
plwikiquote	0.86	**0.98**	0.58	0.83	0.83	0.82
digg reply	0.91	**0.97**	0.84	0.85	0.85	0.9
SMS	0.89	**0.93**	0.63	0.84	0.85	0.74
rt-pol	0.9	**0.98**		0.31	**0.87**	
slashdot	0.94	**0.98**	0.83	0.9	0.9	0.86
wamazon	0.94	**0.99**	0.87	0.86	0.86	0.8
mgwikipedia	0.83	**0.99**		**0.83**	0.81	
tgwiktionary	0.59	**0.99**		**0.9**	0.83	
ltwiktionary	0.7	**0.99**		**0.92**	0.85	

Table 3: Standard (all nodes) and filtered WKT scores for the **TSBM** (Temporal Shortest Betweenness Model) computed using the ordinal tie-breaking strategy. The filtered scores consider only nodes with non-zero ground truth values. Results are reported for the TSBM$_{\mathtt{CF}}$ variant.

Dataset	Top-1%			Top-5%			Top-10%			Top-20%			Jaccard		
	TSFMBM$_{\mathtt{C}}$	TSFMBM$_{\mathtt{CF}}$	MANTRA	TSFMBM$_{\mathtt{C}}$	TSFMBM$_{\mathtt{CF}}$	MANTRA	TSFMBM$_{\mathtt{C}}$	TSFMBM$_{\mathtt{CF}}$	MANTRA	TSFMBM$_{\mathtt{C}}$	TSFMBM$_{\mathtt{CF}}$	MANTRA	TSFMBM$_{\mathtt{C}}$	TSFMBM$_{\mathtt{CF}}$	MANTRA
mathoverflow	**0.85**	0.58		0.84	0.84		0.79	0.79		0.8	0.8		0.95	0.95	
facebook	0.55	0.55	0.83	0.65	0.65	0.81	0.7	0.7	0.76	0.77	0.77	0.61	0.9	0.9	0.65
topology	**0.72**	0.68	0.75	0.68	0.68	0.66	0.7	0.7	0.54	0.71	0.71	0.39	0.77	**0.8**	0.46
mlwikiquote	**0.76**	0.73	0.90	0.8	0.81	0.59	0.76	**0.79**	0.41	0.52	0.52	0.63	0.63	**0.67**	0.46
pliwkiquote	0.66	0.66	0.5	0.74	0.74	0.04	0.64	0.64	0.24	**0.4**	0.37	0.53	0.72	0.73	0.56
digg_reply	0.64	0.64	0.85	0.7	0.7	0.68	0.74	0.74	0.43	0.93	0.93	0.27	0.98	0.98	0.58
SMS	0.65	0.66	0.83	0.64	0.64	0.81	0.67	0.67	0.76	0.85	0.85	0.61	0.84	**0.86**	0.65
rt-pol	0.75	0.75	0.82	0.77	0.77	0.5	0.91	0.91	0.27	0.63	0.62	0.51	0.92	0.92	0.62
slashdot	0.79	0.79	0.74	0.81	0.81	0.61	0.84	0.84	0.37	0.82	0.82	0.37	0.88	0.89	0.56
wamazon	0.96	0.96	0.83	**0.58**	0.55	0.81	**0.71**	0.69	0.76	**0.84**	0.81	0.61	0.99	0.99	0.65
mgwikipedia	0.84	0.84		0.17	0.15		0.1	0.12		0.11	0.13		0.76	0.76	
tgwiktionary	0.84	0.84		0.19	0.2		0.18	0.17		0.22	**0.25**		0.91	0.91	
ltwiktionary	0.11	0.13		0.07	0.03		0.15	**0.3**		0.24	**0.43**		0.92	0.92	

Table 4: Top-N accuracy and Jaccard similarity for the **TSFMBM** (Temporal Shortest-Foremost Betweenness Model) on a diverse set of real-world temporal graphs. TSFMBM$_{\mathtt{C}}$ refers to the model variant using feature concatenation only, while TSFMBM$_{\mathtt{CF}}$ applies FiLM-based conditioning over concatenated features. Bold values highlight the best performance across the GNN methods.

Dataset	WKT (all nodes)			WKT (non zero)		
	TSFMBM$_C$	TSFMBM$_{CF}$	MANTRA	TSFMBM$_C$	TSFMBM$_{CF}$	MANTRA
mathoverflow	**0.92**	0.9		**0.9**	0.88	
facebook	0.91	0.91	0.82	0.86	0.86	0.89
topology	0.9	0.9	0.73	0.85	0.84	0.83
mlwikiquote	0.73	0.72	0.80	0.8	0.8	0.83
plwikiquote	**0.82**	0.72	0.63	0.79	0.79	0.79
digg reply	0.92	0.91	0.8	0.85	0.85	0.86
SMS	0.89	0.89	0.82	0.85	0.85	0.89
rt-pol	**0.89**	0.86	0.81	0.86	0.85	0.86
slashdot	**0.95**	0.93	0.82	0.9	0.9	0.84
wamazon	**0.98**	0.9	0.82	0.86	0.86	0.89
mgwikipedia	**0.8**	0.7		0.81	0.8	
tgwiktionary	0.63	0.63		0.87	0.87	
ltwiktionary	0.62	**0.66**		0.78	0.77	

Table 5: Standard (all nodes) and filtered WKT scores for the **TSFMBM** (Temporal Shortest-Foremost Betweenness Model) computed using the ordinal tie-breaking strategy. The filtered scores consider only nodes with non-zero ground truth values. Results are reported for the TSFMBM$_{CF}$ variant.

5 Integration of a Third Node Feature: `arrival_feat`

To enhance the predictive performance of **TSFMBM** (Temporal Shortest-Foremost Betweenness Model)—whose results using only two node features are shown in Table 4 and Table 5—we introduce a third scalar feature, `arrival_feat`, which encodes the earliest time at which each node becomes reachable.

While the core representation `src_feat` captures structural and temporal context, and `ptd_feat` reflects a node's pass-through frequency, `arrival_feat` introduces a notion of temporal urgency, indicating how early a node can be accessed within the temporal graph. This information is particularly relevant for shortest-foremost paths, which prioritize early arrival over minimal hop count.

Both `ptd_feat` and `arrival_feat` are projected from shape $[B, 1]$ to $[B, 128]$ and concatenated with `src_feat` (of shape $[B, 128]$), resulting in a unified input of shape $[B, 384]$ for the MLP.

We implement this design in two architectural variants:

- `MLPWithThreeFeatC`, which directly concatenates the projected features, corresponding to the TSFMBM$_C$ model;
- `MLPFilmThreeFeatCF`, which applies FiLM-style modulation to `src_feat` using `ptd_feat`, followed by concatenation with both projected `ptd_feat` and `arrival_feat`, corresponding to the TSFMBM$_{CF}$ model.

Results incorporating the `arrival_feat` feature are presented in Table 6 and Table 7.

Dataset	Top-1%		Top-5%		Top-10%		Top-20%		Jaccard	
	TSFMBM$_C$	TSFMBM$_{CF}$	TSFMBM$_C$	TSFMBM$_{CF}$	TSFMBM$_C$	TSFMBM$_{CF}$	TSFMBM$_C$	TSFMBM$_{CF}$	TSFMBM$_C$	TSFMBM$_{CF}$
mathoverflow	0.46	**0.54**	0.84	0.84	0.79	0.79	0.80	0.80	0.95	0.95
facebook	0.54	0.54	0.65	0.65	0.70	0.70	0.77	0.77	0.90	0.90
topology	0.48	**0.51**	0.67	0.67	0.70	0.70	0.71	0.72	**0.78**	0.75
mlwikiquote	**0.73**	0.67	0.81	0.81	**0.78**	0.75	0.72	0.71	**0.65**	0.63
plwikiquote	0.66	0.66	0.74	0.74	0.77	0.77	0.86	0.86	0.73	0.73
digg_reply	0.63	0.64	0.70	0.70	0.74	0.74	0.93	0.93	0.98	0.98
SMS	0.66	0.65	0.64	0.64	0.68	0.68	0.85	0.85	0.82	0.82
rt-pol	0.73	0.74	0.77	0.77	0.91	0.91	0.85	0.90	0.93	0.93
slashdot	0.80	0.79	0.81	0.82	0.84	0.84	0.82	**0.88**	0.88	0.88
wamazon	0.96	0.97	0.90	**0.99**	0.93	**0.99**	0.96	**0.99**	0.99	0.99
mgwikipedia	0.99	1	0.92	**0.94**	0.97	0.98	0.97	**0.99**	0.77	0.76
tgwiktionary	0.99	1	0.91	**0.98**	0.97	0.99	0.97	0.99	0.93	0.93
ltwiktionary	0.99	1	0.91	**0.99**	0.95	**0.99**	0.96	0.99	0.92	0.92

Table 6: Top-N accuracy and Jaccard similarity for the **TSFMBM** (Temporal Shortest-Foremost Betweenness Model) - integrating `arrival_feat` - on a diverse set of real-world temporal graphs. TSFMBM$_C$ refers to the model variant using feature concatenation only, while TSFMBM$_{CF}$ applies FiLM-based conditioning over concatenated features. Bold values highlight the best performance across the GNN methods.

Dataset	WKT (all nodes)		WKT (non zero)	
	$\texttt{TSFMBM}_{\texttt{C}}$	$\texttt{TSFMBM}_{\texttt{CF}}$	$\texttt{TSFMBM}_{\texttt{C}}$	$\texttt{TSFMBM}_{\texttt{CF}}$
mathoverflow	0.94	0.94	0.86	0.87
facebook	0.92	0.92	0.86	0.86
topology	0.90	0.90	0.81	0.82
mlwikiquote	0.88	0.88	0.78	0.78
plwikiquote	0.94	0.94	0.78	0.78
digg reply	0.97	0.97	0.85	0.85
SMS	0.93	0.93	0.85	0.85
rt-pol	0.98	0.98	0.86	0.86
slashdot	0.97	0.97	0.89	0.89
wamazon	0.99	0.99	0.86	0.85
mgwikipedia	0.97	0.98	0.80	0.80
tgwiktionary	0.99	0.99	**0.86**	0.80
ltwiktionary	0.98	0.99	0.83	0.82

Table 7: Standard (all nodes) and filtered WKT scores for the **TSFMBM** (Temporal Shortest-Foremost Betweenness Model) - integrating `arrival_feat` - computed using the ordinal tie-breaking strategy. The filtered scores consider only nodes with non-zero ground truth values. Results are reported for the $\texttt{TSFMBM}_{\texttt{CF}}$ variant.

5.0.1 Supremum Deviation and MAE Analysis

In addition to ranking-based metrics, we report the Supremum Deviation (SD) and Mean Absolute Error (MAE) to quantify the magnitude of prediction errors for temporal betweenness centrality. The Supremum Deviation measures the absolute difference between the mean of the predicted values and the mean of the ground truth values, reflecting any global bias in the model's scale of predictions. Higher SD values indicate that the model systematically overestimates or underestimates the overall magnitude of betweenness scores. Conversely, the MAE assesses the average absolute difference between predicted and true values across all nodes, capturing local prediction accuracy. Low MAE values suggest the model closely approximates the node-level centrality, even if the global scale may be biased.

The SD values indicate that the model exhibits a noticeable global bias in its predictions, while the very low Mean Absolute Error shows that, on average, the model is making highly accurate node-level predictions.

The results, summarized in Table 8, show that although the SD ranges from approximately 0.18 to 0.60, the normalized MAE remains extremely low across datasets, highlighting the strong local accuracy of the model despite some bias in the overall scale of predictions.

Dataset	$\texttt{TSFMBM}_{\texttt{C}}$			$\texttt{TSFMBM}_{\texttt{CF}}$		
	Pred $\leq$ 0, True $>$ 0	**SD**	**MAE**	**Pred $\leq$ 0, True $>$ 0**	**SD**	**MAE**
mathoverflow	0	0.517189	0.001508	33	0.3431932	0.001508
facebook	0	0.59934	0.003499	665	0.416918	0.003499
topology	0	0.4679802	0.001695	26	0.377038	0.001695
mlwikiquote	0	0.343673	0.003043	0	0.090904	0.003077
plwikiquote	0	0.274652	0.000107	119	0.065472	0.000107
digg_reply	0	0.4430229	0.001771	732	0.16643	0.001771
SMS	0	0.4422315	0.0011279	852	0.20441	0.0011280
rt-pol	0	0.321965	0.001571	264	0.09095	0.00157
slashdot	0	0.415216	0.0009534	50	0.13331	0.00095
wamazon	0	0.17757	0.0004942	36	0.00914	0.000428
mgwikipedia	0	0.235975	1.96×10^{-5}	234	0.00601	1.96×10^{-5}
tgwiktionary	0	0.1853245	0.0002576	44	0.00574	0.000223

Table 8: The table reports (i) the number of nodes with predicted betweenness values ≤ 0 while the ground truth betweenness is > 0, (ii) the Supremum Deviation (SD), and (iii) the Mean Absolute Error (MAE) for the **TSFMBM** (Temporal Shortest Foremost Betweenness Model) across the testing datasets.

5.0.2 Discussion

1. For the **TSFMBM** (Temporal Shortest-Foremost Betweenness Model), the results presented in Table 6 and Table 7, which incorporate the `arrival_feat` feature as an additional node input, demonstrate improved performance compared to the baseline results in Table 4 and Table 5 that use only the two node features (`src_feat` and `ptd_feat`). Specifically, the inclusion of the arrival feature leads to consistently higher Top-k Jaccard scores and Weighted Kendall Tau values.

2. In highly imbalanced settings, it is possible for the Top-k accuracy to decrease as k increases, contrary to intuition. This counterintuitive behavior typically occurs when the ground truth contains a large number of zero-valued targets, and the model assigns non-zero (noisy) scores to these unimportant nodes. As k increases, the prediction set may include more of these noisy false positives, thereby reducing precision. While the Top-1% predictions often remain focused on genuinely high-centrality nodes, the wider Top-5% or Top-10% sets can become diluted by misranked low-importance nodes. For instance, this is observed when evaluating the model $\texttt{TSBM}_{\texttt{CF}}$ on the `mathoverflow` dataset, where the Top-1% accuracy reaches 0.50, but the Top-5% drops to 0.82, with Top-10% and Top-20% further declining to 0.76 and 0.80, respectively. The large number of ground-truth-zero nodes in `mathoverflow`, combined with prediction variance among them, leads to reduced performance at broader thresholds despite a strong Top-1% accuracy.

6 Further on FiLM Architecture

6.1 How FiLM Operates Over Features in the Model

Feature-wise Linear Modulation (FiLM) is a conditioning mechanism that adjusts neural network activations using external signals. In our model, FiLM modulates the primary node representation (`src_feat`) using a graph-theoretic input: the pass-through degree (`ptd_feat`), which measures how frequently a node mediates time-respecting paths.

Instead of processing `src_feat` and `ptd_feat` jointly, FiLM treats `ptd_feat` as a controller. It produces per-feature scaling and shifting parameters applied to `src_feat` via:

$$\texttt{modulated\_feat} = \texttt{src\_feat} \times \gamma(\texttt{ptd\_feat}) + \beta(\texttt{ptd\_feat}),$$

where γ and β are learnable projections of the scalar `ptd_feat` to match the dimension of `src_feat`.

This modulation is applied not just at the input, but throughout deeper MLP layers, allowing the model to adapt the importance of features at multiple transformation stages. Notably, `ptd_feat` itself is not transformed or modulated, preserving its role as a stable, interpretable conditioning signal.

Overall, FiLM enables dynamic reweighting of node features while maintaining the interpretability of the external signal. This leads to more expressive and targeted transformations of `src_feat`, especially valuable in capturing node importance in temporal graphs.

Example: `tgwiktionary` Dataset. This dataset contains 33,968 nodes, of which only 235 have non-zero temporal betweenness scores. We expect the model to:

- Predict high values for these top 235 nodes;
- Assign uniform low values to all others.

Under FiLM, the model outputs distinct scores for top nodes (e.g., 0.746, 0.551, 0.337), while collapsing all other predictions to a single low value (e.g., -1.41), clearly separating predictions from noise. By contrast, the concatenation-only model produces scattered negative values for irrelevant nodes (e.g., -0.25, -0.51, -0.29), reducing ranking sharpness. Evaluation results confirm this: FiLM achieves Top-1% = 1.0, while concatenation yields Top-1% = 0.85. (see Table 2)

6.2 When the $_{\text{CF}}$ variant Outperforms or Underperforms $_{\text{C}}$ variant

The combination of FiLM and concatenation performs variably depending on dataset characteristics:

- **In dense or structured graphs** (e.g., SMS, slashdot), the $_{\text{CF}}$ variant tends to outperform $_{\text{C}}$ variant because:
  - FiLM highlights attention-relevant dimensions guided by PTD;
  - Concatenation preserves an explicit, interpretable PTD input;
  - The two mechanisms complement each other: one reshapes internal representations, the other offers direct supervision.
- **In sparse or skewed graphs** (e.g., mathoverflow), $_{\text{CF}}$ variant may underperform:
  - FiLM modulation can distort already informative `src_feat` signals;
  - This can hurt top-1% accuracy, where relevant nodes are rare and highly concentrated.

7 Conclusion

FiLM enables sharper separation between important and irrelevant nodes by modulating the feature space with learned scaling factors. This results in:

- Assign diverse and well-separated predictions to top-ranked, high-centrality nodes;
- Produce consistent low outputs for irrelevant or zero-ground-truth nodes;
- Improve ranking quality, both globally (Kendall Tau) and at the top-k level;
- Enhance the interpretability of node importance by clearly separating predictions from noise.

In highly imbalanced temporal graphs—where few nodes exhibit non-zero betweenness—the model must strongly differentiate between the predicted betweenness values and noise. We compared two strategies for incorporating pass-through degree: concatenation and FiLM. In several datasets, FiLM demonstrated better performance by acting as a feature-wise modulation mechanism that conditions node embeddings based on structural importance. This yields more stable predictions for unimportant nodes and better distinction among the top-ranked nodes.

Appendices

A Temporal Graphs

Definition 1 (Temporal Graph)**.** *A directed temporal graph is an ordered tuple $G = (V, \mathcal{E}, \mathcal{T})$ where:*

- *V is the set of nodes,*
- *$\mathcal{E} = \{(u, v, t) : u, v \in V \wedge u \neq v \wedge t \in \mathcal{T}\}$ is the set of directed temporal edges,*
- *T is the set of time instants [3] t in which at least one temporal edge is present in the network.*

Let $n = |V|$, $M = |\mathcal{E}|$, and $T = |\mathcal{T}|$ denote the number of nodes, edges, and distinct time instants, respectively. Given a subset of nodes $U \subseteq V$, the induced temporal subgraph is defined as $\mathcal{G}[U] := (U, \mathcal{E}')$, where $\mathcal{E}' := \{(u, v, t) \in \mathcal{E} : u, v \in U\}$.

It is also possible to define the *underlying static graph*:

Definition 2 (Underlying Static Graph)**.** *Given a temporal graph $\mathcal{G} = (V, \mathcal{E}, \mathcal{T})$, the underlying static graph is $G_{\mathcal{G}} = (V, E)$, where $E = \{(u, v) : \exists (u, v, t) \in \mathcal{E}\}$.*

The $G_{\mathcal{G}}$ is often a lossy representation of the network $\mathcal{G}$, since it ignores the timing of the events in the entire network.

A.0.1 Temporal Paths

Definition 3 (Temporal Path)**.** *Given a temporal graph $G = (V, \mathcal{E}, \mathcal{T})$ and two nodes $s, z \in V$, a temporal path $tp_{sz} \subseteq V \times V \times T$ is a sequence of time-respecting temporal edges*

$$((u_1, u_2, t_1), (u_2, u_3, t_2), \ldots, (u_{k-1}, u_k, t_{k-1}))$$

such that:

- *$u_1 = s$, $u_k = z$,*
- *$t_i < t_{i+1}$ for all $1 \leq i < k - 1$,*
- *all nodes u_i are distinct.*

This definition ensures that edge timings respect causality. The definition can be adapted to non-strict inequalities ($t_{i+1} \geq t_i$) as needed. A node v is said to be *internal* to tp_{sz} if it appears in a temporal edge and $v \notin \{s, z\}$.

Definition 4 (Optimal Temporal Paths)**.** *Let tp_{sz} be a temporal path from s to z as above. Different optimality criteria are defined as follows:*

- ***Shortest (sh):*** *minimizes the number of edges (hops) in the path.*
- ***Foremost (fm):*** *arrives at z at the earliest possible time.*
- ***Fastest (fs):*** *minimizes the total traversal time $(t_z - t_s)$.*
- ***Shortest-Foremost (sfm):*** *no other path arrives earlier and has fewer transitions.*
- ***Shortest-Fastest (sfs):*** *no other path is faster and has fewer transitions.*
- ***Prefix-Foremost (pfm):*** *is foremost and every prefix of the path is also foremost.*

Figure 1 illustrates examples of the first three path types:

[3] The value $\mathcal{T}$ denotes the life-time of the temporal graph, and, without loss of generality for our purposes, we assume that, for any $t \in \mathcal{T}$, there exists at least one temporal arc at that time and without loss of generality we assume $\mathcal{T} = [1, |\mathcal{T}|]$.

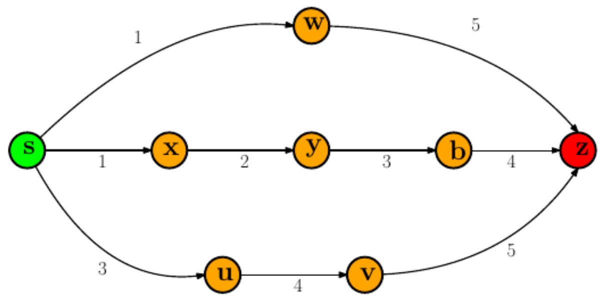

Figure 1: Example of the optimal-temporal paths: shortest, foremost and fastest

- $(s \xrightarrow{1} w \xrightarrow{5} z)$ is shortest,
- $(s \xrightarrow{1} x \xrightarrow{2} y \xrightarrow{3} b \xrightarrow{4} z)$ is foremost,
- $(s \xrightarrow{3} u \xrightarrow{4} v \xrightarrow{5} z)$ is fastest.

A.0.2 Temporal Betweenness Centrality

Just like in static graphs, centrality measures help identify important nodes. In temporal graphs, we focus on *temporal betweenness centrality*, which considers the number of optimal temporal paths a node participates.

Definition 5 (Temporal Betweenness Centrality)**.** *Given a temporal graph $\mathcal{G} = (V, \mathcal{E}, \mathcal{T})$ and a path optimality criterion $(\star)$, let:*

- $\sigma_{sz}^{(\star)}$ *be the number of $(\star)$-temporal paths from s to z,*
- $\sigma_{sz}^{(\star)}(v)$ *be the number of such paths that pass through node v, with $s \neq v \neq z$.*

*Then, the **normalized** $(\star)$**-temporal betweenness centrality** of a node $v \in V$ is:*

$$b_v^{(\star)} = \frac{1}{n(n-1)} \sum_{\substack{s,z \in V \\ s \neq v \neq z}} \frac{\sigma_{sz}^{(\star)}(v)}{\sigma_{sz}^{(\star)}} \in [0, 1].$$

Computing this measure is significantly more complex than in static graphs. For many criteria, it is known to be #P-Hard, so we focus on those variants for which polynomial-time algorithms exist.

References

[1] Ruben Becker, Pierluigi Crescenzi, Antonio Cruciani, and Bojana Kodric. Proxying Betweenness Centrality Rankings in Temporal Networks. In Loukas Georgiadis, editor, *21st International Symposium on Experimental Algorithms (SEA 2023)*, volume 265 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:22, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.SEA.2023.6`.

[2] Marc Brockschmidt. GNN-FiLM: Graph neural networks with feature-wise linear modulation. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1144–1152. PMLR, 13–18 Jul 2020. URL: `https://proceedings.mlr.press/v119/brockschmidt20a.html`.

[3] Antonio Cruciani. Mantra: Temporal betweenness centrality approximation through sampling. In Albert Bifet, Jesse Davis, Tomas Krilavičius, Meelis Kull, Eirini Ntoutsi, and Indrė Žliobaitė, editors, *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 125–143, Cham, 2024. Springer Nature Switzerland.

[4] Tianming Zhang, Junkai Fang, Zhengyi Yang, Bin Cao, and Jing Fan. Tatkc: A temporal graph neural network for fast approximate temporal katz centrality ranking. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 527–538, New York, NY, USA, 2024. Association for Computing Machinery. URL: `https://doi-org.gssi.idm.oclc.org/10.1145/3589334.3645432`, `doi:10.1145/3589334.3645432`.