Temporal Trust & Sentiment Dynamics in TrueSocial

Deep Learning for Social Analytics

Author One[1] Author Two[2]

[1]Affiliation One
[2]Affiliation Two

email1@domain.com, email2@domain.com

Contents

Abstract

We investigate temporal truthfulness dynamics on Truth Social using a directed follower network and their historical posts. We define a rolling user-level truthfulness measure and analyze its structural and temporal patterns. Empirical findings reveal wave-like growth, localized clustering of untruthful users, and strong exposure-dependent transition risks.

To model these dynamics, we propose a Temporal Graph Neural Network (TGNN) that integrates semantic text representations, recurrent temporal updates, and graph-based aggregation. Without relying on previous truthfulness labels, the model achieves a Macro-F1 of 0.855, demonstrating that structural and content-based signals provide substantial predictive power. Incorporating prior labels further improves performance (Macro-F1 0.910), highlighting strong temporal persistence.

Our results emphasize the importance of jointly modeling semantic, structural, and temporal signals when predicting behavioral evolution in polarized online environments.

1 Introduction

Online social platforms play a central role in shaping political discourse and information diffusion. While mainstream platforms have been widely studied, alternative networks with distinct moderation policies and ideologically concentrated user bases remain less underexplored. Understanding how behavioral dynamics evolve in such environments requires jointly modeling content, temporal persistence, and network structure.

Truth Social, launched in 2022 by Trump Media & Technology Group following the suspension of former U.S. President Donald J. Trump from major platforms after the January 6 United States Capitol attack, positions itself as an alternative network emphasizing limited content moderation and political expression. Its politically engaged and relatively homogeneous user base provides a natural setting for studying network-structured behavioral dynamics in polarized contexts.

In this work, we analyze temporal truthfulness dynamics within a large-scale Truth Social follower network. We construct a rolling user-level truthfulness measure and empirically demonstrate wave-like growth patterns, localized structural clustering, and strong exposure-dependent transition risks.

Motivated by these observations, we propose a Temporal Graph Neural Network (TGNN) that integrates semantic text embeddings, recurrent tempo-

ral state updates, and follower-network aggregation. We evaluate the model under two settings: a non-autoregressive configuration that excludes prior truthfulness labels, and an autoregressive variant that includes them. The non-autoregressive model achieves strong predictive performance, indicating that structural and semantic signals alone are highly informative. Incorporating prior labels further improves performance, revealing substantial temporal persistence.

Together, our findings highlight the complementary roles of content, network exposure, and temporal dependence in modeling behavioral evolution in politically polarized online environments.

2 Data

The dataset used in this study is derived from the publicly released corpus described in **?**. Because Truth Social does not provide a public API, data were collected via automated web scraping of publicly accessible user profiles.

The crawl began with the account `@realDonaldTrump` and expanded through follower relationships in a breadth-first manner. For each user, the dataset includes:

- User metadata (e.g., follower and following counts),
- Directed follower relationships,
- Authored posts ("Truths") and associated interactions.

The collection was conducted between September 4 and October 14, 2022, resulting in a network of 65,536 users along with their historical posts available at crawl time.

2.1 Data Model and Graph Construction

The data were stored in a relational schema linking users, posts, and interaction metadata. From this structure, we construct a directed temporal graph $\mathcal{G} = (V, E, \mathcal{T})$, where:

- V denotes the set of users,
- E denotes directed follower relationships,
- $\mathcal{T}$ denotes discrete weekly time intervals.

Because the full platform network is not publicly accessible, $\mathcal{G}$ represents an induced subgraph over the observed users. The follower structure E is treated as static over the observation period, while each node $v \in V$ is associated with time-indexed activity features derived from posting behavior.

This formulation enables joint analysis of structural position and temporal user activity.

Network Structure. The follower network exhibits strong heterogeneity in connectivity. While most users maintain relatively few connections, a small fraction accumulates disproportionately large follower counts. This centralization implies unequal exposure and influence potential across the network.

User Activity. Posting behavior is similarly skewed. A minority of accounts generates a substantial share of total content, whereas many users post only sporadically. Consequently, aggregate behavioral patterns may be driven by a limited subset of highly active users.

Temporal Activity. To characterize aggregate engagement dynamics, we compute the total number of posts per week (Figure 1). Weekly activity fluctuates over time, indicating non-stationary behavior and motivating temporally aware modeling approaches.

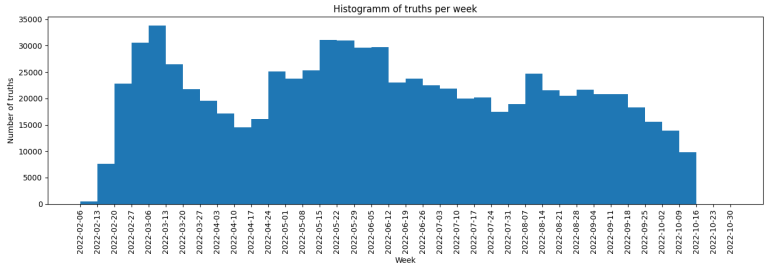


Figure 1: Number of posts per week over the observation period.

Preprocessing. To ensure structural and temporal consistency, we apply the following filters:

- Removal of users without follower or following relationships,
- Removal of users without recorded posts,

- Removal of posts with invalid or missing timestamps.

After filtering, all retained users are embedded in a structurally connected subgraph and associated with temporally valid activity records. The resulting dataset forms the basis for subsequent analysis.

3 Feature Engineering

3.1 Behavioral Features

Behavioral modeling was decomposed into three dimensions: writing style, temporal rhythm, and activity intensity.

3.1.1 Stylometric Clustering

To capture writing style independent of topic, we used character-level TF-IDF representations (3–5 grams), preserving casing to retain stylistic signals such as punctuation, capitalization, and emoji usage.

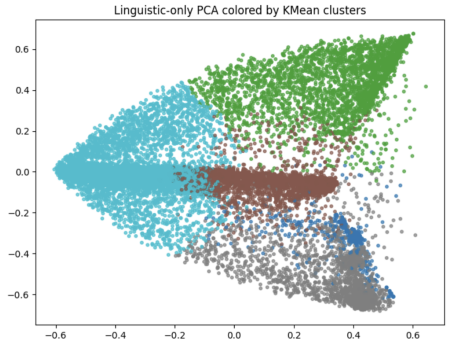
The sparse TF-IDF matrix was reduced using Truncated SVD (50 components) and L2-normalized to operate in cosine space. Silhouette evaluation over $k \in \{4, \ldots, 9\}$ selected $k = 5$ (0.182) as the optimal number of clusters. K-means was applied to obtain the final `style_cluster`.

Visualization and Cluster Distribution The stylometric embedding was projected into two dimensions using PCA (Figure 2a), revealing partially separable stylistic regions with expected overlap in high-dimensional linguistic data.

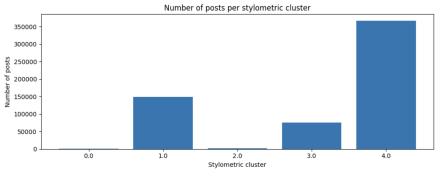
The cluster distribution (Figure 2b) shows one dominant writing style, two moderately represented groups, and two smaller clusters capturing rare or outlier stylistic patterns. Overall, while stylistic variation exists, a prevailing communication norm characterizes the majority of users.

3.2 Temporal Posting Patterns

We evaluated temporal posting behavior using monthly and day-of-week distribution features. Silhouette analysis suggested $k = 2$ as the optimal clustering configuration (0.264). However, cluster distribution revealed that the vast majority of users fall into a single dominant temporal regime, with only a small subgroup exhibiting distinct temporal behavior. This indicates that posting rhythm is largely homogeneous across the platform, with limited



(a) PCA projection of stylometric embeddings colored by K-means clusters ($k = 5$).



(b) Distribution of posts across stylometric clusters ($k = 5$).

Figure 2: Stylometric clustering analysis results.

evidence of strong temporal segmentation. Consequently, temporal features were treated descriptively rather than used as a primary clustering dimension in downstream analysis.

3.2.1 Activity-Based Clustering

User engagement intensity was captured using:

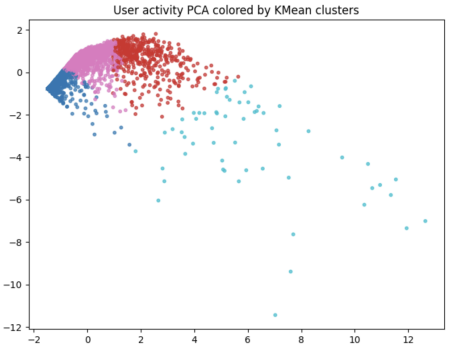
- Total posts
- Active days
- Lifespan (days)
- Posts per active day

After standardization, silhouette analysis showed strong separability (0.519 at $k = 2$). We selected $k = 4$ (0.508) to model engagement tiers while maintaining high cluster quality.

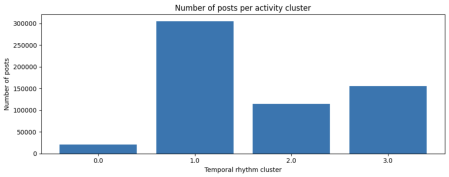
Clear separation indicates strong stratification of users into low, moderate, high, and highly intensive engagement tiers.

3.3 Topic Modeling and Labeling

To identify thematic structure in Truth Social posts, we implemented a semantic topic modeling pipeline combining sentence embeddings, matrix factorization, and LLM-based labeling.



(a) PCA projection of activity features ($k = 4$).



(b) Distribution of users across activity clusters.

Figure 3: Activity clustering analysis results.

Feature Representation. Each post was encoded using the Sentence-Transformers model `all-MiniLM-L6-v2`, producing 384-dimensional sentence embeddings. The embeddings were L2-normalized to operate in cosine space, allowing semantically similar posts to be close in vector space. This approach captures contextual meaning beyond simple keyword frequency.

Topic Extraction. We applied Non-negative Matrix Factorization (NMF) to extract latent topics from the embedding matrix. Since NMF requires non-negative inputs, negative embedding values were clamped to zero before factorization. We initially experimented with $K = 30$ topics; however, qualitative inspection revealed overlapping and semantically redundant themes. Reducing the model to $K = 20$ produced more coherent and distinct topic groupings. The resulting post-topic matrix was normalized so each post forms a probabilistic topic distribution.

Representative Posts and Labeling. To interpret each topic, we selected the top 15 posts with the highest topic weight as representative examples. These exemplars were provided to a locally hosted LLM (LLaMA3 via Ollama), which was prompted to generate a concise 3–6 word topic label without explanation or punctuation. This ensured consistent and human-readable labels across topics.

Dominant Topic Assignment. Each post was assigned a dominant topic via arg max over its topic mixture. Figure 4 shows the distribution of posts

across topics. To examine user-level engagement, we computed the average

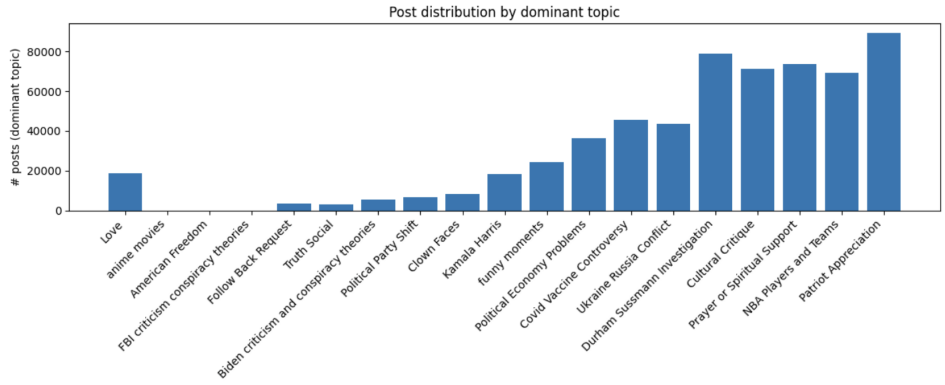


Figure 4: Distribution of posts by dominant topic.

topic weight per author and considered a topic present if its mean weight exceeded 0.05. The number of users associated with each topic is shown in Figure 5.

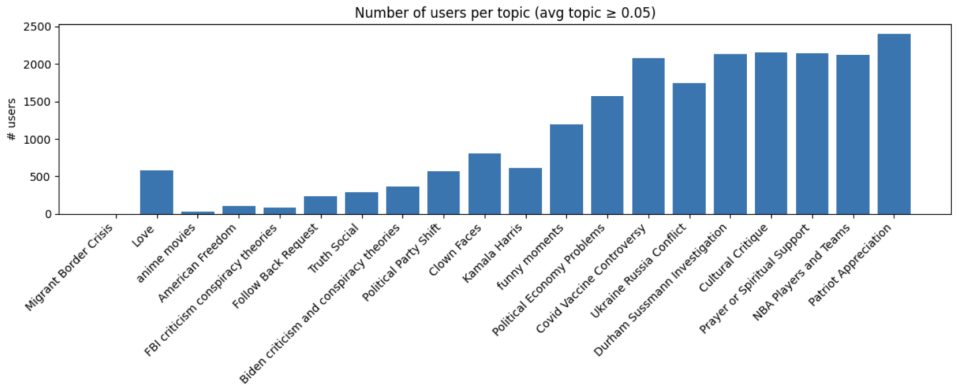


Figure 5: Number of users associated with each topic (average weight ≥ 0.05).

Observations. Both the post-level and user-level distributions exhibit a highly similar and strongly right-skewed structure. The same set of dominant political topics account for the majority of both posts and participating users. The close alignment between the two distributions suggests that high-volume topics are not driven by a small number of highly active users, but rather reflect broad engagement across the user base. Conversely, niche topics

remain marginal both in post volume and in user participation. Overall, the results indicate that thematic dominance is consistent at both the content and user levels.

4 Empirical Characterization of Truthfulness Dynamics

We first define a user-level temporal truthfulness measure and then analyze its structural and temporal behavior within the follower network.

4.1 Temporal User-Level Truthfulness

Let $c_{u,i}$ denote a comment authored by user u in time interval i, and let $\ell(c_{u,i}) \in \{0, 1\}$ be its binary truthfulness label, where 0 represents a truthful comment and 1 otherwise.

To capture short-term behavioral persistence rather than isolated posts, we define a rolling aggregation over three consecutive intervals. For each user u and time t, we compute:

$$y_u(t) = \min\left(\frac{1}{3}\sum_{i=t-2}^{t}\sum_{c\in C_u(i)}\ell(c),\ 1.0\right), \tag{1}$$

where $C_u(i)$ denotes the set of comments authored by user u in interval i.

The resulting score $y_u(t) \in [0, 1]$ represents the recent proportion of untruthful content, smoothed over time. This rolling formulation reduces noise from single comments and reflects short-term behavioral tendencies.

For graph-level analysis, we derive a binary state variable:

$$d_u(t) = \begin{cases} 1 & \text{if } y_u(t) = 1.0 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Users with $d_u(t) = 1$ are considered fully untruthful within the recent window.

The temporal sequence $\{d_u(t)\}$ enables analysis of prevalence, clustering, and transition dynamics.

4.2 Temporal Evolution of Untruthful Behavior

Figure 6 shows the number of users classified as untruthful across time intervals.

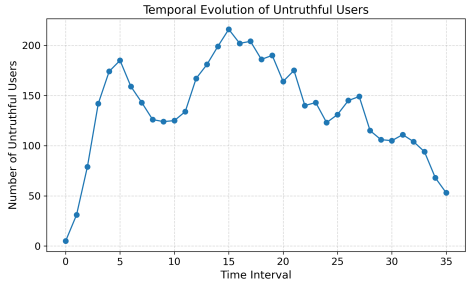


Figure 6: Number of untruthful users over time.

Untruthful prevalence exhibits a two-stage growth pattern, with an initial rapid increase followed by a second, higher peak and subsequent gradual decline. The sharp expansion phase suggests temporally correlated transitions rather than independent behavioral shifts.

Such wave-like dynamics are consistent with diffusion-like processes, though alternative explanations such as exogenous events or platform-level shifts may also contribute.

4.3 Structural Concentration

To assess whether untruthful users cluster in the network, we compute (i) assortativity with respect to $d_u(t)$ and (ii) the fraction of edges connecting two untruthful users.

Figure 7 shows both metrics over time.

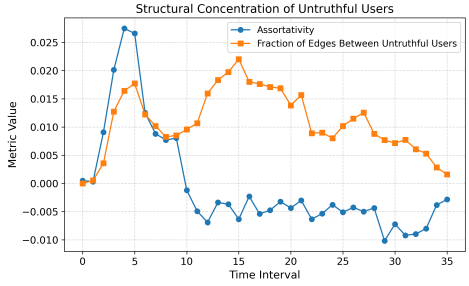


Figure 7: Assortativity and fraction of edges between untruthful users over time.

During the expansion phase, the fraction of within-group edges increases substantially, and the induced subgraph of untruthful users forms a dominant

connected component. This indicates localized structural concentration.

At later intervals, global assortativity becomes slightly negative while the within-group edge fraction remains elevated. This suggests that untruthful users form cohesive regions without complete global segregation of the network.

4.4 Exposure and Transition Risk

To quantify exposure effects, we compute the risk ratio of transitioning from truthful to untruthful in interval $t + 1$ conditional on following at least one untruthful user at time t.

Figure 8 shows the resulting risk ratio over time.

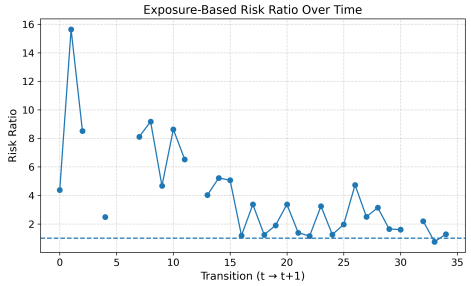


Figure 8: Risk ratio of becoming untruthful given exposure to untruthful neighbors.

In early intervals, the risk ratio exceeds 8 and in some cases surpasses 15, indicating a strong association between exposure and subsequent behavioral change. As overall prevalence increases, the risk ratio declines but remains above 1 for most transitions, consistent with saturation effects in diffusion-like processes.

4.5 Summary of Empirical Findings

The empirical analysis reveals three consistent patterns:

- **Temporal persistence:** Untruthful behavior evolves in waves rather than independently across users.
- **Localized clustering:** Untruthful users form cohesive structural regions without complete polarization.

- **Exposure dependence:** Following untruthful users is strongly associated with increased transition risk, particularly during early growth phases.

Together, these findings indicate that user-level truthfulness exhibits both temporal dependence and network-structured correlation. These observations motivate predictive models that jointly incorporate sequential history and structural neighborhood information, as formalized in the following section.

5 Feature Representation

All feature types were introduced in the Feature Extraction section. Here, we describe how they are converted into numerical vectors and integrated into the temporal graph neural network.

5.1 Text-to-Vector Transformation

Raw comment text is transformed into a fixed-dimensional numerical representation using a pre-trained transformer model.

For each comment, the text is tokenized and passed through the transformer. The contextualized embedding of the `[CLS]` token is extracted as a holistic representation of the comment. To reduce dimensionality and adapt the embedding to the downstream model, the transformer output is passed through a feed-forward projection network, producing a compact fixed-size text vector.

The transformer parameters are kept frozen during training. This ensures stable semantic representations and prevents overfitting, while the projection layer allows task-specific adaptation of the embedding space.

5.2 Numerical Encoding of Structured Features

All non-textual features are converted into numerical form prior to model integration. Categorical variables are represented using one-hot encoding, while continuous variables are standardized via z-score normalization. Count-based engagement features are log-transformed before normalization to reduce skewness. Probabilistic outputs, such as truth or sentiment scores, are retained as continuous values, and multi-cluster assignments are encoded as normalized frequency vectors.

5.3 Feature Concatenation

After conversion, all components are concatenated into a single high-dimensional feature vector:

$$x = [x_{\text{text}}, x_{\text{label}}, x_{\text{categorical}}, x_{\text{sentiment}}, x_{\text{engagement}}, x_{\text{cluster}}].$$

This unified vector preserves semantic content, probabilistic predictions, behavioral attributes, and interaction statistics within a single representation space.

The resulting comment-level vectors serve as input to the comment encoder of the Temporal Graph Neural Network described in the following section.

6 Temporal Graph Neural Network Architecture

To capture the joint temporal and structural dynamics of user-level truthfulness, we propose a Temporal Graph Neural Network (TGNN) that integrates comment-level encoding, temporal state evolution, and graph-based message passing within a unified framework.

6.1 Temporal User Representation

Within each time interval, a user may produce multiple comments. These comment feature vectors are aggregated using a GRU-based encoder to produce a fixed-size representation summarizing the user's activity during that interval. If no comments are present, a zero vector is used.

Each user maintains a hidden state that evolves sequentially across time intervals. The state update is implemented using a GRUCell that integrates the current interval's comment embedding, the previous hidden state, and optionally the prior truthfulness signal. This recurrent formulation captures behavioral persistence and temporal dependencies in user activity.

6.2 Graph-Based Propagation

After the temporal update, user representations are propagated through the follower network. For each user, hidden states of followed accounts are aggregated via mean pooling and combined with the user's own state through learnable linear transformations followed by a non-linear activation.

This step enables structural exposure effects to influence user representations while preserving individual temporal dynamics.

6.3 Prediction and Modeling Perspective

At each time interval, the updated user representation is mapped to a scalar logit and transformed via a sigmoid function to obtain a probability estimate of untruthfulness.

Overall, the proposed TGNN jointly integrates:

- Semantic modeling (text features),
- Temporal modeling (recurrent state evolution),
- Structural modeling (graph-based message passing).

By combining these components, the model captures non-linear behavioral evolution driven by both individual activity and network exposure.

7 Training and Experimental Setup

7.1 Optimization Procedure

The model is trained using the binary cross-entropy loss with logits. This formulation allows direct optimization of probabilistic predictions without requiring an explicit sigmoid layer during training.

Parameters are optimized using the Adam optimizer with weight decay. Gradient clipping is applied to stabilize training and prevent exploding gradients in the recurrent components.

Training is performed for a maximum of E epochs with early stopping based on validation macro-F1. If validation performance does not improve for P consecutive epochs, training is terminated and the best-performing model is retained.

7.2 Temporal Data Split

To preserve chronological consistency, we employ a temporal split. Let T denote the number of time intervals.

- The first 80% of intervals are used for training.
- The remaining 20% are used for validation.

This setup ensures that future information is not used to predict past behavior and reflects a realistic forecasting scenario.

7.3 Decision Threshold Selection

The model outputs a probability estimate of user-level untruthfulness. To obtain binary predictions, a classification threshold τ is applied:

$$\hat{y} = \begin{cases} 1 & \text{if } p \geq \tau \\ 0 & \text{otherwise.} \end{cases}$$

We evaluate multiple thresholds and select the one that maximizes validation macro-F1. The optimal threshold is reported alongside the final results.

7.4 Evaluation Metrics

Model performance is evaluated using class-wise precision, recall, and F1-score, as well as macro-F1 and overall accuracy. Macro-F1 serves as the primary evaluation metric, as it accounts for class imbalance by weighting both truthful and untruthful classes equally.

8 Results

We first report the performance of the temporal graph neural network without incorporating the previous truthfulness state of a user (`use_past_y=False`). This ensures that predictions are based exclusively on textual content and structural information from the network, avoiding autoregressive dependence on prior labels.

The best-performing configuration under this constraint (`threshold=0.35`, `use_graph=True`, `use_comments=True`) achieved a Macro-F1 score of 0.855 and an overall accuracy of 0.967. The F1 score for the untruthful class was 0.728, while the truthful class achieved an F1 score of 0.982. Precision for untruthful users reached 0.75 with a recall of approximately 0.71, indicating that the model maintains a balanced trade-off between false positives and false negatives. At the same time, the extremely high precision and recall for the truthful class demonstrate that the model does not collapse into trivial majority class prediction despite the underlying imbalance.

These results show that combining content-based representations with structural graph aggregation substantially improves predictive performance even in the absence of explicit temporal self-dependence.

The training dynamics further confirm stable convergence. The loss decreases consistently across epochs and stabilizes after roughly 20 epochs without signs of instability. Precision and recall for both classes improve

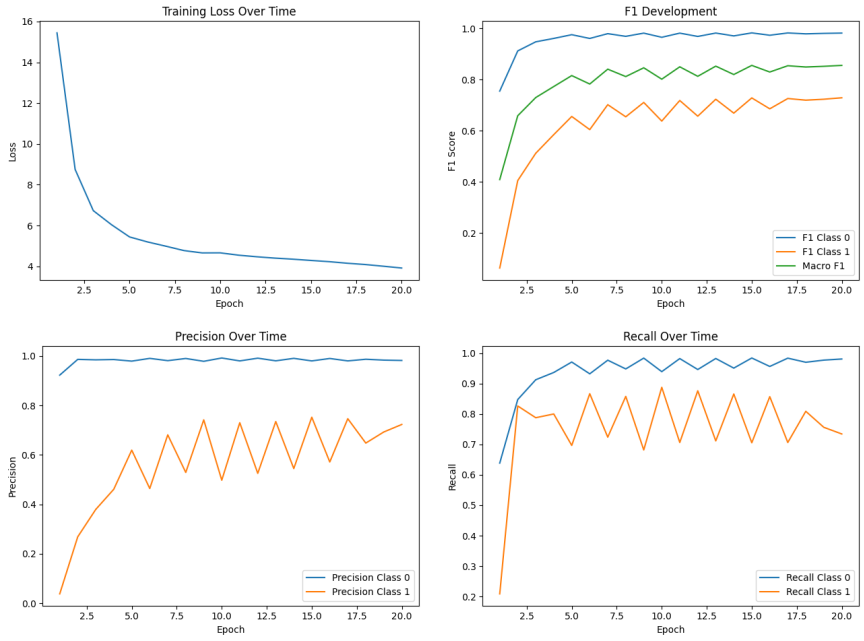


Figure 9: Training dynamics of the temporal graph neural network without temporal self-dependence.

steadily, with a noticeable increase in untruthful-class precision during later epochs, suggesting that the model progressively learns discriminative structural patterns. The Macro-F1 score stabilizes above 0.85 toward the end of training, indicating reliable generalization on the validation set.

When incorporating the previous truthfulness state of a user (`use_past_y=True`), the model achieves a Macro-F1 score of 0.910 and an overall accuracy of 0.976, substantially outperforming the non-autoregressive configuration. The improvement highlights the strong temporal persistence of user-level truthfulness behavior.

However, incorporating previous truthfulness values requires access to ground-truth labels from earlier time steps. If these labels are already available and reliable, the prediction problem becomes largely autoregressive. In such a scenario, one could directly exploit temporal label persistence without necessarily relying on the structural modeling capacity of the graph neural network. In the extreme case where truthfulness is continuously labeled over time, the need for graph-based inference diminishes substantially, as the historical labels themselves carry strong predictive power.

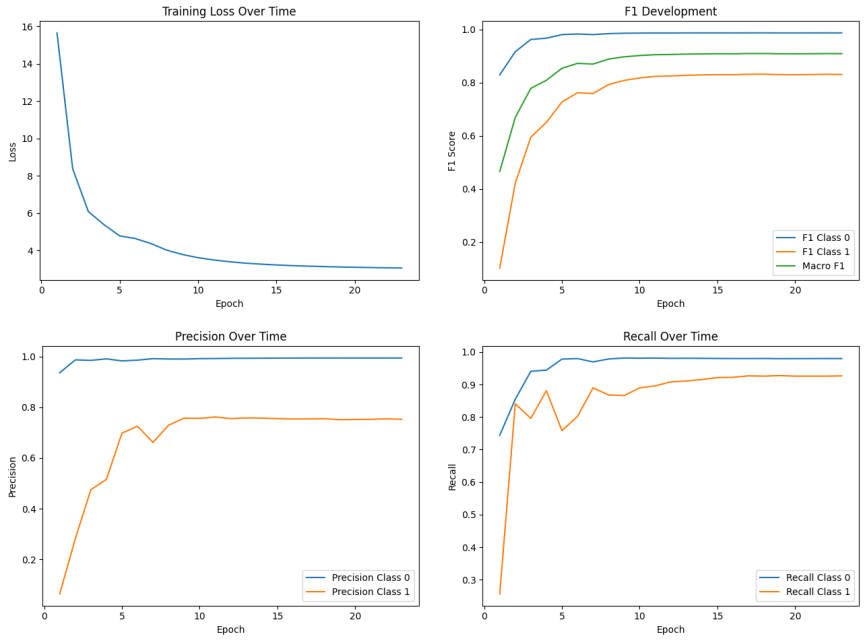


Figure 10: Training dynamics of the temporal graph neural network without temporal self-dependence.

Nevertheless, the high recall of 0.926 for the untruthful class suggests a practically meaningful compromise. Instead of exhaustively labeling all users at every time step, one could apply selective labeling: initial ground-truth annotations could be used to bootstrap the model, which then identifies high-risk users for further verification. Such a semi-supervised strategy preserves the value of structural and content-based modeling while limiting manual annotation effort.

References