SoMeR: Multi-View User Representation Learning for Social Media

Siyi Guo[a], **Keith Burghardt**[a], **Valeria Pantè**[a] **and Kristina Lerman**[a]

[a]Information Sciences Institute, University of Southern California, USA

Abstract. User representation learning aims to capture user preferences, interests, and behaviors in low-dimensional vector representations. These representations have widespread applications in recommendation systems and advertising; however, existing methods typically rely on specific features like text content, activity patterns, or platform metadata, failing to holistically model user behavior across different modalities. To address this limitation, we propose SoMeR, a Social Media user Representation learning framework that incorporates temporal activities, text content, profile information, and network interactions to learn comprehensive user portraits. SoMeR encodes user post streams as sequences of timestamped textual features, uses transformers to embed this along with profile data, and jointly trains with link prediction and contrastive learning objectives to capture user similarity. We demonstrate SoMeR's versatility through two applications: 1) Identifying inauthentic accounts involved in coordinated influence operations by detecting users posting similar content simultaneously, and 2) Measuring increased polarization in online discussions after major events by quantifying how users with different beliefs moved farther apart in the embedding space. SoMeR's ability to holistically model users enables new solutions to important problems around disinformation, societal tensions, and online behavior understanding.

1 Introduction

User representation learning aims to learn low-dimensional vector representations of users that capture their preferences, interests, and behaviors [22]. These representations are used widely in commercial applications, such as personalized recommendation, targeted advertising, and user modeling [2, 42, 5, 17, 37]. Applications of user representation learning extend well beyond commercial sector to the social media domain, where they are used to uncover latent factors of user online behavior that give insights into public attitudes and societal trends, and help understand the formation of online echo chambers [29]. Prior studies have utilized learned representations to detect social bots and inauthentic information operation drivers [26], identify suicidal ideation [36] and detect hate speech [31, 8]. These methods, however, usually depend on specific features, for example, content-based features that use text [14] or combine text with images [29], activity-based features [25, 26], or platform-specific features [1, 7]. As a result, these approaches are not able to holistically model user behavior by combining multiple types features from content, temporal activity and interactions between users.

Multi-view user representation learning for social media that combines multiple streams of evidence poses a number of challenges. User populations on social media are highly heterogeneous, composed of individuals with different beliefs, attitudes, interactions and behaviors. A few users within a population are prolific posters while the vast majority of others post only infrequently. As a result, the temporal user activity is sparse, which presents a challenge for traditional time series analysis methods. Similarly, a few users have large numbers of followers while the vast majority are followed by a handful of others. In addition, for many tasks, ground truth data is not available, or it is difficult to obtain, which makes it hard to train machine classifiers.

To address these challenges, we propose **SoMeR**, a Social Media user Representation Learning framework, which incorporates 1) temporal activities, 2) texts of posts, 3) profile information and 4) network interactions to learn a comprehensive portrait of online users. These features are universal across different social platforms, making our framework very flexible and adaptive. Moreover, this framework allows us to discover similar users in heterogeneous populations with different beliefs, attitudes, and behaviors, thereby creating new solutions to difficult and important problems.

The method works as follows. We first encode the stream of a user's posts as a sequence of triplets composed of $(timestamp, textual feature, value)$. This helps address the challenge of modeling sparse temporal activity data where users post only infrequently. We encode the contextual information of these triplets into an embedding using a transformer-based architecture [40]. We combine this triplet embedding with user profile embedding, and impose two jointly trained objectives: (1) network link prediction to learn interactions between users, and (2) contrastive learning to pull users with similar posting histories closer and push dissimilar users farther away. In the end, the model learns an embedding space that is aware of user similarity and heterogeneity across temporal, textual, network connection and user profile dimensions.

The pre-training step described above learns enriched user representations from multiple features, which can be used in unsupervised settings where annotated data is hard to obtain. The method can also be used within supervised learning. By fine-tuning our model with fewer data, we can adapt them to a range of downstream tasks. For example, by fine-tuning with user ideology, we create an ideologically-aware embedding space to answer questions about polarization.

The framework is extremely versatile and can be adapted to study diverse research questions. We demonstrate framework's generalizability and versatility in two applications. The first application allows us to identify inauthentic accounts associated with online information operations from among a multitude of online activity, i.e., infor-

mation operation (IO) drivers. Social media has emerged as a battleground for IOs, enabling malicious actors to mount coordinated influence campaigns to amplify societal divisions or sway public opinion [44]. The proposed method achieves this goal by identifying accounts that post similar texts at similar times — a signature of coordination. We learn user representations from social media data and train a supervised model to detect IO drivers that share similar content at similar times, which is a signature of coordinated online influence campaigns [28, 26]. When test on known IOs from three countries that have been verified by the platform X (formerly Twitter), our method shows outstanding performance in detecting covert influence campaigns.

Our second application uses learned representations to measure how events widen partisan division and increase polarization of online discussions. Specifically, using data from online discussions about the overturning of federal protections for abortion in the U.S., we show the impact of the decision on the activities of partisans. We use the proposed user representation method to study how populations with different political beliefs responded to the decision. The analysis reveals an increased polarization: users with the same political belief moved closer, whereas users with different believes moved farther apart. These two applications highlight the flexibility and universality of the proposed method.

Our proposed framework helps to bridge the divide between user representation learning and socio-political analysis, providing a promising method to deepen our understanding of user heterogeneity and its dynamics, and paving the way for more informed decisions and interventions in an increasingly interconnected world.

2 Related Works

User Representation Learning User representation learning has gained widespread interest in the recommendation system research area due to its ability to capture meaningful and compact embeddings that represent important characteristics of users' behaviors and preferences [45]. Over the years, researchers have developed many methods [22], including matrix and tensor factorization [2, 15] and deep learning based models, such as auto-encoders [42, 47] and recently more sophisticated transformer-based architectures [5, 17, 37]. Researchers have also advanced the training techniques for these deep learning methods, such as incorporating contrastive learning to learn similarity in data without the need of human labels [27, 5]. Improving from task-specific methods [18, 12] in the early days, many works have also contributed to build universal user representation learning methods which can be generalized to different downstream tasks [45, 17, 37]. The demand for recommendation systems has largely advanced the techniques and methods in user representation learning. However, many of these works formulate the major task to be customization and adaptation of systems to the user's specific needs [22], and most of the methods have only been tested on product preference prediction or user profiling.

User Understanding in Social Domains Beyond recommendation systems, user representation learning is also useful to understand online users and communities through their behaviors and opinions [29]. Mueen et al. [25] and Nwala et al. [26] uses temporal activity features to detect online bot and IO drivers (sometimes known as coordinated users); Hallac et al. [14, 13] experimented with different textual embedding methods, such as TF-IDF, doc2vec and BERT for social media user representation learning; Perozzi et al. [30] uses network features and node prediction to learn user interests; Wang et al. [43] use network embedding method to learn about communities; and Ribeiro et al. [34] use both text and network features to detect hate speech and hateful users online. However, there lacks a universal framework for social analysis that is generalizable to different downstream tasks and incorporates all of the textual, temporal, profile and network features. Our framework, **SoMeR**, addresses these gaps.

3 Methods

We propose a self-supervised framework to learn a latent user embedding space based on the architecture shown in Figure 1. From each user's history, a timeline of texts, we first extract certain textual features, such as the sentence embeddings. Next, to better learn from users with sparse activities, we format the textual features and timestamps into triplets of observations $(timestamp, feature, value)$. These triplets pass through a Triplet Encoder, a transformer-based contextual learning module, and a fusion attention layer, being encoded into a user history embedding. We concatenate it with user's profile embedding from a separate module, obtaining a complete user embedding. We train these encoding modules with two self-supervised objectives: *network link prediction* that learns patterns of interactions, including sharing, following or other connections, and *contrastive loss* that learns user similarity with respect to posting history. This method learns user similarity in a heterogeneous user population without the need for time-consuming human annotations. The method can be easily adapted for various downstream tasks such as supervised learning and unsupervised similarity search.

User History Data Processing In this study, we consider each user's history to be a collection of timestamped texts. Each text can be an original post, a repost, a reply, etc. There can be other types of user activity, such as likes or views, that also provide valuable information. We plan to incorporate these data in future works. To learn from a user's history, first we need to extract the desired textual features that are appropriate for the downstream task. These features can be topics discussed in the posts, emotions expressed in them, or contextual features extracted as text embeddings [11]. In our experiments, we find that using contextual sentence-BERT embeddings [33] leads to a flexible and powerful representation, and hence use this approach. However, BERT embeddings yield more complex features than other methods, which can slow performance. To reduce model size and complexity, we perform dimensionality reduction using principal component analysis (PCA) to reduce BERT embeddings to the first five components and treat them as textual features.

Triplet Data Encoder Social media users exhibit highly diverse posting behaviors. A small fraction of users generate most of the contents, whereas majority of users only have few posts over a long period of time [21]. This leads to a sparse matrix of multivariate time series as the input data, which makes representation learning harder. We take inspiration from [38], and represent user posting history as a collection of triplets $(timestamp, feature, value)$. Thus, a dataset of M users can be represented as U:

$$\mathbf{U} = \{(d^u, \mathbf{X}^u)\}_{u=1}^{M} \; where \; \mathbf{X} = \{(t_n, f_n, v_n)\}_{n=1}^{N} \tag{1}$$

Each user u is characterized by their profile feature vector $d^u \in \mathbb{R}^D$ and their posting history $\mathbf{X}^u$, which is a set of N triplets including the timestamp t_n, the feature category f_n and the value of this feature category v_n. Note that there can be more than one feature-value pairs at one time point, and therefore N does not necessarily equal to the total time points.

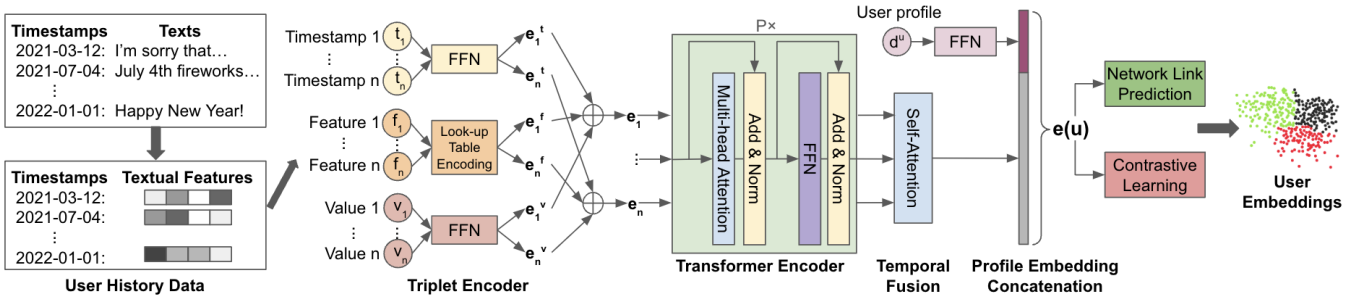

Figure 1. Model Architecture of **SoMeR**. We format a user's posting history into triplets of time, feature, and value, which undergo encoding via a Triplet Encoder, a transformer-based contextual learning module and a fusion attention layer, becoming a user history embedding that is then concatenated to the user profile embedding. Through training with two self-supervised objectives - network link prediction and contrastive loss - our method effectively captures user similarity in the latent space.

Tipirneni and Reddy [38] shows the effectiveness of using a feed-forward network to embed continuous values. We therefore use two separate feed-forward networks to encode timestamp t_n and value v_n into time embedding $e_n^t \in \mathbb{R}^K$ and value embedding $e_n^v \in \mathbb{R}^K$ respectively, where K is the hidden dimension of these embeddings. These networks have one linear layer followed by a tanh activation function. For the feature categories, we use a lookup-table encoder to generate a feature embedding $e_n^f \in \mathbb{R}^K$. Lastly, we add up these three embeddings to be the triplet embedding $e_n^{triplet} \in \mathbb{R}^K$.

$$e_n^t = W_1^t \, tanh(W_2^t t_n + b^t) \quad (2)$$
$$e_n^v = W_1^v \, tanh(W_2^v v_n + b^v) \quad (3)$$
$$e_n^f = LookupEncoder(f_n) \quad (4)$$
$$e_n^{triplet} = e_n^t + e_n^v + e_n^f \quad (5)$$

Transformer Encoder The transformer architecture has been shown to have great performance in representation learning for time series and user behavior sequences [46, 37]. Therefore we choose to use the transformer encoder to better extract a latent representation from the triplet embedding $e_n^{triplet} \in \mathbb{R}^K$. We use L transformer layers. Each layer has H attention heads with learnable key, query and value weights $W^k, W^q, W^v \in \mathbb{R}^{K \times P}$ where P is the hidden dimension size for these weights. After all attention heads are added up and layer normalized, the embedding vector is projected back to K-dimension through a feed-forward network. This network includes two layers with hidden dimension $2K$ and a ReLU activation in the middle. Lastly, layer normalization is applied again. From this transformer module we obtain $e_n^{trans} \in \mathbb{R}^K$. In our experiments and applications, data sizes are within 20 million, and therefore we choose to have a small transformer module with $L = 2$, $H = 4$ and $P = K/H$ which we show in § 4 and § 5 works well for our purposes.

Temporal Fusion After all triplets for a user pass through the triplet encoder and the transformer encoder, we use an attention layer to learn the correlations between different triplets. This gives us the integrated embedding of posting history $e^{hist} \in \mathbb{R}^K$ for each user.

$$a_n = W_1^{attn} \, tanh(W_2^{attn} e_n^{trans} + b^{attn}) \quad (6)$$
$$a_n = \frac{exp(a_n)}{\sum_{i=1}^{N} exp(a_i)} \quad (7)$$
$$e^{hist} = \sum_{n=1}^{N} a_n \, e_n^{trans} \quad (8)$$

where $W_1^{attn} \in \mathbb{R}^{2K}$, $W_2^{attn} \in \mathbb{R}^{2K \times K}$, $b^{attn} \in \mathbb{R}^{2K}$ are trainable weights and intercept.

Profile Embedding Other than the posting history of a user, their profile features, e.g., location and number of followers and friends, can also play an important role. Therefore, we add a feed-forward network to learn a profile embedding $e^{prof} \in \mathbb{R}^K$ from user's profile vector $d^u \in \mathbb{R}^D$, and concatenate it to the user history embedding e^{hist} to obtain a complete user embedding $e^u \in \mathbb{R}^{2K}$.

$$e^{prof} = W_1^{prof} \, tanh(W_2^{prof} d^u + b^{prof}) \quad (9)$$
$$e^u = e^{hist} \oplus e^{prof} \quad (10)$$

where $W_1^{prof} \in \mathbb{R}^{2K}$, $W_2^{prof} \in \mathbb{R}^{2K \times D}$, $b^{prof} \in \mathbb{R}^{2K}$ are trainable weights and intercept, and $\oplus$ is a concatenation operation.

Network Link Prediction Individuals connected in social networks tend to be similar, and their behaviors are often affected by other users in the network [24]. Therefore it's crucial to include network connection features when learning user representations [1]. We design a self-supervised network link prediction objective to train our model to learn interaction activities such as sharing, following and commenting. It is a feed-forward module to perform link prediction, with a binary cross-entropy loss. During training, we consider all pairs of distinct users in each batch. The feature for link prediction is the concatenated embedding of a user pair, and we obtain the links from a self-defined network as the binary labels (e.g. whether a user repost another user). This is described as following:

$$\tilde{y}_{i,j}^{link} = \sigma(W_1^{link} \, ReLU(W_2^{link}(e_i^u \oplus e_j^u) + b^{link})) \quad (11)$$

$$\mathcal{L}_{\text{network}} = -\frac{1}{N_{\text{distinct pairs}}} \sum_{\substack{i,j \in batch \\ i \neq j}} \Big[y_{i,j}^{link} \cdot log(\tilde{y}_{i,j}^{link}) + (1 - y_{i,j}^{link}) \cdot log(1 - \tilde{y}_{i,j}^{link}) \Big] \quad (12)$$

where i, j are indices of two users in a training batch, $W_1^{link} \in \mathbb{R}^{2K}$, $W_2^{link} \in \mathbb{R}^{2K \times 4K}$ and $b^{link} \in \mathbb{R}^{2K}$ are trainable weights and intercept, $\sigma(\cdot)$ is the sigmoid function, $N_{\text{distinct pairs}}$ is the number of all pairs of distinct users in a batch, and y^{link} is the binary label of whether i and j are connected in the network.

Contrastive Learning and Data Augmentation Contrastive learning aims to obtain a latent embedding space in which similar samples are closer and distinct samples are farther from each other. Many prior works in user representation learning have shown its suc-

cess [5, 37]. We adopt self-supervised contrastive learning to learn user similarity in their posting histories. The InfoNCE [27] loss function uses categorical cross-entropy loss to optimize the negative log probability of classifying one positive or similar sample correctly among a set of negative or unrelated samples. In our case, it can be written as:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{\text{batch size}} \sum_{i \in \text{batch}} \left[log \frac{exp(e_i^u e_i^{u^+}/\tau)}{\sum_{j \in batch, i \neq j} exp(e_i^u e_j^u/\tau)} \right] \quad (13)$$

For e_i^u a user history embedding, $e_i^{u^+}$ is its positive pair. $e_j^u \ \forall j \in batch$ where $i \neq j$ are the embeddings of all other users in that randomly retrieved batch, which we consider as negative samples to e_i^u. We further perform temperature scaling with parameter τ, which is tuned during training using grid search in $[0.5, 1, 3, 6]$.

To generate the positive sample paired to each user history embedding, we perform data augmentation on users' triplet data. Although researchers have proposed many time series data augmentation methods [44], many classical methods are not applicable to our scenario. For example, a user history cannot be sub-sequenced or shuffled in time domain. We use another efficient and simple way - bootstrapping with replacement from existing triplets $\{(t_n, f_n, v_n)\}_{n=1}^N$. This ensures the generated set of triplets is similar. We incorporate noise into augmentation by

1. varying the number of random draws between $range(start = (1-\gamma)N,\ end = (1+\gamma)N,\ step = 1)$, where N is the total number of triplets for a user and γ is a hyperparameter to control the noise level,
2. scaling the value v_n by a factor randomly selected between $range(start = 1-\gamma,\ end = 1+\gamma,\ step = 0.5)$, and
3. imposing a lag time, randomly selected between 1 -3 days, on the timestamp t_n of sampled triplets.

During training, we tune γ by grid searching in $[0.1, 0.5, 1, 2, 5]$. The results of the two applications below shows the effectiveness of this augmentation method.

Model Training Finally, the contrastive objective function and the network link prediction objective are jointly trained at the same time. The overall loss is

$$\mathcal{L} = \mathcal{L}_{\text{InfoNCE}} + \lambda \mathcal{L}_{Network} \quad (14)$$

where λ is a hyperparamter to balance between two losses. We perform grid search on λ in $[0.1, 1, 5, 10]$ and decide to use $\lambda = 1$. We also tune the hidden dimension K used in different modules with a grid search in $[32, 64, 128]$ and decide to use $K = 64$. We use Adam optimizer, a learning rate of 5e-5, a cosine decay learning scheduler, an early stopping mechanism monitored by overall loss function, a maximum epoch of 60 and a batch size of 128.

Model Validity To verify that our self-supervised framework indeed learns temporal activity and textual features from the triplet data, we test it on two synthetic datasets. We use synthesized numerical features to mimic textual features in real data. One dataset consists of clusters that vary in timestamps but have the same features and values, and another dataset consists of clusters that vary only in features and values, but having all the same timestamps. On both datasets, our model can successfully detect the clusters, indicating it is able to learn heterogeneity in both temporal activities and numerical feature values (see S.I. for details).

In the next sections, we use two applications - IO driver detection and political polarization analysis - to illustrate the effectiveness and versatility of our framework.

4 Detecting Drivers of Information Operations

Information operations use strategically organized efforts to sway and manipulate public opinions, contaminating the online information ecosystem with disinformation. The rapidly expanding social media platforms provide fertile grounds for these operations [28]. Therefore, it is a critical to detect these IO drivers. IO drivers have different tactics from each other and from non-IO accounts thus leading to distinct behaviors we can identify, such as (1) co-sharing the same posts or URLs, (2) using an identical sequence of hashtags, (3) synchronized behaviors, and (4) high textual similarity in posts. Researchers have developed some effective methods for IO drivers using these features [28, 26, 23].

However, most of these methods are specifically designed for the X platform. As many other platforms, such as Reddit, also host suspicious operations [35], there is a need to develop more generalizable methods for coordination detection. Moreover, IO driver behavior may not be identical to another account, but their behavior could nonetheless be distinguishable. Our user representation learning framework can account for temporal, textual, profile and network connection features, within which synchronized behaviors are ingrained. Moreover, our framework is not limited to any platform-specific features. Here we perform supervised fine-tuning on top of our framework to build an IO driver detector, and test it on three different campaigns to demonstrate its efficacy.

Data We evaluate our method on an X dataset that prior IO driver detection methods have benchmarked on [26, 23]. This set of IO drivers from multiple operations in 21 countries were suspended and released by X because they were associated with malicious IOs and violated the platform terms. We select four campaigns — one in China involving a large number of accounts, one small operation in Egypt and UAE, and two operations in Venezuela which are combined into one dataset for testing the multi-campaign scenario. In future works, we plan to expand our framework to other social platforms.

We collect a set of control users by first collecting the top five keywords and top five hashtags for each IO driver within our X datasets. We then extract 10 random posts that were posted within the timeframe that the IO driver was active (between their first and last post). For each post, we find the post author and query all their posts made within the timeframe the IO driver is active. Table 1 shows the information about these campaigns. We split data by 70%-20%-10% for training, validation and testing.

Table 1. Meta-data of Information Operation Datasets

Campaign	Time Range	# IO Drivers	# Control Users	# Posts
China	2019 - 2021	2016	11366	17M
Egypt-UAE	2016 - 2019	240	2164	4.5M
Venezuela	2017 - 2021	275	4183	10M

Supervised Classification We use a two-step approach to classify IO drivers. First, we pre-train our model in the self-supervised manner with both IO driver and control users using all of their posts and meta-data, but without any IO label. For textual post preprocessing, we remove user mentions, URLs, emojis and all non-ASCII characters, but retain the hashtags. This is a multilingual datasets including more than 50 languages. Therefore we compute the sentence-BERT embedding for each post using the

`stsb-xlm-r-multilingual`[1] and compute the first five components from PCA as its textual features. This is to reduce the number of triplets and to reduce computation complexity. We aggregate data for each user by summing up their PCA embeddings of their posts in 3-day intervals and taking the middle day in the interval as the corresponding timestamp. Other intervals can be used, although this value acts as a reasonable balance for low-activity accounts. This gives us the temporal and textual features. For profile features, we use number of followings and followers, as accounts such as news media outlets with lots of followers can behave very differently from other types users. For network link prediction part, we use the repost network (i.e.retweet network) because co-repost has been shown as a potential indicator of an IO driver [28]. With all these features, we train the model to learn a user embeddings space in which users with similar behaviors are closer together (cf. Figure 1).

In the next step, we perform supervised fine-tuning on the learned model parameters with an additional two-layer feed-forward network for binary classification. This network has a linear layers with a hidden dimension of 128, a ReLU activation, a dropout layer with rate of 0.3, a batch normalization, and a second linear layer that project embeddings onto $\mathbb{R}$ for binary prediction. The sigmoid function is then applied and a binary cross-entropy function is used as the loss. We use same hyperparameters described in § 3.

Baseline Models We compare our method with predominantly used IO driver detection methods [28]. Based on the observation that IO drivers have abnormally similar behaviors, this method first construct similarity networks among users based on various features such as co-sharing, and identify users with similarity in the top-percentile as IO drivers. Following previous works [28, 23], we detect IO drivers based on (1) co-repost and (2) co-URL sharing behaviors if cosine similarity is above or at the 99.5-th percentile, (3) hashtag sequence (using a minimum sequence of 5 identical hashtags in the same order within a post) and (4) text similarity in averaged BERT embeddings over all posts from a user, with a cosine similarity threshold of 0.7.

Table 2. Model Performance on Detecting Information Operation Drivers.

	F1-Scores			ROC-AUC		
	China	**Egypt-UAE**	**Venezuela**	**China**	**Egypt-UAE**	**Venezuela**
BASELINES						
Co-Repost	0.00	0.15	0.26	0.50	0.54	0.58
Co-URL	0.19	0.27	0.30	0.45	0.74	0.52
Hashtag-sequence	0.08	0.20	0.05	0.51	0.56	0.51
Text Similarity	0.13	**0.93**	**0.82**	0.53	0.94	0.85
OURS						
Temporal	0.96±0.01	0.41 ± 0.13	0.57±0.06	0.99±0.01	0.95 ± 0.01	0.90±0.02
Textual	0.97±0.01	0.69 ± 0.06	0.67±0.04	0.99±0.01	0.96 ± 0.02	0.96±0.01
Temporal+Textual	0.98±0.01	0.77 ± 0.05	0.77±0.04	0.99±0.01	0.98 ± 0.01	0.97±0.01
SoMeR	**0.99 ± 0.01**	0.85 ± 0.04	**0.82 ± 0.04**	**0.99 ± 0.01**	**0.99 ± 0.01**	**0.97 ± 0.01**

Model Performance We demonstrate how our framework can effectively learn from user behaviors and network connections, and adequately detect IO drivers. Table 2 show the outstanding performance of our method from the F1-scores and ROC-AUC. Our models are evaluated from 10 random data splits. First we compare our method with the baselines. For all three campaigns, we outperform the co-repost, co-URL and hashtag-sequence baselines. Compared to the text similarity baseline, our method outperforms on China and Venezuela campaigns, and has a higher ROC-AUC score but a lower F1-score on the Egypt-UAE campaign. We hypothesize that our method utilizing only the first five PCA components on post BERT embeddings versus the text similarity baseline using the full BERT embeddings might result in this gap. This can be addressed by using the full BERT embeddings when having more powerful GPUs. Nonetheless, the high F1-scores and AUC scores we achieve demonstrate the effectiveness of our method.

The performance of different baselines also reveal some interesting facts. Co-sharing and hashtag usage seem to be less distinguishable from non-IO accounts in all three campaigns. On the other hand, by just using text similarity can achieve a very high performance on Egypt-UAE and Venezuela datasets, but the text similarity baseline has relatively low performance on the China dataset, indicating that different campaigns utilize different tactics. These observations demonstrate that only by looking at one type of suspicious behavior pattern is not enough when detecting IO drivers.

Ablation Next, we perform an ablation study on our model to dissect the impact of different features and modules we use. The Temporal model only uses timestamps and three-day post counts as the feature but do not use any textual or network features. The Textual model only uses timestamps and the *average* textual embedding over three-day intervals, which does not reflect temporal activity. The Temporal+Textual model uses timestamps and the *summed* textual embeddings over three-day intervals, as described in § 4 Supervised Classification. This reflects both temporal activity and textual features. The full model SoMeR incorporates network feature on top of the temporal and textual features. We observe that the full model has better performance than any of the ablated models, implying the benefit of including the network link prediction objective. In addition, Temporal+Textual model performs better than Temporal and Textual models. These indicate that each of the four features plays a role to detect IO drivers.

5 Uncovering Shifts in Polarized Discussions

The U.S. society has grown increasingly more polarize. Not only do liberals and conservatives hold sharply different opinions on a range of issues [3], but they also have more negative feelings towards members of the other party, compared to members of their own party [19]. These differences show up not only in political speech, but also in the everyday behaviors [10], on on social media platforms, where liberals and conservatives segregate themselves in different echo chambers, they are reflected in the network structure [6]. Prior research has found that events can shift public opinion, polarizing the population [16, 20, 32]. While many alternative methods for measuring polarization on social media exist, e.g., using network analysis [6], interaction behaviors and emotions [9], and community embeddings [41], they are not well suited for measuring shifts in polarization. Leveraging our multi-view user representation learning, we provide a new way to measure changes in polarization after significant events by tracking user embeddings, and apply it to measure polarization in the online discussions about abortion, a highly contentious issue in the American society. Our framework allows us to isolate specific topics and identify ones that grew more polarized.

Data On June 24, 2022, SCOTUS struck down federal protections for abortion rights. This event sparked massive online discussions, in which users with different political ideologies expressed distinct views [32]. We study a public dataset [4] containing tweets with abortion-related keywords, such as “roevswade”, “prochoice”, and “prolife”. The data spans the entire year of 2022. We select English posts in the U.S. from users with at least 20 posts. This gives us about 10 million posts from 0.1 million users. Using methodology described in Rao et al. [32], we identify each user’s political ideology

[1] https://huggingface.co/sentence-transformers/stsb-xlm-r-multilingual

(liberal/conservative), leaving us with 103K liberals and 18K conservatives. We use the learned user embeddings to track how these two ideological populations change in the user embedding space.

Measuring Event-Driven Polarization We start with pre-training, which learns a user representation space for all liberal and conservative users using their posting histories in 2022. We perform the same text preprocessing as described in § 4, except that the sentence-BERT model we use for this English-only dataset is `sentence-transformers/all-mpnet-base-v2`[2]. We use following and follower counts as profile features, and use a repost (i.e., retweet) network to train link prediction objective. Figure 2(a) shows a t-SNE [39] representation of the learned embedding space for all users for the year 2022. We see the separation of populations, with conservatives clustered in some regions, whereas liberals, who are 85% of all data, are distributed across the entire space.

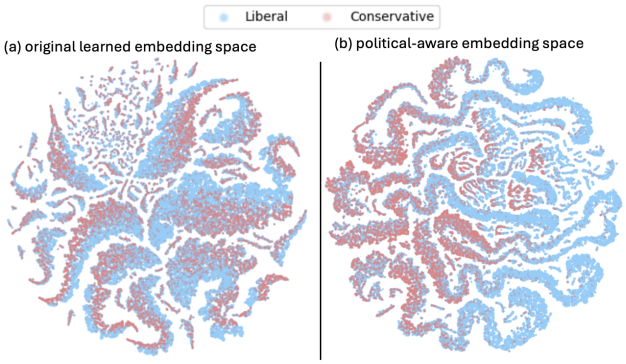

Figure 2. (a) t-SNE [39] of user embedding space learned with both ideology populations over the entire year of 2022. (b) t-SNE of the space in (a) is fine-tuned with data from elite users with known ideologies, making the embedding space more politically-aware.

Next, we fine-tune the model via few-shot learning with political elites.[3] These correspond to well known people, e.g., U.S. politicians, with a known ideology. We found 358 in our abortion data. This fine-tuning step trains the model to predict user ideology, thereby aligning the embedding space to political ideology. For the analysis we discuss below, we test in both the original and the fine-tuned politically-aware embedding and find the same trends. Figure 2(b) shows a t-SNE visualization of the politically-aware user embeddings after few-shot fine-tuning. We see that the conservative users have moved closer in one direction, indicating the effectiveness of fine-tuning at distinguishing these disparate groups.

In this politically-aware embedding space, to identify shifts in ideological polarization driven by the SCOTUS ruling, we measure how embeddings changed for the same users after the ruling. We first establish a baseline period when users were not affected. There was a leak about this ruling on May 3rd, 2022. Therefore we select the period of January 1st to May 2nd 2022 as the baseline period to avoid interference from this event. We then determine the period to observe the impact of ruling to be June 24th to November 11th 2022. We select this end date to reduce the confounding effect of the 2022 US midterm elections. Next, we select users who posted in both time periods, take their posting history in these two periods separately, and project these users onto the same politically-aware embedding space we have learned. By visualizing the embeddings of the same users in the baseline period and after the SCOTUS ruling, we find a clear shift especially in the conservative population (Figure 3). In the baseline period, conservative users were more uniformly distributed in the t-SNE embedding space, but they moved closer together after the SCOTUS ruling. This indicates that the conservative users became more similar in the content of their posts or in their behaviors.

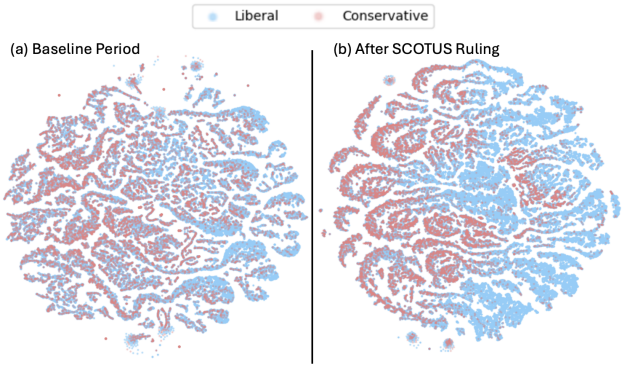

Figure 3. Users shifted in the embeddings space after the SCOTUS abortion ruling. Points in (a) are encoded with user post histories between January 1st to May 2nd, 2022. Points in (b) are encoded with the post histories from the same users between June 24th to November 8th, 2022. Points in (a) and (b) are both projected in the same embedding space.

To further quantify this effect, we find the k-nearest-neighbor (kNN) and check for each user the percentage of neighbors with the same ideology (in-group) and different ideology (out-group).We can infer how clustered each population is by the share of the nearest neighbors who are from their in-group. On the other hand, the share of out-group neighbors tells us how far away the two ideological populations are. Figure 4 shows the percent change in the mean of these four metrics across populations, from the baseline period to the after the ruling. Consistent with Figure 3, we see in the "All Data" row that the share of in-group neighbors increased for both conservatives and liberals and the share of out-group neighbors decreased. These indicate that *users with same ideology moved closer together, and users with different ideologies moved farther apart in the embedding space*, implying that polarization increased. The conservatives especially became more clustered after the ruling. We perform the same analysis with k=50, 200, 500 nearest neighbors, all showing the same trends.

To dig deeper, we explored how this effect depends on the topic users discuss. Rao et al. [32] identified topics discussed in each post, including religion, bodily autonomy, fetal rights and women's health. We created one subset with posts related to liberal-centric topics, i.e. bodily autonomy and women's health, and another subset with posts related to conservative-centric topics, like religion and fetal rights. Then we perform the same analysis for each subset. Figure 4 shows a consistent overall trend that users with same ideology move closer and users with different ideologies move farther away. Interestingly, we also observe that *each population coalesced when discussing partisan topics promoted by the opposite ideology* - in-group neighbors of conservative users increased more on Liberal-centric topics than on Conservative-centric topics, and similarly in-group neighbors of liberals increased more on Conservative-centric topics than on Liberal-centric topics. Users "united against a common enemy" in these online discourses.

Ablation We again perform an ablation study to assess how different types of evidence contribute to the shift of user embeddings we observe. Figure 5 compares three ablated models and the full model.

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[3] https://github.com/sdmccabe/new-tweetscores?tab=readme-ov-file

Figure 4. Users with same ideology moved closer after SCOTUS abortion ruling, and users with different ideologies moved away. The color represents the *percent change in the mean of these nearest neighbor metrics across populations* from baseline period to after ruling period. *** indicates that the means are significantly different in two time periods with p-value < 0.0001.

Using the Temporal model results in very little change in all four metrics, indicating that users did not change their activity patterns much after the ruling. Instead, we see much bigger changes when using the Textual model. This implies that users of different ideologies diverged more in the topics and content they discussed after the ruling. Note that the full model SoMeR gives smaller changes than using Temporal+Textual model. During pre-training, the model learns the repost network aggregated over the entire year of 2022, including baseline period and after ruling period. With the repost network not changing between these two periods, user embeddings change less.

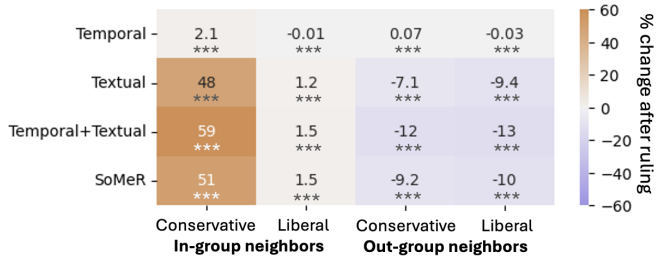

Figure 5. Comparison of changes observed in the embedding spaces learned by ablated models and the full model. Temporal features contributed little whereas textual features are the greater factor. The color represents the *percent change in the mean of these four metrics across populations* from baseline period to after ruling period. *** indicates that the means are significantly different in two time periods with p-value < 0.0001.

6 Conclusion

In this work, we propose a multi-view user representation learning framework, **SoMeR**, which is better tailored in the socio-political domain. Our framework learns from a variety of user features including 1) temporal activities, 2) texts of their posts, 3) profile information and 4) network connections. It is versatile and generalizable to different downstream tasks and across different social platforms. We have demonstrated its effectiveness and generalizability by applying it to detect IO drivers with outstanding performance, and to uncover political polarization in online abortion discourses.

There are certain limitations that we plan to address in future works. First, both of our applications are on X datasets. We plan to apply our framework to other platforms such as Reddit. Second, the IO driver detection application is imperfect, and it might not reach the state-of-the-art for the X platform. Third, although using the network features has been shown to contribute in model learning, the network link prediction objective is only a part of the pre-training process. During inference, if we want to project a user onto the learned embedding space, we do not use this user's network connections. In future works, we plan to address both limitations by improving model architecture to better use the network features, such as using a network graph embedding module. Fourth, in § 5 the machine-labeled user political ideologies are not perfect. However, Rao et al. [32] extensively validated the high accuracy of their method, showing the trustworthiness of these labels when looking at the aggregated population level. Finally, the utilization of temporal features opens up exciting possibilities to track how users and communities evolve over time. Change point detection on user representations will be an interesting direction to expand this framework to a more powerful tool.

Overall, our framework represents a step toward bridging the gap between universal user representation learning and the socio-political domain, offering a potent tool to understand user heterogeneity. We hope it contribute to foster more discerning decision-making and policy interventions in our progressively interconnected world.

References

[1] H. AlMahmoud and S. AlKhalifa. Tsim: a system for discovering similar users on twitter. *Journal of Big Data*, 5(1):39, 2018.

[2] P. Bhargava, T. Phan, J. Zhou, and J. Lee. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *Proceedings of the 24th international conference on world wide web*, pages 130–140, 2015.

[3] P. R. Center. The partisan divide on political values grows even wider. *Trust, facts and democracy*, 2017.

[4] R.-C. Chang, A. Rao, Q. Zhong, M. Wojcieszak, and K. Lerman. # roeoverturned: Twitter dataset on the abortion rights controversy. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 17, pages 997–1005, 2023.

[5] M. Cheng, F. Yuan, Q. Liu, X. Xin, and E. Chen. Learning transferable user representations with sequential behaviors via contrastive pre-training. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 51–60, 2021. doi: 10.1109/ICDM51629.2021.00015.

[6] M. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, F. Menczer, and A. Flammini. Political polarization on twitter. In *Proceedings of the international aaai conference on web and social media*, volume 5, pages 89–96, 2011.

[7] S. Dahiya, G. Kumar, and A. Yadav. A contextual framework to find similarity between users on twitter. In *Proceedings of Second Doctoral Symposium on Computational Intelligence: DoSCI 2021*, pages 793–805. Springer, 2022.

[8] M. Del Tredici, D. Marcheggiani, S. S. i. Walde, and R. Fernández. You shall know a user by the company it keeps: Dynamic representations for social media users in nlp. *arXiv preprint arXiv:1909.00412*, 2019.

[9] M. Del Vicario, G. Vivaldo, A. Bessi, F. Zollo, A. Scala, G. Caldarelli, and W. Quattrociocchi. Echo chambers: Emotional contagion and group polarization on facebook. *Scientific reports*, 6(1):37825, 2016.

[10] D. DellaPosta, Y. Shi, and M. Macy. Why do liberals drink lattes? *American Journal of Sociology*, 120(5):1473–1511, 2015.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[13] I. R. Hallac, S. Makinist, B. Ay, and G. Aydin. user2vec: Social media user representation based on distributed document embeddings. In *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–5, 2019. doi: 10.1109/IDAP.2019.8875952.

[14] I. R. Hallac, B. Ay, and G. Aydin. User representation learning for social networks: An empirical study. *Applied Sciences*, 11(12):5489, 2021.

[15] X. He and T.-S. Chua. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 355–364, 2017.

[16] S. Hebbelstrup Rye Rasmussen and M. B. Petersen. The event-driven nature of online political hostility: How offline political events make online interactions more hostile. *PNAS nexus*, 2(11):pgad382, 2023.

[17] Y. Hou, S. Mu, W. X. Zhao, Y. Li, B. Ding, and J.-R. Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593, 2022.

[18] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338, 2013.

[19] S. Iyengar and M. Krupenkin. The strengthening of partisan affect. *Political Psychology*, 39:201–218, 2018.

[20] J. Jiang, E. Chen, S. Yan, K. Lerman, and E. Ferrara. Political polarization drives online conversations about covid-19 in the united states. *Human Behavior and Emerging Technologies*, 2(3):200–211, 2020.

[21] A. Krishnan, A. Sharma, and H. Sundaram. Insights from the long-tail: Learning latent representations of online user behavior in the presence of skew and sparsity. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 297–306, 2018.

[22] S. Li and H. Zhao. A survey on representation learning for user modeling. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4997–5003, 2021.

[23] L. Luceri, V. Pantè, K. Burghardt, and E. Ferrara. Unmasking the web of deceit: Uncovering coordinated activity to expose information operations on twitter. *arXiv preprint arXiv:2310.09884*, 2023.

[24] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[25] A. Mueen, N. Chavoshi, N. Abu-El-Rub, H. Hamooni, and A. Minnich. Awarp: Fast warping distance for sparse time series. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 350–359. IEEE, 2016.

[26] A. C. Nwala, A. Flammini, and F. Menczer. A language framework for modeling social media account behavior. *EPJ Data Science*, 12(1):33, 2023.

[27] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[28] D. Pacheco, P.-M. Hui, C. Torres-Lugo, B. T. Truong, A. Flammini, and F. Menczer. Uncovering coordinated networks on social media: methods and case studies. In *Proceedings of the international AAAI conference on web and social media*, volume 15, pages 455–466, 2021.

[29] S. Pan and T. Ding. Social media-based user embedding: A literature review. *arXiv preprint arXiv:1907.00725*, 2019.

[30] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[31] J. Qian, M. ElSherief, E. M. Belding, and W. Y. Wang. Leveraging intra-user and inter-user representation learning for automated hate speech detection. *arXiv preprint arXiv:1804.03124*, 2018.

[32] A. Rao, R.-C. Chang, Q. Zhong, K. Lerman, and M. Wojcieszak. Tracking a year of polarized twitter discourse on abortion. *arXiv preprint arXiv:2311.16831*, 2023.

[33] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[34] M. Ribeiro, P. Calais, Y. Santos, V. Almeida, and W. Meira Jr. Characterizing and detecting hateful users on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12, 2018.

[35] M. H. Saeed, S. Ali, J. Blackburn, E. De Cristofaro, S. Zannettou, and G. Stringhini. Trollmagnifier: Detecting state-sponsored troll accounts on reddit. In *2022 IEEE symposium on security and privacy (SP)*, pages 2161–2175. IEEE, 2022.

[36] R. Sawhney, H. Joshi, R. R. Shah, and L. Flek. Suicide ideation detection via social and temporal user representations using hyperbolic learning. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2176–2190, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.176.

[37] K. Shin, H. Kwak, S. Y. Kim, M. N. Ramström, J. Jeong, J.-W. Ha, and K.-M. Kim. Scaling law for recommendation models: Towards general-purpose user representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4596–4604, 2023.

[38] S. Tipirneni and C. K. Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–17, 2022.

[39] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[41] I. Waller and A. Anderson. Quantifying social organization and political polarization in online platforms. *Nature*, 600(7888):264–268, 2021.

[42] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1235–1244, 2015.

[43] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang. Community preserving network embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[44] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

[45] F. Yuan, G. Zhang, A. Karatzoglou, J. Jose, B. Kong, and Y. Li. One person, one model, one world: Learning continual user representation without forgetting. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 696–705, 2021.

[46] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 2114–2124, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467401.

[47] F. Zhuang, Z. Zhang, M. Qian, C. Shi, X. Xie, and Q. He. Representation learning via dual-autoencoder for recommendation. *Neural Networks*, 90:83–89, 2017.