Polarization identification on multiple timescale using representation learning on temporal graphs in Eulerian description

Victor Chomel*
CNRS - ISCPIF, EHESS, Paris, France
victor.chomel@iscpif.fr

Nathanaël Cuvelle-Magar*
CNRS - ISCPIF, Ecole polytechnique, Paris, France
nathanael.cuvelle-magar@polytechnique.edu

Maziyar Panahi
CNRS - ISCPIF, Paris, France
maziyar.panahi@iscpif.fr

David Chavalarias
CNRS - ISCPIF, EHESS, Paris, France
david.chavalarias@iscpif.fr

Abstract

Social media is often described as both reflecting and distorting real-life debates. Indeed, social division occurs not only offline but also online on various political topics. Several studies propose tools to identify and quantify online controversies through stance detection or polarization measures. While polarization is typically studied as a “snapshot" in time of a social network, we consider it as a temporal process. Moreover, the recent evolution in temporal graph representation learning provides new tools to directly combine time, content and graph topology. Current techniques that characterize polarization are beginning to use these tools, typically using a Lagrangian description which focuses on user trajectories. In this article, we make a case for approaching these problems with a Eulerian description, the concurrent description in fluid mechanics. In this description, the temporal evolution of nodes embeddings is represented with a deformation of velocity vector fields. Finally, we validate our method on a retweet graph from the last French presidential election campaign.

1 Introduction

The COVID-19 crisis shed light on scientific controversies in real-time. Among them were the effectiveness of masks and drugs such as hydroxychloroquine, or the risk-benefit balance of vaccination. Unsurprisingly, these controversies quickly became part of public discourse, sometimes leading to public health issues. As a result, the general public re-discovered the fact that research and politics or media discourse are on different time scales. Political and media expectations have contributed to a politicisation of some scientific controversies for which scientific consensus was being built at the time.The consequence is a polarisation of the subject along pre-existing political dividing lines. For example, in the United States, Republicans and Democrats appear to greatly differ in their approaches towards vaccination against COVID-19 (Fridman et al. [2021]). These dividing lines also appear online where evolution may be measured. There exists a direct link between opinion dynamics and the study of social dividing lines, as a shift of these lines indicates that users switch sides.
Polarization is usually evaluated on static networks, once communities are already divided. However, this approach does not characterize the process under which this polarization arises. Instead, using temporal graphs, we treat polarization as an evolution of social community boundaries inside the network, to detect it before being too late. This paper develops an approach based on representation

*These authors contributed equally to this work.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

learning of temporal graphs. Like recent works, the goal is to embed graph evolution inside a latent space to study community evolution (Li et al. [2022]). Our work is based on an implementation of Temporal Graph Networks (Rossi et al. [2020]) to embed retweet graphs. However, instead of using a Lagrangian description, i.e. following trajectories of users, polarization is then seen through a Eulerian description from fluid mechanics and represented with "velocity" vector fields, focusing on specific locations in the latent space.

2 Related works

2.1 Measure polarization

Many measures of polarization have been proposed, most of them coming from models of opinion dynamics (Gurukar et al. [2020], Baumann et al. [2020]) but not always suited for graph topology. A large literature studies stance detection to approach polarization. It can be based on content (tweets or hashtags), interactions (follows, retweets, etc) or users directly (name, location, bio, etc). For Twitter, interaction-based solutions generally provide better results (Magdy et al. [2016]). If supervised approaches exist (Borge-Holthoefer et al. [2014], Kutlu et al. [2018]), more recent solutions are unsupervised using, for instance, force-directed layouts (Darwish et al. [2019]). Lai et al. [2019] work on dynamic stance detection on various interaction graphs, exhibiting the importance of retweets.
Polarization can also be measured directly on the graph, using boundaries between communities for instance (Guerra et al. [2013]). Another solution from Morales et al. [2015] defines a metric based on the relative sizes of detected communities and distances between their centroids. Solutions based on embeddings are now vastly used (Rashed et al. [2020]) but they are mostly limited to static graphs. Finally, Raghavendra et al. [2022] models the evolution of signed networks with a dynamic Graph Neural Network (GNN), these networks being well suited for societal applications such as polarization measurement (Bonchi et al. [2019]).
So far, solutions based on polarization evolution (Kearney [2019]) (opposed here to polarization as a static situation) do not fully benefit from more recent development in representation learning on dynamic graphs. This paper aims to reconcile the study of polarization evolution with recent representation learning techniques by creating dynamic embeddings and computing their deformation.

2.2 Eulerian approach

Each new interaction changes nodes embeddings. By gathering these changes, we can model embeddings deformations. Velocity vector fields are used to represent them. Instead of following users or communities along the time and trying to predict their trajectories, we focus on areas of the latent space and computing the direction of users passing through areas without following them. This is inspired from the dichotomy between Lagrangian and Eulerian descriptions in fluid mechanics.
Eulerian descriptions of complex systems have been used in the study of road traffic or crowd flow (van Wageningen-Kessels et al. [2016], Ma et al. [2016]). To the best of our knowledge, the most recent Eulerian approaches for social network are from Mirtabatabaei et al. [2014], Canuto et al. [2012]. This approach attempts to overcome Lagrangian limits of opinion dynamics models such as the Hegselmann-Kraus model. Due to the Eulerian description, they manage to prove properties of the models, reusing ideas from fluid mechanics with operators such as divergence for instance. This changing paradigm focuses on flows rather than objects, exhibiting almost instantly fault lines compare to Lagrangian solutions .

2.3 Representation learning on dynamic graphs

As presented by Kazemi et al. [2020] in their survey on representation learning for dynamic graphs, existing models can be seen as an encoder/decoder scheme. The encoder associates nodes to their embeddings, from which the decoder performs a specific task, such as link prediction or node classification. As stated in Rossi et al. [2020], deep learning models on dynamic graphs can be classified according to the temporal dimension representation, discrete or continuous. The discrete time setting poses several difficulties: limited temporal granularity, lost order of appearance of edges, and impossible identification of links that frequently appear together (Skarding et al. [2021]).

Different approaches have thus been proposed to model dynamic graphs in a continuous setting. Nguyen et al. [2018] and Bastas et al. [2019] build network representation learning methods using temporal random walks to generate structure and continuous time-aware embeddings. ATiSE (Xu et al. [2019]) leverages time series method for temporal knowledge graph embedding model. Specific dynamic graph neural networks architectures have also been developed, such as Dyrep (Trivedi et al. [2019]), which produces dynamic low-dimensional node embeddings via a learned set of functions, Jodie (Kumar et al. [2019]), which uses coupled RNNs to learn node embedding trajectories, or Temporal Graph Attention (TGAT) (Xu et al. [2020]), which introduces a new temporal graph attention layer allowing it to efficiently aggregate temporal and topological features. These three architectures can be seen as special cases of general framework, Temporal Graph Networks (Rossi et al. [2020]), that adds a memory module to TGAT. TGN's framework being the most generic, we choose to use it as a reference model for our study.

3 Method Description

As detailed in 2.3, the tool we have selected to model the evolution of online behaviors is the TGN framework [*], the goal being to analyze the observable dynamics in its latent space. We therefore start by presenting the general workflow of the algorithm.

The approach proposed by TGN to record the history of a node, i.e. the set of events in which it took part (interactions, change of features, etc.), relies on the use of a memory module. The latter, based on a RNN, allows to update a memory vector encoding the past behaviors of the nodes. It is coupled to an embedding module allowing to aggregate the neighborhood information and which can be realized by different functions: identity, a sum, an attention mechanism (Veličković et al. [2018]), etc. These two modules, memory and embedding, constitute an encoder associating to a dynamic graph a vector representation of its nodes. This representation can then be used by a decoder to perform a specific task. In this study, we will use a decoder associated to link prediction, the training being then self-supervised. Indeed, the model learns, for each batch, to predict the probability of next interactions. A more detailed description of the algorithm is provided in appendix A.1.
By removing the final MLP, TGN can be used to create node embeddings dynamically. Depending on the wanted time scale, a time step is defined (from minutes to a week). Embeddings can then be compared between two instants to understand the geometric evolution of the landscape.

As stated before, a Lagrangian description with direct reading of trajectories is not the only way to describe polarization. In our Eulerian perspective, we compute the evolution of nodes in areas at the wanted granularity. To do so, the solution adopted consists of defining a square mesh on the latent space [*]. For each square, the average move between two time steps is evaluated. To this end, the difference between positions of two time steps is computed and averaged over users located in the square at the beginning. This calculation is only performed for squares whose numbers of users exceed a threshold in order to ensure a certain significance. The move per square is represented with a vector, leading to a vector field (Figure 1), where angles are associated with varying colors.

4 Experiments and results

4.1 Preliminary analysis

Our usecase is based on a French political dataset from Twitter. Experiment's settings and data collection are detailed in Appendix A.2 and A.3.

The first step is to provide meaningful interpretations for the embedding reduced dimensions. Interestingly, the embeddings obtained provide similar results to other representations (Ramaciotti Morales et al. [2021], Chomel et al. [2022]) such as force-directed algorithms leading to interpretable axis of the PCA. For instance, on Figure 2, communities are computed using Louvain algorithm on the static graph to exhibit the political colors of users. Extreme right parties appear on the left side (dark blue) and left parties appear in the bottom (red and pink). Lastly, the government party is on the top right (yellow). It seems that specific regions of the latent space can be linked to political communities and and we will use this information to link deformations to political events.

[*] `https://github.com/twitter-research/tgn`, licensed under Apache-2.0 license.

[*] Instead of square, we should speak of a hypercube here. In practice, the solution developed is used after a dimension reduction to be visualized in two dimensions, hence the use of square.

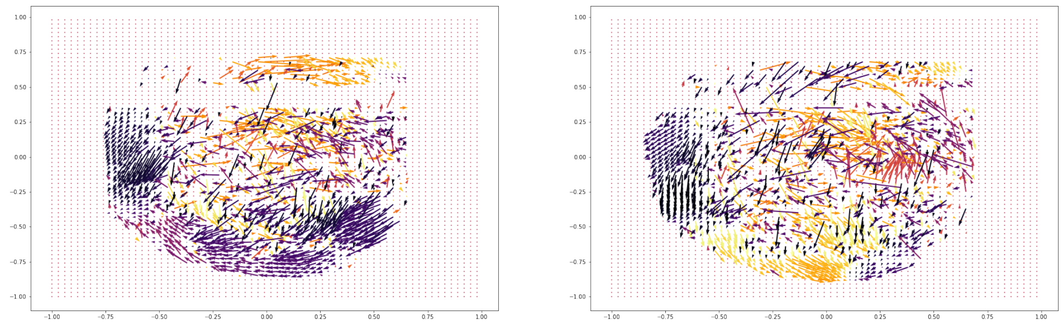

Figure 1: Example of two successive deformations, with a one hour time step, on April 5th 2022. The threshold is 10 nodes per tile. The axis are the same as the PCA used in Figure 2.

4.2 Deformation examples

The first striking result is the apparent disorganization in regions with less nodes such as the center. In these regions, without strongly defined communities, there is no reason for close users to move in identical directions. We will thus focus the analysis on dense regions of the Figure 2, as they provide consistent and explainable evolutions. A first example is studied with a time step of about an hour. It exhibits a pendulum swing that commonly occurs on the platform with an event bringing two groups closer before they take their distances again (Figure 1).

On the left panel, nodes at the bottom are going to the left. Politically speaking, it means that users from the extreme left users are getting closer during this hour to alt right ones. This matches perfectly the network activity as several major alt right accounts including the head of the movement, Florian Philippot, has been largely retweeted. In the meantime, extreme left figures have been less active, leading to the large departure from this area. During the same time interval, the extreme right around Zemmour promoted his role in the help of a victim of aggression, gathering users of the community.

During the next time step, leaders of the extreme left, such as Jean-Luc Mélenchon, were massively retweeted when they criticized the government. At the same time, the alt right was less active, pushing users from the bottom to the right of the plot. Another example, at a different time scale, can be found in Appendix A.4. Longer time scale solutions can be used to identify polarization directly.

5 Discussion

The method proposed in this study, based on the visual analysis of the deformations of the latent space, can be used to identify social phenomena such as polarization*. Nevertheless, two questions appear naturally from the identification of such phenomena.

Actors/Users Which actors/users are involved and can explain these evolutions? An in-depth analysis of the role of leaders has been provided for the use case, but can it be automated?
The aforementioned deformations correspond to a synthetic representation of the dynamics of nodes embeddings, allowing an efficient visualisation and detection of social phenomena. Once detected, it would thus be interesting to perform a dynamic clustering on nodes embeddings instead of deformations, focused on the identified period and location.

Content/Topics Is it possible to find which topics are covered by these deformation?
The focus is now on the semantic aspect of opinion dynamics, requiring to interpret the space of nodes representations in a meaningful way. Analogously to the opinion space used in Mirtabatabaei et al. [2014], the idea is thus to give an intrinsic meaning and not simply relative to the positions of nodes in this space. An approach would be to decode nodes embeddings via the decoder associated with the text encoder used to generate the node and edge features, with the hope that the information contained in these text embeddings has been preserved by TGN.

*These tools are meant to be used to measure and identify it in short term evolutions. Because of the complexity of social networks and their unpredictability on the long run, it cannot really be used to predict the impact of a few tweets directly, making it of no use to manipulate opinion.

A Appendix

A.1 Details on TGN

A.1.1 Memory

In practice, the memory of a node is the hidden vector, initially zero, of a RNN, which is updated after each event involving this node. More precisely, when a node interacts, a message storing information about this interaction is computed and used as the input vector of the RNN, with the previous memory of the node serving as the input hidden vector. The output hidden vector then becomes the new memory of the node. In fact, as interaction data are processed in batches, it is necessary to define an aggregation rule for the messages produced within the same batch. Here, only the last computed message is retained.

More formally, for an interaction involving nodes i and j, the messages m_i and m_j will be the following concatenations:

$$\begin{cases} m_i(t) = s_i(t^-)||s_j(t^-)||e_{ij}(t)||\phi(\Delta t_i) \\ m_j(t) = s_j(t^-)||s_i(t^-)||e_{ij}(t)||\phi(\Delta t_j) \end{cases}$$

where $s_k(t^-)$ refers to the memory of node $k \in \{i, j\}$ before its update and Δt_k the time difference between the last update time of node $k \in \{i, j\}$ and that of the interaction at the origin of the message.

A.1.2 Embedding module

The general form adopted for this module in Rossi et al. [2020] is:

$$z_i(t) = \sum_{j \in \mathcal{N}_i^l([0,t])} h(s_i(t), s_j(t), e_{ij}, v_i(t), v_j(t)),$$

where $\mathcal{N}_i^l([0,t])$ denotes the time neighborhood at rank l of node i on period $[0,t]$, $v_i(t)$ and $v_j(t)$ denote nodes i and j features, and h a learnable function. In the TGN configuration adopted for this study, the representation of a node $i \in [\![1, n(t)]\!]$ is obtained by applying the temporal graph attention layer developed in Xu et al. [2020].

A.1.3 Final MLP and Loss Function

The decoder is composed of an MLP followed by a sigmoid function, allowing the prediction of link probability between nodes $i, j \in [\![1, n(t)]\!]^2$ at time $t \geq 0$ via the following formula:

$$\hat{\mathbb{P}}_{ij}(t) = \sigma(MLP(z_i(t)||z_j(t))).$$

In training, the estimated probabilities of the edges of the batch being processed and of those obtained by negative sampling (noted $(p_i^+)_{i \in [\![1,M]\!]}$ and $(p_i^-)_{i \in [\![1,M]\!]}$, where M is the batch size) are injected into a Binary Cross Entropy loss function:

$$\ell = -\frac{1}{M} \sum_{i=0}^{M-1} [\log(p_i^+) + \log(1 - p_i^-)].$$

A.2 TGN settings and computing resources

The experiments presented in this section are performed with the TGN-attn configuration of Rossi et al. [2020], i.e. nodes memory is used, the memory updater is a GRU, the embedding module is an attention layer considering the 10 last neighbors, message aggregation is done by keeping only the most recent one and identity is used as message function. The hyperparameters are the same as those used in the original paper. The code allowing the reproduction of these experiments* is available online *. The configuration providing the better results has been selected.

*Experiments were conducted on a laptop (Intel i5-8265U (1.60 GHz) CPU and 8 GB of RAM) and on a computer server (Nvidia Quadro P5000 GPU, 40*Xeon E5-2630 (2.20 GHz) CPUs and 128 GB of RAM).

*`https://github.com/TGL-sub/Polarization-identification-on-temporal-graphs-in-Eulerian-description`

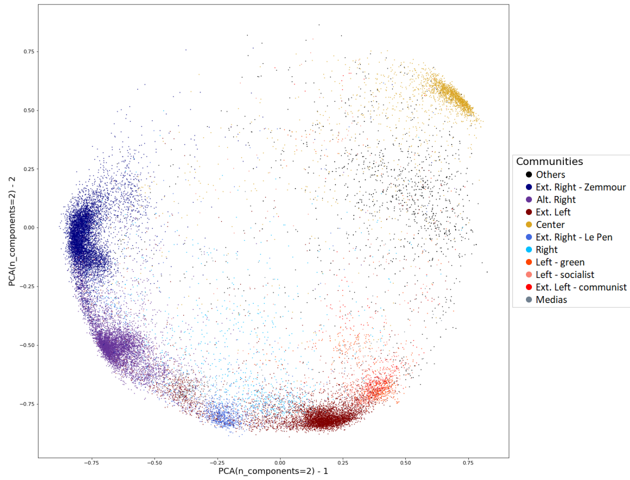

Figure 2: Embedding of the retweet graph with the French political communities (Left in red, Center in yellow and Right in blue). A PCA is used to reduce the dimensions.

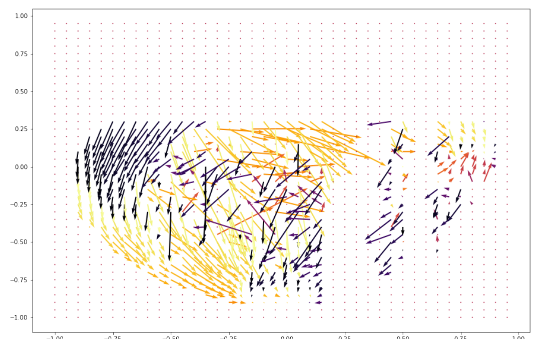

Figure 3: Embeddings deformation from April 7th to April 8th 2022. The threshold is 20 nodes per tile, explaining the large empty areas. The axis are the same as the PCA used in Figure 2.

Before considering embeddings deformation, one must ensure that these embeddings almost converged to their *true* position. As expected, during the first time steps, embeddings quickly evolve before stabilizing (or more exactly moving more slowly) and exhibiting interesting deformations.

A.3 Datasets

This dataset is collected through the Twitter API, by following 3000 personalities from the French political world. More precisely, all their immediate neighbors are considered, as well as the interactions that link them (*mentions*, *retweet*, *quote*, etc.). In addition, tweets and authors mentioning several *hashtags* related to the 2022 presidential elections are also collected. A more detailed description of this data collection is available as it has been used to work on the last two French presidential elections (Gaumont et al. [2018]). Any other graph with features could also be used.

From this dataset, a graph is built. It is obtained by selecting a period and by choosing the considered interaction (defining what an edge corresponds to). In our case, a dynamic graph is thus obtained, noted *Politics*, spanning 431999 seconds, including 30682 nodes and 1088815 interactions. Edges features are BERT embeddings of retweeted texts (dimension 300). Node features are zero vectors of dimension 172.

A.4 Second example

A second example has been run with a one day time step. Instead of observing the impact on the network of a few tweets, this larger time step allows us to identify larger political movements. A consistent result across the last week before the first round is the exodus from one extreme right community (Zemmour) to the other one (Le Pen). It can be identified with the flow from the left to the bottom of the space. This result is consistent with political analysis from media, polls and eventually outcomes from the French presidential election of 2022.

References

Nikolaos Bastas, Theodoros Semertzidis, Apostolos Axenopoulos, and Petros Daras. evolve2vec: Learning network representations using temporal unfolding. In *MMM*, 2019.

Fabian Baumann, Philipp Lorenz-Spreen, Igor M. Sokolov, and Michele Starnini. Modeling echo chambers and polarization dynamics in social networks. *Phys. Rev. Lett.*, 124(4):048301, January 2020. doi: 10.1103/PhysRevLett.124.048301. URL `https://link.aps.org/doi/10.1103/PhysRevLett.124.048301`.

Francesco Bonchi, Edoardo Galimberti, A. Gionis, Bruno Ordozgoiti, and Giancarlo Ruffo. Discovering polarized communities in signed networks. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

Javier Borge-Holthoefer, Walid Magdy, Kareem Darwish, and Ingmar Weber. Content and network dynamics behind egyptian political polarization on twitter, 2014. URL `https://arxiv.org/abs/1410.3097`.

C. Canuto, F. Fagnani, and P. Tilli. An eulerian approach to the analysis of krause's consensus models. *SIAM Journal on Control and Optimization*, 50(1):243–265, 2012. doi: 10.1137/100793177. URL `https://doi.org/10.1137/100793177`.

Victor Chomel, Maziyar Panahi, and David Chavalarias. Manipulation during the french presidential campaign: coordinated inauthentic behaviors and astroturfing analysis on text and images. Proceedings of Complex Networks and their Applications, 2022.

Kareem Darwish, Peter Stefanov, Michaël Aupetit, and Preslav Nakov. Unsupervised user stance detection on twitter, 2019. URL `https://arxiv.org/abs/1904.02000`.

Ariel Fridman, Rachel Gershon, and Ayelet Gneezy. Covid-19 and vaccine hesitancy: A longitudinal study. *PloS one*, 16(4):e0250123, 2021.

Noe Gaumont, Maziyar Panahi, and David Chavalarias. Methods for the reconstruction of the socio-semantic dynamics of political activist Twitter networks. *PLoS ONE*, 13(9), September 2018. doi: 10.1371/journal.pone.0201879. URL `https://hal.archives-ouvertes.fr/hal-01575456`. La version française est disponible en V1 de cette archive.

Pedro Henrique Calais Guerra, Wagner Meira Jr, Claire Cardie, and Robert D. Kleinberg. A measure of polarization on social media networks based on community boundaries. In *ICWSM*, 2013.

Saket Gurukar, Deepak Ajwani, Sourav Dutta, Juho Lauri, Srinivasan Parthasarathy, and Alessandra Sala. Towards quantifying the distance between opinions, 2020. URL `https://arxiv.org/abs/2001.09879`.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, Pascal Poupart, and Karsten M. Borgwardt. Representation learning for dynamic graphs: A survey. *J. Mach. Learn. Res.*, 21:70:1–70:73, 2020.

Michael Wayne Kearney. Analyzing change in network polarization. *New Media & Society*, 21(6): 1380–1402, 2019. doi: 10.1177/1461444818822813. URL `https://doi.org/10.1177/1461444818822813`.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2019.

Mucahid Kutlu, Kareem Darwish, and Tamer Elsayed. Devam vs. tamam: 2018 turkish elections, 2018. URL `https://arxiv.org/abs/1807.06655`.

Mirko Lai, Marcella Tambuscio, Viviana Patti, Giancarlo Ruffo, and Paolo Rosso. Stance polarity in political debates: A diachronic perspective of network homophily and conversations on twitter. *Data & Knowledge Engineering*, 124:101738, 2019. ISSN 0169-023X. doi: https://doi.org/10.1016/j.datak.2019.101738. URL https://www.sciencedirect.com/science/article/pii/S0169023X19300187.

Jinning Li, Huajie Shao, Dachun Sun, Ruijie Wang, Yuchen Yan, Jinyang Li, Shengzhong Liu, Hanghang Tong, and Tarek Abdelzaher. Unsupervised belief representation learning with information-theoretic variational graph auto-encoders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, jul 2022. doi: 10.1145/3477495.3532072. URL https://doi.org/10.1145%2F3477495.3532072.

Yuxin Ma, Tao Lin, Zhendong Cao, Chen Li, Fei Wang, and Wei Chen. Mobility viewer: An eulerian approach for studying urban crowd flow. *IEEE Transactions on Intelligent Transportation Systems*, 17(9):2627–2636, 2016. doi: 10.1109/TITS.2015.2498187.

Walid Magdy, Kareem Darwish, Norah Abokhodair, Afshin Rahimi, and Timothy Baldwin. Isisisnotislam or deportallmuslims? predicting unspoken views. In *Proceedings of the 8th ACM Conference on Web Science*, WebSci '16, page 95–106, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342087. doi: 10.1145/2908131.2908150. URL https://doi.org/10.1145/2908131.2908150.

Anahita Mirtabatabaei, Peng Jia, and Francesco Bullo. Eulerian opinion dynamics with bounded confidence and exogenous inputs. *SIAM Journal on Applied Dynamical Systems*, 13(1):425–446, 2014. doi: 10.1137/130934040. URL https://doi.org/10.1137/130934040.

A. J. Morales, J. Borondo, J. C. Losada, and R. M. Benito. Measuring political polarization: Twitter shows the two sides of venezuela. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(3):033114, March 2015. doi: 10.1063/1.4913758. URL https://doi.org/10.1063/1.4913758.

Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen Ahmed, Eunyee Koh, and Sungchul Kim. Dynamic network embeddings: From random walks to temporal random walks. *2018 IEEE International Conference on Big Data (Big Data)*, pages 1085–1092, 2018.

Mr. Raghavendra, Kartik Sharma, Mariappan Anandkumar, and Srijan Kumar. Signed link representation in continuous-time dynamic signed networks. *ArXiv*, abs/2207.03408, 2022.

Pedro Ramaciotti Morales, Jean-Philippe Cointet, and Gabriel Muñoz Zolotoochin. Unfolding the dimensionality structure of social networks in ideological embeddings. In *2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Amsterdam, Netherlands, November 2021. URL https://hal.archives-ouvertes.fr/hal-03315759.

Ammar Rashed, Mucahid Kutlu, Kareem Darwish, Tamer Elsayed, and Cansın Bayrak. Embeddings-based clustering for target specific stances: The case of a polarized turkey. 2020. doi: 10.48550/ARXIV.2005.09649. URL https://arxiv.org/abs/2005.09649.

Emanuele Rossi, Benjamin Paul Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. *ArXiv*, abs/2006.10637, 2020.

Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access*, 9:79143–79168, 2021.

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

Femke van Wageningen-Kessels, Ludovic Leclercq, Winnie Daamen, and Serge P. Hoogendoorn. The Lagrangian coordinate system and what it means for two-dimensional crowd flow models. *Physica A: Statistical Mechanics and its Applications*, 443:272–285, 2016. doi: 10.1016/j.physa.2015.09.048. URL https://hal.archives-ouvertes.fr/hal-01215762.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJXMpikCZ`.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. Temporal knowledge graph embedding model based on additive time series decomposition, 2019. URL `https://arxiv.org/abs/1911.07893`.

Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *ArXiv*, abs/2002.07962, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 4 for a use case based on the announced approach.
   (b) Did you describe the limitations of your work? [Yes] See Section 5 for a discussion of improvements that could be made.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See footnote 5 in Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See footnote 4 in Section A.2.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sections A.3 for datasets description and A.2 for TGN settings and hyperparameters.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See footnote 3.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] For TGN, see the end of Section 2.3.
   (b) Did you mention the license of the assets? [Yes] See footnote 1 in Section 3.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The data collection was described in a reference and collected using the Twitter API.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The collected data are tweets and retweets from French politics. User ids are kept separately from users activities. The user decoding is only used on nodes with highest degrees as they are public personalities. It is used to make the political affiliation of communities.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]