Ranking Online Social Users by their Influence

Anastasios Giovanidis, Bruno Baynat, Clémence Magnien, Antoine Vendeville

Ranking Online Social Users by their Influence

Anastasios Giovanidis, Bruno Baynat, Clémence Magnien, and Antoine Vendeville

***Abstract*—We introduce an original mathematical model to analyse the diffusion of posts within a generic online social platform. The main novelty is that each user is not simply considered as a node on the social graph, but is further equipped with his own Wall and Newsfeed, and has his own self-posting and re-posting activity. As a main result using our developed model, we derive in closed form the probabilities that posts originating from a given user are found on the Wall and Newsfeed of any other. These are the solution of a linear system of equations. Comparisons with simulations show the accuracy of our model and its robustness with respect to the modelling assumptions. Using the probabilities from the solution we define a new measure of per-user influence over the entire network, the Ψ-score, which combines the user position on the graph with the user (re-)posting activity. Furthermore, we compare the new model and its Ψ-score against the empirical influence measured from very large data traces (Twitter, Weibo). The results illustrate that these new tools can accurately rank influencers for such real world applications.**

***Index Terms*—online social network, ranking, influence, model, Markov chain, graph, Twitter, Weibo.**

I. Introduction

Online Social Platforms (OSPs) play a major role in the way individuals communicate with each other, share news and get informed. Today such platforms host billions of user profiles. Although OSPs differ from one another, most of them share a common structure, which allows users to post messages on their Wall and read posts of others on a separate Newsfeed. Most OSPs also permit re-posting from Newsfeed to Wall, in order to facilitate information diffusion. With each re-post (or "share", or "re-tweet") the information becomes visible to a new audience, which may choose to adopt it or not, thus spreading further the post or halting its diffusion. In this way, posts originally generated by some user circulate inside the social network [2]. When the post is gradually adopted by a considerable proportion of the users, we see large cascades of information appear, and we call such posts "viral" [3].

Understanding how information spreads through OSPs is very important as it affects the opinion of the population over several subjects of every-day social life. Companies want to determine the set of most influential users ("influencers") for better marketing of their products [4], and they would like to predict information cascades [5]. Such research is critical also because spreading of influence can have malevolent purposes instead [6], such as the spread of misinformation ("fake news"). To be able to develop defence mechanisms against such social attacks, a concrete mathematical analysis of post diffusion through OSPs is necessary.

The preliminary version of this article appeared in the Proc. of INFOCOM 2019 [1]. This material is based in part upon work supported by the Agence Nationale de la Recherche (ANR) under grant ANR-19-CE25-0011-01, project "FairEngine". The authors are with the LIP6 Laboratory, Sorbonne University and French National Center for Scientific Research (UMR 7606 Sorbonne University - CNRS), Paris, France, Email: {firstname.lastname}@lip6.fr

Existing literature on the topic has mainly focused on models for opinion dynamics given the social graph as input. These include the *voter* model [7], the SI(R) [11, Ch. 17], the *threshold and cascade* models [4] and the *DeGroot* [8] model, among others. In each of these, the social graph structure, together with simplified user interaction, has been assumed sufficient to describe the diffusion *of a single opinion*. However, the authors of the highly cited paper about the "million follower fallacy" [9], argue that graph topological measures alone reveal very little about the true influence of a user in a platform; they use large traces from Twitter to support their claim. The authors in [10] further use Facebook data to identify real indicators of user interactions, beyond social links. Our paper has the ambition to fill this gap between analysis and data-driven conclusions by introducing a new dynamic model which combines the information over the social graph together with user activity and the OSP structure, in order to explain more accurately how posts from different origins diffuse and compete among each other inside the social platform.

Viral marketing wants to identify users with high social influence [4]. To this aim, users are ranked based on certain impact measures, which mainly depend on user graph position (e.g. number of follower links), similar to the existing opinion models discussed above. For example, [11, Ch. 7] presents degree, eigenvector and Katz centrality, as well as PageRank score [12], and [13] alternatives for large-scale graphs. We claim that such measures are not suitable to rank the influence of social users, because they do not include user activity, or OSP structure and they will hence mislead when used to identify "influencers". We propose instead a new Ψ-score to rank users, based on our proposed model. Our main contributions are summarised as follows.

A. Main Contributions and Paper Structure

Our main contributions are summarised as follows:

- The entire OSP is described as a continuous-time Markov chain. This model is original in the sense that it combines (i) the social graph, with (ii) dynamic user posting and re-posting activity, and incorporates elements of (iii) the platform structure (Walls, Newsfeeds and the Newsfeed suggestion algorithm). The model can include various Newsfeed mechanisms (FIFO, Random, TTL) as well as user post sharing behaviour. Also, competition among posts to gain the user's attention on the Newsfeed is naturally included.
- By analysing the above chain we result in a linear system of equations, which exactly describes the chain's behaviour in steady-state (Theorem 1, Theorem 2). This has as unknown variables the influence of a given user

on the Wall and Newsfeed of any other. This system actually consists of the balance equations of posting activity on each Wall and Newsfeed and can be solved for an arbitrary input graph and arbitrary user activity rates. Theorem 3 provides its solution.
- An iterative method (Theorem 4) proposed to compute the system solution facilitates numerical implementation. It allows to implement a sparse algorithm, which scales well as the size of the social graph increases.
- The solution gives rise to a new way to rank OSP users by their influence. We call the new ranking metric, the "Ψ-score". The performance of our model and the Ψ-score is tested on two large real-world traces from famous social platforms, one from Twitter and another from Weibo.

The implementation code is made available online [14].

The paper is organised as follows. The social platform under study and the performance metrics of interest are introduced in Section II. The Markovian model describing the generic OSP and its balance equations are given in Section III. Here, we explain how these linear equations naturally constitute the Newsfeed and Wall balance equations, and show their exactness and general validity. The system's closed-form solution is provided in Section IV. In the same section we provide an iterative method that is computationally cheap and converges to this solution. The ranking algorithm based on the Ψ-score and its implementation for large data-traces is detailed in Section V-A. Extended numerical experiments using synthetic data in Section VI verify the model's validity and robustness over modelling assumptions. Massive real-world traces from Twitter and Weibo are used to evaluate the Ψ-ranking and the sparse algorithm, in Section VII. Our Ψ-ranking is further compared against standard user ranking metrics (out-degree, PageRank, and user posting rate). Conclusions are drawn in Section VIII.

B. Related Literature

In most relevant research on opinion dynamics, individuals are seen as agents whose relation is described by a social graph. Each agent has a certain opinion and at each step this opinion is updated through interaction with his direct peers. Such models can be grouped into two general categories. *1) Dynamics with Binary opinions:* There are only two possible opinions that agents can take. A large amount of work descends from the *voter model* [7], where opinion dynamics are based on imitation. The work in [15] studies a variation that includes agents with persisting opinions. For further extensions, see also [16]. Another group of work is related to epidemic spread. An agent is "susceptible" when his opinion is 0 and becomes "infected" when he adopts opinion 1, through social interaction [11]. In [4] two opinion update mechanisms are studied: the threshold and the cascade. *2) Dynamics with Continuous opinions:* Several works in the literature have inherited and extended the original model of DeGroot [8]. In this, each agent updates his continuous opinion by forming per-step a weighted linear combination of the current opinions of his peers. Variations of this model consider the inclusion of persistent agents [17]. In [18] this update mechanism is used to formulate and solve an opinion manipulation problem. To account for more realistic social behaviour, the authors in [19] consider opinion dynamics where agents interact in pairs only when their opinions are already close.

Data, OSPs, and Cascades: Instead of modelling opinion dynamics, recent works rather use available data to investigate more practically how posts spread within OSPs. The authors in [2] describe diffusion patterns that arise in specific online domains. Data analysis of large Facebook cascades is performed in [3]. Interestingly, the authors in [5] propose ways to predict cascade growth using machine learning tools. The insufficiency of using graph-based only information to evaluate user influence is studied in [9] and [10].

User activity: In [20] the authors identify user activity as an important control tool for influence maximisation. Making extensive use of datasets, they study the appropriate times for a user to post or re-post in an OSP in order to maximise the probability of audience response. An interesting analytical effort to relate user activity with OSP design and post diffusion is made in [21]. The authors use temporal point processes to model posting and re-posting activity of a user. They highlight the importance of the Newsfeed in post propagation and map user activity to post visibility, building on the idea that a post can be adopted by a follower when it is visible on his Newsfeed and not pushed away by competing posts. Their model, however, treats only a single user Newsfeed and does not consider the dynamics of the entire social graph. Furthermore, the dynamics of the Newsfeed list are inaccurately mimicked by a FIFO queue. Another relevant line of research includes [22] and [23], where the authors study the bias of Facebook's News Feed algorithm. They consider a bipartite graph of a set of users following a set of publishers, and model post activity as Poisson. Newsfeeds are here again approximated by infinite queues with TTL or FIFO service.

Compared to these works, we propose here a more correct and complete OSP model; we accurately model Newsfeeds as lists, we consider here an arbitrary graph of any size and include re-posting – among other realistic features. Finally, we verify our model's validity by large real-world data traces.

II. System description

Let us first describe a generic social network platform, such as Facebook, Twitter or Weibo. A set of users generate and share some content, denoted as *posts*, through the platform. Each user has a list of *followers* and a list of *leaders*. A user can simultaneously be follower and/or leader of others. As a follower, he (she) is interested in the content posted by his (her) leaders. With each user two lists of posts are associated, namely a *Newsfeed* and a *Wall*. A user's Newsfeed is constantly fed by the content that all of his leaders post on their Walls. A user's Wall is fed (i) by his self-generated posts that draw influence from the "outside world", and (ii) by posts that he shares from his Newsfeed. Hence, a user's Wall is a list of self-posts and re-posts. The generic social network platform is illustrated in Figure 1.

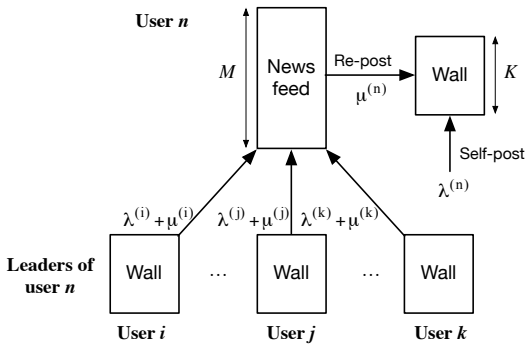


Fig. 1. The social platform from the point of view of user n.

A. Assumptions on the system and notations

We consider a constant number N of active users, forming the set $\mathcal{N}$. Users are labelled by an index $n = 1, \ldots, N$. We denote by $\mathcal{F}^{(n)}$ and $\mathcal{L}^{(n)}$ the list of followers and the list of leaders of user n. Without loss of generality, we draw the directed Leader-graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. Each pair of nodes $(i, j) \in \mathcal{E}$, corresponds to a directed edge from i to j, when i is a leader of j, i.e., $i \in \mathcal{L}^{(j)}$. We denote by $\mathbf{L}$ the $N \times N$ adjacency matrix of the Leader-graph, whose coefficients are given by: $\ell_{i,j} = \mathbf{1}_{\{i \in \mathcal{L}^{(j)}\}}$, where $\mathbf{1}_{\{.\}}$ is the indicator function. We assume that each user n has at least one leader, i.e., $\mathcal{L}^{(n)} \neq \emptyset$, $\forall n$. The Follower-matrix is by definition $\mathbf{F} := \mathbf{L}^T$.

The sizes of both Wall and Newsfeed are considered to be constant. We thus fix $K \geq 1$ the size of a Wall (total number of posts on the Wall of each user) and $M \geq 1$ the size of a Newsfeed. This is reasonable if we assume that only a certain number of most recent posts is considered relevant, and users don't tend to scroll down to access older posting history.

We denote by $\lambda^{(n)}$ [posts/unit time] the rate with which user n generates new posts on his Wall, and by $\mu^{(n)}$ the rate with which user n visits his Newsfeed and selects one of the M entries to re-post on his Wall (note here that each visit implies re-posting). As a result, posts arrive on the n-th Wall with a total rate $\lambda^{(n)} + \mu^{(n)}$ [posts/unit time]. Additionally, we make the assumption that content posted on a users's Wall instantaneously appears on the Newsfeeds of his followers. As a result, the input rate of posts in the n-th Newsfeed, is $\sum_{j \in \mathcal{L}^{(n)}} (\lambda^{(j)} + \mu^{(j)})$. Given that the two lists associated per user have fixed size, then with each new entry one element has to be removed from the list and replaced by the new one. For the user activity we require $\lambda^{(n)} + \mu^{(n)} > 0$, $\forall n$.

Finally, any post originally generated by a given user n takes as label the author's index n, and will keep this label throughout its lifespan inside the network.

B. Influence metric of interest

The aim is to estimate the influence of a specific user, say user i, over the entire network. In order to define the metric of interest, we first define the *influence* of user i on user n, denoted by $q_i^{(n)}$, as the expected percentage of posts of origin i found on the Wall of user n. They obviously satisfy for each Wall n, $\sum_{i=1}^{N} q_i^{(n)} = 1$, $\forall n$. We can also interpret $q_i^{(n)}$ as the probability that, when picking at random a post from Wall n, this post is of origin i. These performance quantities will be the output of the developed models. With the above, we propose the following metric of influence,

$$\Psi_i \quad = \quad \frac{1}{N-1} \sum_{n \neq i} q_i^{(n)} \quad \in [0, 1] \,. \tag{1}$$

It corresponds to the average percentage of posts of origin i on the Walls of any user $n \neq i$. The suggested metric averages over all users in the network, but excludes the original user i.

Other metric definitions are also possible. As an example, we could use the probability to find at least one post of label i on the Wall of user n. This is equal to $1 - (1 - q_i^{(n)})^K$, based on which we can define an alternative metric.

In any case, by associating an influence score to each user, the social users can be ranked by decreasing order of their influence. From now on, we will call the expression in (1) which quantifies the influence of a user in the platform, the *Ψ-score of user i*. In this work we will focus only on this metric, leaving other alternatives for future investigations.

III. Model

A. Markovian model

The model relies on the following assumptions:

- **Poisson arrivals**. For any user n the generation of new posts on his Wall follows a Poisson process with rate $\lambda^{(n)}$ and the re-posting activity from his Newsfeed follows a Poisson process with rate $\mu^{(n)}$.
- **Random selection**. When a user visits his own Newsfeed, we assume that he selects at random one of the M entries to re-post on his Wall.
- **Random eviction**. A novel entry on the Wall or Newsfeed list will push out an older entry of random position.

Thanks to these assumptions, the resulting models developed in the following are Markovian. Indeed, all inter-arrival times between posts and re-posts are exponential and all choices are probabilistic. The *random selection/random eviction* policy will be assumed throughout the solution process to derive the Newsfeed and Wall balance equations. *Random selection* models the case where users pick a post at random from their Newsfeeds, i.e. without order of preference. We will show later that our solution is actually robust to other selection choices, like the *newest selection* where a user always picks up the object from the top of his Newsfeed list. *Random eviction* models platforms which put new posts to Newsfeeds (and less realistically to Walls) in a random order. In Facebook, content curation algorithms decide the order of content appearance based on some background machine learning algorithms. In Twitter, however, both Newsfeeds and Walls normally show posts in a First-In-First-Out fashion, so the appropriate policy in this case would be the *FIFO eviction* i.e., the oldest object is removed from the list and the fresh content is placed at the top. We will show that such eviction policy satisfies the same balance equations as the random one. More complicated options for the selection and eviction policies will be compared to our solution by simulation in Section VI-A.

B. Detailed model

The full state-description for this system is an N-tuple $\mathbf{U} := (\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)})$, where $\mathbf{U}^{(n)} = (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ is the state of user n (at a given time t, omitted in notations for sake of clarity). $\mathbf{x}^{(n)}$ is the state of his Newsfeed and $\mathbf{y}^{(n)}$ the state of his Wall. The random eviction and random selection assumptions allow to describe the system-state evolution without using information over the order of posts in the lists. Then, $\mathbf{x}^{(n)} = (x_1^{(n)}, \ldots, x_N^{(n)})$, where $x_i^{(n)}$ counts the number of posts with user-origin i found on the Newsfeed of user n. Similarly, $\mathbf{y}^{(n)} = (y_1^{(n)}, \ldots, y_N^{(n)})$, where $y_i^{(n)}$ counts the number of posts with origin i found on the Wall of user n.

With all the assumptions described in Section III-A, the stochastic process with full state $\mathbf{U}$ is a continuous-time Markov chain model with finite state-space. This process obtains a unique stationary distribution. However, even for very small values of the system parameters the number of states will be enormous, whereas the state of a user's Newsfeed and Wall is coupled with the state of other users. As a result, any numerical method to find the solution, would be computationally intractable. For this reason we first introduce in the next subsections a state aggregation and a simple decoupling of the state-space that considerably reduce the solution complexity. Following that, we prove that the resulting balance equations we find are exact for the detailed model.

It is important to understand where the coupling between states of different users appears in the detailed model, before presenting the aggregated and decomposed model. Consider user n and focus on label i posts. A leader k of user n will re-post from his own Newsfeed to his own Wall a post of label i with probability $x_i^{(k)}/M$, due to the random selection policy. This post will appear immediately on the Newsfeed of user n, thus changing its state $\mathbf{x}^{(n)}$. Hence, the evolution of the state of user n depends not only on his own current state and on his own activity, but also on the current states of his leaders (in this example $x_i^{(k)}$).

C. State aggregation

To simplify the solution process we first need to describe the state-space in a more compact way. To do so, we focus on posts from a particular user i and calculate the influence of this user i on the entire network. Of course, one can successively apply the technique to all $i = 1, \ldots, N$ in order to determine eventually the influence and Ψ-scores of everyone.

The state aggregation is as follows. On all N Walls and N Newsfeeds we consider only two types of posts; those of origin i, and those issued from other users labelled as $-i$. In other words we aggregate the effect of all users except i. Remember that the detailed state of user's n Newsfeed was the N-dimensional vector $\mathbf{x}^{(n)}$. By applying the state-aggregation this is now described by $(x_i^{(n)}, x_{-i}^{(n)})$, whose sum is equal to the Newsfeed size M, so that $x_{-i}^{(n)} = M - x_i^{(n)}$. As a result, the state of the Newsfeed of user n is reduced to a single integer $\mathbf{x}^{(n)} = x_i^{(n)}$ with values ranging from 0 to M. Similarly, the state of user n's Wall becomes also 1-dimensional $\mathbf{y}^{(n)} = y_i^{(n)}$ with values ranging from 0 to K.

D. Decomposition by mean-field approximation

After state aggregation, the states $(x_i^{(n)}, y_i^{(n)})$ of different users n are always coupled among each other. We decompose here the state-description, to obtain $2N$ independent 1-dimensional Markov Chains, each one associated with the Newsfeed and the Wall of a user. To do so, we use a "mean-field" approximation [24]: for a given user n, the state transitions of his Newsfeed and Wall will still be a function of his own current state and activity, as well as the activity of all of his leaders. But they will not depend anymore on the current Newsfeed and Wall states of the user's leaders $x_i^{(k)}(t)$, $y_i^{(k)}(t)$ but rather on their *average probabilities in steady-state*, which at this stage are unknown values.

More precisely, let us consider user n. We denote by $p_i^{(n)}$ the steady-state probability for a post on Newsfeed n to be of label i, i.e., to originate from user i. Similarly, we have already defined in Section II-B $q_i^{(n)}$ as the steady-state probability for a post on the Wall of user n to be of label i. These quantities for $n = 1, \ldots, N$ are the model *unknowns* after aggregation. Note that these probabilities are actually the ergodic means of the related user states, i.e., $p_i^{(n)} = \mathbb{E}[\frac{X_i^{(n)}}{M}]$ and $q_i^{(n)} = \mathbb{E}[\frac{Y_i^{(n)}}{K}]$.

We distinguish here between the Newsfeed and Wall of user i (particularized user) and the Newsfeeds and Walls of users $n = j \neq i$. Consider the Newsfeed of user j; the state $x_i^{(j)}$ can evolve as follows:

$$x_i^{(j)} \xrightarrow{f_+^{(j)}(x_i^{(j)})} x_i^{(j)} + 1 \qquad \& \qquad x_i^{(j)} \xrightarrow{f_-^{(j)}(x_i^{(j)})} x_i^{(j)} - 1.$$

In the above $f_+^{(j)}$ and $f_-^{(j)}$ are the transition rates between the states, respecting the range 0 to M. The rate $f_+^{(j)}(x_i^{(j)})$ is

$$f_+^{(j)} = \left(\lambda^{(i)} \mathbf{1}_{\{i \in \mathcal{L}^{(j)}\}} + \sum_{k \in \mathcal{L}^{(j)}} \mu^{(k)} p_i^{(k)} \right) \frac{M - x_i^{(j)}}{M}. \quad (2)$$

Indeed, $\lambda^{(i)}$ is the rate with which user i generates a new own post on his Wall, post that instantaneously appears on the Newsfeed of user j, if $i \in \mathcal{L}^{(j)}$. A post of label i can as well appear in the Newsfeed of j through re-posting. This occurs when one of the j's leaders visits his own Newsfeed and re-posts a label i post on his own Wall. For leader k such event occurs with rate $\mu^{(k)} \frac{x_i^{(k)}}{M}$, following the random selection policy. We propose that $\mu^{(k)} p_i^{(k)}$ is an estimation of this rate and this is where the *"mean-field" approximation* that decomposes the state-space lies. Finally, $\frac{M - x_i^{(j)}}{M}$ is the probability that an incoming post (with label i) replaces an old post of label $-i$, by the principle of random eviction.

The rate $f_-^{(j)}(x_i^{(j)})$ is defined in a similar manner as

$$f_-^{(j)} = \left(\sum_{k \in \mathcal{L}^{(j)}, k \neq i} \lambda^{(k)} + \sum_{k \in \mathcal{L}^{(j)}} \mu^{(k)} (1 - p_i^{(k)}) \right) \frac{x_i^{(j)}}{M}. \quad (3)$$

As a consequence, the rate transitions from $x_i^{(j)}$ to $+1$ or -1 depend only on the current state of Newsfeed j and not that of its leaders. We can thus describe $x_i^{(j)}$ as a 1-dimensional Markov Chain. Similar arguments hold for the Newsfeed state of user $n = i$. These simple Markov Chains are shown at

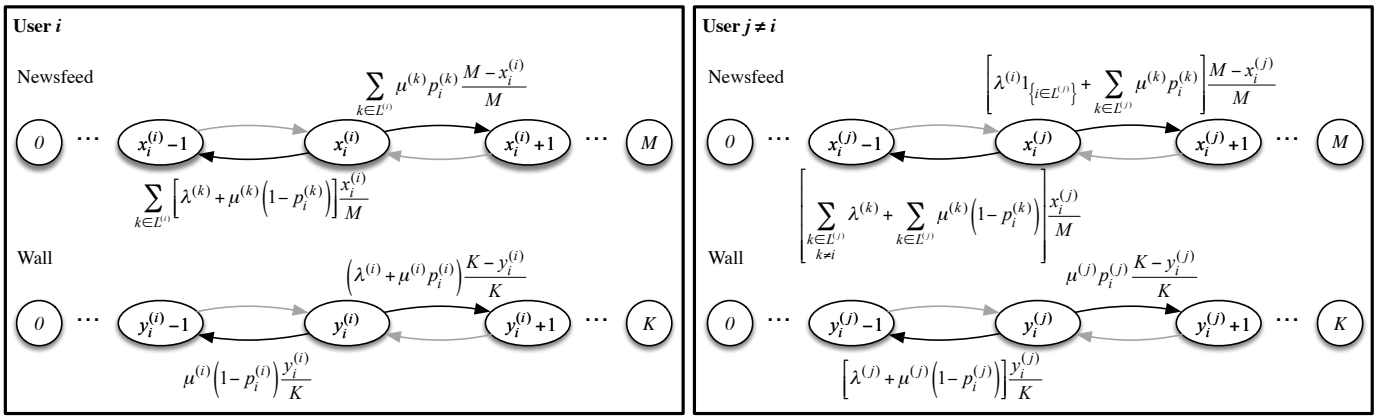


Fig. 2. Aggregated Markov Chain model.

the top part of Figure 2. From the stationary probabilities of the chain for Newsfeed n, we can derive the steady-state probabilities $p_i^{(n)}$:

$$p_i^{(n)} = \sum_{x_i^{(n)}=0}^{M} \pi(x_i^{(n)}) \frac{x_i^{(n)}}{M}. \tag{4}$$

Observe that the unknown probabilities $p_i^{(n)}$ depend on the steady-state solution of the 1-dimensional Markov chain, whose transition rates depend in their turn on the probabilities $p_i^{(.)}$ (see e.g. (2)). As a result, the probabilities $p_i^{(n)}$ result from the solution of a *fixed-point problem*.

For the Walls of users we can follow a similar process. In the Wall of user $j \neq i$, the state $y_i^{(j)}$ can evolve as follows:

$$y_i^{(j)} \xrightarrow{g_+^{(j)}(y_i^{(j)})} y_i^{(j)} + 1 \qquad \& \qquad y_i^{(j)} \xrightarrow{g_-^{(j)}(y_i^{(j)})} y_i^{(j)} - 1.$$

In the above $g_+^{(j)}$ and $g_-^{(j)}$ are the transition rates between the states, respecting the range 0 to K. The rate $g_+^{(j)}(y_i^{(j)})$ is

$$g_+^{(j)}(y_i^{(j)}) \quad = \quad \mu^{(j)} p_i^{(j)} \frac{K - y_i^{(j)}}{K}. \tag{5}$$

Indeed, the state of posts i on the Wall of user $j \neq i$ can only evolve by reposting. Such posts enter the Wall j with average rate $\mu^{(j)} p_i^{(j)}$, because the user re-posts with rate $\mu^{(j)}$ and has $p_i^{(j)}$ probability to choose posts of label i, due to random selection. This is again the *"mean-field" approximation*. The incoming post will replace an old post of label $-i$ with probability $\frac{K-y_i^{(j)}}{K}$ due to random eviction. The corresponding rate $g_-^{(j)}(y_i^{(j)})$ is defined in a similar fashion. Hence, the state evolution of posts i on the Wall of user j can be described by an independent 1-dimensional Markov Chain. We proceed similarly for the Wall of user $n = i$. These simple Markov chains are illustrated in detail at the bottom part of Figure 2. From the stationary probabilities of the chain associated with user's n Wall, we can derive the $q_i^{(n)}$:

$$q_i^{(n)} = \sum_{y_i^{(n)}=0}^{K} \pi(y_i^{(n)}) \frac{y_i^{(n)}}{K}. \tag{6}$$

Note from (5) and Fig. 2 that the Wall probabilities $q_i^{(n)}$ do not result from a fixed-point solution, because they are directly expressed as a function of the Newsfeed probabilities $p_i^{(n)}$.

E. Derivation of the balance equations

Here we further develop and simplify the equations (4) and (6). We first consider the Markov chain associated with the Newsfeed of user $n = j \neq i$ (see top right part of Fig. 2). To solve this birth-and-death process we define the quantity

$$\tau_j := \frac{\lambda^{(i)} \mathbf{1}_{\{i\in\mathcal{L}^{(j)}\}} + \sum_{k\in\mathcal{L}^{(j)}} \mu^{(k)} p_i^{(k)}}{\sum_{k\in\mathcal{L}^{(j)}, k\neq i} \lambda^{(k)} + \sum_{k\in\mathcal{L}^{(j)}} \mu^{(k)}(1-p_i^{(k)})}.$$

Note that τ_j depends on all probabilities $p_i^{(k)}$ for $k \in \mathcal{L}^{(j)}$. The steady-state probability of the Markov chain associated with the Newsfeed of user j can then be derived using τ_j:

$$\pi(x_i^{(j)}) = \pi(0) \begin{pmatrix} M \\ x_i^{(j)} \end{pmatrix} \tau_j^{x_i^{(j)}},$$

where $\pi(0)$ is obtained by normalization (thanks to the Binomial formula):

$$\pi(0) = \frac{1}{(1+\tau_j)^M}.$$

Then, applying this result in (4) we get:

$$\begin{aligned} p_i^{(j)} \quad &= \quad \frac{1}{(1+\tau_j)^M} \sum_{m=1}^{M} \begin{pmatrix} M \\ m \end{pmatrix} \tau_j^m \frac{m}{M} \\ &\overset{m':=m-1}{=} \quad \frac{\tau_j}{(1+\tau_j)^M} \sum_{m'=0}^{M-1} \begin{pmatrix} M-1 \\ m' \end{pmatrix} \tau_j^{m'} \\ &\overset{Binomial}{=} \quad \frac{\tau_j}{1+\tau_j}. \end{aligned}$$

By replacing the expression for τ_j, we obtain the following very simple expression for posts of origin i in the Newsfeed of user j. The fixed-point is now visible:

$$\left[\sum_{k\in\mathcal{L}^{(j)}} \left(\lambda^{(k)} + \mu^{(k)}\right)\right] p_i^{(j)} = \lambda^{(i)} \mathbf{1}_{\{i\in\mathcal{L}^{(j)}\}} + \sum_{k\in\mathcal{L}^{(j)}} \mu^{(k)} p_i^{(k)}. \tag{7}$$

Following a similar reasoning, we get for the Newsfeed of any user $n = i$, the following equation:

$$\left[\sum_{k\in\mathcal{L}^{(i)}}\left(\lambda^{(k)}+\mu^{(k)}\right)\right]p_i^{(i)} = \sum_{k\in\mathcal{L}^{(i)}}\mu^{(k)}p_i^{(k)}. \quad (8)$$

As a result, the set of N equations (7) and (8) constitute the new equations of the fixed point, whose solution gives the required Newsfeed probabilities $p_i^{(n)}$ for $n = 1, \ldots, N$.

In the same fashion, the steady-state probabilities for the Wall can be directly derived from the steady-state probabilities for the Newsfeed through the following equations:

$$\left(\lambda^{(j)}+\mu^{(j)}\right)q_i^{(j)} = \mu^{(j)}p_i^{(j)}, \quad (9)$$

$$\left(\lambda^{(i)}+\mu^{(i)}\right)q_i^{(i)} = \lambda^{(i)}+\mu^{(i)}p_i^{(i)}. \quad (10)$$

F. Explanation of the balance equations

Interestingly, equations (7)-(8) and (9)-(10) allow for a simple intuitive interpretation: they balance the incoming and outgoing flow of posts of origin i on each Newsfeed and Wall list. More precisely, equation (7) equalizes the incoming rate and the outgoing rate of posts of origin i in the Newsfeed of user j (for $j \neq i$). Here, $\sum_{k\in\mathcal{L}^{(j)}}\left(\lambda^{(k)}+\mu^{(k)}\right)$ is the average number of posts per unit of time that enter the Newsfeed of user j. From the random eviction policy, each of these arriving posts replaces a post of origin i with probability $p_i^{(j)}$. Indeed, by assuming that post and re-post processes are Poisson, the PASTA property holds which tells us that arriving posts see the Newsfeed in steady-state. As a result, the left-hand side of equation (7) is just the outgoing rate of posts of origin i in the Newsfeed of user j. Now looking at the right-hand side of this equation, $\mu^{(k)}$ is the average number of posts per unit of time that arrive on the Newsfeed of user j because a leader k of j reposts something on his Wall. Each of these posts is of origin i with probability $p_i^{(k)}$, due to the random selection policy in Newsfeeds. In addition, if i is a leader of j, the $\lambda^{(i)}$ self-posts of i per unit of time also appear on the Newsfeed of j. As a result, the right-hand side of equation (7) is the incoming rate of posts of origin i in the Newsfeed of user j.

Similarly, equation (8) equalizes the incoming rate and the outgoing rate of posts of origin i in the Newsfeed of user i. The only difference is that a new post that has just been created by i does not appear on his own Newsfeed.

Equation (9) equalizes the incoming rate and the outgoing rate of posts of origin i in the Wall of user j. Indeed $\lambda^{(j)}+\mu^{(j)}$ is the average number of posts per unit of time that enter the Wall of user j. Each of these posts replaces a post of origin i with probability $q_i^{(j)}$, due to the random eviction policy in Walls and the PASTA property. As a result, the left-hand side of equation (9) is the outgoing rate of posts of origin i from the Wall of user j. Obviously, $\mu^{(j)}p_i^{(j)}$ is the average number of posts of origin i per unit of time that arrive on the Wall of user j, due to the random selection policy in Newsfeeds.

Similarly, equation (10) equalizes the incoming and outgoing rate of posts of origin i on the Wall of user i. We just have to add at the incoming rate the $\lambda^{(i)}$ self-posts per unit of time from i.

Theorem 1 (Exactness). *For Poisson posting and re-posting activity and for the random selection / random eviction policy, the equations (7)-(8) and (9)-(10) describe exactly the original detailed model in steady-state.*

The proof can be found in the Appendix.

The balance equations further give an important structural property of the steady-state solution as a side product.

Corollary 1 (Insensitivity in list size). *In view of (7)-(8) and (9)-(10), the steady-state probabilities to find posts from user i on the Newsfeed of any user n ($p_i^{(n)}$, $n = 1, \ldots, N$) as well as on the Wall of any user n ($q_i^{(n)}$, $n = 1, \ldots, N$), depend neither on the size M of the Newsfeed, nor on the size K of the Wall.*

G. Alternative selection and eviction policies

Looking at the model through the balance equations enables us to relax random selection and random eviction assumptions, and introduce alternative policies. We will show now that a number of different policies satisfy the same balance equations, thus granting generality to the result.

1) Random selection / FIFO eviction: Let us modify the eviction policy in both Newsfeeds and Walls, and replace "Random" by a more realistic FIFO policy: now, a new post enters at the top of the list and evicts the oldest post out of the list. We thus have to modify only the left-hand side of equation (7). To do so, we define $\phi_i^{(j)}$ as the new outgoing rate of posts of origin i in the Newsfeed of user j. This will replace the left-hand side of eq. (7)). From Little's law,

$$\phi_i^{(j)} = \frac{\overline{X}_i^{(j)}}{\overline{T}_i^{(j)}}, \quad (11)$$

where $\overline{X}_i^{(j)}$ is the average number of posts of origin i in the Newsfeed of user j, and $\overline{T}_i^{(j)}$ is the average time a post of origin i stays in the Newsfeed of user j.

The total arrival rate of posts in the Newsfeed of user j is $\sum_{k\in\mathcal{L}^{(j)}}\left(\lambda^{(k)}+\mu^{(k)}\right)$. The mean time between two successive arrivals in the Newsfeed of user j is thus the inverse of this quantity. As a result any post arriving in the Newsfeed of user j will stay on average M times this mean value:

$$\overline{T}_i^{(j)} = \frac{M}{\sum_{k\in\mathcal{L}^{(j)}}\left(\lambda^{(k)}+\mu^{(k)}\right)}. \quad (12)$$

Since $\overline{X}_i^{(j)} = Mp_i^{(j)}$ per definition, we conclude that:

$$\phi_i^{(j)} = p_i^{(j)}\sum_{k\in\mathcal{L}^{(j)}}\left(\lambda^{(k)}+\mu^{(k)}\right) \quad \text{[FIFO evict]}, \quad (13)$$

which has the same expression as the left-hand side of equation (7). In other words, when we replace the "Random" eviction policy by the FIFO eviction policy in the Newsfeed, equation (7) does not change. We can easily show by a similar reasoning that equations (8), (9) and (10) remain also unchanged under the FIFO eviction policy.

We now show that the set of balance equations remain the same also if we choose a TTL (Time-To-Live) eviction principle [23], [25]. Here, each post stays at the Newsfeed for

a fix amount of time T before leaving. In this case, the size of the list is not constant M, but rather varies over time. By Little's law, the mean Newsfeed size is equal to

$$\overline{M}^{(j)} \quad = \quad T \sum_{k \in \mathcal{L}^{(j)}} (\lambda^{(k)} + \mu^{(k)}), \tag{14}$$

where again $\sum_{k \in \mathcal{L}^{(j)}} (\lambda^{(k)} + \mu^{(k)})$ is the total arrival rate of posts in Newsfeed j. Then in (11) we substitute $\overline{X}_i^{(j)} = \overline{M}^{(j)} p_i^{(j)}$ and $\overline{T}_i^{(j)} = T$, to get

$$\phi_i^{(j)} = p_i^{(j)} \sum_{k \in \mathcal{L}^{(j)}} \left(\lambda^{(k)} + \mu^{(k)} \right) \quad \text{[TTL evict]}. \tag{15}$$

2) Newest selection / Random eviction: We come back to our original model with a "Random" eviction policy, and where Newsfeeds are of limited size M and Walls are of limited size K. We have proved in Theorem 1 that the steady-state probabilities $p_i^{(j)}$ and $q_i^{(j)}$, being solutions of the system (7)-(10)) depend neither on M nor on K. In order to change the selection policy from "Random" to "Newest", we just have to take $M = 1$. Indeed, when the size of Newsfeeds is unitary, the "Random" selection will necessarily choose the newest (i.e. latest, freshest) entry of the Newsfeed to repost on the Wall of a user. As a result, the system (7)-(10) remains true also under the "Newest" selection policy.

IV. Closed Form solution

A. Linear system

We can re-write (8)-(7) and (10)-(9) for posts with label i in a compact form and summarize our findings as follows.

Theorem 2 (Linear System). *The unknown column vectors* $\mathbf{p}_i := (p_i^{(1)}, \ldots, p_i^{(N)})^T$ *and* $\mathbf{q}_i := (q_i^{(1)}, \ldots, q_i^{(N)})^T$ *are the solution of the following linear system*

$$\mathbf{p}_i \quad = \quad \mathbf{A} \cdot \mathbf{p}_i + \mathbf{b}_i \tag{16}$$

$$\mathbf{q}_i \quad = \quad \mathbf{C} \cdot \mathbf{p}_i + \mathbf{d}_i. \tag{17}$$

In the above, $\mathbf{A}$ and $\mathbf{C}$ are $N \times N$ matrices independent of i, whereas $\mathbf{b}_i$ and $\mathbf{d}_i$ are N-column vectors that depend on i. Hence, a standard linear system should be resolved for each i. The entries of the above matrices and vectors are summarised in Table I. It is interesting to note that $a_{j,j} = 0$ for all j, $b_{i,i} = 0$, $\mathbf{C}$ is diagonal, and also there is a unique positive $d_{j,i}$ entry for $i = j$.

The matrix $\mathbf{A}$ is non-negative. In addition, it is row sub-stochastic, meaning that the sum of all its rows is less than or equal to 1, with at least one row sum strictly less than 1 (if we reasonably assume that at least one user injects self-posts). Another interesting property is that $\mathbf{A}$ is a weighted version of the Follower-matrix $\mathbf{F} = \mathbf{L}^T$, so that if $\mathbf{1}_{\left\{j \in \mathcal{F}^{(k)}\right\}} = \mathbf{1}_{\left\{k \in \mathcal{L}^{(j)}\right\}} = 0 \Rightarrow a_{j,k} = 0$. There are cases however where j follows ℓ, but $a_{j,\ell} = 0$ in the matrix $\mathbf{A}$, because $\mu^{(\ell)} = 0$. Hence, users that never re-post from their Newsfeed alter the possibilities of post propagation in the graph. This is why we call $\mathbf{A}$, the *propagation matrix*.

TABLE I
Entries for the matrices/vectors of the linear system.

$\mathbf{A}$	$a_{j,k} := \frac{\mu^{(k)}}{\sum_{\ell \in \mathcal{L}^{(j)}} (\lambda^{(\ell)} + \mu^{(\ell)})} \mathbf{1}_{\{k \in \mathcal{L}^{(j)}\}}$
$\mathbf{b}_i$	$b_{j,i} := \frac{\lambda^{(i)}}{\sum_{\ell \in \mathcal{L}^{(j)}} (\lambda^{(\ell)} + \mu^{(\ell)})} \mathbf{1}_{\{i \in \mathcal{L}^{(j)}\}}$
$\mathbf{C}$	$c_{j,k} := \frac{\mu^{(j)}}{\lambda^{(j)} + \mu^{(j)}} \mathbf{1}_{\{j=k\}}$
$\mathbf{d}_i$	$d_{j,i} := \frac{\lambda^{(i)}}{\lambda^{(i)} + \mu^{(i)}} \mathbf{1}_{\{j=i\}}$

B. Closed-form solution

We would like to know under which conditions a solution to the linear system in (16) - and as a consequence (17) - exists. To this aim we first recall the following known Lemma, where $\mathbf{I}_N$ is the $N \times N$ identity matrix. It relates the solution of our system to the spectral radius of $\mathbf{A}$, denoted by $\rho(\mathbf{A})$.

Lemma 1. *[26, Chapter 6, Lemma 2.1] Given a nonnegative matrix* $\mathbf{T} \in \mathbb{R}_+^{N \times N}$*, its spectral radius is* $\rho(\mathbf{T}) < 1$ *if and only if* $(\mathbf{I}_N - \mathbf{T})^{-1}$ *exists, which can be written as the series*

$$(\mathbf{I}_N - \mathbf{T})^{-1} \quad = \quad \sum_{n=0}^{\infty} \mathbf{T}^n \geq 0. \tag{18}$$

From the specific structure of the non-negative matrix $\mathbf{A}$ we have the following property.

Lemma 2. *It holds* $\rho(\mathbf{A}) \leq 1$*. Strict inequality is guaranteed in the following non-exclusive non-exhaustive cases (cs):*

(cs1) $\lambda^{(n)} > 0$, $\forall n \in \mathcal{N}$.

(cs2) For every cycle in the Leader-graph, at least one participating user has a leader k *with positive self-post rate.*

Proof. Let us denote the row sums of $\mathbf{A}$ by $r(j)$, $j = 1 \ldots N$. Then $r(j) \leq 1$ by definition from Table I. It is known that ([28, Theorem 8.1.22]) the following bounds are valid for the spectral radius of a non-negative matrix: $\min_{j=1}^{N} r(j) \leq \rho(\mathbf{A}) \leq \max_{j=1}^{N} r(j)$. The right-hand side in our case is 1 and the first part is proven.

(cs1) When $\lambda^{(n)} > 0$, $\forall n$, then $\forall j$ and $k \in \mathcal{L}^{(j)}$, $a_{j,k} < \mu^{(k)} / \sum_{\ell \in \mathcal{L}^{(j)}} \mu^{(\ell)}$, so that $r(j) < 1$, $\forall j$. Then the matrix is *strictly* sub-stochastic, and $\rho(\mathbf{A}) \leq \max_{j=1}^{N} r(j) < 1$.

(cs2) In this case, suppose the length of a particular cycle is $\gamma > 1$ and the participating nodes are $n_1, \ldots, n_\gamma$. Then at least one row sum $r(j) < 1$, $j \in \{n_1, \ldots, n_\gamma\}$. By direct application of the Al'pin, Elsner, van den Dreissche bound [27, Theorem A], we conclude that $\rho(\mathbf{A}) < 1$. An additional condition for this bound is that $r(j) > 0$, $\forall j$, which is satisfied when $\mathcal{L}^{(j)} \neq \emptyset$, $\forall j \in \mathcal{N}$ and not all leaders of some user have $\mu^{(k)} = 0$. □

Remark 1. *A special instance of (cs2) is when* $\mathbf{A}$ *is irreducible and* $\lambda^{(j)} > 0$ *for at least one* $j \in \mathcal{N}$.

Theorem 3 (**Solution**). *For the two cases of Lemma 2, the solution of the linear system (16)-(17) is unique, and given by*

$$\mathbf{p}_i = (\mathbf{I}_N - \mathbf{A})^{-1}\mathbf{b}_i \quad (19)$$

$$\mathbf{q}_i = \mathbf{C}(\mathbf{I}_N - \mathbf{A})^{-1}\mathbf{b}_i + \mathbf{d}_i. \quad (20)$$

Proof. Lemma 2 guarantees that $\rho(\mathbf{A}) < 1$ in both cases, so that from Lemma 1 the inverse $(\mathbf{I}_N - \mathbf{A})^{-1} \geq 0$ exists and the solution is unique. □

An interesting observation is that the inverse $(\mathbf{I}_N - \mathbf{A})^{-1}$ involved in the derivation of $\mathbf{p}_i$ (relation (19)) is independent of i. Thus, in the solution process the inverse should be calculated only once, and then applied to the expressions in (19)-(20) for labels $i = 1, \ldots, N$.

C. Fixed-point algorithm

For large N it can be practically very difficult to calculate the inverse $(\mathbf{I}_N - \mathbf{A})^{-1}$. A different way to proceed in order to solve the system (16) is to use an iterative approach.

Theorem 4. *For the two cases of Lemma 2 and any initialization vector $\mathbf{p}_i(0)$, the discrete-time linear system (21) converges towards the fixed-point solution (19) when $t \to \infty$.*

$$\mathbf{p}_i(t) = \mathbf{A} \cdot \mathbf{p}_i(t-1) + \mathbf{b}_i \quad (21)$$

Proof. We first write $\mathbf{p}_i(t)$ as a function of $\mathbf{p}_i(0)$ and t,

$$\mathbf{p}_i(t) = \mathbf{A}^t \mathbf{p}_i(0) + \left(\sum_{n=0}^{t-1} \mathbf{A}^n\right)\mathbf{b}_i.$$

We need to find the limiting value $\mathbf{p}_i := \lim_{t\to\infty} \mathbf{p}_i(t)$. For the two cases in Lemma 2 we have $\rho(\mathbf{A}) < 1$, so that from [28, pp.137–138, or Theorem 5.6.12] it holds $\mathbf{A}^\infty := \lim_{t\to\infty} \mathbf{A}^t = \mathbf{0}$. Additionally, from Lemma 1 the limit of the matrix series for $t \to \infty$ converges to $(\mathbf{I}_N - \mathbf{A})^{-1}$. Hence, the iteration converges to the solution (19), and is independent of the initialisation $\mathbf{p}_i(0)$. □

Note that once the Newsfeed-vector $\mathbf{p}_i := \lim_{t\to\infty} \mathbf{p}_i(t)$ has been obtained, the Wall-vector $\mathbf{q}_i$ can be calculated from relation (17). The influence score for user i, i.e. the value Ψ_i is then directly derived from (1). We need to solve for all i's to derive all scores $\{\Psi_i\}_{i=1}^N$, however notice that the matrices $\mathbf{A}$ and $\mathbf{C}$ are the same for all users, and only $\mathbf{b}_i$ and $\mathbf{d}_i$ differ.

V. Implementation and Numerical Aspects

For the numerical implementation we coded the following programs, that we make available in [14]: (A) an algorithm to derive the Ψ-rank for each user from the balance equations of the model; two versions are coded, one for small OSP sizes and a sparse one for real-world sizes, (B) a discrete-event simulator, to simulate over time the behaviour of an OSP with arbitrary input traffic and user/platform policies, and (C) an emulator, which takes a real data-trace as input and outputs empirical Ψ-scores.

A. Ψ-ranking by the model

We remind the reader that the Ψ-score of user i's influence in the social platform was introduced in (1), as a function of the $q_i^{(j)}$'s ($j = 1, \ldots, N$), i.e. the steady-state Wall probabilities. These values have been derived in closed form, through the balance equations. The two methods to calculate them, one using Theorem 3 with matrix inversion and a second using Theorem 4, are coded in [14] for small OSP sizes. The algorithm takes as input the vector of all posting and reposting rates $\lambda = (\lambda_1, \ldots, \lambda_N)$ and $\mu = (\mu_1, \ldots, \mu_N)$, as well as the leader-graph $\mathcal{G}$ and outputs the Ψ-scores.

Since the size N of real-world social graphs is of the order of millions of users, an efficient algorithm that runs in reasonable time-scales is necessary to calculate these scores for all users in the platform. A tedious matrix inversion $(\mathbf{I}_N - \mathbf{A})^{-1}$ is not recommended for such cases, as it is computationally very expensive - typically of the order of $\mathcal{O}(N^3)$. We have programmed here another algorithm to calculate the Ψ-score in large graphs, which implements a sparse version of the iterative solution introduced in Theorem 4. It computes first the p_i's by involving at each iteration a matrix-vector multiplication $\mathbf{A} \cdot \mathbf{p}_i$ and then a vector addition $\mathbf{b}_i$, which take into account the sparsity of both the propagation matrix $\mathbf{A}$, and each vector $\mathbf{b}_i$. Sparsity comes from the fact that the number of leaders and followers of any user i is very small compared to the total population N; as a second step - to calculate the q_i's we also use the fact that $\mathbf{C}$ is diagonal and sparse, and $\mathbf{d}_i$ has a single non-zero element. We can break the user set in subsets of users and parallelise the computational process on several machines, because solving for the influence of one user i does not affect the process of solving for others.

B. Discrete-event simulator

Additionally, we have developed our own discrete-event simulator (also available in [14]) to validate the mathematical analysis through simulation, and to evaluate the robustness of the modelling assumptions presented in Section III-A against alternative traffic and policies. Unlike the code in the previous section V-A which solves the model's balance equations, the simulator precisely implements the behaviour of the generic OSP over time as described in Section II: i) The global state description consists of dynamic lists (of length K for Walls and M for Newsfeeds); ii) A variety of selection and eviction policies are implemented ("Random", in a first phase, and "Newest", FIFO, "Popular" later to evaluate robustness); iii) Self- and re-posts can be generated according to Poisson or other processes. As such, the simulator *does not* decouple the state space, *does not* estimate average probabilities, and *does not* rely on Markovian assumptions. For each simulation we set $M = 20$ and $K = 10$ and ran long enough simulations to reach the steady-state with small confidence intervals. More specifically, in all experiments, we let the simulator run for a total of 300 000 events (self- and re-posts).

C. Emulator (Trace-based empirical influence)

Finally, we have coded the emulator, which uses a real data-trace as input from Twitter, Weibo or other platform. The

emulator differs from the simulator in the sense that it does not simulate the post propagation or specific policies, rather it directly outputs numerical values of influence, as read from the post sequence in the trace. To do so, we first pre-process the available data trace, so that each line of the input is just the quadruple [PostID, TimeStamp, UserID, RePostID], where the fourth entry is -1 in case of an original post, else the PostID of the original post which was reposted. The program calculates the influence of each user (author) on another user directly from data: *this is equal to the percentage of time that posts of a specific origin occupy the first position on a Wall of another user, assuming a FIFO principle*. By finding all these empirical values $\tilde{q}$[user][author], we can derive the empirical Psi-score Ψ_i^{emu} for all users in the trace. Note here that for the emulator, the empirical influence is determined by the Wall occupancy periods divided by the total duration of the data-trace, and no further information about the social graph is necessary. We provide the code for the emulator also in [14].

VI. Numerical Evaluation

We plot the numerical values of the influence metric (1) from the model (see V-A) for three different user-graph configurations and increasing N: complete graph, grid and ring. The results are shown in Figure 3a-c. We also include the score values from the simulator (see V-B), with $T\_iter = 300,000$.

a) Complete graph: In this case, each user follows all other users. All users have the same activity tuple (λ, μ). As the network is totally symmetric, we plot the influence of any user over the network for three different values of $\frac{\lambda}{\mu}$ (0.5, 1 and 2), as a function of the network size N. We observe that the influence of a given user decreases as the size of the network increases. This is reasonable since the more the users in the network, the larger the competition between users to influence the Wall of each other, thus the smaller the influence per user. Furthermore, we observe that the smaller the $\frac{\lambda}{\mu}$, the higher the influence. This result shows that in a fully symmetric network, when everyone decreases his self-post activity, there is more space left on his Wall for being influenced by others.

b) Grid graph: Depending on his position in the grid, a user may have 4, 3 or 2 leaders. All users have the same activity tuple (λ, μ), here set to $(5, 3)$. On Fig. 3b, we plot the influence metric for three different types of users : the central user (with 4 leaders), a user at the middle of an edge (with 3 leaders) and a user at a corner (with 2 leaders), as a function of the network size, $N = 9, 25, 49, 81, 121$. We observe that the corner user has less influence than the edge user, who in turn has less influence than the central user. This is an obvious qualitative result, considering the different numbers of followers each user has, but here we quantify the impact of the position through our Ψ-score metric.

c) Ring graph: Users are arranged on a circle and each user i has R_i leaders on his right ($i+1$, ..., $i+R_i$) and R_i leaders on his left ($i-1$, ..., $i-R_i$), R_i being denoted as the *radius* of user i. For the comparison we use $N = 31$ users. Each one has been given a random uniform radius in $\{1, ..., 15\}$ and a random uniform activity tuple (λ_i, μ_i) in $[0.1, 10] \times [0.1, 10]$. We plot the influence for each user in Fig.3c. We observe a very good fit between model and simulation, even for the user who has the least accurate estimation (user "20" with a relative error of 2.5%).

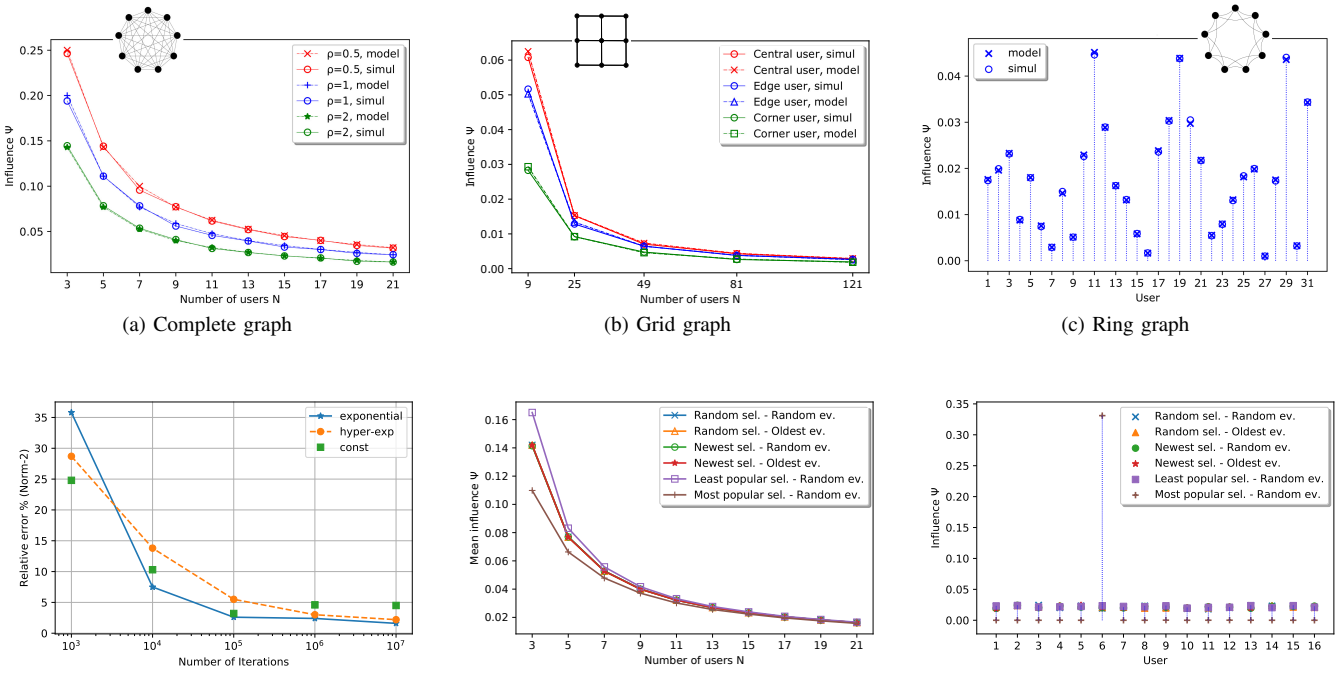
d) Convergence: We evaluate the convergence speed of the simulator to the solution of the balance equations by increasing the number of iterations T_iter and plotting the relative error; this is the ratio of the 2-norm of the difference between simulated and analytical values, divided by the 2-norm of the analytical values. We choose for the experiment a ring graph of $N = 8$ with random λ_i, μ_i per user and random radius R_i, as above. The convergence till a relative error of 1.5% is shown in Fig. 3d.

A. Robustness

We further evaluate the robustness of the model with respect to the modeling assumptions. For this purpose, we modified our simulator to take into account inter-arrival distributions alternative to Poisson, as well as selection and eviction policies other than random.

a) Alternative inter-arrival times: In Fig. 3d, we plot in addition to the exponential inter-arrival case, also the convergence of the simulator to the solution from the balance equations when applying alternatively the following two distributions for both posting and re-posting: (i) hyper-exponential (with same mean but higher variance than Poisson) and (ii) deterministic (with same mean and zero variance). The figure shows that the hyper-exponential converges to the same solution as the Poisson albeit more slowly. It exhibits a 1.6% relative error compared to the analytical solution for $T\_iter = 10^7$. The convergence for fixed interval process is non-monotone with a relative error of 5% for $T\_iter = 10^7$.

b) Policies: We use here a complete graph with a varying number N of users, and set $(\lambda, \mu) = (10, 5)$. We programmed our simulator (see V-B) to test alternative policies for user selection and post eviction, based on age and post-popularity. In (III-G) we showed that certain alternatives give the same balance equations as the basic "Random/Random" policy; specifically the "newest selection" policy, where a user always chooses to re-post the most recent post on his Newsfeed, and the "FIFO (oldest) eviction" where the new post enters the top of the list and pushes out the oldest one. Figure 3e shows that indeed "Random/Random", "Newest/Random" and "Random/FIFO (oldest)" have the same average performance, as these curves coincide. Furthermore, we test the "least popular (resp. most popular) selection", where the user chooses to re-post from his Newsfeed the post with the maximum (resp. minimum) current global number of re-posts. This policy uses extra information on re-post history. We observe in Fig. 3e that with such choice, the absolute difference with a random policy becomes higher. The comparison is more pronounced when we observe the individual influences on a complete graph with 16 users in Fig. 3f. Interestingly, all policies behave similarly, except the "most popular" selection, which tends to give all influence to a single user. We can conclude that our model is very robust regarding the choice of selection and eviction policies when there is no history information involved.



(a) Complete graph (b) Grid graph (c) Ring graph

(d) Convergence for various arrival distributions (e) Selection/Eviction policies (f) Influence per user

Fig. 3. Top (a)-(c): Ψ-scores in three different graphs. Bottom (d)-(f): Sensitivity with respect to modeling assumptions.

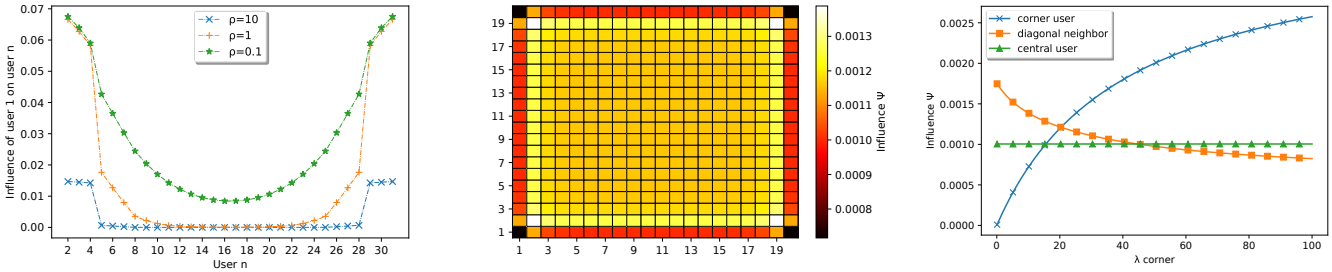


Fig. 4. Model exploitation: (a) left: Ring graph, (b) centre: Grid - fixed symmetric activity, (c) right: Grid - influence evolution.

B. Model exploitation

Having demonstrated the accuracy and the robustness of our model, we now investigate how the influence of a given user is related to his position in the graph and to the relative values of his own activity compared to the activity of other users.

a) Direct vs. indirect influence: We first consider a ring with $N = 31$ users, each one having the same radius $R = 3$, and we plot the influence of user "1" on the other users. Since $R = 3$, user "1" has six followers: users $\{$"2", "3", "4", "29", "30", "31"$\}$. In Fig. 4a we represent three curves corresponding to three different values of the ratio $\rho = \frac{\lambda}{\mu}$, namely $\rho = 0.1$, 1, and 10, assuming that each user in the network has the same self-post rate λ and the same re-post rate μ. First, we observe that all the curves are symmetrical, which comes from the symmetry of the user graph. More importantly, we see that user "1" has a greater influence on his direct followers, than on the other users. And obviously, the greater the distance from user "1", the less influence of user "1" on the considered user. Here user "16" is the one that is less influenced by user "1".

b) Influence and graph position: We consider in Fig. 4b a grid graph with 400 users, each one having the same activity $(\lambda, \mu) = (10, 10)$. Each square represents a user and is colored according to his influence Ψ_i over the network. As expected, the peripheral users, i.e., the users located on the outer edges, and in a more pronounced manner, the corner users are the ones with the smallest influence. This is due to the fact that these users have 3 or 2 direct followers, whereas all others have 4. But contrary to intuition, the central users are not the most influential. In fact, the users with the highest influence (white colour) are the four diagonal neighbours of the four corner users. We have verified this property on different graphs. As an example, on a tree, leaves are the less influential users, whereas the parents of leaves are the most influential users. As a conclusion, in a social network, being a leader of users with few other leaders increases one's influence. Obviously

this is partly due to our definition of influence, but alternative metrics have shown to follow the same trend.

c) Influence and activity: We now want to see if a user with a position that gives him a low influence in the network can counterbalance his bad placement by increasing his posting activity. To this aim we consider again the grid of the previous subsection. We set $(\lambda, \mu) = (10, 10)$ for all users except for the south-west corner user who is given the same re-post rate, $\mu_{corner} = 10$, but that can adjust his self-post rate λ_{corner}. In Fig. 4c, we let only λ_{corner} vary from 0.1 to 100, keeping (λ, μ) and μ_{corner} fixed, and plot the activity of the considered corner user, as well as the activity of the central user and of the diagonal neighbor of the corner user (the most influential in a network with symmetric activities).

First of all, we observe that the corner user becomes more and more influential as his posting rate increases, and he eventually becomes the most influential user in the network. The central user is too far away from the corner user to be affected by the evolution of his posting rate, so his influence remains constant. However, observe that the raise of λ_{corner} causes a drop in the influence of his diagonal direct neighbor. As a conclusion, the answer is positive: one can counterbalance his bad position by increasing his self-posting activity.

VII. Numerical results from Real-world Traces

Traces: We evaluate our model and Ψ metric using two real datasets, one from Twitter and the second from the Weibo social platform. The first trace comes from Kaggle referred to as `Russian`[1]. It contains roughly 2 million (re-)tweets emitted from $180,000$ users during the Russian presidential elections of 2018. The second trace referred to as `Weibo`[2] comes from [30] and contains roughly 34 million messages exchanged over the Chinese microblogging platform Sina Weibo. For both traces, user IDs are anonymised. Posts are ordered in time and their content is removed. Each line contains: [PostID, TimeStamp, UserID, RePostID]. Basic statistics for both datasets are summarised in Table II.

Methodology: Before starting, we need to extract some information from the traces. For each available data trace the input vectors of user activity λ_i, μ_i, $\forall i$ can be estimated as the sample means of each user's posting activity, $\hat{\lambda}_i$, $\hat{\mu}_i$, $\forall i$ i.e., the ratio of the number of posts or re-posts over the total trace duration; the social graph $\mathcal{G}$ – if not provided – should also be inferred from the available traces, as we will show later in the case of `Russian`.

For each trace, we want to show how well our model can evaluate user influence in the respective platform. First, we rank the users based on the model influence Ψ^{model}, where we use the sparse version of the code described in Section V-A with input $(\hat{\lambda}, \hat{\mu}, \mathcal{G})$, and compare this list with the ranking based on user empirical influence, as derived from the emulator Ψ^{emu} in Section V-C. As a second step we compare the model ranking, with the ranking based on alternative influence measures: (i) the user out-degree, (ii) the user post activity $\hat{\lambda}$ and (iii) the user PageRank [12].

[1]Available at https://www.kaggle.com/borisch/russian-election-2018-twitter

[2]Available at https://aminer.org/influencelocality from the paper [30].

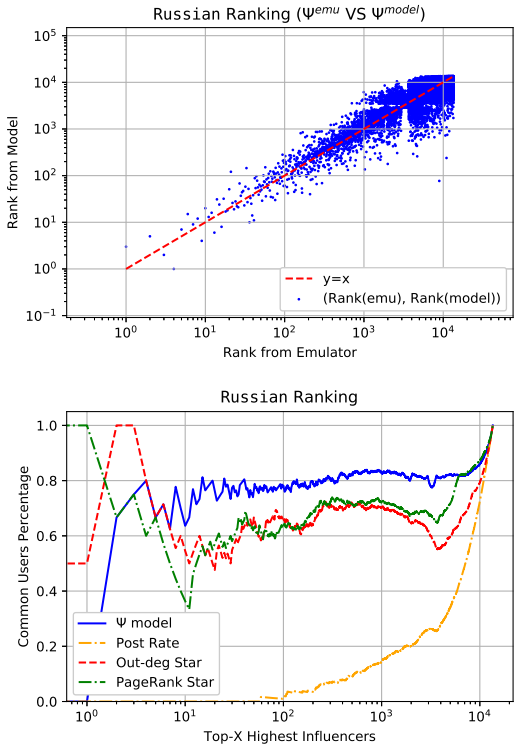


Fig. 5. User ranking comparison for `Russian`. (a) top: Scatter plot between Rank(Ψ^{model}) and Rank(Ψ^{emu}) – points on the left rank higher ($1st$, $2nd$, etc.), (b) bottom: Common users proportion with reference to Rank(Ψ^{emu}).

To compare two ranking lists between each other we use two types of plots: A 2D scatter plot, where each point corresponds to a user and is the tuple of his predicted rank based on the Ψ^{emu} (x-axis) and the Ψ^{model} (y-axis); in such plots we also visualise the distance of the rankings from the line $x = y$, which describes the ideal perfect match of ranks. The second type of plot illustrates a metric similar to the Jaccard index to compare the two rankings, called "Common users proportion". More precisely, if $\{u_1, \ldots, u_X\}$ are the top-X UserIDs according to the emulator list, whereas $\{v_1, \ldots, v_X\}$ the top-X UserIDs for the model list, we define the proportion of common users at depth X of the emulator list by

$$\mathcal{C}_X = \frac{|\{u_1, \ldots, u_X\} \cap \{v_1, \ldots, v_X\}|}{X}.$$

Note that this quantity converges to 1 as X grows to N, because the two full lists contain the same set of users. But for some given $X < N$ (e.g. top-10), the curve shows how well the model manages to rank users in relation to the emulator in the top-X positions.

A. First Dataset — `Russian`

To apply our sparse algorithm (see V-A) we estimate user activity by the sample means $(\hat{\lambda}, \hat{\mu})$, i.e. the number of posts and reposts per user in the trace divided by the total time window (57 days). The user graph needed as input is not directly available for `Russian` and we have to find a way to recover it. For this we use the following heuristic:

TABLE II
BASIC STATISTICS ON BOTH DATASETS.

	Russian	**Weibo**
Time window	57 days	1 216 days
# users	181 621	1 340 816
# posts	674 292	232 978
# reposts	1 271 073	33 307 189
Mean #posts/user	3.71	0.17
Mean #reposts/user	7.00	24.84
Max #posts	4 834	3 718
Max #reposts	2 811	1 032
% users with #posts > 0	54.45	03.55
% users with #reposts > 0	63.60	99.54

TABLE III
STATISTICS OF INPUT SOCIAL GRAPHS.

	Russian (Star)	**Weibo** (Real)
#nodes	181 621	1 340 816
#edges	517 421	291 761 716
mean out-degree	5.70	236.90
max out-degree	7 868	431 385
max in-degree	389	8 107

TABLE IV
TOP-10 INFLUENCERS IN RUSSIAN AS RETURNED BY THE EMULATOR AND COMPARED WITH THE MODEL (STAR GRAPH).

User ID #	$\Psi^{\mathbf{emu}}$ 10^{-3}	$\Psi^{\mathbf{model}}$ 10^{-3}	**Rank** emu	**Rank** model	**Out-dg** Star	$\boldsymbol{\lambda}[s^{-1}]$ 10^{-7}
20905367	12.02	10.23	1	3	6 676	42.9
82299300	11.29	7.03	2	5	7 833	96.0
494076761	7.35	13.80	3	2	6 963	439.3
615422017	5.33	13.82	4	1	5 474	639.5
174953869	5.13	5.81	5	7	1 742	118.5
711363811	4.79	4.12	6	15	1 309	8.2
36309919	4.44	5.23	7	9	4 516	118.5
1867848452	4.15	4.60	8	12	1 235	6.1
34200559	3.68	7.96	9	4	3 571	982.8
50597428	3.36	3.64	10	20	1 156	8.2

Star $\mathcal{G}$ User i is considered to be a follower of user j iff i has reposted a post of origin j at least once.

With *Star* we make the simplifying assumption that a user only reposts content created by his direct leaders. This approach short-circuits the diffusion paths of posts as it links directly original authors to all the re-posters, drawing a network with star-shaped communities. The *Star* graph statistics are given in Table III. We chose to follow this approach because the data-trace does not contain extra information over the IDs of "relay"-users. In such traces we can only know and store the original author and the total set of users each post reached, but not the paths. Note here that inferring graphs from data is a subject of active research and alternative methods can be found in the recent literature [29]. For the time being we will use the Star-network as input; in the `Weibo` section, we will benefit from a real known graph.

The evaluation of the `Russian` model against the emulator is provided in Table IV, and in Figure 5. The top plot in Figure 5 shows the scatter plot of ranking based on Ψ^{model} and Ψ^{emu}; note here that the points at the left correspond to higher influence rank ($1st$, $2nd$,...) compared to points at the right, which are less significant. We observe a very good fit between the two rankings in the entire domain, and even for the top ranks ($1st$, $2nd$,...) at the left part of the plot, which are of practical interest, e.g. for a company targeting high-influencers. At the bottom plot we illustrate four curves: each of them plots the Common users proportion between the user ranking from the emulator and the ranking from (i) user out-degree from the model, (ii) user PageRank from the Star graph, (iii) user post rate, and (iv) the model influence Ψ^{model} using the Star graph. We observe that our model (iv) explains much better than the other metrics the ranking by the emulator, and is able to find 80% of emulator top-X users, in the largest part of the X range. The two curves (i) and (ii) which describe only the graph structure perform much lower. Finally, in this specific example, the curve (iii) about user posting rate has no considerable importance in explaining influence, as such ranking seems unrelated to the emulator.

Precise results for the top-10 Ψ^{emu} ranked users are shown in Table IV. We observe a very good fit between actual values of Ψ^{model} and Ψ^{emu}. Specifically, the model manages to find 7-out-of-10 influencers in the emulator top-10 list. An important observation comes from the two last columns of the table, namely the user out-degree and the user posting rate. We see that neither the out-degree nor the posting frequency follows the ranked order of the top influencers, in support of our observations already in Figure 5. By inspection, the user with maximum out-degree (7833 followers in Star = 7833 users who shared posts originating from him) is ranked 2^{nd} by the emulator and 5^{th} by the model, whereas the user with maximum posting rate is ranked 9^{th} by the emulator and 4^{th} by the model. Impressively, we find in the list a top-influencer (ranked 8^{th} by the emulator) with low posting rate (e.g. $6.13 \cdot 10^{-7}$ [posts/sec] ≈ 0.00221 [posts/day]) and relatively low out-degree (1235 "followers" in Star). Hence, neither the out-degree as a measure of importance in the social graph, nor the posting rate as an activity measure are alone sufficient to rank users by influence. Our Ψ-score mixes user activity with graph position.

For `Russian`, an important part of the user influence is already present in the Star-graph, due to the way we chose to draw it (all users who shared a post are assumed as followers of its original author). The correlation between the influence score Ψ^{model} and *Star* out-degree, is very high, equal to 0.82, whereas the correlation between Ψ^{model} and posting activity λ is only 0.11. For the above reasons, we study the second dataset (`Weibo`), accompanied by its true social graph.

B. Second Dataset — `Weibo`

For the second dataset (`Weibo`) we have access to the underlying friendship graph i.e., who follows who. The graph statistics are given in Table III[3]. We use this *Real* graph to compute values of influence from our model Ψ^{model}, whereas the Ψ^{emu} is derived directly from the trace. The `Weibo` trace is special, because of the way it was collected to serve the study of cascades (see [30]); the authors isolated and kept in

[3]The friendship graph contains users that do not appear in the trace; since no further information is available over their activity we ignore them.

the dataset only certain (approx. $200K$) microblog episodes, which were massively re-posted. Specifically, as can be seen from Table II there are $10\times$ more users than the Russian, but only 3.5% of users post, in comparison to 99.5% who re-post. The number of original posts is very small compared to much larger ($100\times$) number of re-posts, meaning that the trace has a small user set of potentially very large influence. Here, we expect user activity to play an important role in determining influence, not just graph structure. On the other hand, the Real graph contains 291 million (M) edges (see Table III), most of which are never observed to be active in the available trace for post forwarding. In fact the trace contains ≈ 33M re-posts. Even if each repost passed through a different edge from the Real graph, there would be $291 - 33 = 258$M edges unused. Let us see how our model behaves in this particular case.

User ranking for Weibo using the model and the emulator are shown in Figure 6 and Table V. Specifically, Figure 6 (left) presents the scatter plot for ranking by $\Psi^{\rm model}$ and $\Psi^{\rm emu}$. The fit for high influencers is good, but worsens as we move to the right in the low influence ranks, but the points are always centered around the line $x = y$. This behaviour is partly due to the specific trace and partly due to numerical issues; the massive size and density of the Weibo graph and the asymmetry in activity slowed-down computation of $\Psi^{\rm model}$, when using our sparse algorithm for the model, and we had to trade-off accuracy for run-time.

Figure 6 (centre) shows the Common users proportion metric between the rank list from $\Psi^{\rm emu}$ and the ranking from: (i) out-degree, (ii) PageRank, (iii) post rate, and (iv) the model $\Psi^{\rm model}$. The performance of $\Psi^{\rm model}$ is again the best and it can explain around 60% of the user top-X ordering by $\Psi^{\rm emu}$. The reason for the lower performance compared to Russian *Star* is that, in the Real graph most of the edges do not participate in the post diffusion described in the Weibo trace. Our model in this paper does not include contextual preferences towards users or topics, but rather gives equal probability to all posts visible in the Newsfeed to be re-posted. We believe that a pre-processing of the real graph to keep only active relationships (edges) could significantly improve the model performance. Even in this unfavourable situation, however, we observe once again that our model using the Real graph outperforms the other three measures in explaining the top-X influencers found in the emulator ranking list. Interestingly, the graph-related measures of out-degree and PageRank behave very badly, whereas the post rate explains much better the empirical influence; this is to be expected because only cascades of posts from specific origins are kept in the trace. The Ψ metric from our model combines both graph position and user activity to give a better estimation of social influence. Finally, Figure 6 (right) compares ranking by the model with ranking by the post rate and the out-degree. Although $\Psi^{\rm model}$ and post rate performance seem very close in the (centre) plot, less than 80% of the $\Psi^{\rm model}$ ranking can be explained just by the post rate. This means that the $\Psi^{\rm model}$ cannot be replaced simply by the post rate - even in this special case of dataset, because it contains different information over user influence.

Finally, referring to Table V, we see that the model with Real graph input finds 5-out-of-10 top influencers common with the emulator. From the out-degree and activity λ columns, we verify again that the influence Ψ-score cannot be explained by out-degree or activity only, but rather by an appropriate combination of the two, summarised in $\Psi^{\rm model}$.

TABLE V
TOP-10 INFLUENCERS IN WEIBO AS RETURNED BY THE EMULATOR AND COMPARED WITH THE MODEL (REAL GRAPH).

User ID #	$\Psi^{\rm emu}$ 10^{-3}	$\Psi^{\rm model}$ 10^{-3}	**Rank** *emu*	**Rank** *model*	**Out-deg** *Real*	$\boldsymbol{\lambda}[s^{-1}]$ 10^{-6}
519514	37.08	63.88	1	2	459	31.7
490872	24.68	93.99	2	1	595	35.4
1004172	13.30	7.05	3	23	520	2.0
482551	11.53	47.56	4	3	1247	11.0
110361	7.07	13.19	5	5	288	10.3
244531	7.05	12.33	6	7	312	10.4
296675	6.77	8.30	7	17	347	8.7
980392	6.70	5.94	8	27	230	6.2
153610	6.22	2.93	9	54	81	3.6
821785	6.21	11.32	10	11	1084	2.7

VIII. CONCLUSIONS

In this work we have introduced an original Markovian model that analyzes the diffusion of posts in a generic social platform, and quantifies the influence of a given user over any other. By resolving it we have derived closed-form expressions for metrics of influence, which allow to rank users based on the novel Ψ-score; the latter summarises the combined effect of user position in the graph with user activity. These results constitute a novel powerful toolbox that can be further exploited to understand and design social platforms. To highlight the importance of these results we have implemented a sparse version of the solution algorithm in [14] and have applied it to massive data traces from real OSPs (Twitter and Weibo). The model-based Ψ-ranking is verified to correspond well with the empirical influence as read from the traces, a fact which validates our model for real world applications. As a consequence, we believe that the model can serve as *a test and prediction tool* for many social platform and user activity scenarios, when no trace is available. It is flexible enough to further include extra OSP features (post-filtering, "likes", etc). The $\Psi-$score, itself, is shown more suitable to rank user influence compared to standard centrality metrics (out-degree, PageRank, user activity) and cannot be substituted by any of them. Hence, it enriches the literature with a novel important measure that can combine user position in the graph with user activity to adequately rank user influence inside a platform.

REFERENCES

[1] A. Giovanidis, B. Baynat, and A. Vendeville. *Performance Analysis of Online Social Platforms*, IEEE Conference on Computer Communications (IEEE INFOCOM 2019) Paris, France, pp. 2413-2421, 2019.

[2] S. Goel, D.J. Watts, D.G. Goldstein, *The structure of Online Diffusion Networks*, 13th ACM Conference on Electronic Commerce (EC), Valencia, Spain, 2012.

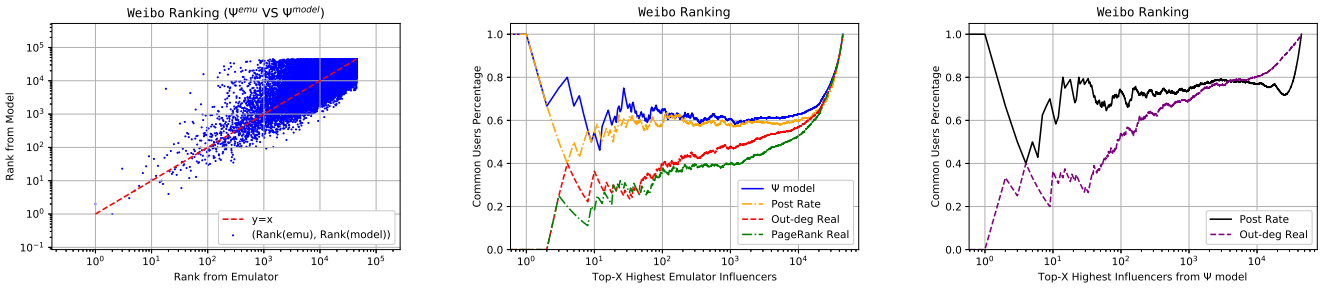


Fig. 6. User ranking comparison for `Weibo`. (a) left: Scatter plot between Rank(Ψ^{model}) and Rank(Ψ^{emu}) – points on the left rank higher ($1st$, $2nd$, etc.), (b) centre: Common users proportion with reference to Rank(Ψ^{emu}), (c) right: Common users proportion with reference to Rank(Ψ^{model}).

[3] P. Alex Dow, L.A. Adamic, A. Friggeri. *The Anatomy of Large Facebook Cascades*, 7th int. AAAI Conference on Weblogs and Social Media (ICWSM), 2013.

[4] D. Kempe, J. Kleinberg, É. Tardos. *Maximizing the Spread of Influence Through a Social Network*, ACM KDD'03, New York, NY, USA, pp.137–146, Aug. 2003.

[5] J. Cheng, L.A. Adamic, P.A. Dow, J. Kleinberg, J. Leskovec. *Can cascades be Predicted?*, 23rd int. conf. on World wide web (WWW), Seoul, Korea, 2014.

[6] S. Zannettou, M. Sirivianos, J. Blackburn, N. Kourtellis. *The Web of False Information: Rumors, Fake News, Hoaxes, Clickbait, and Various Other Shenanigans*, ACM J. Data and Information Quality, no.3, vol.11, July 2019.

[7] R. A. Holley, T. M. Liggett. *Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model*, Journal of the American Statistical Association, Vol.3, No.4, pp.643–663, 1975.

[8] M.H. DeGroot. *Reaching a Consensus*, Journal of the American Statistical Association, Vol.69, No.345, pp.118–121, Mar. 1974.

[9] M. Cha, H. Haddadi, F. Benevenuto, K.P. Gummadi. *Measuring user influence in Twitter: The million follower falacy*, ICWSM, pp. 10-17, Menlo Park, USA, 2010.

[10] C. Wilson, A. Sala, K. Puttaswamy, B. Zhao. *Beyond Social Graphs: User Interactions in Online Social Networks and their Implications*, TWEB, 6(4): 17:1-17:31, 2012.

[11] M.E.J. Newman. *Networks: An Introduction*, Oxford University Press, 2010.

[12] S. Brin, and L. Page. *The anatomy of a large-scale hypertextual web search engine*, Computer Networks and ISDN systems, 30(1-7), pp. 107-117, 1998.

[13] U. Kang, S. Papadimitriou, J. Sun, and H. Tong. *Centralities in Large Networks: Algorithms and Observations*, SIAM SDM 2011, Arizona, USA, pp. 119-120, 2011.

[14] https://github.com/yokaiAG/social-platform-model

[15] E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, A. Scaglione. *Binary Opinion Dynamics with Stubborn Agents*, ACM Transactions on Economics and Computation, Vol.1, No.4, Article 19, pp.19:1–19:30, Dec. 2013

[16] V. S. Varma, I.-C. Morarescu, Y. Hayel. *Continuous time opinion dynamics of agents with multi-leveled opinions and binary actions*, INFOCOM, Honolulu, USA, 2018.

[17] M. Grabisch, A. Mandel, A. Rusinowska, and E. Tanimura. *Strategic Influence in Social Networks*, Mathematics of Operations Research, 43(1):29–50, 2018.

[18] A. Silva. *Opinion Manipulation in Social Networks*, Network Games, Control, and Optimization (NETGCOOP), Springer, pp. 187–198, 2017.

[19] F. Baccelli, A. Chatterjee, S. Vishwanath. *Pairwise stochastic bounded confidence opinion dynamics: Heavy tails and stability*, INFOCOM, pp. 1831–1839, 2015.

[20] N. Spasojevic, Z. Li, A. Rao, P. Bhattacharyya. *When-To-Post on Social Networks*, ACM KDD'15, Sydney, NSW, Australia, pp.2127–2136, Aug. 2015

[21] M. R. Karimi, E. Tavakoli, M. Farajtabar, L. Song, M. Gomez Rodriguez. *Smart Broadcasting: Do You Want to Be Seen?*, ACM KDD'16, San Francisco, CA, USA, pp.1635–1644, Aug. 2016

[22] A. Reiffers-Masson, E. M. Hargreaves, E. Altman, W. Caarls, D.S. Menasché. *Timelines are Publisher-Driven Caches: Analyzing and Shaping Timeline Networks*, SIGMETRICS Perform. Eval. Rev., 44(3): 26-29, 2016.

[23] E. Hargreaves, C. Agosti, D. Menasche, G. Neglia, A. Reiffers-Masson, E. Altman. *Fairness in Online Social Network Timelines: Measurements, Models and Mechanism Design*, SIGMETRICS Perform. Eval. Rev. 46(3): 68-69, 2019.

[24] N. Gast. *Refinements of Mean Field Approximation*, Habilitation à diriger des recherches (HdR), Université Grenoble Alpes, tel-02509756 , 2020.

[25] N. Gast, B. Van Houdt. *TTL approximations of the cache replacement algorithms LRU(m) and h-LRU*, Perform. Evaluation 117: 33-57, 2017.

[26] A. Berman, R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*, SIAM Classics in Applied Mathematics; 9, 1994.

[27] L. Elsner, P. van den Driessche. *Bounds for the Perron root using max eigenvalues*, Linear Algebra and its Applications 428, pp. 2000–2005, 2008.

[28] E. A. Horn, C. A. Johnson. *Matrix Analysis*, Cambridge University Press, 1985.

[29] M. E. J. Newman. *Network Structure from Rich but Noisy Data*, Nature Physics , vol. 14, pp. 542-545, 2018.

[30] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. *Social Influence Locality for Modeling Retweeting Behaviors*, IJCAI 2013 pp. 2761-2767, 2013.

[31] P. Brémaud. *Probability Theory and Stochastic Processes*, Springer Universitext, 2020.

APPENDIX

The proof is based on the *conservation law of posts in the Newsfeed*. We will show that (7) is exact (i.e., holds without any approximation). The same methodology can be used to prove exactness for the other three equalities (8), (9) and (10).

[**Proof of Theorem 1**] Let us observe the Newsfeed of user n. This user has a set of leaders $\mathcal{L}^{(n)}$ with index $k = 1, \ldots, L$. Each leader has a posting activity, described by a homogeneous Poisson process (HPP) $N_p^{(k)}$ with rate $\lambda^{(k)}$ and a re-posting activity described by a HPP $N_r^{(k)}$ with rate $\mu^{(k)}$ respectively. The superposition of all $2L$ processes is itself a HPP N of rate $\sum_{k\in\mathcal{L}^{(n)}}(\lambda^{(k)}+\mu^{(k)})$. This is the process of total incoming posts in the Newsfeed of user n. By fixing a time interval $[0,T]$ we count $N([0,T])$ incoming posts

$$N([0,T]) \quad = \quad \sum_{k=1}^{L}\left(N_p^{(k)}([0,T]) + N_r^{(k)}([0,T])\right). \quad (22)$$

We focus on posts of origin i. The state of the Newsfeed n related to this type of posts at $t \in [0,T]$ is denoted by $X_i^{(n)}(t) \leq M$ and is itself a continuous-time process. The states of the leaders are denoted by $X_i^{(k)}(t) \leq M$, $k = 1, \ldots, L$. Furthermore, the arrival process of posts of origin i through leader k is the counting process $N_{r,i}^{(k)}$, for $k = 1, \ldots, L$. It holds $N_{r,i}^{(k)}([0,T]) \leq N_r^{(k)}([0,T])$.

• **Incoming posts of origin i to Newsfeed n:** Each leader k when re-posting will choose at random a post from his own Newsfeed. This is the *random selection* assumption. We can model the choice of user k to re-post content with label i at time $t < T$ as a Bernoulli random variable

$$Z_i^{(k)}(t) \quad = \quad \begin{cases} 1 & \text{, with probability } \frac{X_i^{(k)}(t)}{M} \\ 0 & \text{, with probability } 1 - \frac{X_i^{(k)}(t)}{M} \end{cases} \quad (23)$$

This is itself a random process, which depends on the current state of the leader's Newsfeed $X_i^{(k)}(t)$. The counting process $N_{r,i}^{(k)}$ results from thinning the re-post process $N_r^{(k)}$ of user k based on the random variable (23). Now, assume one realisation of $N_r^{(k)}[0,T]$, where user k reposted exactly $S \geq 1$ times within $[0,T]$, at time instants $t_1 < \ldots < t_s < T$. Then,

$$\mathbb{E}\left[N_{r,i}^{(k)}([0,T]) \mid t_1, \ldots, t_s\right] \quad = \quad \sum_{s=1}^{S}\mathbb{P}[Z_i^{(k)}(t_s) = 1].$$

Since $N_r^{(k)}$ is a HPP with rate $\mu^{(k)}$ the number S of re-posts within $[0,T]$ and their exact instants are random. Applying the *smoothing formula* [31, Th.7.1.7], the expected number of incoming posts of origin i through leader k up to time T is

$$\begin{aligned} \mathbb{E}\left[N_{r,i}^{(k)}([0,T])\right] &= \int_{t=0}^{T}\mathbb{P}(Z_i^{(k)}(t)=1)\mu^{(k)}dt = \\ &= \int_{t=0}^{T}\frac{X_i^{(k)}(t)}{M}\mu^{(k)}dt. \end{aligned} \quad (24)$$

For the expected total incoming number of posts of origin i in Newsfeed n, we consider re-posts of origin i through the L leaders, plus self-posts by user i, if user i is a leader of n,

$$in = \lambda^{(i)}T\mathbf{1}_{\left\{i\in\mathcal{L}^{(n)}\right\}} + \sum_{k=1}^{L}\int_{t=0}^{T}\frac{X_i^{(k)}(t)}{M}\mu^{(k)}dt, \quad (25)$$

where we used the fact $\mathbb{E}[N_p^{(i)}([0,T])] = \lambda^{(i)}T$ because it is a HPP of rate $\lambda^{(i)}$.

• **Outgoing posts of origin i from Newsfeed n:** Each of the posts of any origin entering the Newsfeed of user n (say at time t) will replace at random an existing post present on the list. This is the *random eviction* assumption. The evicted post will be of origin i with probability $X_i^{(n)}(t)/M$. We define as above the Bernoulli random variable

$$Z_i^{(n)}(t) \quad = \quad \begin{cases} 1 & \text{, with probability } \frac{X_i^{(n)}(t)}{M} \\ 0 & \text{, with probability } 1 - \frac{X_i^{(n)}(t)}{M} \end{cases} \quad (26)$$

This is itself a random process, which depends on the current state of the Newsfeed n. The outgoing process of posts of origin i from Newsfeed n results from thinning the total incoming process $N([0,T])$ shown in (22) based on (26). Similarly to (24), the expected number of posts of origin i which leave Newsfeed n up to time T is

$$\begin{aligned} out &= \sum_{k=1}^{L}\int_{t=0}^{T}\mathbb{P}(Z_i^{(n)}(t)=1)\lambda^{(k)}dt + \\ &+ \sum_{k=1}^{L}\int_{t=0}^{T}\mathbb{P}(Z_i^{(n)}(t)=1)\mu^{(k)}dt = \\ &= \int_{t=0}^{T}\frac{X_i^{(n)}(t)}{M}dt\left[\sum_{k=1}^{L}\left(\lambda^{(k)}+\mu^{(k)}\right)\right]. \end{aligned} \quad (27)$$

• **Newsfeed conservation law:** The way we modelled the Newsfeed, no post is lost and hence all incoming posts will eventually leave the Newsfeed. The conservation law for posts of origin i up to time T states that

$$X_i^{(n)}(0) + in \quad = \quad out + X_i^{(n)}(T), \quad (28)$$

where in is given in (25), out is given in (27) and $X_i^{(n)}(0)$, $X_i^{(n)}(T)$ is the count of origin i posts on the Newsfeed n at time $t = 0$ and $t = T$, respectively.

• **Taking the limit:** To derive the balance equation for Newsfeed n found in (7), we divide both sides of the conservation equality (28) by T and take the limit as $T \to \infty$. Then, $\lim_{T\to\infty} X_i^{(n)}(0)/T = 0$ and $\lim_{T\to\infty} X_i^{(n)}(T)/T = 0$. At the left-hand side of (28) we get (see (25))

$$\lambda^{(i)}\mathbf{1}_{\left\{i\in\mathcal{L}^{(n)}\right\}} + \sum_{k=1}^{L}\mu^{(k)}\lim_{T\to\infty}\frac{1}{T}\int_{t=0}^{T}\frac{X_i^{(k)}(t)}{M}dt,$$

The detailed Markovian model has finite state-space (finite users and size M of Newsfeeds), and we assume in this proof that the leader graph is strongly connected (can be relaxed). Then this system is ergodic and it holds [31, Th.7.4.18]

$$\lim_{T\to\infty}\frac{1}{T}\int_{t=0}^{T}\frac{X_i^{(k)}(t)}{M}dt \quad = \quad \mathbb{E}\left[\frac{X_i^{(k)}}{M}\right] = p_i^{(k)}. \quad (29)$$

Similarly, for the right-hand side of (28) using out in (27)

$$\lim_{T\to\infty}\frac{1}{T}\int_{t=0}^{T}\frac{X_i^{(n)}(t)}{M}dt \quad = \quad \mathbb{E}\left[\frac{X_i^{(n)}}{M}\right] = p_i^{(n)}, \quad (30)$$

which completes the proof. □