Gateway Entities in Problematic Trajectories

Xi (Leslie) Chen
Columbia University
New York City, NY, USA
xc2450@columbia.edu

Abhratanu Dutta
Northwestern University
Ames, IA, USA
abhratanu.dutta@gmail.com

Stratis Ioannidis
Northeastern University
Boston, MA, USA
ioannidis@ece.neu.edu

Sindhu Ernala
Meta
Menlo Park, CA, USA
sindhuernala@meta.com

Shankar Kalyanaraman
Meta
Menlo Park, CA, USA
kshankar@meta.com

Israel Nir
Meta
Menlo Park, CA, USA
rouli@meta.com

Udi Weinsberg
Meta
Menlo Park, CA, USA
udi@meta.com

ABSTRACT

Social media platforms like Facebook and YouTube connect people with communities that reflect their own values and experiences. People discover new communities either organically or through algorithmic recommendations based on their interests and preferences. We study online journeys users take through these communities, focusing particularly on ones that may lead to problematic outcomes. In particular, we propose and explore the concept of *gateways*, namely, entities associated with a higher likelihood of subsequent engagement with problematic content. We show, via a real-world application on Facebook groups, that a simple definition of gateway entities can be leveraged to reduce exposure to problematic content by 1% without any adverse impact on user engagement metrics. Motivated by this finding, we propose several formal definitions of gateways, via both frequentist and survival analysis methods, and evaluate their efficacy in predicting user behavior through offline experiments. Frequentist, duration-insensitive methods predict future harmful engagements with an 0.64–0.83 AUC, while survival analysis methods improve this to 0.72–0.90 AUC.

CCS CONCEPTS

• **Human-centered computing** → **Social media**.

KEYWORDS

Gateway, Survival Analysis, Recommender System

ACM Reference Format:
Xi (Leslie) Chen, Abhratanu Dutta, Stratis Ioannidis, Sindhu Ernala, Shankar Kalyanaraman, Israel Nir, and Udi Weinsberg. 2023. Gateway Entities in Problematic Trajectories. In *Proceedings of the ACM Web Conference 2023 (WWW '23), May 1–5, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3543507.3583283

WWW '23, May 1–5, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583283

1 INTRODUCTION

People increasingly turn to social media platforms such as Facebook, Instagram, TikTok, and YouTube for entertainment, communication, and information. Recently, there has been growing attention to the proliferation of misinformation on social media platforms [27, 58], raising concerns that algorithmic recommendations can lead to migrations from moderate to more extreme content [14, 30, 50, 53, 55]. Colloquially, these online journeys are sometimes referred to as rabbit-holes (an appropriation of an otherwise innocuous term used in Lewis Carroll's *Alice in Wonderland* [12]). From a platform's perspective, there is a need to understand these journeys better, especially if they lead people to problematic content through the mechanism of *preference amplification* [33].

We introduce the notion of *gateways*: these are non-problematic entities that are nevertheless associated with a higher likelihood of future engagement with problematic content. Gateways can be communities, such as Facebook Groups or Pages, public accounts on Instagram or Twitter, or creator feeds on YouTube and TikTok. By definition, gateways *are not problematic per se*; however, online interactions with them, especially when repeated over longer periods, may correlate with subsequent harmful engagements and/or a shift towards problematic content. An example includes interactions with wellness communities, that are sometimes followed with engagements with content related to conspiracy theories [5, 60].

Identifying gateways poses a fundamental conceptual challenge: by definition, they may not contain textual or other features explicitly linking them to problematic behavior. As a result, a classifier trained on manual labels to identify problematic entities [4, 26] (e.g., policy-violating content such as hate-speech, drug use, violent content, or nudity) will–and rightfully should–fail to classify gateways as problematic. Nevertheless, identifying gateways is of great importance to online platforms. This identification can allow for the adoption and deployment of mitigation strategies, that can help reduce subsequent engagement with problematic content. Mitigation strategies can range from, e.g., identifying and inhibiting pathways

leading from gateways to problematic content, particularly those enabled by algorithmic recommendations of an online platform, to more active early interventions such as steering the user towards non-problematic content, displaying warnings to discourage engagement with problematic content etc. Such interventions require the development of algorithmic, quantifiable means of identifying gateways from historical data of user interactions.

Our goal is to address this challenge by proposing and exploring different *gateway scores*, i.e., scores that capture the influence that innocuous interactions have towards downstream interactions with harmful content. We make the following contributions:

- We conduct a large-scale online experiment on Facebook Groups, demonstrating that even a simple definition of gateways, combined with a simple recommender system mitigation strategy, can yield significant dividends. Filtering gateway entities from the recommender's seeding mechanism yields a 1% statistically significant decrease in the prevalence of harmful content in the platform, without any observable loss in user engagement.
- This experiment motivates us to define and study more nuanced definitions of gateway scores. To that end, we introduce and formally define several methods to compute a gateway score that quantifies the impact that a user interaction with an entity can have towards a subsequent problematic engagement, using both frequentist methods and survival analysis models.
- We compare these methods through experiments on a 12-month dataset from Facebook Groups. Gateway scores generated through frequentist approaches can be used to predict future harmful joins with an 0.64–0.83 AUC, while scores based on survival analysis and neural network-based methods improve this to 0.72–0.90 AUC.

2 RELATED WORK

Several works study user trajectories in online platforms. Fabbri et al. [23] use a graph-based approach to mitigate radicalization pathways. Gallacher and Bright [25] examine whether one form of hate speech (e.g., Islamophobia) can lead to a different form (e.g., anti-immigrant hate speech). Hosseinmardi et al. [30] attribute political radicalization to a combination of user preferences and platform features rather than social media recommendations. In contrast, Papadamou et al. [50] find that recommendation algorithms might play an active role in steering users toward Incel-related content. Ribeiro et al. [53] observed that users migrate from milder to more extreme content on aggregate, but did not analyze individual pathways. Restrepo et al. [52] discovered that alternative health entities were the interconnecting link between mainstream parenting entities and conspiracy theory entities during 2020. However, the study differs from ours as it is based on bonds between communities rather the online behavior of individuals themselves.

To the best of our knowledge, little work has been done on identifying gateway entities through pathways users directly participate in. Our work also sheds light on how interfering with pathways to problematic content can ultimately protect users (see Sec. 3). A related concept is that of reachability of an entity, defined via the existence of a user path towards engagement, typically as a result of interacting with a recommender system [17]. However, the focus is on the existence of a path terminating at an entity, while we quantify the impact of entities the user encounters along this path.

We rely on survival analysis to define some of our gateway scores. The Kaplan-Meier estimator [35] is the most widely used non-parametric method for estimating the survival function. It is efficient to estimate but difficult to interpret and does not incorporate users' covariates. Parametric models [40, 59] assume survival times follow a well-known parametric distribution, which may introduce bias in the estimation process. The Cox hazard model [16] and its variants [10, 13, 18, 36, 64] is the most commonly used semi-parametric model. Katzman et al. [36] propose a nonlinear extension called DeepSurv and use a neural network to model the interaction between covariates. Several recent works have proposed deep learning methods to learn the hazard rate or survival function directly [29, 39, 51]. Finally, survival analysis has been successfully applied in different domains. Ren et al. [51] apply deep learning based survival method to predict the time elapsed from the last visit of one user to their next visit on a music streaming platform. Zeithaml et al. [63] and Berger and Nasr [9] use survival analysis to predict the probability of a user purchasing from a certain service supplier within t days. Barbieri et al. [8] and Yin et al. [61] model the probability that a user clicks on the advertisement within a given time. Our work adds to this growing list of applications.

3 A MOTIVATING EXAMPLE

Our working hypothesis is that identifying gateways is important, as it can lead to the design of mitigation strategies; these can range from reactive measures, e.g., attempting to steer a user away from problematic content after they visit a gateway, to proactive measures, like, e.g., altering the platform's recommendation pipeline. To validate this hypothesis, we conducted a real-world experiment over the recommendations of Facebook groups: these are public or closed communities, wherein users share content around a topic of interest. Users discover new groups either organically or through platform recommendations. We use a simple definition of gateways to alter the recommendation pipeline via a mild, conservative intervention. We observe a statistically significant decrease in propensity to engage with harmful groups, without any visible effect on user engagement. We describe this experiment in detail below.

Recommendation Pipeline. The ranking algorithms underlying group recommendations in the platform are similar in spirit to those described by Thorburn et al. [57]. Recommendations follow four distinct stages, illustrated in Fig. 1. In the first stage, *seed generation*, a relatively small (10s-100s) set of "seed" groups that should be relevant to the user is selected. These seeds are based on a range of signals, such as the user's past history of interactions, group membership, friends, etc. In the *candidate generation* stage, seeds are expanded by adding groups based on various similarity metrics w.r.t. to seeds, yielding a much larger set. In the *distillation* stage, candidates are filtered via lightweight ML models; finally, a more computationally expensive ranking and value model produces the recommendation presented to the user in the final *ranking* stage.

Integrity classifiers pre-trained on human-labelled data detect potentially harmful groups (distinct from the ranking pipeline classifiers in Fig. 1). Detection leads to enforcement, ranging from ranking penalties on harmful groups within the recommendation pipeline, to removal from the platform altogether. We refer to groups as *non-recommendable* if they remain in the platform but are not eligible

for discovery through the platform's recommendation algorithm. Note that non-recommendable groups, by definition, are detectable by the platform's integrity classifiers.

A Gateway-Based Intervention. To assess the importance and impact of gateways in this pipeline, we conduct the following experiment. We define a group to be a gateway if there is a high probability that a user who joins it subsequently joins also a non-recommendable group; this corresponds to our first frequentist gateway score (see Defn. 5.1) in Section 5. By definition, such groups *are not themselves non-recommendable*; they merely have a high chance of leading towards such a group. Using this definition, we design the following mild intervention to the recommendation pipeline. We pick the top 1 percentile of gateway groups w.r.t. this metric and filter them out of the seed generation stage. This is based on the intuition that gateways are on the "path" to non-recommendable groups, and thereby are inappropriate seeds. Notice that our mitigation strategy does not directly prevent gateway groups from being recommended. We remove them from seed generation, but they may however be reintroduced at the candidate generation stage.

Impact of Gateways. We ran this experiment on 12.3M users split evenly across control and treatment groups, and observed engagement with groups over a period of two weeks. We summarize results in Table 1. Our main finding is a drop in non-recommendable groups prevalence by as much as 0.6%. This is the proportion of total impressions of groups recommended that were subsequently deemed as non-recommendable by integrity classifiers. Similarly, we saw a 2.5% decrease in impressions of groups that needed additional enforcement (e.g., filtering, additional strikes against the group) and a 1.3% decrease in "conversions" (i.e., users wanting to join the group after seeing a recommendation) on groups that had received some form of prior enforcement. This validates the assertion that by applying our mild intervention at the seed filtering stage, we are reducing the likelihood of groups that are potentially non-recommendable from being considered as candidates. Most importantly, these improvements occurred without any significant impact on daily user engagement.

Generalizability of Gateway-Based Intervention. The group recommendation pipeline we describe in Fig 1 first generates a short list of entities from a large corpus via a light-weight preselection algorithm, and then uses a more accurate but also computationally expensive ranking algorithm to distinguish the relative importance to the user. This two-stage paradigm is used in most of large-scale industrial recommendation systems, as applying the more accurate algorithm directly to the corpus is computationally intractable [34]. Examples include Youtube's video recommendation [15, 19, 32, 34], Pinterest's related pin and newsfeed recommendation [22, 62], Linkedin's job recommendation [11], and Taobao's product recommendation [42, 44, 65, 66]. The majority of methods for generating a short list of candidates can be divided into two categories: 1. Graph-traversal methods that use previous activity of users as seeds for initiating walks on the graph [19, 22]; 2. Deep-learning based methods that learn user representations from user characteristics and activities [15, 32, 42, 44, 62, 65, 66] to distinguish among entities. The mitigation strategy presented above can be applied to both such systems. While it naturally generalizes to seed removal in graph taversal methods, for deep-learning based methods, user activities related to gateway entities could be withheld or weighed down as inputs of the deep neural network.

Metric	Treatment effect %
Impressions of actioned groups	-2.5^* [-3.7, -1.3]
Impressions of reported groups	-0.37^* [-0.64, -0.1]
Conversions to demoted groups	-1.3^* [-2.3, -0.2]
Non-rec. group prevalence	-0.61^* [-0.72, -0.52]
Daily engagement	−0.011 [-0.093, 0.071]

Table 1: Experiment results of Gateway/Seed Mitigation Strategy. We report 95% confidence intervals in parentheses. Metrics with statistically significant movement are indicated with an asterisk.

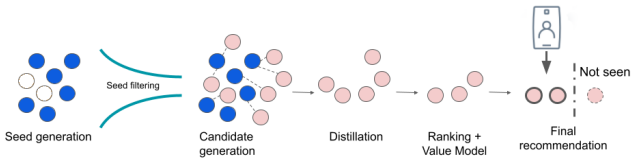


Figure 1: The group recommendation pipeline. In the *seed generation* stage, a small set of "seed" groups relevant to the user is selected. In *candidate generation*, this set is expanded into candidate groups based on similarity to seeds. In the *distillation* stage, candidates that are filtered via lightweight ML models, to be finally ranked during the *ranking* stage by a deep classifier. Our intervention acts on the first stage, filtering seeds deemed as gateways (shown in white).

4 PROBLEM STATEMENT

Motivated by the experiment above, we turn our attention to studying more nuanced, formal definitions of gateway entities. Formally, we assume access to a dataset of interactions between a set of users $\mathcal{U}$ and a set of entities $\mathcal{E}$, spanning an observation period $[0, T]$. We denote by $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{x}_e \in \mathbb{R}^{d'}$ the feature vectors of users and entities, respectively. During the observation period, users "follow" entities they are interested in. Here, entities may be Facebook groups, topic-driven channels on YouTube, personal feeds, etc. Correspondingly, "follow" maps to a range of actions, such as subscribing to a channel, following a person or joining a social group. Each user $i \in \mathcal{U}$ is associated with: (a) a *trajectory sequence* $\mathcal{S}_i = \{e_j^i\}_{j=1}^{n_i} \subseteq \mathcal{E}$ of entities that i followed during the observation period, and (b) a corresponding *time sequence* $\mathcal{T}_i = \{t_j^i\}_{j=1}^{n_i} \subset [0, T]$ at which these "follow" events took place. For simplicity, we assume that an entity e appears in $\mathcal{S}_i$ at most once. Note that we do not consider "follows" that happened (or will happen) outside of the observation period $[0, T]$. There exists a subset of *target entities* $\mathcal{B} \subset \mathcal{E}$, which are known to be problematic. Trajectories $\mathcal{S}_i$, $i \in \mathcal{U}$, may contain 0, 1, or more targets from the target set $\mathcal{B}$.

For every entity $e \in \mathcal{E}^0 \equiv \mathcal{E} \setminus \mathcal{B}$ we would like to assign a score $s_e \in \mathbb{R}_+$ that quantifies the association of following e with *subsequent* follows of targets in $\mathcal{B}$. Sorting entities by that score, and selecting the top k ranking entities, will give us the set of *gateway entities* with respect to the topic or activities associated with $\mathcal{B}$. As illustrated by our motivating example in Section 3, identifying such gateways can then used to design mitigation strategies.

$\mathcal{U}$	set of users
$\mathcal{E}$	set of entities
T	length of observation period
$\mathcal{B}$	target entities
i	user in $\mathcal{U}$
e	entity in $\mathcal{E}$
$\mathcal{S}_i$	trajectory sequence of entities followed by user i
$\mathcal{T}_i$	time sequence "follow" events of user i
$\mathbf{x}_i$	d-dimensional feature vector of user i
n_i	number of entities followed by user i
$\mathcal{E}^0$	candidate gateway set, i.e., $\mathcal{E} \setminus \mathcal{B}$.
$\mathcal{U}_e$	set of users that followed entity e
$\mathcal{U}_\mathcal{B}$	set of users that followed at least one target in $\mathcal{B}$
$\mathcal{E}_i^{e+}$	set of entities that i followed after following e
$\mathcal{E}_i^{\mathcal{B}+}$	set of entities that i followed after following any of the targets in $\mathcal{B}$
$\mathcal{B}_i^{e+}$	set of targets i followed after following e

Table 2: Notation Summary

5 METHODOLOGY

We discuss two different approaches for constructing scores that measure the contribution of a gateway entity to a subsequent target "follow": (a) duration-insensitive scores, that focus only on the order of temporal events and ignore the length of an interval until a "follow" happens, and (b) survival analysis scores, that also take the time between follow events into account.

5.1 Duration-Insensitive Scores

A Frequentist Approach. The simplest way to estimate how likely a user in the candidate gateway $e \in \mathcal{E}^0$ will subsequently follow an entity in $\mathcal{B}$ is via a frequentist approach. We define several frequentist variants below. For a given entity $e \in \mathcal{E}$, let $\mathcal{U}_e \equiv \{i \in \mathcal{U} \text{ s.t. } e \in \mathcal{S}_i\}$ be the users that follow e. For every user $i \in \mathcal{U}_e$, let $\mathcal{E}_i^{e+}$ be the set of entities that user i followed *after* following e, i.e.:

$$\mathcal{E}_i^{e+} \equiv \left\{ e_j^i \in \mathcal{S}_i \text{ s.t. } t_j^i > t_{je}^i \right\}, \tag{1}$$

where t_{je}^i is the time at which i followed e. Also, let $\mathcal{B}_i^{e+} \equiv \mathcal{E}_i^{e+} \cap \mathcal{B}$ be the targets that i followed after following e.

Our duration-insensitive, frequentist scores are as follows:

Definition 5.1. Our first frequentist score, also used in Section 3, is the empirical estimate of the probability a user follows at least one $b \in \mathcal{B}$ conditioned on following $e \in \mathcal{E}^0$. That is:

$$s_e^{\mathsf{F1}} \equiv \frac{1}{|\mathcal{U}_e|} \sum_{i \in \mathcal{U}_e} \mathbb{1}_{\mathcal{B}_i^{e+} \neq \emptyset}. \tag{2}$$

Definition 5.2. Our second frequentist score is the empirical estimate of the conditional probability that, given that a "follow" event occurs after a user follows $e \in \mathcal{E} \setminus \mathcal{B}$, there is a "follow" of a target in $\mathcal{B}$. Formally,

$$s_e^{\mathsf{F2}} \equiv \sum_{i \in \mathcal{U}_e} |\mathcal{B}_i^{e+}| / \sum_{i \in \mathcal{U}_e} |\mathcal{E}_i^{e+}|. \tag{3}$$

Definition 5.3. Our third score is the empirical estimate of the conditional probability on a per user basis, averaged across $\mathcal{U}_e$, i.e.,

$$s_e^{\mathsf{F3}} \equiv \frac{1}{|\mathcal{U}_e|} \sum_{i \in \mathcal{U}_e} \left(|\mathcal{B}_i^{e+}| / |\mathcal{E}_i^{e+}| \right). \tag{4}$$

Figure 2 illustrates the differences between each of these frequentist definitions (5.1-5.3).

Reducing Variance Using Beta-Binomial Fitting. One possible shortcoming of scores s_e^{F3} given by (4) is that group members with very small number of subsequent joins $|\mathcal{E}_i^{e+}|$ in the denominator will have very high variance, which in turn will increase the variance of the estimation of s_e^{F3}. We take an empirical Bayes approach of fitting a beta-binomial distribution to alleviate this issue. We describe this in detail in Appendix A.

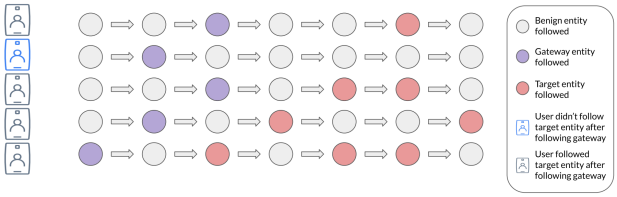


Figure 2: A schematic representation of follow events between a gateway entity e and target set $\mathcal{B}$. A purple circle denotes the entity e for which we are computing the gateway scores while the red circle denotes any of the target entities belonging to the set $\mathcal{B}$. Under Definition 5.1, the gateway score is simply the fraction of users who followed an entity in $\mathcal{B}$ after following e, $4/5 = 0.8$. Using Definition 5.2, we compute the score to be the fraction of follows *after* following e that culminated in $\mathcal{B}$: $8/24 = 0.33$, and with Definition 5.3, this would be an average per-user fraction of follows culminating in $\mathcal{B}$: $1/5 \cdot (1/4 + 0 + 2/4 + 2/5 + 3/6) = 0.33$.

5.2 Survival Analysis Scores

Notice that frequentist scores ignore the time interval between an gateway and a target follow into account. In this section, we use *survival analysis* [1, 48] to produce scores s_e that take into account not only the probability that a user follows a target entity, but also the time elapsed until they do so.

Let $\mathcal{U}_\mathcal{B} \equiv \bigcup_{b \in \mathcal{B}} \mathcal{U}_b$ be the set of users who followed any of the target entities. Hence, $\mathcal{U}_e \cap \mathcal{U}_\mathcal{B}$ is the set of users who have followed both the gateway entity e and any of the target entities. For every user $i \in \mathcal{U}_\mathcal{B}$, let

$$t_\mathcal{B}^i \equiv \min\{t_j^i \text{ s.t. } s_j^i \in \mathcal{B}\}, \tag{5}$$

be the first time i follows a target in $\mathcal{B}$. Then, given a candidate gateway $e \in \mathcal{E} \setminus \mathcal{B}$, we define the *survival time* of $i \in \mathcal{U}_e \cap \mathcal{U}_\mathcal{B}$ as:

$$\tau_e^i \equiv t_\mathcal{B}^i - t_e^i. \tag{6}$$

We assume that survival times after following e are independent across users and distributed according to a random variable $X_e \in \mathbb{R}_+$, whose density $f_{X_e} : \mathbb{R}_+ \to \mathbb{R}_+$ is parameterized by some $\boldsymbol{\xi}_e \in \mathbb{R}^k$;

We denote the *survival function* of r.v. X_e as $S_{X_e}(t)$ and the hazard function as $h_{X_e}(t)$ (Appendix B). We now introduce our scores motivated from survival analysis.

Parametric Survival Model. We assume that the survival time after following e has one of the following three parametric distributions: $\texttt{LogNormal}(\mu_e, \sigma_e^2)$, $\texttt{Exponential}(\lambda_e)$, or $\texttt{Weibull}(\lambda_e, \rho_e)$; for example for $\texttt{LogNormal}$, parameters to be learned are $\boldsymbol{\xi}_e \equiv \mu_e, \sigma_e^2$). We learn parameters $\boldsymbol{\xi}_e$ via maximum likelihood estimation (MLE) [1, 48] (Appendix B). After estimating parameters, we compute scores as follows:

Definition 5.4. We define the per-entity log-normal, exponential, and Weibull scores (s_e^{LN}, s_e^{EXP}, and s_e^{WB}, respectively) to be

$$s_e^{\text{dist}} = 1 - S_{X_e}(T), \quad \text{for dist} \in \{\mathsf{LN}, \mathsf{EXP}, \mathsf{WB}\}. \tag{7}$$

I.e., this is the target join probability at the entire, duration of the dataset, as specified by the corresponding parametric distribution.

Joint Parametric Survival Model. One drawback of the above survival analysis scores (and frequentist models as well) is that per-entity parameters are learned *in isolation*. In determining an entity's influence, the above approach *ignores all other entities the user may have also interacted with*. This can have distorting effects: the influence of an entity may be overestimated when it co-occurs with a highly influential entity. We illustrate this with an example in Appendix C.2.

To account for this, we propose a model that accounts for joint occurrence among entities. We make the following probabilistic assumptions on how target follows occur. Whenever a user follows an entity $e \in \mathcal{E}^0$, this triggers the following process. First, a "clock" X_e starts, again sampled from a well-known parametric distribution, parameterized by $\boldsymbol{\xi}_e \in \mathbb{R}^k$. Once the clock expires, the user follows the target. Finally, if a user interacts with multiple entities, *they all generate independent clocks*; the target event then happens whenever *the first of these clocks expires*. The latter assumption is the main departure from standard survival analysis, and has two consequences: first, it induces a *coupling* or *joint effect* of entities joined prior to following the target. Put differently, it spreads the attribution of a target follow to past joins of a user, in a principled fashion. Second, the "independent clocks" assumption has the following intuitive technical advantage:

Theorem 5.5. *Assume that clocks $\{X_e\}_{e\in\mathcal{E}^0}$ are continuous random variables. Let τ^i be the survival time of a user i that has so far followed entities $\mathcal{E}' \subseteq \mathcal{E}^0$. Then the hazard function of τ^i is the sum of the hazard functions h_{X_e}, $e \in \mathcal{E}'$, and its survival function is the product of the survival functions S_{X_e}, $e \in \mathcal{E}'$, respectively.*

We prove this theorem in Appendix C.3. An immediate consequence is that parameters $\boldsymbol{\xi}_e$ can be learned again by maximizing a likelihood of the form in Eq. (14), using however the corresponding density and survival functions of τ^i, as specified by Theorem 5.5. After parameters $\boldsymbol{\xi}_e$ have been computed this way, they can subsequently be used to compute per entity scores exactly as in Defn. 14. This estimation *couples* the learning across entities and, contrary to Eq. (14), MLE is *no longer separable across parameters* $\boldsymbol{\xi}_e$, $e \in \mathcal{E}_0$, that need to be trained jointly (see Appendix C.4).

DeepSurv Model. The final score we use is based on DeepSurv [36], which itself is based on the Cox proportional hazard model. Applying the latter to our setting, the hazard function characterizing the survival time of a user i after following e takes the form:

$$h_e^i(t) = \lambda_0(t) r_e^i \tag{8}$$

where λ_0 is a baseline function and r_e^i is a parametric function of user and entity features. More specifically, DeepSurv uses $r_e^i = f(\mathbf{x}_i, \mathbf{x}_e; \mathbf{w}) \geq 0$, where $\mathbf{x}_i$, $\mathbf{x}_e$ are the feature vectors of the user and the entity, respectively, and $f(\cdot; \mathbf{w})$ is a neural network parameterized by w. DeepSurv trains the neural network by maximizing the Cox partial likelihood function, while the baseline function is determined via a Kaplan-Meier estimate. After training this deep model, we compute per-entity scores using the following definition:

Definition 5.6. We define s_e^{DS} to be the average of quantities r_e^i across $\mathcal{U}_e$, i.e., $s_e^{\mathsf{DS}} = \frac{1}{|\mathcal{U}_e|} \sum_{i\in\mathcal{U}_e} r_e^i$.

5.3 Directionality

An additional property that may help evaluate different methods for identifying gateway entities is "directionality". This measures, for an entity e, the fraction of all users who followed it and one of the targets, those that followed e first. Formally, let us denote $\mathcal{E}_i^{\mathcal{B}+}$ to be the set of entities user i followed after following any of the entities in $\mathcal{B}$. That is:

$$\mathcal{E}_i^{\mathcal{B}+} \equiv \left\{ e_j^i \in \mathcal{S}_i \text{ s.t. } t_j^i > t_{\mathcal{B}}^i \right\}, \text{ where } t_{\mathcal{B}}^i \text{ is given by Eq. (5).} \tag{9}$$

We define the directionality of an entity e as:

$$d_e \equiv 1 - \frac{1}{|\mathcal{U}_e \cap \mathcal{U}_{\mathcal{B}}|} \sum_{i\in\mathcal{U}_e\cap\mathcal{U}_{\mathcal{B}}} \mathbb{1}_{e\in\mathcal{E}_i^{\mathcal{B}+}}. \tag{10}$$

Intuitively, $d_e = 0$ indicates that all users in $i_{\mathcal{B}}$ that "followed" e did so after following a target, while $d_e = 1$ indicates that all users in $\mathcal{U}_{\mathcal{B}}$ that "followed" e did so before following any target. This is an imperfect proxy and does not capture causality, which is best estimated through online experiments such as the one described in Sec. 3. Furthermore, it can be sensitive to the observation period or the nature of target entities: for example, it ignores that a target entity may not have been so at the time a user followed it.

6 EVALUATION

6.1 Experiment Setup

We present an offline evaluation of our gateway entity scores conducted using Facebook Groups.

6.1.1 Dataset.

We identify three different types of public target sets H1, H2, and BF, two of which correspond to problematic content, and the last one corresponding to benign content. For each of the three target sets, we focus on a 6-month observation period (from 2021-07-01 to 2021-12-31) for training. We randomly select 1000 candidate gateway groups with more than 100 new members during the 6-month observation period. For each candidate gateway group, we monitor the trajectories of at most 5000 new users. Our test set is constructed over a separate 6-month observation period (from 2022-01-01 to 2022-07-01), immediately succeeding the training period. We monitor the same 1000 candidate gateway groups in this period and randomly select at most 5000 new users in each group. Then we measure how well scores computed on the training set can predict whether "follows" at a gateway also result to "follows" of a target group in the test set (see section *Performance Metrics* below). We extract user and group embeddings using PyTorch-BigGraph [41]. We provide more details of the datasets in Appendix D.

6.1.2 Evaluation Methodology & Performance Metrics.

Algorithms. We implement all three frequentist scores (s_e^{F1}, s_e^{F2}, s_e^{F3}); for the last score (s_e^{F3}), we also perform variance reduction via the Bayesian approach presented in Section 5.3 (denoted by s_e^{F3} (B)). We also compute the four survival analysis scores (s_e^{LN}, s_e^{EXP}, s_e^{WB} s_e^{DS}) as well as the directionality (d_e) of each candidate

	Test s_e^{F1}			Test s_e^{F2}		
	H1	H2	BF	H1	H2	BF
s_e^{F1}	0.3798	0.5837	0.7363	0.3491	0.4847	0.7131
s_e^{F2}	0.1631	0.5008	0.7088	0.3357	**0.6608**	**0.7960**
s_e^{F3}	0.1456	0.4912	0.6972	0.3103	0.6067	0.7679
s_e^{F3} (B)	0.1538	0.5214	0.7053	0.3223	0.6426	0.7807
s_e^{LN}	0.3890	0.6160	0.7642	0.3637	0.4891	0.7290
s_e^{EXP}	**0.4207**	**0.6514**	0.7586	**0.3892**	0.5085	0.7241
s_e^{WB}	0.3925	0.6226	**0.7671**	0.3709	0.4934	0.7303
s_e^{DS}	0.3666	0.6399	0.7104	0.3684	0.4788	0.6257

Table 3: Spearman correlation between training and test scores on H1, H2, and BF datasets. The exponential model performs the best with respect to Test s_e^{F1} on H1 and H2 datasets. s_e^{F2} performs the best with respect to Test s_e^{F2} on H2 and BF datasets.

	Train			Test		
	H1	H2	BF	H1	H2	BF
s_e^{F1}	0.7918	0.7270	0.8218	0.7758	0.6534	0.8287
s_e^{F2}	0.7782	0.6709	0.7994	0.7653	0.6367	0.8162
s_e^{F3}	0.7825	0.6712	0.8010	0.7694	0.6376	0.8139
s_e^{F3} (B)	0.7829	0.6789	0.8040	0.7727	0.6416	0.8163
$\mathsf{PD}_{ie}^{\mathsf{LN}}$	0.8209	0.7605	0.8438	0.8142	0.7191	0.8537
$\mathsf{PD}_{ie}^{\mathsf{EXP}}$	0.8246	0.7613	0.8443	0.8117	0.7226	0.8536
$\mathsf{PD}_{ie}^{\mathsf{WB}}$	0.8217	0.7615	0.8442	0.8155	0.7221	0.8539
$\mathsf{PD}_{ie}^{\mathsf{DS}}$	**0.8916**	**0.8468**	**0.9302**	**0.8463**	**0.7840**	**0.9029**

Table 4: ROC-AUC score on H1, H2, and BF datasets. The DeepSurv model performs the best across all three datasets on Train and Test sets.

gateway entity. We use the lifelines package [20] to compute the log-normal, exponential, and Weibull fits, and the PySurvival package [24] to compute DeepSurv scores. Additional details regarding the DeepSurv NN architecture and hyperparameters are in Appendix B.

Performance Metrics We evaluate the gateway scores on the following two tasks:

Task 1. For each candidate gateway entity, we measure s_e^{F1} and s_e^{F2} *on the test set*: these correspond to the *death/non-survival probability* (i.e., the empirical probability that a user that followed a candidate gateway e followed a target in $\mathcal{B}$ during the test period), and the conditional "follow" probability (i.e., the conditional probability that a "follow" in the test period was to a target). We then evaluate how gateway scores we computed over the training set correlate to these two test metrics. We measure this through Spearman correlation.

Task 2. For each candidate gateway entity, we use the scores computed on the training set to predict whether a user in the test set will join a target entity within the observation period. In particular, for every user in the test set, we observe all join events in $\mathcal{E}_0$ that occur during the test observation period, excluding target joins. We then use this information, as well as the scores extracted from the training set, to predict whether the user indeed joins a target group during the observation period.

For frequentist scores (s_e^{F1}, s_e^{F2}, s_e^{F3}), we construct a simple classifier per user in the test set by summing up all the scores of entities in $\mathcal{E}_0$. Note that this ignores the time at which joins happen. For survival analysis scores, we exploit the underlying generative model to makes predictions that also take into account the time interval between the gateway join time and the end of the observation period. For each score, we define a *probability of death* (PD) as:

$$\mathsf{PD}_{ei}^{\text{dist}} = 1 - S_{X_e}(T - t_e^i), \quad \text{for dist} \in \{\mathsf{LN}, \mathsf{EXP}, \mathsf{WB}, \mathsf{DS}\}, \tag{11}$$

which takes into account t_e^i the time a user followed e. For parametric models, this quantity depends on $\boldsymbol{\xi}_e$; for DeepSurv, this can be computed directly from (8). Finally, as for frequentist scores, we sum these probabilities across entities in $\mathcal{E}_0$ the user followed, and use this quantity to predict target joins.[1] In both frequentist and survival analysis cases, we evaluate the performance of each method using Area Under the Receiver Operating Characteristics Curve (ROC-AUC).

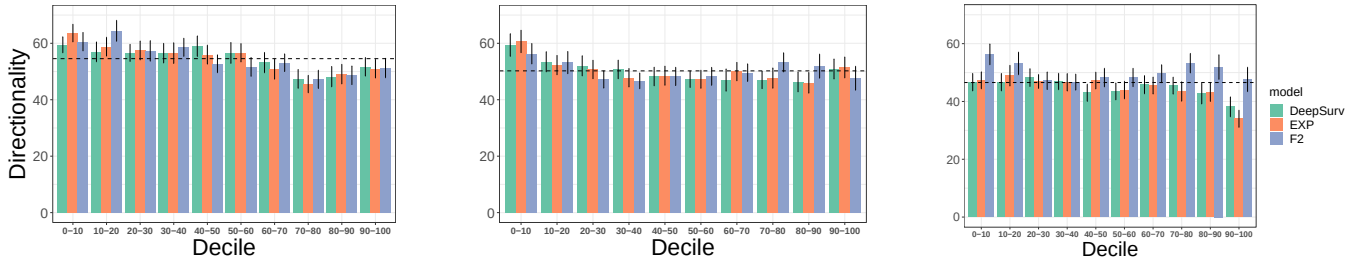
6.2 Score Comparison

Anticipating Future Behavior. Table 3 shows the performance of each algorithm across three datasets with respect to the Spearman correlation with the two test scores (s_e^{F1} and s_e^{F2}). Survival models perform better with respect to Test s_e^{F1} across all datasets. Exponential model performs the best among survival models. Spearman correlation with Test s_e^{F2} however is optimized by s_e^{F2} in two datasets (H1 and BF). This is not surprising, as other methods largely ignore non-target event frequencies, which are taken into account in this metric. Finally, we observe that the Bayesian version of s_e^{F3} (namely, s_e^{F3} (B)) always outperforms the non-Bayesian version, indicating that variance reduction indeed helps.

Table 4 shows the ROC-AUC score of each algorithm with respect to predicting whether a user i in a candidate gateway group e will join a target group within the observation period. Parametric survival models and the DeepSurv model perform better than frequentist scores on the train and test set across all three target datasets. This might be because the frequentist scores are computed over all users in the dataset and do not take durations into consideration. Instead of assigning a uniform score for all users, the survival models make a personalized prediction based on the join time of each user. Not surprisingly, the DeepSurv model performs better than parametric survival models across all target datasets. The exceptional performance of DeepSurv might be because it uses user features in training and learns to distinguish users at risk based on their embedding. In conclusion, DeepSurv works the best at predicting whether a new user will join one of the target groups within the observation period. The Exponential survival model and frequentist method F2 perform the best with respect to ranking groups by the likelihood of group members joining a target group.

Directionality. An ideal evaluation of our techniques would be against a golden set of ground truth labels acquired manually about whether an entity is truly a gateway to the target set $\mathcal{B}$. This however is infeasible for a few reasons. Firstly, the notion of "gatewayness" is somewhat subjective and this can introduce noise in the human labels. Secondly, an entity cannot immediately be determined to be a gateway just on the basis of the content it carries or the people following it. Although directionality as described in Section 5.3 is imperfect as a proxy and may still not sufficiently capture this "gatewayness", we nonetheless use it to evaluate the different methods for the sake of completeness.

[1] We explore alternative ways of combining scores in Sec. 6.3



(a) BF (b) H1 (c) H2

Figure 3: Barplots for directionality across gateway score buckets for BF, H1, H2 datasets. We bucket groups according to their respective gateway score decile. Error bars indicate 95% confidence intervals and dotted lines denote average directionality

	PD	MaxD	LD	PD^{LN}	PD^{EXP}	PD^{WB}	PD^{LNJ}	PD^{EXPJ}	PD^{WBJ}
H1	0.8496	0.8290	0.7235	0.8563	0.8558	0.8564	0.8762	**0.8771**	**0.8771**
BF	0.8496	0.8471	0.8145	0.8645	0.8635	0.8647	0.8701	0.8700	**0.8705**

Table 5: Predictive performance, w.r.t. AUC, of joint parametric survival models (LNJ, EXPJ, WBJ) against other baselines, over the H1 and BF datasets. Comparing these results to Table 4, we observe that assuming independence at test time outperforms simple PD methods. Moreover, joint training (shown in the last three columns) outperforms individual training through MLE (14) (shown in the middle three columns). Lastly, though joint training outperforms DeepSurv on H1, it does not on BF; LD also performs much better on BF. These two observations seem to indicate that joint training is not as important on BF, an observation that is consistent also with the length of trajectories in Appendix Fig. 6.

Figures 3a, 3b, and 3c show the directionality scores across the three types (BF, H1, H2). We bucket groups according to their score percentiles according to each best performing method with respect to Task 1 and 2 (s_e^{DS}, s_e^{EXP}, s_e^{F2}). We observe the mean to be around 0.5 across three datasets, indicating roughly half of the users joined the gateway group before joining a target group: one possible explanation is that gateways joined after the target are targets themselves, that have not been classified as such. We also observe the directionality scores for the three methods (s_e^{DS}, s_e^{EXP}, s_e^{F2}) are relatively constant across deciles and comparable to the mean, whereas the directionality scores of F2 is slightly higher on the H2 dataset. Except for the H1 dataset, gateway groups in the 90% percentile of F2 maintain a directionality score close to or above the average. Notwithstanding problems with directionality as a metric, the fact that these methods do not perform much better than average indicates a lot more room for improvement in the methodology used to identify gateways, potentially using more content and graph-based features.

6.3 Effect of Joint Parametrization

We also explore the effect of joint vs. isolated parameter estimation in survival analysis parametric models. We focus on H1 and BF: in H2, no user joined more than 3 entities; in contrast, user trajectories lengths $|\mathcal{S}_i|$ spanned far more values in H1 and BF (see Fig. 6). This creates the opportunity of improving learned parameters via joint estimation in these two datasets.

To test this hypothesis, we train per-entity parameters $\xi_e, e \in \mathcal{E}_0$, through the joint parametric model, as described in Sec. 5.2. We then use these to compute the probability of death (**PD**) on the test set for each of the three parametric distributions (LNJ, EXPJ, WBJ), assuming the independent clocks model: we describe this formally in Appendix C.1. We use this to predict again whether a user in the test set joined a target group. We also report the same prediction but with parameters ξ_e learned in isolation (LN, EXP, WB), as in previous sections: this is effectively the "independent clocks" model at test time, but with parameters ξ estimated in an isolated fashion via MLE (14). As additional baselines, we also use the following scores as classifiers; all three are based on s_e^{F1} estimated over the training set. *Probability of Death* **(PD)**: This is given by the product of $1 - s_e^{F1}$ across all $e \in \mathcal{E}_0$ the user interacted with. Note that, like the "independent clocks" model, this assumes independence across entities at test time, but ignores join times. *Maximum Death* **(MaxD)**: This is the maximum s_e^{F1} across all $e \in \mathcal{E}_0$ the user interacted with; it assumes only the "most influential" entity matters. *Last Death* **(LD)**: This is $1 - s_e^{F1}$ for the very last entity e the user interacted with during the observation period.

The quality of prediction w.r.t. ROC-AUC is indicated in Table. 5. Comparing this to Table 4, we observe that estimating PD under the independent clocks model improves predictive power in general, even for the simple PD method based on s_e^{F1}. Both this and remaining PD scores assume independence at test time; this indicates that the simplifying independence assumption does not introduce significant bias on these two datasets.

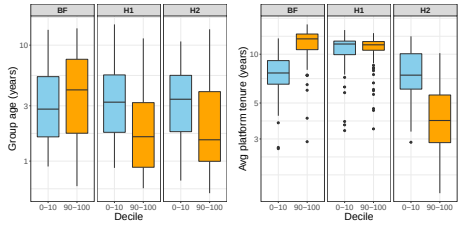
Moreover, we observe that joint training (indicated by suffix J) always outperforms individual estimation through Eq. (14). In H1, joint estimation surpasses the predictive quality of DeepSurv scores (c.f. Table 4), but not on BF. **LD** is worst-performing, indicating that the entire trajectory, rather than the most recent join event, has a role to play in a user joining a target; however, **LD** is somewhat better in BF. These two observations are consistent with each other: both indicate that joint information from the entire trajectory is not as important in BF as in H1; this also agrees with the fact that trajectory lengths are shorter in BF (see Appendix Fig. 6).

6.4 Qualitative Analysis

We also perform a qualitative analysis on a set of Breastfeeding target groups. In this case the trajectory is obviously not harmful, and the intuition is that "gateways" to breastfeeding groups should be groups related to pregnancy. Table 6 shows the top-5 ranked English-speaking groups using the best performing individual scores with respect to Task 1 and 2 ($s_e^{F1}, s_e^{F2} s_e^{F3}, s_e^{DS}, s_e^{EXP}$). We

Rank	s_e^{F1}	s_e^{F2}	s_e^{F3} (B)	s_e^{DS}	s_e^{EXP}
1	**Exclusively Pumping Moms**	**Babywearing Doctors UK**	**Exclusively Pumping Moms**	**Exclusively Pumping Moms**	**Exclusively Pumping Moms**
2	**Babywearing Doctors UK**	**Exclusively Pumping Moms**	**Babywearing Doctors UK**	Due in September 2021 UK	**Babywearing Doctors UK**
3	Due in September 2021 UK	BabyBuddha Community	Due in September 2021 UK	Exclusive Pumping Moms Canada	Exclusive Pumping Moms Canada
4	Positive Pregnancy & Birth Australia	Exclusive Pumping Moms Canada	Exclusive Pumping Moms Canada	**Babywearing Doctors UK**	How to get pregnant faster and easy.
5	Exclusive Pumping Moms Canada	Breast Bottle & Beyond -Support Group	BabyBuddha Community	Positive Pregnancy & Birth Australia	Positive Pregnancy & Birth Australia

Table 6: Top gateway groups for Breastfeeding. We highlight two of the the overlapping groups across methods in black and blue.



(a) Group age **(b) Avg platform tenure**

Figure 4: Comparing top vs. bottom gateway groups

observe that gateway models work as expected, and the resulting gateway groups are indeed closely related to breastfeeding: some of them are pregnancy support groups (which directly precedes breastfeeding), while some are breast-pump brands and breastfeeding support groups. We also observe a significant overlap between the resulting groups across methods. This extends to lower ranked groups, and across different topics.

In Figure 4, we compare some characteristics between top and bottom decile gateway groups as ranked by our methods. For simplicity, we show plots only for ranking gateways based on fitting an exponential survival model (EXP) although the plots are similar for other methods. Figure 4a shows the average age of groups (i.e., time since creation of the group itself) for the three datasets; while in the case of H1 and H2, top gateways tend to be newer, top gateways for BF are relatively older than bottom groups, suggesting that over time gateways to harm can potentially devolve to being problematic themselves and enforced on while gateways to benign topics are likely to flourish. When looking at the average tenure on the platform for users that join these gateway groups (Figure 4b), those in top gateways for H2 are newer than their counterparts in bottom gateways, whereas for BF users belonging to top gateways have more tenure on the platform. This is representative of the fact that gateways for Breastfeeding naturally cater to people interested in parenting and child-rearing, and hence are likely to index higher on age (and tenure, correspondingly).

7 ETHICAL CONSIDERATIONS

The use of gateways to alter the recommendation pipeline, as described in Sec. 3, represents can be viewed as a content moderation strategy; as with any such strategy, there is the risk of limiting the reach for non-harmful content [2, 49]. Gillespie [28] argues that content moderation systems should take into account privacy, legal and ethical considerations in balancing speech and interests of various groups.

Although we observed no significant daily engagement drop in our experiments in Sec. 3, interventions such as the one we proposed may lead to less accurate predictions and, in turn, to opportunity costs by not providing valuable information to users. Regular monitoring that measures how much such a system reduces the spread of harmful content can ensure that its outcomes align with its aims. The results should also be compared to any negative externalities. This may include reduced voice [56] and access to valuable information [43]. For example, the impressions of critical information, such as medical resources and news, should be monitored to ensure that morally responsible distribution and access is not hampered by such content moderation strategies.

Algorithmic enforcement on content and freedom of expression is a current topic of research [6, 7, 38]. As Gillespie [28] writes "moderation is not an ancillary aspect of what platforms do. It is essential, constitutional, and definitional. Not only can platforms not survive without moderation, they are not platforms without it." So it is not a question of whether content moderation should happen, but how best to do so while considering the free speech rights and interests of users and societies. Keller [37] distinguishes freedom of speech from right to amplification and argues that "private companies have no obligation to host their users' speech, or to provide it with additional reach via amplification." Algorithmic amplification is often an artifact of how recommender systems are designed [21, 45]. When implementing such systems, an effort must be made to understand and control for such side effects. The experiment in Sec. 3 was an attempt to limit the exacerbating effects of algorithmic amplification without taking down the responsible content completely.

8 CONCLUSION AND DISCUSSION

We studied several quantitative scores assessing the impact of user interactions with online entities towards subsequent "follows" of potentially problematic content. Then we quantified their performance over real-life datasets as well as through a deployment in Groups Recommendation; the latter led to a significant decrease in non-recommendable group prevalence [3, 46].

Proposing new scores, as well as additional experiments to evaluate "gateway" quality, is an interesting future direction; so is identifying expert labeling processes to assess ground truth in this setting. From a computational perspective, scaling parametric inference of more complex scores (such as survival analysis or scores regressed via deep neural networks), remains also an interesting direction to explore. Finally, capturing both formally and experimentally joint effects, such as the collective impact of exposure to sequences of gateway entities, is also an open problem.

In identifying gateways systematically, our work is a first step in leveraging network signals for content moderation and is not a blueprint for devising a complete moderation strategy. Understanding how gateway scores could be used to that end where they might yield the most benefits in a pipeline, while not hampering access or introducing undue censoring, is an open area of investigation.

REFERENCES

[1] Odd Aalen, Ornulf Borgan, and Hakon Gjessing. 2008. *Survival and event history analysis: a process point of view.* Springer Science & Business Media.

[2] Ahmed Abouzeid, Ole-Christoffer Granmo, Christian Webersik, and Morten Goodwin. 2022. Socially fair mitigation of misinformation on social networks via constraint stochastic optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11801–11809.

[3] Tom Alison. 2021. Changes to Keep Facebook Groups Safe. https://about.fb.com/news/2021/03/changes-to-keep-facebook-groups-safe/

[4] Lora Aroyo, Lucas Dixon, Nithum Thain, Olivia Redfield, and Rachel Rosen. 2019. Crowdsourcing subjective tasks: the case study of understanding toxicity in online discussions. In *Companion proceedings of the 2019 world wide web conference*. 1100–1105.

[5] Stephanie Alice Baker. 2022. Wellness as a Gateway to Misinformation, Disinformation and Conspiracy. In *Wellness Culture*. Emerald Group Publishing Limited, 115–151.

[6] Jack M Balkin. 2017. Free speech in the algorithmic society: Big data, private governance, and new school speech regulation. *UCDL rev.* 51 (2017), 1149.

[7] Jack M Balkin. 2018. Free speech is a triangle. *Colum. L. Rev.* 118 (2018), 2011.

[8] Nicola Barbieri, Fabrizio Silvestri, and Mounia Lalmas. 2016. Improving Post-Click User Engagement on Native Ads via Survival Analysis. In *Proceedings of the 25th International Conference on World Wide Web* (Montréal, Québec, Canada) *(WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 761–770. https://doi.org/10.1145/2872427.2883092

[9] Paul D Berger and Nada I Nasr. 1998. Customer lifetime value: Marketing models and applications. *Journal of interactive marketing* 12, 1 (1998), 17–30.

[10] Harald Binder and Martin Schumacher. 2008. Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models. *BMC bioinformatics* 9, 1 (2008), 1–10.

[11] Fedor Borisyuk, Krishnaram Kenthapadi, David Stein, and Bo Zhao. 2016. CaSMoS: A framework for learning candidate selection models over structured queries and documents. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 441–450.

[12] Lewis Carroll. 1865. *Alice's Adventures in Wonderland.* Macmillan.

[13] Travers Ching, Xun Zhu, and Lana X Garmire. 2018. Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data. *PLoS computational biology* 14, 4 (2018), e1006076.

[14] Alessandro Cossard, Gianmarco De Francisci Morales, Kyriaki Kalimeri, Yelena Mejova, Daniela Paolotti, and Michele Starnini. 2020. Falling into the echo chamber: the Italian vaccination debate on Twitter. In *Proceedings of the International AAAI conference on web and social media*, Vol. 14. 130–140.

[15] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[16] David R Cox. 1972. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* 34, 2 (1972), 187–202.

[17] Mihaela Curmei, Sarah Dean, and Benjamin Recht. 2021. Quantifying Availability and Discovery in Recommender Systems via Stochastic Reachability. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 2265–2275. https://proceedings.mlr.press/v139/curmei21a.html

[18] G Kleinbaum David and Klein Mitchel. 2012. *Survival Analysis: A Self-Learning Text.* Spinger.

[19] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*. 293–296.

[20] Cameron Davidson-Pilon. 2019. lifelines: survival analysis in Python. *Journal of Open Source Software* 4, 40 (2019), 1317. https://doi.org/10.21105/joss.01317

[21] Sarah Dean and Jamie Morgenstern. 2022. Preference Dynamics Under Personalized Recommendations. In *Proceedings of the 23rd ACM Conference on Economics and Computation* (Boulder, CO, USA) *(EC '22)*. Association for Computing Machinery, New York, NY, USA, 795–816. https://doi.org/10.1145/3490486.3538346

[22] Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. 2018. Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In *Proceedings of the 2018 world wide web conference*. 1775–1784.

[23] Francesco Fabbri, Yanhao Wang, Francesco Bonchi, Carlos Castillo, and Michael Mathioudakis. 2022. Rewiring What-to-Watch-Next Recommendations to Reduce Radicalization Pathways. In *Proceedings of the ACM Web Conference 2022*. 2719–2728.

[24] Stephane Fotso et al. 2019–. PySurvival: Open source package for Survival Analysis modeling. https://www.pysurvival.io/

[25] JD Gallacher and J Bright. 2021. Hate Contagion: Measuring the spread and trajectory of hate on social media. (2021).

[26] Amira Ghenai and Yelena Mejova. 2017. Catching Zika fever: Application of crowdsourcing and machine learning for tracking health misinformation on Twitter. *arXiv preprint arXiv:1707.03778* (2017).

[27] Amira Ghenai and Yelena Mejova. 2018. Fake cures: user-centric modeling of health misinformation in social media. *Proceedings of the ACM on human-computer interaction* 2, CSCW (2018), 1–20.

[28] Tarleton Gillespie. 2018. *Custodians of the Internet: Platforms, Content Moderation, and the Hidden Decisions That Shape Social Media.* Yale University Press.

[29] Eleonora Giunchiglia, Anton Nemchenko, and Mihaela van der Schaar. 2018. RNN-SURV: A deep recurrent model for survival analysis. In *International Conference on Artificial Neural Networks*. Springer, 23–32.

[30] Homa Hosseinmardi, Amir Ghasemian, Aaron Clauset, Markus Mobius, David M Rothschild, and Duncan J Watts. 2021. Examining the consumption of radical content on YouTube. *Proceedings of the National Academy of Sciences* 118, 32 (2021).

[31] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.

[32] Manas R Joglekar, Cong Li, Mei Chen, Taibai Xu, Xiaoming Wang, Jay K Adams, Pranav Khaitan, Jiahui Liu, and Quoc V Le. 2020. Neural input search for large scale recommendation models. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2387–2397.

[33] Dimitris Kalimeris, Smriti Bhagat, Shankar Kalyanaraman, and Udi Weinsberg. 2021. *Preference Amplification in Recommender Systems.* Association for Computing Machinery, New York, NY, USA, 805–815. https://doi.org/10.1145/3447548.3467298

[34] Wang-Cheng Kang and Julian McAuley. 2019. Candidate generation with binary codes for large-scale top-n recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1523–1532.

[35] Edward L Kaplan and Paul Meier. 1958. Nonparametric estimation from incomplete observations. *Journal of the American statistical association* 53, 282 (1958), 457–481.

[36] Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. 2018. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC medical research methodology* 18, 1 (2018), 1–12.

[37] Daphne Keller. 2021. Amplification and its discontents: why regulating the reach of online content is Hard. *J. Free Speech L.* 1 (2021), 227.

[38] Kyle Langvardt. 2017. Regulating online content moderation. *Geo. LJ* 106 (2017), 1353.

[39] Changhee Lee, William R Zame, Jinsung Yoon, and Mihaela van der Schaar. 2018. Deephit: A deep learning approach to survival analysis with competing risks. In *Thirty-second AAAI conference on artificial intelligence*.

[40] Elisa T Lee and John Wang. 2003. *Statistical methods for survival data analysis.* Vol. 476. John Wiley & Sons.

[41] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. 2019. Pytorch-biggraph: A large-scale graph embedding system. *arXiv preprint arXiv:1903.12287* (2019).

[42] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2615–2623.

[43] Taylor Lorenz. 2022. Twitter labeled factual information about covid-19 as misinformation. https://www.washingtonpost.com/technology/2022/08/25/twitter-factual-covid-info-labeled-misinformation/

[44] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *Advances in neural information processing systems* 32 (2019).

[45] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Feedback Loop and Bias Amplification in Recommender Systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 2145–2148. https://doi.org/10.1145/3340531.3412152

[46] Meta. 2022. Reducing the distribution of problematic content. https://transparency.fb.com/enforcement/taking-action/lowering-distribution-of-problematic-content/

[47] Meta. 2022. What are recommendations on Facebook? https://web.facebook.com/help/1257205004624246?_rdc=1&_rdr

[48] Melinda Mills. 2010. *Introducing survival and event history analysis.* Sage.

[49] Selman Özdan. 2021. The Right to Freedom of Expression Versus Legal Actions Against Fake News: A Case Study of Singapore. *The Epistemology of Deceit in a Postdigital Era: Dupery by Design* (2021), 77–94.

[50] Kostantinos Papadamou, Savvas Zannettou, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Michael Sirivianos. 2020. Understanding the incel community on youtube. *arXiv preprint arXiv:2001.08293* (2020).

[51] Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu, and Yong Yu. 2019. Deep recurrent survival analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4798–4805.

[52] Nicholas J Restrepo, Lucia Illari, Rhys Leahy, Richard F Sear, Yonatan Lupu, and Neil F Johnson. 2021. How Social Media Machinery Pulled Mainstream Parenting Communities Closer to Extremes and their Misinformation during Covid-19. *IEEE Access* (2021).

[53] Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgílio AF Almeida, and Wagner Meira Jr. 2020. Auditing radicalization pathways on YouTube. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 131–141.

[54] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[55] Lu Tang, Kayo Fujimoto, Muhammad Tuan Amith, Rachel Cunningham, Rebecca A Costantini, Felicia York, Grace Xiong, Julie A Boom, and Cui Tao. 2021. "Down the Rabbit Hole" of Vaccine Misinformation on YouTube: Network Exposure Study. *Journal of Medical Internet Research* 23, 1 (2021), e23262.

[56] Dias Oliva Thiago, Antonialli Dennys Marcelo, and Alessandra Gomes. 2021. Fighting hate speech, silencing drag queens? artificial intelligence in content moderation and risks to lgbtq voices online. *Sexuality & culture* 25, 2 (2021), 700–732.

[57] Luke Thorburn, Priyanjana Bengani, and Jonathan Stray. 2022. How Platform Recommenders Work. https://medium.com/understanding-recommenders/how-platform-recommenders-work-15e260d9a15a

[58] Aleksandra Urman, Mykola Makhortykh, Roberto Ulloa, and Juhi Kulshrestha. 2022. Where the Earth is flat and 9/11 is an inside job: A comparative algorithm audit of conspiratorial information in web search results. *Telematics and informatics* 72 (2022), 101860.

[59] Ping Wang, Yan Li, and Chandan K Reddy. 2019. Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)* 51, 6 (2019), 1–36.

[60] Eva Wiseman. 2021. The dark side of wellness: the overlap between spiritual thinking and far-right conspiracies. *The Guardian* (2021).

[61] Peifeng Yin, Ping Luo, Wang-Chien Lee, and Min Wang. 2013. Silence is Also Evidence: Interpreting Dwell Time for Recommendation from Psychological Perspective. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Chicago, Illinois, USA) *(KDD '13)*. Association for Computing Machinery, New York, NY, USA, 989–997. https://doi.org/10.1145/2487575.2487663

[62] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.

[63] Valarie A Zeithaml, Katherine N Lemon, and Roland T Rust. 2001. *Driving customer equity: How customer lifetime value is reshaping corporate strategy*. Simon and Schuster.

[64] Hao Helen Zhang and Wenbin Lu. 2007. Adaptive Lasso for Cox's proportional hazards model. *Biometrika* 94, 3 (2007), 691–703.

[65] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. 2017. Scalable graph embedding for asymmetric proximity. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[66] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1079–1088.

A REDUCING VARIANCE USING BETA-BINOMIAL FITTING

We take an empirical Bayes approach of fitting a beta-binomial distribution to alleviate variance in s_e^{F3} given by (4). In particular, we use the following steps To fit a beta prior, we use the following steps :

(1) First, using the method of moments, for each entity e, we fit a beta prior

$$f(x;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

to the empirical distribution of $\frac{|\mathcal{B}_i^{e+}|}{|\mathcal{E}_i^{e+}|} \in [0,1]$ of all users $u \in \mathcal{U}_e$ s.t. $|\mathcal{E}_i^{e+}| \geq 5$.

(2) Then, for each user $u \in \mathcal{U}_e$, we use the beta prior computed in the previous step and do Bayesian updates to this distribution with each subsequent joins. That is, we treat each "follow" event as a Bernoulli random variable whose parameter is sampled from this prior. A "follow" is labeled a success if it is to a target in $\mathcal{B}$, and fail otherwise. Note that, as Beta is conjugate to the binomial distribution, the resulting posterior is also a beta distribution.

(3) We treat the mean of the posterior beta distribution as the point estimate of $\frac{|\mathcal{B}_i^{e+}|}{|\mathcal{E}_i^{e+}|}$ for each user. Using (4), we average across these point estimates to obtain s_e^{F3}.

B SURVIVAL MODELING

We denote the *survival function* of r.v. X_e, for $t \geq 0$, as:

$$S_{X_e}(t) \equiv \mathbf{P}(X_e \geq t) = \int_t^\infty f_{X_e}(s)ds, \tag{12}$$

and the *hazard function* of X_e, for $t \geq 0$, as:

$$h_{X_e}(t) \equiv \lim_{\delta\to 0} \frac{1}{\delta}\mathbf{P}\left(X_e \in [t, t+\delta])|X_e \geq t\right) = \frac{f_{X_e}(t)}{S_{X_e}(t)}, \tag{13}$$

and the maximum likelihood function is:

$$\mathcal{L}(\boldsymbol{\xi}) = \prod_{e\in\mathcal{E}_0} \prod_{i\in\mathcal{U}_e\cap\mathcal{U}_\mathcal{B}} f_{X_e}(\tau_e^i) \prod_{i\in\mathcal{U}_e\setminus\mathcal{U}_\mathcal{B}} S_{X_e}(T - t_e^i). \tag{14}$$

The neural network used in DeepSurv model contains three fully connected layers with ReLU activation function followed by batch normalization [31] and a dropout layer [54]. The numbers of units in each hidden layer are 250, 200, and 100. The dropout rate is 0.5, and the neural network is trained with an Adam optimizer with learning rate 0.01.

C JOINT PARAMETRIC SURVIVAL MODEL

C.1 Probability of Death via the "Independent Clocks" Model

The probability of death assuming the independent clocks model is the probability that at least one of the "clocks" by each e encountered during observation period expires within the observation interval. This is determined by the (product) survival function in Thm. (5.5). In particular, it is given by

$$\mathsf{PD}_i = 1 - S_i(T), \tag{15}$$

where $S_i(\cdot)$ is given by Eq. (C.3).

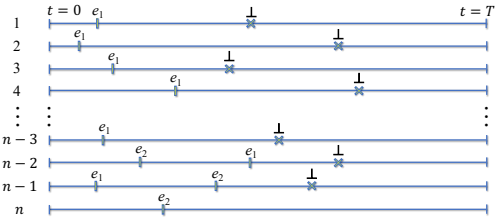


Figure 5: A scenario where studying entities in isolation may lead to wrong conclusions. Studying e_2 *in isolation* gives the impression that e_2 is also influential, as it is very likely to cause a target follow ($\perp$): the survival probability is just 33%. However, both users that eventually interact with the target (namely, $n-1$ and $n-2$) also interact with e_1. Taking into account that e_1 is very likely to cause a target interaction, joins by users $n-2$ and $n-1$ should be attributed to e_1, rather than e_2, thereby "discrediting" the latter's influence. The joint parametric survival model accounts for this, but isolated estimation does not.

C.2 Overestimation of Influence via Parametric Survival Analysis

Under simple parametric survival analysis models, the influence of an entity may be overestimated when it is followed at the same time as another, highly influential entity. An example illustrating this can be found in Fig. 5. Every user that follows entity e_1 eventually follows with the target, so e_1 is highly influential towards target interactions. On the other hand, out of the the three users that interact with entity e_2, two of them also interact with a target. Studying e_2 *in isolation* gives the impression that e_2 is also influential, as it is very likely to cause a target follow: the survival probability is just 33%. However, both users that eventually interact with the target ($n-1$ and $n-2$) also interact with e_1. Taking into account that e_1 is very likely to cause a target interaction, joins by users $n-2$ and $n-1$ should be attributed to e_1, rather than e_2, thereby "discrediting" the latter's influence. Approaches that operate in isolation, like frequentist scores, but also survival analysis parameter estimation via (14), do not capture this, and would rank e_2 highly.

The joint parametric survival model accounts for joint follows, but isolated estimation does not. In particular, parameter estimation in this setting *couples* the learning across entities and, contrary to Eq. (14), MLE is *no longer separable across parameters* $\boldsymbol{\xi}_e$, $e \in \mathcal{E}_0$, that need to be trained jointly (see also Appendix C.4). In turn, this changes how entities in examples like the one shown in Fig. 5 are treated: entities that co-occur with highly influential (i.e., high-hazard) entities contribute less to loss Eq. (14), and are therefore not considered as important.

C.3 Proof of Thm. 5.5

Denote by

$$\mathcal{R}_i^{\leq t} \equiv \left\{e_j^i \in \mathcal{S}_i \cap \mathcal{E}^0 \text{ s.t. } t_j^i <= t\right\}, \tag{16}$$

the trajectory of i, excluding the interaction with the target entity, up to and including time t.

Independence implies that, when $\mathcal{R}_i^{\leq t} \neq \emptyset$:

$$\mathbf{P}(\tau_i > t) = \prod_{e\in\mathcal{R}_i^{\leq t}} \mathbf{P}(X_e > t - t_{i^e}^i) \overset{(12)}{=} \prod_{e\in\mathcal{R}_i^{\leq t}} S_e(t - t_{i^e}^i).$$

On the other hand, since the distribution of clock X_e is continuous, we have that $\mathbf{P}[X_e \in [t, t+\delta]] = f_{X_e}(t) \cdot \delta + o(\delta)$ and, hence,

$$\begin{aligned}\mathbf{P}(\tau^i > t+\delta) &= \prod_{e \in \mathcal{R}_i^{\leq t+\delta}} S_e(t - t^i_{j^e} + \delta) \\ &= \prod_{e \in \mathcal{R}_i^{\leq t+\delta}} \left(S_e(t - t^i_{j^e}) - f_{X_e}(t - t^i_{j^e}) \cdot \delta + o(\delta) \right)\end{aligned}$$

Observe that, for $\delta > 0$ small enough, we have that $\mathcal{R}_i^{\leq t+\delta} = \mathcal{R}_i^{\leq t}$. Thus, for small enough $\delta > 0$,

$$\begin{aligned}&\mathbf{P}\left(\tau^i \in [t, t+\delta]\right) = \mathbf{P}\left(\tau_i \geq t\right) - \mathbf{P}\left(\tau^i \geq t+\delta\right) \\ &= \prod_{e \in \mathcal{R}_i^{\leq t}} S_e(t - t^i_{j^e}) - \prod_{e \in \mathcal{R}_i^{\leq t}} \left(S_e(t - t^i_{j^e}) - f_{X_e}(t - t^i_{j^e}) \cdot \delta + o(\delta) \right) \\ &= \delta \cdot \sum_{e \in \mathcal{R}_i^{\leq t}} \left[f_{X_e}(t - t^i_{j^e}) \prod_{\substack{e' \in \mathcal{R}_i^{\leq t} \\ e'' \neq e}} S_e(t - t^i_{j^{e'}}) \right] + o(\delta) \\ &\overset{(13)}{=} \delta \sum_{e \in \mathcal{R}_i^{\leq t}} h_e(t - t^i_{j^e}) \prod_{e \in \mathcal{R}_i^{\leq t}} S_e(t - t^i_{j^e}) + o(\delta).\end{aligned}$$

The theorem follows, as $f_{\tau^i}(t) = \lim_{\delta \to 0} \mathbf{P}\left(\tau^i \in [t, t+\delta]\right)/\delta$, and the hazard is the ratio of the density and survival functions. In particular, we get that τ^i has the following survival and hazard functions: $S_i(t) = \prod_{e \in \mathcal{R}_i^{\leq t}} S_e(t - t^i_{j^e})$, $h_i(t) = \sum_{e \in \mathcal{R}_i^{\leq t}} h_e(t - t^i_{j^e})$.

C.4 Joint Parameter Inference via MLE

In the joint parametric survival model, parameters $\boldsymbol{\xi}$ can be estimated again via maximum likelihood estimation. An immediate consequence of Theorem 5.5 is that the negative log-likelihood loss in our model has the following form:

$$\begin{aligned}\mathcal{L}_{\mathsf{MLE}}(\boldsymbol{\xi}) &= -\log\Big(\prod_{i \notin \mathcal{U}_{\mathcal{B}}} \mathbf{P}(\tau^i > T) \cdot \prod_{i \in \mathcal{U}_{\mathcal{B}}} f(\tau^i) \Big) \\ &\overset{(C.3)}{=} - \sum_{i \notin \mathcal{U}_{\mathcal{B}}} \sum_{e \in \mathcal{R}_i^{\leq T}} \log(S_e(T - t^i_{j^e})) \\ &\quad - \sum_{i \in \mathcal{U}_{\mathcal{B}}} \Big(\log h_i(\tau^i) + \sum_{e \in \mathcal{R}_i^{\leq \tau^i}} \log S_e(\tau^i - t^i_{j^e}) \Big)\end{aligned} \tag{17}$$

where the aggregate hazard function h_i is given by Eq. (C.3). Comparing this to Eq. (14), we observe that they differ in the treatment of hazards: the aggregate hazard rate h_i *couples* the optimization across entities. Contrary to Eq. (14), Eq. (17) is *not separable across parameters* $\boldsymbol{\xi}_e, e \in \mathcal{E}_0$. As a result, parameters $\boldsymbol{\xi}_e, e \in \mathcal{E}_0$ need to be trained jointly. In turn, this changes how entities in examples like the one shown in Fig. 5 are treated: the concavity of the log means that entities that co-occur with highly influential (i.e., high-hazard) entities contribute less to loss Eq. (17), and are therefore penalized.

D DATASET

Target Sets. The problematic content sets, termed H1 an H2, contain Facebook Groups that have been deemed non-recommendable [47] (see also Sec. 3). We also construct a benign set, BF, comprising groups about breastfeeding using regular expression terms to match names of groups containing words like "breastfeeding", "breast pump", "elvie pump", "willow pump". We look at group memberships over a 12-month period and report membership statistics in Table 7.

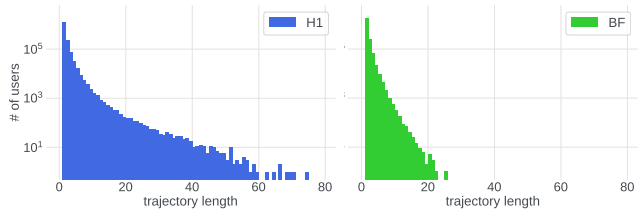


Figure 6: Histograms of trajectory lengths $|S_i|$ in the H1 and BF datasets, respectively. Though the majority of trajectories have length one, both datasets include trajectories of the order of tens of entities per user, though the tail for BF decays faster. In contrast, H2 (not shown) contains no trajectory of length larger than 3 entities.

Label	# Groups $\lvert\mathcal{B}\rvert$	Avg # Users $\overline{\lvert\mathcal{U}_b\rvert}$ (Train)	Avg # Users $\overline{\lvert\mathcal{U}_b\rvert}$ (Test)
H1	1334	265.34	98.74
H2	373300	1301.88	1314.23
BF	13102	217.84	269.21

Table 7: Target Set Overview

Target	# Groups $\lvert\mathcal{E} \setminus \mathcal{B}\rvert$	Avg # Users $\overline{\lvert\mathcal{U}_e\rvert}$ (Train)	Avg. # Users $\overline{\lvert\mathcal{U}_e\rvert}$ (Test)
H1	1000	2603.93	1729.43
H2	1000	1201.85	976.20
BF	1000	3125.68	2078.40

Table 8: Candidate Gateways Overview

Candidate Gateways. For each of the three target sets, we focus on a 6-month observation period (from 2021-07-01 to 2021-12-31) for training. We randomly select 1000 candidate gateway groups with more than 100 new members during the 6-month observation period. We require each candidate gateway group to have more than 10 new members and more than 0.1% of new members joining one of the target groups after joining the candidate gateway. For each candidate gateway group, we monitor the trajectories of at most 5000 new users to avoid computational overruns. Summary statistics are provided in Table 8.

Test Set. The training set is used to compute our proposed gateway scores for each candidate gateway entity. Our test set is constructed over a separate 6-month observation period (from 2022-01-01 to 2022-07-01), immediately succeeding the training period. We monitor the same 1000 candidate gateway groups in this period and randomly select at most 5000 new users in each group. Then we measure how well scores computed on the training set can predict whether "follows" at a gateway also result to "follows" of a target group in the test set (see section *Performance Metrics* below).

User and Group Features. We extract user and group features, using PyTorch-BigGraph [41], a node embedding method which embeds all Facebook users, Pages, Groups and topics into a single latent space. This graph embedding is trained on interactions between entities including likes, comments, shares, reactions, and group membership, restricted to the training set.