Social Reinforcement Learning to Combat Fake News Spread

Mahak Goindani
Department of Computer Science
Purdue University,
West Lafayette, IN
mgoindan@purdue.edu

Jennifer Neville
Departments of Computer Science and Statistics
Purdue University,
West Lafayette, IN
neville@purdue.edu

Abstract

In this work, we develop a *social reinforcement learning* approach to combat the spread of fake news. Specifically, we aim to learn an intervention model to promote the spread of true news in a social network—in order to mitigate the impact of fake news. We model news diffusion as a Multivariate Hawkes Process (MHP) and make interventions that are learnt via policy optimization. The key insight is to estimate the response a user will get from the social network upon sharing a post, as it indicates her impact on diffusion, and will thus help in efficient allocation of incentive. User responses also depend on political bias and peer-influence, which we model as a second MHP, interleaving it with the news diffusion process. We evaluate our model on semi-synthetic and real-world data. The results demonstrate that our proposed model outperforms other alternatives that do not consider estimates of user responses and political bias when learning how to allocate incentives.

1 INTRODUCTION

[Allcott and Gentzkow, 2017] defines fake news as “fabricated articles that are intentionally and verifiably false, and could misled readers”. [Fourney et al., 2017] showed that during the 2016 U.S. Presidential Elections, social media was a major source for fake news dissemination, and voting patterns were highly correlated with the average number of users visiting websites showing fake news. [Allcott and Gentzkow, 2017] found that roughly half of the users on Facebook who viewed fake news stories believed them. This indicates a pressing need to combat fake news spread in social networks.

Fake news mitigation is a multifaceted problem. Much of previous work has focused on detection of fake news using linguistic, demographic, and community based features. There has been relatively less work on limiting the *spread* of fake news. Some recent work has considered mitigating fake news by identifying potential purveyors of fake news to block their posts [Shu et al., 2017]. However, it may not be feasible to take forceful actions such as censoring users posts, since it can violate users’ rights (Bill [H.R.492, 2019]). To address this, we use an approach similar to [Farajtabar et al., 2017], which aimed to mitigate the impact of fake news by making interventions to the *true* news diffusion process. In addition, we also consider the user responses as *feedback* to determine the efficacy of users, and model both the news diffusion and user responses as stochastic processes.

Stochastic point processes are widely used to model user activities in social networks, where events are both self-exciting and mutually-exciting. For example, in Twitter, if a user tweets more about certain types of news articles in the past, then she is likely to tweet more about similar articles in the future. Also, the more a user is exposed to tweets from her followees, the more likely she will (re)tweet similar information. And, tweets of users with more followers tend to get retweeted more [Rizoiu et al., 2017]. We consider different stochastic process models for news diffusion, based on social network structure, history of events, and user interactions.

Given a model of news diffusion, we aim to increase the spread of true news among people exposed to fake news. This is based on the conjecture that increased exposure to true news will increase suspicion and mistrust for fake news, leading to a potential decrease in fake news spread in the future. Users’ perception of information credibility increases if her peers also perceive it as credible, and increases with multiple exposures to same information [Sharma et al., 2019, Garimella et al., 2017]. Hence, the more a user is exposed to true news, the more she will tend to believe such news.

To capture this, we develop a *social reinforcement learning* approach that learns how to incentivize users to spread true news so that people exposed to more fake news are also exposed to more true news. This incentive is realized by a multi-stage intervention to the process of true news diffusion, which increases the probability of a user sharing true news. We assume a fixed incentive budget, and hence aim to learn an optimal strategy to efficiently allocate incentive to users.

Social reinforcement refers to the process where acceptance and praise from others reinforces behaviors/preferences of an individual (see e.g., [Jones et al., 2011]). We propose to model feedback from peers to learn better incentivization policies. Rewards on social media (i.e., 'likes') are a form of acceptance and appreciation from peers, which affects the regions of the brain responsible for decision-making and thus leads to a change in their behavior [Meshi et al., 2015, Crone and Konijn, 2018]. Specifically, we use the number of 'likes' obtained on sharing a post since [Lee and Lim, 2015] found that users provide a positive reinforcement by hitting the 'like' button. 'Likes' have also been used as an important feature in classification of news as fake or true [Wang et al., 2016].

To learn how to efficiently allocate incentives, we consider estimates of user *feedback* and user *political bias*. [Silverman, 2016] observed that around 46% of the fake news stories circulated on Facebook were on U.S. Politics and Elections and a recent analysis of Twitter showed average user polarization was higher for tweets marked with hashtag 'fake news' [Ribeiro et al., 2017]. We also know that reactions of people are more significant for topics related to politics, than other topics such as movies or weather [Kahan et al., 2017], and that people tend to agree more with the information that aligns with their belief, even if the information is false [Allcott and Gentzkow, 2017]. Since the response a user provides for a tweet is likely to depend on their degree of *political bias*, we conjecture that estimates of user response (as a function of political bias) can help to efficiently select people to incentivize to promote true news. To incorporate these effects, we consider a user's leaning towards the Democratic and Republican Parties.

We model user response using a Multivariate Hawkes Process (MHP), whose base intensity is proportional to their political bias, and interleave it with the news diffusion processes (also modeled as an MHP). We estimate a user's initial political bias using a community detection algorithm and propose a model to update the bias over time. The goal is then to learn an intervention strategy by selecting the optimal set of people to be incentivized (based on bias and likely extent of response), and efficiently allocate incentives among them, under a specified budget constraint. We pose the problem as a policy optimization problem in a Reinforcement Learning (RL) framework, with specifically designed reward function targeted at maximizing the true news spread among people exposed to fake news. The RL framework helps to accommodate the objective easily in the form of rewards, and consider reward not only from current stages but also from future stages. By integrating the MHPs in the RL framework, we can model both excitation events and social reinforcement.

Our setting is a cooperative multi-agent RL problem (MARL), where the number of agents is large and the state and action spaces are continuous, which makes the problem more challenging. Much of the previous work in MARL focuses on learning a separate model for each user independently, or learning jointly by considering the full state and action spaces across all users. However, both these approaches are computationally intensive for a large number of agents. We avoid this by decoupling of the post and response processes to approximate the joint action space more efficiently. We dynamically optimize the intensity for the MHP corresponding to post events, but only estimate parameters for the response events from historical data. By doing this, we reduce the number of parameters and avoid noisy policy estimates.

To evaluate the performance of our model, we use two real-world Twitter datasets. Since we have access to limited real-world data, and we cannot make real-time intervention, we perform experiments on semi-synthetic data demonstrating the results with respect to different network properties for fake and true news diffusion likely to hold in real-world. The results show that adding intervention to increase the spread of true news is beneficial for mitigating the impact of fake news relative to providing no incentive. And compared to other baselines that do not consider estimates of user response and political bias, our model is able to achieve increased true news diffusion, in terms of maximizing the number of people reached and the number of *mitigated* users.

2 RELATED WORK

Most of the previous work has focused on the detection of fake news using different features such as linguistic, demographic, community based. [Yang et al., 2013] studied network properties such as clustering coefficient, closeness and betweenness centrality, neighbor based features like number of followers and followees, to identify users likely to spread fake news on Twitter. [Liu and Wu, 2018] tried to classify the propagation path of news to detect fake news at early stages of diffusion. [Shu et al., 2019] considers mitigating fake news by identifying potential provenances and persuaders of fake news, so that their posts can be blocked, however this is difficult in practice due to ethical issues.

Stochastic point processes have gained popularity in modeling user activities in social networks [Farajtabar et al., 2017, Xiao et al., 2017]. Specifically, Hawkes process models [Rizoiu et al., 2017] have been used widely since their mathematical form naturally captures the self-exciting nature of events. [Farajtabar et al., 2017] proposed to mitigate the impact of fake news by making interventions to true news diffusion process modeled as MHP, and mapping the problem to a Markov Decision Process (MDP).

Our work is motivated by their approach, but we extend their model to incorporate a feedback component between pairs of users modeled using a separate MHP, and interleave it with the news diffusion MHP. We believe that feedback is important in selection of users for efficient incentive allocation under budget constraints. The feedback provided to users can be thought of as a *reward shaping* technique, which is used in multi-agent credit assignment and resource allocation problems where it is important to determine the contribution of each agent towards the common system goal for learning better policies [Mannion et al., 2017]. However, our approach is different from standard reward shaping techniques, which consider a separate feedback for each user in the reward function. Since that requires a separate model for learning each agent's policy function, it is computationally intensive for large number of agents. To avoid this issue, we provide user feedback as input to the policy function approximator.

[Upadhyay et al., 2018] uses deep reinforcement learning with marked temporal point processes for incentivizing agents in personalized teaching and viral marketing domains. Similar to our approach, they use feedback events to improve policy learning. However, their events are application specific and are assumed to be generated from a black box distribution. In contrast, we propose a process governing generation of feedback events, and evaluate it using events from real data. Moreover, [Upadhyay et al., 2018] trains a separate model for each user *independently*, which is computationally intensive. In contrast, we *decouple* the news diffusion and response processes to learn an *approximate* model. This reduces the size of the joint action space and helps to avoid noisy estimates, in addition to reducing the number of parameters (compared to the full joint).

Our approach to *social reinforcement learning* is also related to previous work on multi-agent RL (MARL). However, much of the work on MARL focuses on small number of agents (< 50) (e.g., [He et al., 2016]). The standard approaches to train a complex model for each user independently (e.g., [Devlin et al., 2014]) are impractical for thousands of agents. Moreover, the joint action space grows with the number of agents, so joint learning (e.g., [Mannion et al., 2016] is also impractical. Our scenario involves a large number of agents (> 1000) in a social network, with relatively few interactions between them. We use RL to encourage users to share more true news related posts based on our conjecture stated in Sec 1 that with an increased exposure to true news, the impact of fake news will decrease.

3 APPROACH

3.1 PROBLEM DEFINITION

In this work, we consider the task of combating fake news dissemination in online social media systems. Under the assumption we can characterize the diffusion of news over the network by some stochastic process, and that the diffusion of true news is independent from the diffusion of fake news, our aim is to mitigate the spread of fake news by increasing the spread of true news.

We consider the following setting. Let there be N users and let A represent the followers network, where $A_{ij} = 1$ if $j = i$, or j follows i, and 0 otherwise. We consider tweets corresponding to news stories, labeled fake (F) or true (T). We consider the act of tweeting and retweeting by users as a news sharing event and do not differentiate between them. The data contains a temporal stream of events $e = (t, i, c)$, where t is the time-stamp at which user i (re)tweets a post with label c = F or T corresponding to fake or true news. Let $\mathcal{F}_i(t)$ and $\mathcal{T}_i(t)$ be the number of times user i shares posts corresponding to fake and true news, respectively up to time t. We use $\mathcal{N}_i(t)$ as a generic notation to represent the number of times user i shares posts by time t where $\mathcal{N} = \mathcal{F}$ or $\mathcal{T}$ depending on whether we are considering fake or true news. We consider an observation time window of length T divided into K stages of length Δ, where stage k corresponds to the time interval $[\tau_k, \tau_{k+1})$ such that $\tau_{k+1} - \tau_k = \Delta$.

The impact of fake and true news can be measured in terms of number of people who are exposed, that has also been used in [Shu et al., 2019, Farajtabar et al., 2017]. We can compute the number of times user i is exposed to news by time t is given as $A_{.i} \cdot \mathcal{N}(t)$. Since it is difficult to stop the spread of fake news, we want to ensure that users receive at least as much true news as they do fake news (i.e., $A_{.i} \cdot \mathcal{F}(t) \simeq A_{.i} \cdot \mathcal{T}(t)$). We believe that an increased exposure to true news can increase skepticism for fake news, as explained in Sec 1.

The goal is then to incentivize users to share true news in a targeted fashion such that the people who are exposed more to fake news are also exposed to true news. From an algorithmic perspective, we want to learn how to efficiently allocate the incentives, assuming a budget constraint. Specifically, we have a fixed budget that can be provided as incentives and thus, appropriate selection of users and efficient allocation among those is important. The response a user receives on sharing some post is an important indicator of her effectiveness in spreading news further, and can help to determine the amount of incentive to spend on the user. For example, in social networks, this response can be quantitatively measured in terms of number of “likes”[1] received by the user. Our data contains “like” events $l(u, i, t)$ where t is the timestamp at which user i likes user u’s post.

We consider news related to U.S. Politics, and measure *political bias* as a user’s political leaning towards communities of two polarities: Democratic (D) and Republican (R) Party. Each user i has bias values $b_i^R, b_i^D \in [0, 1]$, for R and D, respectively. To compute initial bias values, we run a random walk based community-detection algorithm [Ribeiro et al., 2017, Calais Guerra et al., 2011], using A, with starting seeds for the two communities as the official profiles of politicians whose political affiliation is already known. The bias is estimated as user’s proximity to the two sets of seeds for D and R such that $b_i^R + b_i^D = 1$.

Since we will only incentivize sharing of true news, we evaluate our intervention strategy by computing the correlation between exposures to fake and true news. We also measure the distinct number of people mitigated and asses the effectiveness of users selected by the strategy to spread true news.

3.2 NEWS DIFFUSION PROCESSES

A number of diffusion models have been developed to capture the spread of information in social networks. Many of these models are based on stochastic processes that use *intensity functions* governing the sharing rate per user. Some intensity functions depend only on network structure, while others take into account the effect of previous events and interactions between users. We considered several alternative processes and evaluate which better characterizes the diffusion of news in our real data.

Generative Process Since the process of fake and true news diffusion is the same except for parameters, we provide a generic expression for intensity. Let λ_i^T be the intensity for user i sharing a post corresponding to true news. (The process for fake news is defined analogously.) The generative process to determine timestamps at which user i makes posts corresponding to true news, given their respective intensities, is described as follows. Let $\{t^T_{i,j}\}_{j\geq 1}$ be the time-stamps for user i corresponding to true news events. Define $\{\sqcup^T_{i,j}\}_{j\geq 1}$ to be the inter-arrival times, which are assumed to be independent for all processes. Assuming the diffusion processes start at time 0, we can write, $t^T_{i,m} = \sum\limits_{n=1}^{m} \sqcup^T_{i,n}$. $\mathcal{T}_i(t)$ are the number of times user i shares posts corresponding to true news, by time t. We have $\mathcal{T}_i(t) = \sum\limits_{m\geq 1} \mathbb{I}(t \geq t^T_{i,m})$. Let κ_i^T be the fraction of true news tweets up to time T by user i. These values are computed beforehand from the data. The sampling method to generate inter-arrival times depends on the type of diffusion process and is explained for each type below.

3.2.1 Diffusion based on Network Structure

DEG Intensity depends on the user’s number of followers and followees: $\lambda_i = \kappa_i(\sum\limits_{u=1}^{N} A_{iu} + \sum\limits_{u=1}^{N} A_{ui})$

CEN Intensity is proportional to closeness centrality [Chen et al., 2012]. Let δ_{iu} be the shortest distance from i to u in A: $\lambda_i = \kappa_i(\sum\limits_{u=1}^{N} \delta_{iu})^{-1}$

Generative Process The above processes are homogeneous poisson processes, whose inter-arrival times are exponentially distributed, $f_{\sqcup_i}(t) = \lambda_i e^{-\lambda_i t}$, with inverse cdf is given by $F^{-1}_{\sqcup_i}(u) = \frac{-\ln u}{\lambda_i}$. Since $F^{-1}_{\sqcup_i}(t)$ has a closed form expression, we use inverse transform sampling to sample $\sqcup_{i,j} = \frac{-\ln u_j}{\lambda_i}$, where $j \geq 1, u_j \sim \mathcal{U}(0, 1)$. After we obtain the inter-arrival times $\sqcup_{i,j}$, we can generate the event times $t_{i,j} = \sum\limits_{n=1}^{j} \sqcup_{i,n}$.

3.2.2 Diffusion based on History and Influence

Multivariate Hawkes Process (MHP) We consider an N-dimensional MHP, where each dimension corresponds to a user i. MHP naturally capture the phenomenon of self and mutual excitations between events discussed in Sec. 1.

$$\lambda_i(t) = \mu_i + \sum_{j=1}^{N} \int_0^t \Phi_{ji}\left(\omega e^{-\omega t}\right) d\mathcal{N}_j(s) \tag{1}$$

Here μ_i is user i’s base exogenous intensity. The second term considers the effect of previous events and mutual excitations among users, where Φ is a kernel adjacency matrix that captures the impact user j has on user i. We use the standard Hawkes exponential kernel $\omega e^{-\omega t}$ to capture the decaying effect of history over time, where ω is the hyper-parameter governing the rate of decay. μ and Φ are estimated from the data (see Sec. 3.2.4).

Generative Process The simulations of MHP are performed using “tick” python library ([Bacry et al., 2017]) that uses Ogata’s Thinning Algorithm [Ogata, 1981] to generate event times by sampling inter-arrival times using rejection sampling. The idea is to first generate

[1] Information about likes is obtained using the Twitter API.

events from a homogeneous poisson process with a rate greater than the desired rate, and then reject an appropriate fraction of events generated to achieve the desired rate [Lewis and Shedler, 1979]. After this, we assign a dimension $i \in [1, N]$ to each of the time-stamps generated with probability proportional to λ_i.

3.2.3 Diffusion based on Political Bias

Apart from network properties and user interactions, we believe that user's political bias is an important factor governing the probability of her sharing a post. Hence we model political bias and its change over time based on [Del Vicario et al., 2017]. The idea is that when two users interact, it changes their degree of bias. Let $\mathcal{A}_i = \{j | A_{ij} = 1\}$ be the set of followers of user i. Let $\mathcal{I}_k$ be the list of events $\{e = (t, i, c)\}_{\tau_k \leq t \leq \tau_{k+1}}$ that occurred during stage k, sorted in chronological order. Let $b^D_{i,k}$ and $b^R_{i,k}$ be the bias of user i, respectively, for stage k. We say that a user i, in stage k, has polarity $p_{i,k} = D$ if $b^D_{i,k} > b^R_{i,k}$, and $p_{i,k} = R$ otherwise. We assume that the bias is constant in the interval $[\tau_k, \tau_{k+1})$ and update it at the end of stage k (time τ_{k+1}), taking into account the cumulative effect of interactions during the interval. See Alg. 1 in the Supplementary Material for details.

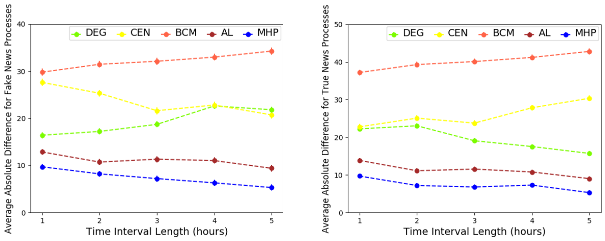
Aligned (AL) If a user i has polarity D in stage k, then her intensity for stage k+1 will be set to her bias (at stage k) for D, otherwise the intensity will be set to her bias for R: $\lambda^{k+1}_i = \mathbb{I}(p_{i,k} = R)\, b^R_{i,k} + \mathbb{I}(p_{i,k} = D)\, b^D_{i,k}$

BCM Similar to AL, but the bias at stage k is computed using Bounded Confidence Model (BCM) [Deffuant et al., 2000, Lorenz, 2007] that has been widely used to capture opinion dynamics in social networks. More details on BCM are in the Supplementary Material.

Generative Process Given the bias values computed for stage k, the diffusion process during stage $k + 1$, for each user i, is a homogeneous Poisson process. Therefore, we sample the inter-arrival times as $\sqcup^{k+1}_{i,j} = \frac{-\ln \mathcal{U}(0,1)}{\lambda^{k+1}_i}$. For stage $k + 1$, we can write $t^{k+1}_{i,j} = \tau_{k+1} + \sum\limits_{n=1}^{j} \sqcup^{k+1}_{i,n}$.

3.2.4 Evaluation of Proposed Processes

Our goal is to quantitatively assess which of the above processes better characterizes news diffusion in real-world data. For this, we use a portion of the data as training data to infer parameters, and then simulate processes for later stages, with the assumption that parameters learnt from historical data (past stages) continue to describe the process in the future. We compare characteristics of the simulated data with the real data to evaluate the various processes.



(a) Fake News Processes

(b) True News Processes

Figure 1: Difference (expected and observed number of events)

We use two real-world datasets, Twitter 2016 and Twitter 2015 [Ma et al., 2017, Liu and Wu, 2018], with 749 and 2051 users in the networks, respectively. We observed that in our data around 75% of the news last for 40 hours, so we set $T = 40$ hours, with 40 stages of $\Delta T = 1$ hour.

We evaluate which of the proposed processes better captures the real data. The training/test framework is shown in Figure 2. Using the parameters learnt from the first 10 stages, we simulate the process for later stages. Details regarding parameter estimation and choice of ω are provided in the Supplementary Material. Let $\mathcal{N}^D_i(t)$ be the number of events of user i up to time t in the real data, and let $\mathcal{N}^P_i(t)$ be the number of events of user i up to time t obtained from the simulating process P. $\mathcal{N}^P_i(t) = \mathcal{F}_i(t)$ for fake news diffusion and $\mathcal{N}^P_i(t) = \mathcal{T}_i(t)$ for true news diffusion defined in Section 3.2. For a given interval of length Δ, we define error $\mathcal{E}^P_\Delta$ as the absolute difference between the number of events generated from the simulated process P and the number of events in the real data in the interval Δ, averaged over all users:

$$\mathcal{E}^P_\Delta = \frac{1}{N} \sum_{i=1}^{N} |[\mathcal{N}^D_i(t' + \Delta) - \mathcal{N}^D_i(t')] - [\mathcal{N}^P_i(t' + \Delta) - \mathcal{N}^P_i(t')]| \qquad (2)$$

where $t' > T_{PE} (= 10)$. Figure 1 shows the error, for each process, corresponding to different values of Δ, where we average over 10 different time intervals for each value of Δ by taking different values of $t' \in [11, 40]$. We observe that MHP achieves the least error, and that it decreases with increasing interval length, for both fake and true news diffusion. This can be attributed to the fact that MHP considers history of previous events, and mutual excitations. Thus, We can say that MHP closely models the diffusion of fake and true news in real-world data, and use it as the process characterizing news diffusion in our model described next.

3.3 INCENTIVIZATION MODEL

Let s^k be the state of the network at stage k. We define actions $\mathbf{a}^k \in \mathbb{R}^N$, where $a^k_i \geq 0$ is the incentive provided to user i to promote true news, during stage

k. Specifically, $\mathbf{a}^k = \pi(s^{k-1})$. We learn the function $\pi : s^{k-1} \rightarrow a^k$ by using policy optimization problem in a Markov Decision Process (MDP) ([Bellman, 1957]), such that the reward (objective) defined in Section 3.3.2 is maximized. MDP based methods take into account the reward achieved on applying the policy, from the current stage as well as from the future stages. We add $\mathbf{a}^k$ as an intervention to the intensity function for true news diffusion modeled using MHP.

$$\lambda_i^T(t) = \mu_i^T + a_i^k + \sum_{j=1}^{N} \int_0^t \Phi_{ji} \left(\omega^T e^{-\omega^T t}\right) d\mathcal{T}_j(s) \quad (3)$$

where $\tau_k \leq t < \tau_{k+1}$ for the k^{th} stage. Since the total amount of incentive provided is usually limited, we impose budget constraint by fixing the sum of incentives for all users at stage k to be C^k, $(\sum_{i=1}^{N} a_i^k = C^k)$. We consider $\mathbf{a}^k$ as actions in the MDP, where the space of all possible actions is given by $O^k = \{a \in R^N | a \geq 0, ||a||_1 = C^k\}$

Generative Process The process to generate events after applying intervention is the same as in Sec. 3.2.2, except the time-stamps are generated for every stage k using the corresponding intensity for the stage, similar to the diffusion based on political bias (Sec. 3.2.3).

3.3.1 State Features

We represent the state of the network s^k for stage k as $s^k = (z_k, \nu^k)$. Here z_k^F refers to the number of previous fake news events and z_k^T refers to the number of previous true news events. ν^k refers to user responses in terms of the number of likes received.

Number of Events in Previous Stages As shown in previous work [Parikh et al., 2012, Qin and Shelton, 2015, Farajtabar et al., 2017], a common choice of features to parameterize point processes is the number of events in the previous stage. Hence, we define $z_k^F \in \mathbb{R}^N$ and $z_k^T \in \mathbb{R}^N$, such that $z_{k,i}^F = \mathcal{F}_i(\tau_{k+1}) - \mathcal{F}_i(\tau_k)$ and $z_{k,i}^T = \mathcal{T}_i(\tau_{k+1}) - \mathcal{T}_i(\tau_k)$.

User Response We consider the news diffusion and response processes to be inter-leaving, and measure response for a user at the end of each stage. Let $\mathbf{W}(t) = [W_{ui}(t)]_{u,i=1,u\neq i}^N$, where $W_{ui}(t)$ is the number of times user i likes the (re)tweets by user u up to time t. $W_u^k = \sum_{i=1,u\neq i}^{N} \int_{\tau_k}^{\tau_{k+1}} dW_{ui}(s)$ is the total likes received by user u during stage k. Hence, the *feature vector* representing the feedback received by users is $\nu^k = \{W_u^k\}_{u=1}^N$.

We cannot make real-time interventions on Twitter and do not know apriori the response (number of likes) a user would receive on (re)tweeting under the news diffusion model. Hence we model the environment generating user responses using another MHP, motivated by [Farajtabar et al., 2015] that modeled the number of times user i retweets source u. We extend their approach, in our case, to model the number of likes given by a user i to source u, by incorporating the stage (time) dependent political bias as the base exogenous intensity explained below. For each pair of users, we have corresponding intensity modeled using MHP, given as $\{\psi_{u,i}(t)\}, u, i \in [1, N], u \neq i)$.

Aligned Bias User Response

If source u and her follower i have the same political leaning (polarity), then the probability (intensity) of i "liking" u's post increases, whereas if they have different polarity, then the probability decreases. To realize this, we adjust the base intensity depending on the bias values.

$$\psi_{u,i}^{k+1}(t) = \chi_i + \mathbb{I}(p_{i,k} = p_{u,k}) b_{i,k}^{p_{i,k}} - \mathbb{I}(p_{i,k} \neq p_{u,k}) b_{i,k}^{\neg p_{i,k}} + \sum_{j \in A_{.i}}^{N} \int_0^t \omega A_{ji}\, e^{-\omega t}\, dW_{uj}(s) \quad (4)$$

where $t \in [\tau_k, \tau_{k+1})$. χ_i is the base intensity estimated from the data that is independent of the history. $A_{.i}$ is the set of followees of i. Using above, we try to accumulate the response a user receives from her direct and indirect followees, by aggregating the likes by followees of user i to the post of user u. The more frequently the followees of i like u's posts, the more she tends to "like" u's posts. When i likes u's post, $W_{ui}(t)$ gets incremented, furthur increasing the chances of liking u's post among followers of i. We simulate the above process for each stage k by first computing the users u who shared true news during stage k, i.e., users with $z_{u,k}^T > 0$, and then generate feedback events using $\psi_{u,i}^k$, only for those users. For simplicity, we set $w = 1$.

To evaluate how well the above model captures "like" events in the network, we use a similar setting as in Sec. 3.2, and observed that the Aligned Bias User Response model outperforms an alternative user response model that does not take into account bias (see Supplementary Material for more detail).

3.3.2 Reward

We use the correlation between exposures to fake and true news ([Farajtabar et al., 2017]) to quantify our objective that people exposed more to fake news are also exposed more to true news. As described in Sec. 3.1, number of exposures by time t is given by $A_{.i} \cdot \mathcal{N}_i(t)$. Hence, the number of exposures in stage k is given as $A_{.i} \cdot \mathcal{N}_i(\tau_{k+1}) - A_{.i} \cdot \mathcal{N}_i(\tau_k)$. Based on the feature representation in Section 3.3.1, the number of exposures to true and fake news, for stage k, is Az_k^T and Az_k^F, respectively. Thus, we write the reward for stage k as,

$$R^k(s^k) = \frac{1}{N}(Az_k^T)^T A z_k^F = \frac{1}{N}(z_k^T)^T\, A^T A\, z_k^F \quad (5)$$

3.3.3 Policy Learning and Optimization

Our goal is to learn policy π to determine the intervention to be applied at each stage for true news diffusion process such that the total expected discounted reward for all stages, $J = \sum_{k=1}^{K} \gamma^k \mathbb{E}[R^k(s^k, \mathbf{a}^k)]$ is maximized, where $\gamma \in (0, 1]$ is the discount rate. $\mathbb{E}[R^k(s^k, \mathbf{a}^k)] = \sum_{\mathbf{a}^k \in \mathcal{O}} R^k(s^k) \cdot P(s^k|\mathbf{a}^k)$, where $P(s^k|\mathbf{a}^k)$ is the probability of being in state s^k after applying intervention a^k in stage $k-1$, and $\mathcal{O}$ is the space of possible actions.

$$\begin{aligned} \mathbb{E}[R^k(s^k, \mathbf{a}^k)] &= \frac{1}{N}\mathbb{E}[(z_k^T)^T A^T A z_k^F] \\ &= \frac{1}{N}\mathbb{E}[z_k^T]^T A^T A \mathbb{E}[z_k^F] \end{aligned} \tag{6}$$

The expected reward for fake and true news diffusion processes can be decomposed due to the independence assumption. Due to space limitations, we provide the details to compute $\mathbb{E}[z_k^T]$, $\mathbb{E}[z_k^F]$ and $\mathbb{E}[\nu_k]$ in the Supplementary Material.

We represent the policy as a function of state (s^k), parameterized by weights θ, that is, $\mathbf{a}^{k+1} = \pi(s^k; \theta)$, where π is the function we want to learn. Each state is associated with a value $V(s^k)$, that is, the total expected reward when in the given state following policy π, $V(s^k) = \mathbb{E}[\sum_{j=k}^{K} \gamma^j R^j|(s^k, \pi)]$. Since it is computationally expensive to compute $V(s^k)$ from future rewards for every possible policy π, we approximate the value as a function of the state parameterized by weights ϕ, that is, $V(s^k) = f(s^k; \phi)$, as in [Kurutach et al., 2018, Zheng et al., 2018]. Policy gradient methods are more effective in high dimensional spaces, and can learn continuous policies. Thus, we use state-of-the-art advantage actor-critic algorithm [Schulman et al., 2015] to find the optimal policy. The details are presented in Algorithm 1.

Setup Figure 2 shows the complete training/test setup for our model. In the figure, $e[t, t']$ and $l[t, t']$ represent the post and like (feedback) events between time t and t'. We use MHP$_\lambda$ to denote the MHP defined in Sec. 3.2.2, and MHP$_\psi$ to denote the MHP defined in Sec. 3.3.1. We use the data from time $[0, T_{PE}(= 10))$ to learn the parameters. Then we divide the remaining data corresponding to time interval $[T_{PE}(= 10), T(= 40)]$ into three parts, data from $[T_{PE}(= 10), T_{Tr}(= 20))$ corresponds to *training dataset* used to learn the policy, data from $[T_{Tr}(= 20), T_{Te}(= 30))$ corresponds to *evaluation dataset* used to evaluate the learnt policy by measuring reward obtained, and data from $[T_{Te}(= 30), T(= 40)]$ is used as *held-out dataset* for experiments in Section 4.2.

We obtain the training dataset and evaluation dataset by generating post (tweet) and feedback events using MHP$_\lambda$ and MHP$_\psi$, respectively. Generating data using MHPs is supported by our observation that MHP$_\lambda$ and MHP$_\psi$ better capture the diffusion and feedback processes as shown in Sections 3.2.4 and 3.3.1. Moreover, since we cannot make real-time intervention to test the policy, we use a simulated environment (using MHPs) as a proxy for online interventions to measure the reward using evaluation data. In order to make the training and evaluation environment similar, we use the events generated by simulating MHPs with parameters learnt from real data. Moreover, we compute the expected value of reward in the future (next stage) assuming that the diffusion process follows the MHP.

Let there be K_{Tr} stages in the training data. Given features for stage k, we find the policy to be applied for stage $k+1$. We use a multi-layer feed-forward network to learn this policy π. Due to space constraints, we describe the architecture for Neural Network (NN) in Supplementary Material. After we obtain the policy as output of the NN, we impose budget constraint by normalizing, as shown in line 7, and compute the expected reward for stage $k+1$ using $\mathbf{a}^{k+1}$ (line 8). $z^{k'}$ and $\nu^{k'}$ are the expected feature vector for the state $s^{k'}$ obtained after applying the policy (line 9), and we find the expected value of the next state $V(s^{k'})$ in line 10. r^k represents the expected reward that could be obtained by applying policy $\mathbf{a}^{k+1}$ given state s^k, in line 11, that comes from the Bellman Optimality Equation [Bellman, 1957]. Instead of simply using the expected reward to optimize the policy, we use an advantage function that is obtained by subtracting the value of state as baseline from the expected reward. This helps to reduce the variance in estimates. Lines 15-16 show the computation of advantage function for each state s^k. In lines 20-21, we learn the optimal parameters θ and ϕ, initialized randomly, using stochastic gradient descent, with learning rates η_θ and η_ϕ, respectively. We use the same NN to learn parameters θ and ϕ for approximating policy and value function, respectively, however the parameters are learnt independently of each other.

3.3.4 Evaluation

To evaluate the learnt policy, we find the intervention $\mathbf{a} = \pi(s)$ where $s = (z, \nu)$ obtained from events in the evaluation data. We simulate MHP$_\lambda$ after adding $\mathbf{a}$ to the base exogenous intensity μ^T for true news diffusion, and compute the following evaluation metric.

Evaluation Metric To compare the performance of different methods, we consider the reward along with the fraction of users exposed to fake news that become exposed to true news. The latter helps to assign more importance to the selection of distinct users over selection of few users with high exposures. Let L_k^T and L_k^F be

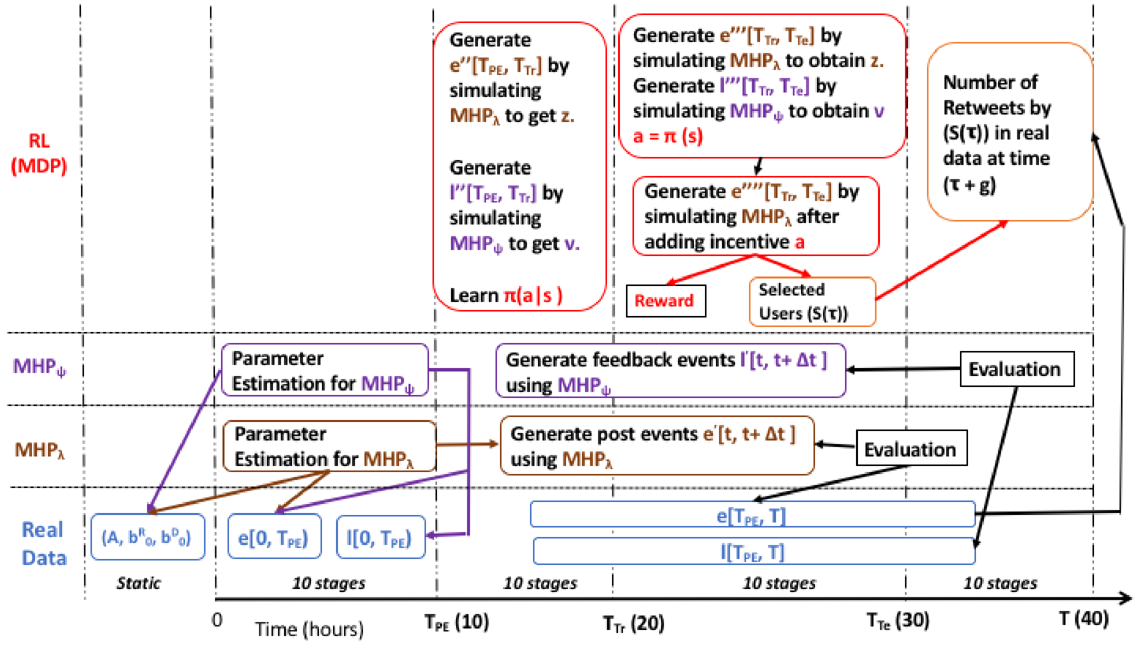

RL (MDP)
MHPψ
MHPλ
Real Data
Generate e''[T_PE, T_Tr] by simulating MHPλ to get z.
Generate l''[T_PE, T_Tr] by simulating MHPψ to get v.
Learn π(a|s)
Generate e'''[T_Tr, T_Te] by simulating MHPλ to obtain z.
Generate l'''[T_Tr, T_Te] by simulating MHPψ to obtain v
a = π (s)
Generate e''''[T_Tr, T_Te] by simulating MHPλ after adding incentive a
Reward
Selected Users (S(τ))
Number of Retweets by (S(τ)) in real data at time (τ + g)
Parameter Estimation for MHPψ
Parameter Estimation for MHPλ
Generate feedback events l'[t, t+ Δt] using MHPψ
Generate post events e'[t, t+ Δt] using MHPλ
Evaluation
Evaluation
(A, b^R_0, b^D_0)
e[0, T_PE)
l[0, T_PE)
e[T_PE, T]
l[T_PE, T]
Static
10 stages
10 stages
10 stages
10 stages
0
Time (hours)
T_PE (10)
T_Tr (20)
T_Te (30)
T (40)

Figure 2: Complete Framework

the sets of users exposed to true and fake news, respectively, during stage k. $L^T_k = \{i|i \in [1,N], z^T_k > 0\}$ and $L^F_k = \{i|i \in [1,N], z^F_k > 0\}$. Define performance as $\mathcal{P} = \sum_{k=0}^{K} R^k \times \frac{|L^T_k \cap L^F_k|}{|L^F_k|}$ We measure the performance of a method relative to that obtained by applying no intervention, in order to assess the gains by making interventions. Specifically, we report the difference between the performance after applying the learnt policy and that without applying a policy. Note that since we cannot make real time interventions, we cannot explicitly test if there is a reduction in fake news spread.

4 EXPERIMENTS

For experiments, we used $C^k \sim N{\cdot}\mathcal{U}(0,1)$, and $\gamma = 0.7$. We compare our model, that we call MHP-U, against different baselines described below. To evaluate a policy on test dataset, we simulate the network under that policy 10 times, and report the average.

4.1 BASELINES

Vanilla MHP (V-MHP) Policy is a function of user events (tweets), similar to [Farajtabar et al., 2017]), does not consider bias or feedback.

Exposure-based Policy (EXP) To mitigate users who shared more posts related to fake news in past [Farajtabar et al., 2017]: $a^k_i \propto \sum_{l=0}^{k} A\, z^F_{l,i}$.

DEG $a_i \propto$ degree of a user i (Section 3.2).

CEN $a_i \propto$ closeness centrality of a user i (Section 3.2).

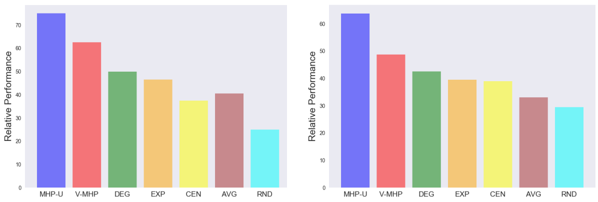
AVG $a^k_i = \frac{C^k}{N}$, that is, the average budget per user.

Random (RND) $a^k_i \propto \mathcal{U}(0, \frac{C^k}{N}]$

4.2 RESULTS

Figure 3 shows the relative performance of different methods. The correlation between fake and true news exposures is higher for MHP-U, and it also maximizes distinct number of users exposed to fake news. The gain in performance comes from the feedback (MHP$_\psi$) used.

Fig 4 shows a change in performance with respect to the ratio of decay parameter for true and fake news diffusion. As this ratio increases, the performance decreases exponentially. This is due to the fact that in the later stages, we have less true news events compared to fake news events. We mark the ratio corresponding to the settings described in Section 3.2.4, with a vertical line.



(a) Twitter 2016

(b) Twitter 2015

Figure 3: Relative Performance on Twitter Datasets

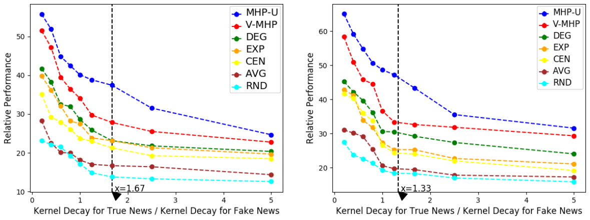
The above results serve as a proof of concept that providing incentives helps to increase the spread of true news even among the people exposed to fake news. However, since we cannot make any real-time interventions, we compare different methods by measuring

Algorithm 1 Policy Learning and Optimization

```
Input: $\{s^k\}_{k=1}^{K_{Tr}}, \mu, \Phi, \chi, \omega, \gamma, \eta_\theta, \eta_\phi$
Output: $\theta^*$
repeat
    for $k = 1, ..., K_{Tr} - 1$ do
        $\mathbf{a}^{k+1} = \pi(s^k; \theta)$
        $V(s^k) = f(s^k; \phi)$
        $\mathbf{a}^{k+1} = \frac{\mathbf{a}^{k+1}}{||\mathbf{a}^{k+1}||_1} \times C^k$ /*budget constraint*/
        Compute $\mathbb{E}[R^{k+1}(\mathbf{a}^{k+1})]$ (Eq. 5)
        $z^{k'} = \mathbb{E}[z^k]$, $\nu^{k'} = \mathbb{E}[\nu^k]$, $s^{k'} = (z^{k'}, \nu^{k'})$
        $V(s^{k'}) = f(s^{k'}; \phi)$
        $r^k = \mathbb{E}[R^{k+1}(\mathbf{a}^{k+1})] + \gamma V(s^{k'})$
    end for
    $L_\theta = 0, \quad L_\phi = 0$
    for $k = 1, ..., K_{Tr} - 1$ do
        Let $D^k = \sum_{j=k}^{K-1} \gamma^k r^k$
        $B^k = D^k - V(s^k)$ /*Compute Advantage*/
        $L_\theta = L_\theta + B^k$
        $L_\phi = L_\phi + ||V(s^k) - D^k||_2$
    end for
    $J_\theta = L_\theta, \quad J_\phi = -L_\phi$
    $\theta = \theta + \eta_\theta \nabla_\theta J_\theta, \quad \phi = \phi + \eta_\phi \nabla_\phi J_\phi$
until $|\Delta\theta| < 0.1$
$\theta^* = \theta$
return $\theta^*$
```

the impact of nodes selected on held-out dataset (Section 3.3.3). Let $S(\tau)$ be the set of users who spread true news according to the model by time τ, that is, $S(\tau) = \{i|(\mathcal{T}_i(\tau) - \mathcal{T}_i(T_{Te})) > 0\}$. We call these as *selected* users, and the remaining users are considered *missed* ($M(\tau)$) by the model. We consider $\tau \in [20, 30]$. Given users in $S(\tau)$ and $M(\tau)$, we calculate the total number of users who retweeted the posts of these users between time $[\tau^{'}, \tau^{'} + \Delta)$ where $\tau^{'} = \tau + g$, in order to measure the impact of the selected and missed nodes in terms of the people actually reached out in real data. $g = \{0, 2, 5, 8\}$ indicates the gap or number of stages after which we want to measure the impact (in the future). We considered different values of $\Delta \in \{1, 2, 3, 4, 5\}$ and report the average values in Table 1. We see that the impact of selected nodes (S) is greater than that of missed nodes (M) for MHP-U, and V-MHP by a large margin.

We also conducted semi-synthetic data experiments using subsets of the Twitter data to study performance with respect to different network parameters, including degree, centrality and user bias. See Supplementary Material for detailed discussion of the results, which show our method outperforming the baselines by a large margin in all scenarios, particularly those that closely match real-world observations.



(a) Twitter 2016

(b) Twitter 2015

Figure 4: Relative Performance vs Ratio of Decay

Table 1: Sum of Retweets at $\tau + k$ for Users Selected at τ

MODEL	$\tau^{'} = \tau + 0$		$\tau^{'} = \tau + 2$		$\tau^{'} = \tau + 5$		$\tau^{'} = \tau + 8$	
	S	M	S	M	S	M	S	M
MHP-U	**1000.5**	**400.8**	**830.2**	**223.3**	**630.8**	170	**553.3**	**140.4**
V-MHP	700.4	416.6	553.7	250.4	420.9	**162.2**	369.1	148.8
DEG	590.6	500.3	445.3	350.1	330.3	259.8	296.6	233.4
EXP	600.2	575.7	299.1	437.8	210.5	298.8	200.2	291.7
CEN	538.5	569.9	369.2	334.6	283.7	257.6	246.1	223.3
BCEN	501.1	511.2	350.5	355.6	276.2	250.3	233.6	237.2
CLC	456.4	545.8	328.2	422.9	242.1	272.9	218.4	281.8
AVG	420.7	598.2	260.3	449.1	215.4	325.6	173.3	300.2
RND	410.1	518.4	205.7	459.2	178.7	340.2	136.1	306.4

5 CONCLUSION

In this paper, we presented a social reinforcement learning approach that can be used to combat the dissemination of fake news by learning how to incentivize users to spread more true news. Our key insight was to estimate likely feedback for each user based on both their network structure and the political bias of their followers, and then combine those estimates with the observed events while learning an incentivization policy. Experiments show that our proposed approach achieves better performance in terms of expected reward and number of distinct mitigated users. Our performance gain comes from the appropriate selection of users and efficient allocation of incentive among them. We tested the efficacy of the users selected by our model and results show that the selected users are able to achieve greater number of retweets, leading to an increased true news spread. Moreover, in semi-synthetic experiments, we observed that it is difficult to encourage people to spread news that does not align with their ideology as has been observed in [Allcott and Gentzkow, 2017], thus justifying our conjecture to model the likelihood of user response as a function of political bias.

Acknowledgements

We thank Ying-Chun Lin for constructive discussions. This research is supported by NSF and AFRL under contract numbers IIS-1618690 and FA8650-18-2-7879.

References

[Allcott and Gentzkow, 2017] Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36.

[Bacry et al., 2017] Bacry, E., Bompaire, M., Gaïffas, S., and Poulsen, S. (2017). tick: a Python library for statistical learning, with a particular emphasis on time-dependent modeling. *ArXiv e-prints*.

[Bellman, 1957] Bellman, R. (1957). A markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684.

[Calais Guerra et al., 2011] Calais Guerra, P. H., Veloso, A., Meira Jr, W., and Almeida, V. (2011). From bias to opinion: a transfer-learning approach to real-time sentiment analysis. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158. ACM.

[Chen et al., 2012] Chen, D., Lü, L., Shang, M.-S., Zhang, Y.-C., and Zhou, T. (2012). Identifying influential nodes in complex networks. *Physica a: Statistical mechanics and its applications*, 391(4):1777–1787.

[Crone and Konijn, 2018] Crone, E. A. and Konijn, E. A. (2018). Media use and brain development during adolescence. *Nature communications*, 9(1):588.

[Deffuant et al., 2000] Deffuant, G., Neau, D., Amblard, F., and Weisbuch, G. (2000). Mixing beliefs among interacting agents. *Advances in Complex Systems*, 3(01n04):87–98.

[Del Vicario et al., 2017] Del Vicario, M., Scala, A., Caldarelli, G., Stanley, H. E., and Quattrociocchi, W. (2017). Modeling confirmation bias and polarization. *Scientific reports*, 7:40391.

[Devlin et al., 2014] Devlin, S., Yliniemi, L., Kudenko, D., and Tumer, K. (2014). Potential-based difference rewards for multiagent reinforcement learning. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 165–172. International Foundation for Autonomous Agents and Multiagent Systems.

[Farajtabar et al., 2015] Farajtabar, M., Wang, Y., Rodriguez, M. G., Li, S., Zha, H., and Song, L. (2015). Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems*, pages 1954–1962.

[Farajtabar et al., 2017] Farajtabar, M., Yang, J., Ye, X., Xu, H., Trivedi, R., Khalil, E., Li, S., Song, L., and Zha, H. (2017). Fake news mitigation via point process based intervention. *ICML*.

[Fourney et al., 2017] Fourney, A., Racz, M. Z., Ranade, G., Mobius, M., and Horvitz, E. (2017). Geographic and temporal trends in fake news consumption during the 2016 us presidential election. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2071–2074. ACM.

[Garimella et al., 2017] Garimella, K., Gionis, A., Parotsidis, N., and Tatti, N. (2017). Balancing information exposure in social networks. In *Advances in Neural Information Processing Systems*, pages 4663–4671.

[He et al., 2016] He, H., Boyd-Graber, J. L., Kwok, K., and Daumé, H. (2016). Opponent modeling in deep reinforcement learning. In *ICML*.

[H.R.492, 2019] H.R.492 (2019). Biased algorithm deterrence act of 2019. *https://www.congress.gov/bill/116th-congress/house-bill/492/.*

[Jones et al., 2011] Jones, R. M., Somerville, L. H., Li, J., Ruberry, E. J., Libby, V., Glover, G., Voss, H. U., Ballon, D. J., and Casey, B. (2011). Behavioral and neural properties of social reinforcement learning. *Journal of Neuroscience*, 31(37):13039–13045.

[Kahan et al., 2017] Kahan, D. M., Peters, E., Dawson, E. C., and Slovic, P. (2017). Motivated numeracy and enlightened self-government. *Behavioural Public Policy*, 1(1):54–86.

[Kurutach et al., 2018] Kurutach, T., Clavera, I., Duan, Y., Tamar, A., and Abbeel, P. (2018). Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592*.

[Lee and Lim, 2015] Lee, Y. and Lim, Y.-k. (2015). Understanding the roles and influences of mediators from multiple social channels for health behavior change. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1070–1079. ACM.

[Lewis and Shedler, 1979] Lewis, P. W. and Shedler, G. S. (1979). Simulation of nonhomogeneous poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413.

[Liu and Wu, 2018] Liu, Y. and Wu, Y.-F. B. (2018). Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Lorenz, 2007] Lorenz, J. (2007). Continuous opinion dynamics under bounded confidence: A survey. *International Journal of Modern Physics C*, 18(12):1819–1838.

[Ma et al., 2017] Ma, J., Gao, W., and Wong, K.-F. (2017). Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 708–717.

[Mannion et al., 2017] Mannion, P., Devlin, S., Duggan, J., and Howley, E. (2017). Multi-agent credit assignment in stochastic resource management games. *The Knowledge Engineering Review*, 32.

[Mannion et al., 2016] Mannion, P., Mason, K., Devlin, S., Duggan, J., and Howley, E. (2016). Dynamic economic emissions dispatch optimisation using multi-agent reinforcement learning. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS 2016)*.

[Meshi et al., 2015] Meshi, D., Tamir, D. I., and Heekeren, H. R. (2015). The emerging neuroscience of social media. *Trends in cognitive sciences*, 19(12):771–782.

[Ogata, 1981] Ogata, Y. (1981). On lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31.

[Parikh et al., 2012] Parikh, A. P., Gunawardana, A., and Meek, C. (2012). Conjoint modeling of temporal dependencies in event streams.

[Qin and Shelton, 2015] Qin, Z. and Shelton, C. R. (2015). Auxiliary gibbs sampling for inference in piecewise-constant conditional intensity models. In *UAI*, pages 722–731.

[Ribeiro et al., 2017] Ribeiro, M. H., Calais, P. H., Almeida, V. A., and Meira Jr, W. (2017). "everything i disagree with is #fakenews": Correlating political polarization and spread of misinformation. *arXiv preprint arXiv:1706.05924*.

[Rizoiu et al., 2017] Rizoiu, M.-A., Lee, Y., Mishra, S., and Xie, L. (2017). A tutorial on hawkes processes for events in social media. *arXiv preprint arXiv:1708.06401*.

[Schulman et al., 2015] Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

[Sharma et al., 2019] Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., and Liu, Y. (2019). Combating fake news: A survey on identification and mitigation techniques. *arXiv preprint arXiv:1901.06437*.

[Shu et al., 2019] Shu, K., Bernard, H. R., and Liu, H. (2019). Studying fake news via network analysis: detection and mitigation. In *Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining*, pages 43–65. Springer.

[Shu et al., 2017] Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.

[Silverman, 2016] Silverman, C. (2016). This analysis shows how viral fake election news stories outperformed real news on facebook. *BuzzFeed News*, 16.

[Upadhyay et al., 2018] Upadhyay, U., De, A., and Gomez-Rodriguez, M. (2018). Deep reinforcement learning of marked temporal point processes. *arXiv preprint arXiv:1805.09360*.

[Wang et al., 2016] Wang, Y., Luo, J., Niemi, R., Li, Y., and Hu, T. (2016). Catching fire via" likes": Inferring topic preferences of trump followers on twitter. In *ICWSM*, pages 719–722.

[Xiao et al., 2017] Xiao, S., Farajtabar, M., Ye, X., Yan, J., Song, L., and Zha, H. (2017). Wasserstein learning of deep generative point process models. In *Advances in Neural Information Processing Systems*, pages 3247–3257.

[Yang et al., 2013] Yang, C., Harkreader, R., and Gu, G. (2013). Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Transactions on Information Forensics and Security*, 8(8):1280–1293.

[Zheng et al., 2018] Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N. J., Xie, X., and Li, Z. (2018). Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 167–176. International World Wide Web Conferences Steering Committee.