

Discovering Influencers in Opinion Formation Over Social Graphs

VALENTINA SHUMOVSKAIA ¹ (Graduate Student Member, IEEE),
MERT KAYAALP ¹ (Graduate Student Member, IEEE), MERT CEMRI² (Student Member, IEEE),
AND ALI H. SAYED ¹ (Fellow, IEEE)

¹École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

²Bilkent University, 06800 Ankara, Turkey

CORRESPONDING AUTHOR: VALENTINA SHUMOVSKAIA. (e-mail: valentina.shumovskaia@epfl.ch).

Mert Cemri contributed to the work as a visiting student at EPFL, Switzerland. This work was supported in part by the Swiss National Science Foundation under Grant 205121-184999. A short version of this work appears in the conference publication [1]

ABSTRACT The adaptive social learning paradigm helps model how networked agents are able to form opinions on a state of nature and track its drifts in a changing environment. In this framework, the agents repeatedly update their beliefs based on private observations and exchange the beliefs with their neighbors. In this work, it is shown how the sequence of publicly exchanged beliefs over time allows users to discover rich information about the underlying network topology and about the flow of information over the graph. In particular, it is shown that it is possible (i) to identify the influence of each individual agent to the objective of truth learning, (ii) to discover how well-informed each agent is, (iii) to quantify the pairwise influences between agents, and (iv) to learn the underlying network topology. The algorithm derived herein is also able to work under non-stationary environments where either the true state of nature or the graph topology are allowed to drift over time. We apply the proposed algorithm to different subnetworks of Twitter users, and identify the most influential and central agents by using their public tweets (posts).

INDEX TERMS Social learning, social influence, explainability, inverse modeling, online learning, graph learning, Twitter.

I. INTRODUCTION AND RELATED WORK

The social learning paradigm is a popular non-Bayesian formulation that enables a group of networked agents to learn and track the state of nature. It has motivated several studies in the literature with many useful variations under varied modeling assumptions (see, e.g., [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]). Under this framework, agents observe streaming data and share information with their immediate neighbors. Through a process of localized cooperation, the agents continually update their *beliefs* about the underlying state. These beliefs describe the agents' confidence on each possible hypothesis. The main question in social learning is whether agents are able to learn the truth eventually, i.e., whether the beliefs on the wrong hypotheses vanish.

In principle, each agent in the network could consider pursuing a *fully* Bayesian solution to learn and track the state of

nature. However, this solution is intractable and generally NP-hard [10], [11], [18]. This is because it requires that each agent has access to the data from the entire network, in addition to their knowledge of the full graph topology. These pieces of information are rarely available in a decentralized setting. For this reason, non-Bayesian approaches have been devised as effective alternatives [2], [3], [4], [5], [6], [7], [8], [9]. In this formulation, the agents first perform a local Bayesian update using their newly received private observations, and then fuse their beliefs with those of their neighbors either linearly or geometrically [3], [6], [8], [9], [19]. This approach allows for diverse data models across the agents, and helps preserve the privacy of the individual observations. In this work, we adopt the *Adaptive Social Learning* (ASL) strategy from [6], which showed how to extend traditional non-Bayesian learning under *fixed* truth to *dynamic* scenarios where the state of nature is allowed to drift with time. Under ASL, the agents will be

able to track these drifts rather effectively with performance guarantees.

Now, given a collection of networked agents tracking the state of nature by means of the adaptive social learning (ASL) strategy, our main objective is to focus on two questions related to *explainability* and *inverse modeling*. In particular, by observing the sequence of publicly exchanged beliefs, we would like to discover the underlying graph topology (i.e., how the agents are connected to each other). We would also like to discover each agent’s contribution (or influence) to the network’s learning process.

The question of explainability over graphs is actually a challenging task, and it has been receiving increasing attention (e.g., [20], [21], [22], [23], [24], [25], [26]). These works approach explainability from different perspectives, and with different aims. For instance, in [25], [26], the authors aim at building a framework for human understanding of why a black-box method arrives at a particular solution. The work [20] argues that a better understanding of multi-agent reinforcement learning can help to limit the search space. Other works suggest modifying the learning algorithm for better interpretability [23]. Overall, higher transparency and a better understanding of the solutions are generally crucial for critical applications, such as using artificial intelligence in healthcare or in autonomous devices (such as vehicles).

We have performed some prior work on explainability for social learning [27], [28]. In these works, given the evolution of beliefs and assuming some partial prior knowledge about the distribution of private observations, it was shown how to identify pairwise influences inside the network (i.e., how strongly pairs of agents influence each other), as well as how to recover the underlying graph topology. One of the key differences with the previous works is that here, we consider a more *limited* (and, therefore, more challenging) information scenario, where we do not require any information about the probability distribution of the observations (or their likelihoods). Despite being more challenging to analyse, the algorithm nevertheless becomes more practical and applicable to real-world data, where distributions of privately received observations remain unknown. We will show that, under this more demanding scenario, we are still able to identify the contribution of each agent to truth learning, assess its level of informativeness, as well as learn the underlying graph and the pairwise influences between agents. The modeling conditions we consider are generally commonplace in real-world social networks, such as Twitter [29], [30], [31], [32], [33], [34], [35], [36]. We consider a Twitter application in Section VII. In this application, users publicly exchange their opinions on Twitter, which are therefore observable. However, we do not have access to additional information about sources affecting people’s opinion during these exchanges such as News or discussions occurring outside Twitter and, therefore, we do not have information about the signal distributions. Identifying the most influential users and their communication patterns can provide valuable insights in social network analysis [37]. Actually, the problem of identifying the most influential nodes

in a network is increasingly relevant [38], [39], [40], [41], [42], [43], especially following the rise of online social networks. Once identified, this information can be useful in many contexts. For example, it can be used to enhance recommendations for marketing purposes [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], where the objective is to maximize the number of influenced nodes.

There have also been several important contributions to graph learning such as [27], [28], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63]. This is because the graph structure plays an important role in distributed learning [64], [65], [66], [67], and its identification can provide valuable information about relationships within the network [27]. One approach is based on exploiting correlations and similarities between vertices [68], [69], [70]. Other works assume a defined model behind the graph signal, such as the heat diffusion process [52], [54], [55]. Structural constraints (such as sparsity, connectivity, symmetry) can also be introduced through regularization [51], [56], [60]. The present work examines the problem of identifying a graph in the adaptive social learning setting. Learning the graph from social interactions requires a different approach, as already illustrated in [28], due to the special form of the non-stationary observation signals. Importantly, our study will relax certain assumptions introduced in that work.

The paper is organized as follows. We describe the adaptive social learning model in Section II. Then, we propose an algorithm for learning the combination matrix and the agents’ informativeness in Section III. In Section IV, we justify the fact that each agent’s contribution to truth learning is proportional to its relative centrality and its informativeness (the KL-divergence between the marginal likelihood of the truth and the other hypotheses). We provide theoretical performance guarantees of the algorithm in Section V. In Section VI, we illustrate performance in different settings by means of numerical simulations. Finally, in Section VII, we apply the algorithm to real-world data from Twitter.

II. SOCIAL LEARNING MODEL

We refer to Fig. 1 and consider a collection of agents \mathcal{N} performing peer-to-peer exchanges of beliefs according to some combination matrix A_\star with non-negative entries, $[A_\star]_{\ell,k} = a_{\ell k} \geq 0$. Agent ℓ is able to communicate with agent k when $a_{\ell k}$ is positive; this scalar refers to the weight that agent k assigns to the information received from agent ℓ . We assume the matrix A_\star is *left-stochastic* and corresponds to a *strongly connected* graph [3], [6], [8]. The first assumption means that the entries on any matrix column $k \in \mathcal{N}$ add up to one, $\sum_{\ell \in \mathcal{N}} a_{\ell k} = 1$. The second assumption means that there exists a path with positive weights between any two agents, and there is at least one agent in the network that does not ignore its own observation, i.e., $a_{kk} > 0$ for at least one $k \in \mathcal{N}$. This implies that the combination matrix is primitive, i.e., for any $\ell, k \in \mathcal{N}$, there exists $t > 0$ such that $[A_\star^t]_{\ell,k} > 0$. It follows from the Perron-Frobenius theorem [71, Chapter 8], [72] that the power matrix A_\star^t converges to $u\mathbb{1}^T$ as $t \rightarrow \infty$ at an exponential

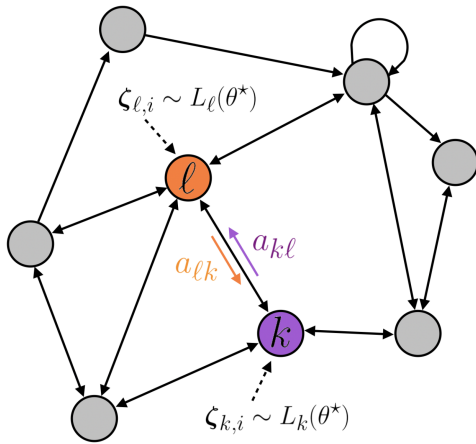


FIGURE 1. An illustration of the network model.

rate, where u is the Perron eigenvector that satisfies:

$$A_* u = u, \quad u_\ell > 0, \quad \sum_{\ell \in \mathcal{N}} u_\ell = 1, \quad (1)$$

where the u_ℓ denote the individual entries of u . Each of these entries describes the centrality of the corresponding agent in the graph (i.e., its level of contribution to the inference task).

We assume that there exists one true state of nature θ^* belonging to a finite set of hypotheses, denoted by Θ . Initially, each agent k starts with a *private belief* vector $\mu_{k,0} \in [0, 1]^{|\Theta|}$, where each entry $\mu_{k,0}(\theta)$ describes how confident agent k is that θ corresponds to the true hypothesis θ^* . As befitting of a true probability mass function, the total confidence sums up to one, $\sum_{\theta} \mu_{k,0}(\theta) = 1$. To ensure that no hypothesis is excluded beforehand by any of the agents, we assume that $\mu_{k,0}(\theta) > 0, \forall \theta \in \Theta$.

At each time instant i , each agent k observes a measurement $\zeta_{k,i}$. We assume initially that each agent $k \in \mathcal{N}$ has access to private likelihood functions, $L_k(\zeta|\theta)$, which describe the distribution of the observation ζ conditioned on each potential model θ . With a slight abuse of notation, we sometimes denote $L_k(\cdot|\theta)$ by $L_k(\theta)$. The observations $\zeta_{k,i}$ are assumed to be independent and identically distributed (i.i.d.) over time. In order to be able to distinguish the true hypothesis θ^* from any other hypothesis $\theta \neq \theta^*$, we need to assume that for any $\theta \neq \theta^*$, there exists at least one clear-sighted agent $k \in \mathcal{N}$ that has strictly positive KL-divergence relative to the true likelihood, i.e., $D_{\text{KL}}(L_k(\theta^*)||L_k(\theta)) > 0$. The following boundedness assumption on the likelihood is common in the literature [7], [27]; it essentially amounts to assuming that the likelihoods share support regions.

Assumption 1 (Bounded likelihoods): There exists a finite constant $b > 0$ such that, for all $k \in \mathcal{N}$:

$$\left| \log \frac{L_k(\zeta|\theta)}{L_k(\zeta|\theta')} \right| \leq b \quad (2)$$

for all $\theta, \theta' \in \Theta$, and ζ .

Next, we describe the ASL strategy from [6]. At each time step i , each agent k performs a local update based on the newly received observation and forms the *intermediate (public)* belief:

$$\psi_{k,i}(\theta) = \frac{L_k^\delta(\zeta_{k,i}|\theta) \mu_{k,i-1}^{1-\delta}(\theta)}{\sum_{\theta' \in \Theta} L_k^\delta(\zeta_{k,i}|\theta') \mu_{k,i-1}^{1-\delta}(\theta')}, \quad k \in \mathcal{N}. \quad (3)$$

Here, $\delta \in (0, 1)$ is a step-size parameter that controls the adaptation capacity, i.e., δ controls the importance of newly received data relative to the past history. This intermediate belief is shared with the *follower* agents of k , i.e., with all agents ℓ for which $a_{k\ell} > 0$. Subsequently, agent k fuses the beliefs received from its neighbors, i.e., from all agents for which $a_{\ell k} > 0$. We denote this set by \mathcal{N}_k . One fusion rule is to fuse the beliefs geometrically as follows in order to obtain the *private* beliefs [3], [6], [8], [19]:

$$\mu_{k,i}(\theta) = \frac{\prod_{\ell \in \mathcal{N}_k} \psi_{\ell,i}^{a_{\ell k}}(\theta)}{\sum_{\theta' \in \Theta} \prod_{\ell \in \mathcal{N}_k} \psi_{\ell,i}^{a_{\ell k}}(\theta')}, \quad k \in \mathcal{N}. \quad (4)$$

The term in the denominator in (4) is used to normalize $\mu_{k,i}(\theta)$ to a probability mass function. Once (4) is performed, the true state θ^* can be estimated by agent k at time i using the maximum a-posteriori construction over either the private or public beliefs, for example:

$$\hat{\theta}_{k,i} \triangleq \arg \max_{\theta \in \Theta} \psi_{k,i}(\theta). \quad (5)$$

It was shown in [6], [27] that this social learning scheme has a powerful performance guarantee. Specifically, the probability of error goes to zero for both the private and public beliefs as the step-size δ approaches zero, namely,

$$\lim_{\delta \rightarrow 0} \lim_{i \rightarrow \infty} \mathbb{P}(\arg \max_{\theta \in \Theta} \mu_{k,i}(\theta) \neq \theta^*) = 0, \quad \forall k \in \mathcal{N} \quad (6)$$

and

$$\lim_{\delta \rightarrow 0} \lim_{i \rightarrow \infty} \mathbb{P}(\arg \max_{\theta \in \Theta} \psi_{k,i}(\theta) \neq \theta^*) = 0, \quad \forall k \in \mathcal{N}. \quad (7)$$

Thus, the agents' confidence on hypothesis θ^* being the true hypothesis converges to one. In other words, the agents are able to learn the truth eventually.

Following [27], we introduce two matrices Λ_i and \mathcal{L}_i in order to represent the recursions (3)–(4) in a more compact matrix form as follows:

$$\Lambda_i = (1 - \delta) A_*^\top \Lambda_{i-1} + \delta \mathcal{L}_i. \quad (8)$$

The matrices are of size $|\mathcal{N}| \times (|\Theta| - 1)$, and their entries are log-belief and log-likelihood ratios and given by:

$$[\Lambda_i]_{k,j} \triangleq \log \frac{\psi_{k,i}(\theta_0)}{\psi_{k,i}(\theta_j)} \quad (9)$$

$$[\mathcal{L}_i]_{k,j} \triangleq \log \frac{L_k(\zeta_{k,i}|\theta_0)}{L_k(\zeta_{k,i}|\theta_j)} \quad (10)$$

where the reference state $\theta_0 \in \Theta$ can be chosen at will by the designer.

The matrices \mathcal{L}_i are i.i.d. over time due to the i.i.d. assumption on the observations. Again, following similar previous

approaches [6], [27], we introduce the following condition on the higher-order moments of \mathcal{L}_i .

Assumption 2 (Positive-definite covariance matrix): The covariance matrix $\mathcal{R}_{\mathcal{L}}$ is uniformly positive-definite for all $i \geq 0$, i.e., there exists $\tau > 0$ such that:

$$\mathcal{R}_{\mathcal{L}} \triangleq \mathbb{E}(\mathcal{L}_i - \mathbb{E}\mathcal{L}_i)(\mathcal{L}_i - \mathbb{E}\mathcal{L}_i)^\top \geq \tau I. \quad (11)$$

Iterating (8), we find that

$$\mathbf{\Lambda}_i = (1 - \delta)^i (\mathbf{A}_\star^i)^\top \mathbf{\Lambda}_0 + \delta \sum_{t=0}^{i-1} (1 - \delta)^t (\mathbf{A}_\star^t)^\top \mathcal{L}_{i-t}. \quad (12)$$

For large i , it can be shown that $\mathbf{\Lambda}_i$ converges in distribution to a limit value given by [27, Lemma 1]:

$$\mathbf{\Lambda}_i \xrightarrow{d} \mathbf{\Lambda} \triangleq \delta \sum_{t=0}^{\infty} (1 - \delta)^t (\mathbf{A}_\star^t)^\top \mathcal{L}_t. \quad (13)$$

For further analysis, we introduce the following useful result for log-beliefs ratios. The standard laws of large numbers cannot be applied directly to the expression for $\mathbf{\Lambda}_i$ in (12) since there are dependencies between the random variables.

Lemma 1 (Law of large numbers for log-belief ratios): After a sufficient number of iterations i (i.e., for $i > M \gg 1$), the average of the log-belief matrix converges in probability as follows:

$$\frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j \xrightarrow{M \rightarrow \infty} \mathbb{E}\mathbf{\Lambda}. \quad (14)$$

Proof: See Appendix A. ■

III. INVERSE LEARNING FROM PUBLIC BELIEFS

In this section, we introduce an algorithm for learning the graph combination matrix \mathbf{A}_\star from observations of the public beliefs, as well as for assessing the level of informativeness of the various agents.

A. PROBLEM STATEMENT

The data available from the social network might be limited for various reasons, including privacy. Therefore, in this work, we assume that we can only observe the evolution of the public beliefs over time:

$$\{\psi_{k,i}(\theta)\}_{i \geq 1}, \quad \forall k \in \mathcal{N}. \quad (15)$$

This assumption is motivated by the fact that agents *share* their intermediate beliefs computed by (3) during the collaborative process, in contrast to the private beliefs in (4). Here, by $i \gg 1$ we underline that we are observing $\mathbf{\Lambda}_i$ after a sufficient amount of iterations, i.e., after $\mathbf{\Lambda}_i$ has reached its steady-state distribution (13).

A good illustration for this setting is the social network of Twitter users. From each post (or tweet) that a user posts to their followers, we can extract an intermediate belief $\psi_{k,i}$ based on sentiment analysis, a.k.a., opinion mining. After that,

each user k reads the posts of its followers and constructs the private belief $\mu_{k,i}$ according to (4).

Next, we will show that by observing public beliefs, we can recover many useful network properties such as (i) the combination matrix, which determines the confidence levels that agents have about each other, (ii) the KL-divergences $D_{\text{KL}}(L_k(\theta^\star) || L_k(\theta))$, which assess the capacity of each agent to distinguish the true hypothesis from the other possibilities, (iii) the pairwise influences of agents on each other, and the (iv) global influencers across the network. We allow both the network and the true state θ^\star to drift over time, and the algorithm will be able to track these changes too.

B. ALGORITHM DEVELOPMENT

The previous work on learning the combination matrix \mathbf{A}_\star in [27] assumes that the expected log-likelihood matrix $\tilde{\mathcal{L}} \triangleq \mathbb{E}\mathcal{L}_i$ is known. It can be verified that

$$\begin{aligned} [\tilde{\mathcal{L}}]_{k,j} &\triangleq [\mathbb{E}\mathcal{L}_i]_{k,j} \\ &= D_{\text{KL}}(L_k(\theta^\star) || L_k(\theta_j)) - D_{\text{KL}}(L_k(\theta^\star) || L_k(\theta_0)). \end{aligned} \quad (16)$$

Using this knowledge along with (8), the following objective function was then minimized to learn \mathbf{A}_\star :

$$Q(\mathbf{A}; \mathbf{\Lambda}_i, \mathbf{\Lambda}_{i-1}) \triangleq \frac{1}{2} \|\mathbf{\Lambda}_i - (1 - \delta)\mathbf{A}^\top \mathbf{\Lambda}_{i-1} - \delta \tilde{\mathcal{L}}\|_F^2 \quad (17)$$

in terms of the squared Frobenius norm. In this work, we do not assume that $\tilde{\mathcal{L}}$ is known beforehand, and will instead estimate $\tilde{\mathcal{L}}$ at each iteration i by using

$$\hat{\mathcal{L}}_{i-1}(A) = \frac{1}{\delta M} \sum_{j=i-M}^{i-1} \left(\mathbf{\Lambda}_j - (1 - \delta)\mathbf{A}^\top \mathbf{\Lambda}_{j-1} \right), \quad (18)$$

where A will be the estimate that is available for \mathbf{A}_\star at that point in time. This step allows us to keep any information about the privately received data $\zeta_{k,i}$ hidden from the algorithm, which makes potential applications more feasible. Accordingly, the cost function is replaced by:

$$\begin{aligned} \hat{Q}(A; \mathbf{\Lambda}_i, \mathbf{\Lambda}_{i-1}, \hat{\mathcal{L}}_{i-1}) &= \frac{1}{2} \|\mathbf{\Lambda}_i - (1 - \delta)\mathbf{A}^\top \mathbf{\Lambda}_{i-1} - \delta \hat{\mathcal{L}}_{i-1}\|_F^2 \\ &= \frac{1}{2} \left\| \mathbf{\Lambda}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j - (1 - \delta)\mathbf{A}^\top \right. \\ &\quad \times \left. \left(\mathbf{\Lambda}_{i-1} - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_{j-1} \right) \right\|_F^2 \\ &= \frac{1}{2} \|\mathbf{\Lambda}_i - (1 - \delta)\mathbf{A}^\top \mathbf{\Lambda}_{i-1}\|_F^2 \end{aligned} \quad (19)$$

where we introduced

$$\mathbf{\Delta}_i \triangleq \mathbf{\Lambda}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j \quad (20)$$

The corresponding risk function is then given by:

$$J_i(A) \triangleq \mathbb{E} \widehat{Q}(A; \mathbf{\Lambda}_i, \mathbf{\Lambda}_{i-1}, \widehat{\mathcal{L}}_{i-1}) \quad (21)$$

We will show in Lemma 2 that the unique minimizer of this risk function gets closer to the true combination matrix as M grows.

Now, assume that we observe a sequence of public beliefs (15), therefore a sequence $\{\mathbf{\Lambda}_i\}_{i \gg 1}$. We define the objective function as a sum over N observed time indices $i \gg 1$:

$$\min_A J(A) \triangleq \frac{1}{N-M} \sum_i J_i(A) \quad (22)$$

To solve (22), we apply stochastic gradient descent (SGD) with constant step-size $\mu > 0$. At each iteration i , the estimate A_i for the combination matrix is updated via:

$$\begin{aligned} A_i &= A_{i-1} + \mu(1-\delta)\mathbf{\Lambda}_{i-1} \\ &\times \left(\mathbf{\Lambda}_i^\top - (1-\delta)\mathbf{\Lambda}_{i-1}^\top A_{i-1} - \delta \widehat{\mathcal{L}}_{i-1}^\top \right) \end{aligned} \quad (23)$$

We sample the observations in the direct order $i, i+1, i+2, \dots$. This online nature of the algorithm, as well as the use of a constant step-size instead of a vanishing step-size, will allow the algorithm to track changes in A_* .¹ We list the procedure in Algorithm 1.

IV. GLOBAL INFLUENCE IDENTIFICATION

In this section, we establish a strong connection between the probability of error for truth learning and the *network divergence*. The network divergence is defined in terms of the Perron eigenvector of A_* , and the KL-divergences between the likelihoods:

$$K(\theta^*, \theta) \triangleq \sum_{k \in \mathcal{N}} u_k D_{\text{KL}}(L_k(\theta^*) || L_k(\theta)) > 0. \quad (24)$$

Note that this quantity is a function of the hypothesis θ . As remarked before (5), the true state estimator can be deduced from the public beliefs. The probability of error is defined as the probability of selecting a wrong hypothesis $\theta \neq \theta^*$:

$$p_{k,i} \triangleq \mathbb{P} \left(\arg \max_{\theta \in \Theta} \psi_{k,i}(\theta) \neq \theta^* \right). \quad (25)$$

Note that if θ^* does not maximize a public belief, then there exists at least one $\theta \neq \theta^*$ such that

$$\log \frac{\psi_{k,i}(\theta^*)}{\psi_{k,i}(\theta)} \leq 0. \quad (26)$$

Thus, we can equally define the probability of error $p_{k,i}$ as:

$$p_{k,i} = \mathbb{P} \left(\exists \theta \neq \theta^* : \log \frac{\psi_{k,i}(\theta^*)}{\psi_{k,i}(\theta)} \leq 0 \right). \quad (27)$$

For this section alone, we will additionally assume that the observations $\{\zeta_{k,i}\}$ are independent over space (and not only

¹In our algorithm, we assume deterministic A_* . However, recursion (23) allows adapting to changes in A_* .

Algorithm 1: Graph Social Learning (GSL).

Data: At each time i :

$$\{\psi_{k,i}(\theta)\}_{k \in \mathcal{N}, \delta}$$

Result: Estimated combination matrix A_N ,
estimated expected log-likelihood ratios $\widehat{\mathcal{L}}_N$.

initialize $A_0, \widehat{\mathcal{L}}_0$

repeat

 Compute matrices $\mathbf{\Lambda}_i$:

for $k \in \mathcal{N}, j = 1, \dots, |\Theta|$ **do**

$$[\mathbf{\Lambda}_i]_{k,j} = \log \frac{\psi_{k,i}(\theta_0)}{\psi_{k,i}(\theta_j)}$$

 Combination matrix update:

$$\begin{aligned} A_i &= A_{i-1} + \mu(1-\delta) \left(\mathbf{\Lambda}_{i-1} - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_{j-1} \right) \\ &\times \left(\mathbf{\Lambda}_i^\top - (1-\delta)\mathbf{\Lambda}_{i-1}^\top A_{i-1} - \delta \widehat{\mathcal{L}}_{i-1}^\top \right). \end{aligned}$$

 Log-likelihoods matrix update:

$$\widehat{\mathcal{L}}_i = \frac{1}{\delta M} \sum_{j=i-M+1}^i \left(\mathbf{\Lambda}_j - (1-\delta)A_i^\top \mathbf{\Lambda}_{j-1} \right)$$

$i = i + 1$

until sufficient convergence;

over time). Now, we know from [6, Th. 3] that each random variable (for any k) of the form (26) can be approximated by a Gaussian random variable in the steady state with the following moments:

$$\log \frac{\psi_{k,i}(\theta^*)}{\psi_{k,i}(\theta)} \approx \mathcal{G} \left(K(\theta^*, \theta) + O(\delta), \delta C + O(\delta^2) \right) \quad (28)$$

for some finite and constant covariance matrix, C . Thus, the probability of error (27) becomes the probability of the Gaussian random variable (28) assuming negative values for at least one $\theta \in \Theta$. This Gaussian random variable concentrates around its mean (i.e., the network divergence in (24)), which is positive. The larger network divergence is, the smaller the probability of error for each individual agent will be.

If we examine expression (24) for the network divergence, we observe that each individual agent k contributes with a term of the form

$$K_k(\theta^*, \theta) = u_k D_{\text{KL}}(L_k(\theta^*) || L_k(\theta)) \quad (29)$$

which is scaled by the Perron entry u_k . This entry reflects the centrality of agent k , and serves as a measure of how well it is connected to other nodes in the network. The larger the value of (29) is, the stronger the contribution of this agent will be towards moving the network away from an erroneous decision. We therefore say that (29) helps convey the amount of information that agent k has about θ disagreeing with θ^* .

Since (29) is a function of θ , we can define the level of informativeness of agent k to the learning process by considering the aggregate of its contributions for all θ , which we denote by

$$I_k \triangleq \sum_{\theta \in \Theta} K_k(\theta^*, \theta) = u_k \sum_{\theta \in \Theta} D_{\text{KL}}(L_k(\theta^*) || L_k(\theta)). \quad (30)$$

This quantity serves as a measure of *influence*, since agents with large I_k contribute the most to learning the truth by the network.

In what follows, we describe how to estimate the quantities I_k by the learning algorithm. First, to obtain the Perron eigenvector for A_i , we need to normalize any of its eigenvectors corresponding to the eigenvalue at 1. Subsequently, we identify $\hat{\theta}_{k,i}$ that maximizes $\psi_{k,i}(\theta)$. From (7) we know that $\hat{\theta}_{k,i}$ tends to θ^* almost surely as $i \rightarrow \infty$ and $\delta \rightarrow 0$. After a sufficient number of iterations i_N of the Algorithm 1, we let j' denote the index within the hypothesis set Θ that corresponds to:

$$\hat{\theta}_{j'} = \arg \max_{\theta \in \Theta} \psi_{k,i_N}(\theta). \quad (31)$$

Returning to (16), we can then approximate the KL-divergences by

$$D_{\text{KL}}(L_k(\theta^*) || L_k(\theta_0)) \approx -[\hat{\mathcal{L}}_{i_N}]_{k,j'}, \quad (32)$$

$$D_{\text{KL}}(L_k(\theta^*) || L_k(\theta_j)) \approx [\hat{\mathcal{L}}_{i_N}]_{k,j} + [\hat{\mathcal{L}}_{i_N}]_{k,j'}, \quad (33)$$

where $\hat{\mathcal{L}}_{i_N}$ is an estimate for $\bar{\mathcal{L}}$.

To conclude, the sequence of public beliefs contains rich information about the network. Using the GSL algorithm, we are not only able to identify the graph topology, but can also find answers to the explainability question: which agents were the most responsible (or the main drivers) for the overall network learning process?

V. THEORETICAL RESULTS

First, we establish some useful properties of the risk function (21).

Lemma 2 (Risk function properties): After sufficient number of iterations i , the risk function $J_i(A)$ is strongly convex and has Lipschitz gradient with constants ν_i and κ_i given by:

$$\begin{aligned} \nu_i &\triangleq (1 - \delta)^2 \lambda_{\min} \left(\mathbb{E} \Delta_{i-1} \Delta_{i-1}^T \right) \\ &\geq \tau \delta^2 (1 - \delta)^2 + O \left(1/\sqrt{M} \right) \end{aligned} \quad (34)$$

$$\begin{aligned} \kappa_i &\triangleq (1 - \delta)^2 \lambda_{\max} \left(\mathbb{E} \Delta_{i-1} \Delta_{i-1}^T \right) \\ &\geq \tau \delta^2 (1 - \delta)^2 + O \left(1/\sqrt{M} \right) \end{aligned} \quad (35)$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ are the minimum and maximum eigenvalues. In other words, it holds that:

$$\nu_i I \leq \nabla_A^2 J_i(A) \leq \kappa_i I. \quad (36)$$

Moreover, the difference between the true combination matrix A_* and the unique minimizer $A_{\min,i}$ of $J_i(A)$ is on the order of:

$$\|A_{\min,i} - A_*\|_{\text{F}}^2 \leq O(1/\delta^2 M^2) \quad (37)$$

Proof: See Appendix B. ■

Result (37) states that the gap between A_* and the unique minimizer of $J_i(A)$ is negligible as $M \gg 1/\delta$ grows.

To investigate the steady-state performance of recursion (23), we adopt the following independence assumption, which is typical in the study of adaptive systems [73], [74], [75].

Assumption 3 (Separation principle): We denote the estimation error by $\tilde{A}_i \triangleq A_* - A_i$, and assume the step-size μ is small enough to allow $\|\tilde{A}_i\|_{\text{F}}^2$ to attain a steady-state distribution. The separation principle states that the error \tilde{A}_i is independent of the observations $\Lambda_i, \dots, \Lambda_{i-M}$, conditioned on the history of previous observations. ■

The following theorem shows that in steady-state, the mean-squared error is $O(\mu) + O(1/\delta^3 M^2)$ in expectation.

Theorem 1 (Steady-state performance): The mean-square deviation (MSD) converges exponentially fast with asymptotic convergence rate:

$$\alpha \triangleq 1 - \mu(2\nu + O(\delta^3)) + O(\mu^2) \quad (38)$$

where

$$\nu \triangleq (1 - \delta)^2 \lambda_{\min} \left(\lim_{i \rightarrow \infty} \mathbb{E} \Delta_{i-1} \Delta_{i-1}^T \right) \quad (39)$$

In the limit, the MSD satisfies:

$$\limsup_{i \rightarrow \infty} \mathbb{E} \|\tilde{A}_i\|_{\text{F}}^2 \leq O(\mu) + O(1/\delta^3 M^2) \quad (40)$$

Proof: See Appendix C. ■

Naturally, the convergence rate for learning the combination matrix is dependent on the strong convexity constant, ν . Usually, the limiting MSD expression is of the form $O(\mu)$, so that we can reduce the deviation by using arbitrary small step-size μ . In our case, we also have the additional term $O(1/\delta^3 M^2)$, which is due to the difference (37) between the unique minimizer and the true combination matrix. We can control the number of samples M . Note that by selecting $M = O(1/\sqrt{\mu})$, the MSD in (40) becomes $O(\mu)$.

Finally, we study how well the algorithm approximates $\bar{\mathcal{L}}$.

Theorem 2 (Steady-state log-likelihood learning): The MSD converges exponentially fast with

$$\begin{aligned} &\limsup_{i \rightarrow \infty} \mathbb{E} \|\hat{\mathcal{L}}_i - \bar{\mathcal{L}}\|_{\text{F}}^2 \\ &\leq \frac{1}{M} \text{Tr}(\mathcal{R}_{\mathcal{L}}) + O(\mu/\delta^2) + O(1/\delta^5 M^2) \end{aligned} \quad (41)$$

where

$$\mathcal{R}_{\mathcal{L}} = \mathbb{E} (\mathcal{L}_i - \bar{\mathcal{L}}) (\mathcal{L}_i - \bar{\mathcal{L}})^T \quad (42)$$

is independent of i due to i.i.d. observations.

Proof: See Appendix D. ■

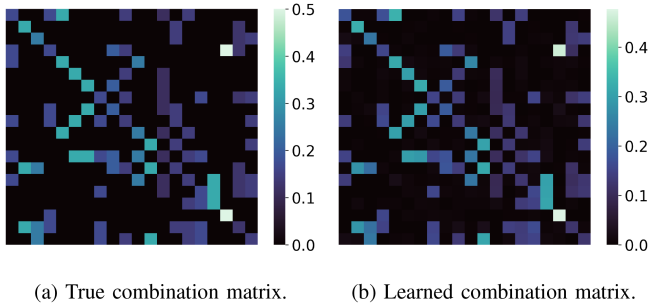


FIGURE 2. True combination matrix and the learned matrix using the GSL algorithm with $M = 50$.

Since we simultaneously learn the combination matrix A_* and $\tilde{\mathcal{L}}$, the limiting MSD for $\tilde{\mathcal{L}}$ has similar expression to Theorem 1. We would like to note that usually, the hyper parameter δ is a network property we don not have a direct influence on. Thus, the final MSD is controlled by the step-size μ and it decreases as we use higher number of samples M .

VI. COMPUTER SIMULATIONS

In this section, we illustrate how well the proposed algorithm is able to identify the true combination matrix A_* and the expected log-likelihood matrix $\tilde{\mathcal{L}}$. We also experiment with different M in (18) to see how the convergence rate changes. Additionally, we will see how well the learned combination matrix and KL-divergences identify the influences (30).

We generate a graph with $|\mathcal{N}| = 20$ agents according to the Erdos-Renyi model with an edge probability of $p = 0.2$. We set the adaptation hyperparameter to $\delta = 0.05$. Then, we generate the combination weights (see Fig. 2(a)) with uniform weights in the column, such that the resulting matrix is left-stochastic. We consider $|\Theta| = 5$ states, where the likelihood models $L_k(\theta)$ for each agent $k \in \mathcal{N}$ are assumed to follow a binomial distribution with randomly generated parameters (for details, see Appendix E). We generate likelihood models such that we observe only 3 agents with high informativeness. Later, we illustrate that the algorithm allows for identifying these agents.

First, we consider how well the combination matrix is learned for different $M \in \{1, 10, 50\}$. We additionally compare with [27], where the expectation $\tilde{\mathcal{L}}$ was assumed to be known beforehand. For $M = 50$, we use $\mu = 0.1$, for $M = 10$, we use $\mu = 0.01$, and for $M = 1$, we use $\mu = 0.001$ for better convergence. In Fig. 3, we plot the reconstruction error with respect to the iteration number:

$$\|\tilde{A}_i\|_{\mathbb{F}}^2 = \|A_i - A_*\|_{\mathbb{F}}^2 \quad (43)$$

We notice that the higher M improves the limiting MSD as reflected in Theorem 1.

Next, we illustrate how well the learned $\tilde{\mathcal{L}}_i$ approximates $\tilde{\mathcal{L}}$ for different $M \in \{1, 10, 50\}$. In Fig. 4, we show how the reconstruction error evolves with i :

$$\|\tilde{\mathcal{L}}_i\|_{\mathbb{F}}^2 = \|\hat{\mathcal{L}}_i - \tilde{\mathcal{L}}\|_{\mathbb{F}}^2. \quad (44)$$

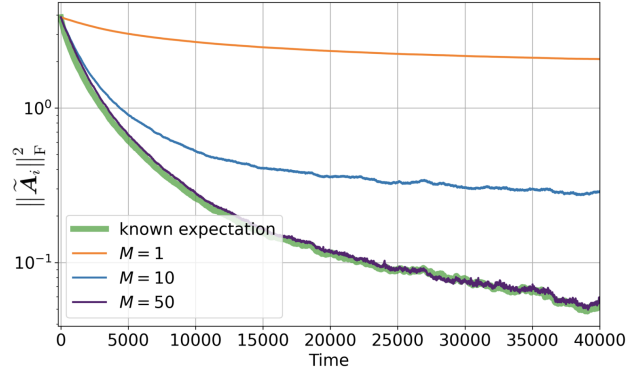


FIGURE 3. Algorithm performance when $\tilde{\mathcal{L}}$ is known and when it is estimated by (18) for different $M \in \{1, 10, 50\}$.

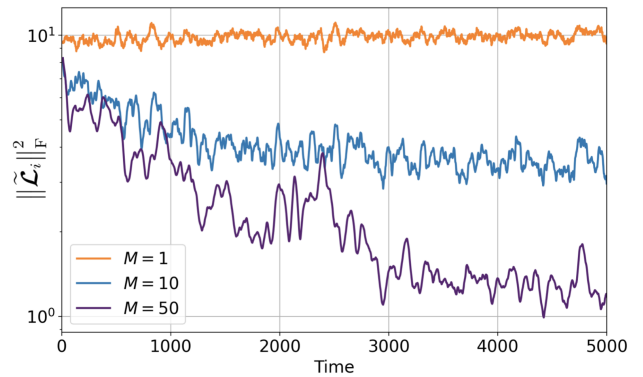


FIGURE 4. $\tilde{\mathcal{L}}$ reconstruction error estimated by (18) for different $M \in \{1, 10, 50\}$. To reduce the variance and therefore for better interpretability, we plot the rolling mean with window size equal to 50.

We notice that with M growing, we can approximate $\tilde{\mathcal{L}}$ more precisely, which aligns with the result of Theorem 2, and explains Fig. 3.

We illustrate the recovered combination matrix in Fig. 2(b). We use $M = 50$ since it has better convergence. Comparing Fig. 2(a) and (b), we observe an almost perfect recovery of the true combination weights.

Fig. 5 illustrates how well the learned KL-divergences and combination matrix can recover the global influences (30). For better interpretability, we normalize the values so that they add up to one. We see that for some agents, the algorithm does not perfectly recover these components, but yet allows us to identify that the first agents are driving the learning the most. This property allows us to search for agents that are the most contributing to learning the true state.

In Fig. 6, we illustrate the connection between the probability of error (25) and the presence of agents with a significant contribution I_k (30) to the network divergence. We plot the rate of correctly classified states at each point i :

$$r_i = \frac{1}{i} \sum_{t=1}^i \mathbb{I}\{\hat{\theta}_t = \theta_*\} \quad (45)$$

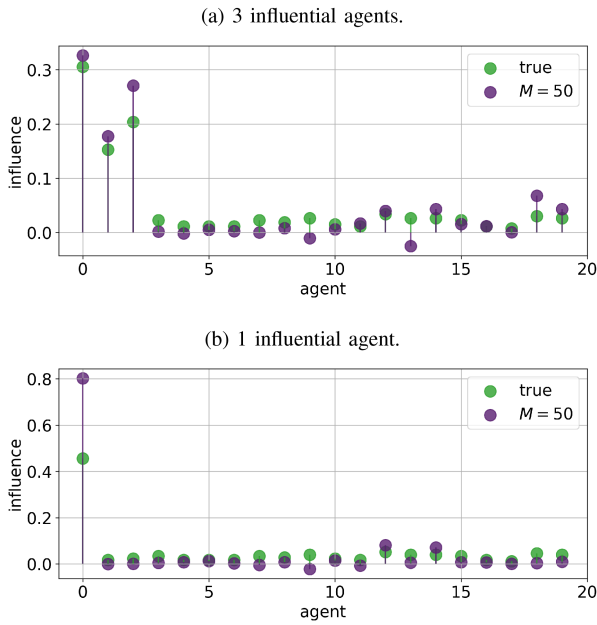


FIGURE 5. Agents' influences (30) based on the learned graph and KL-divergences.

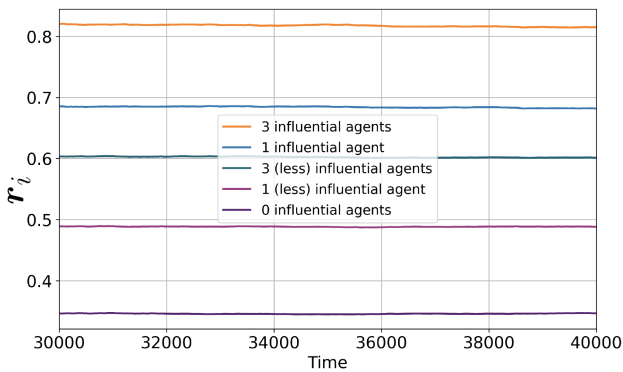
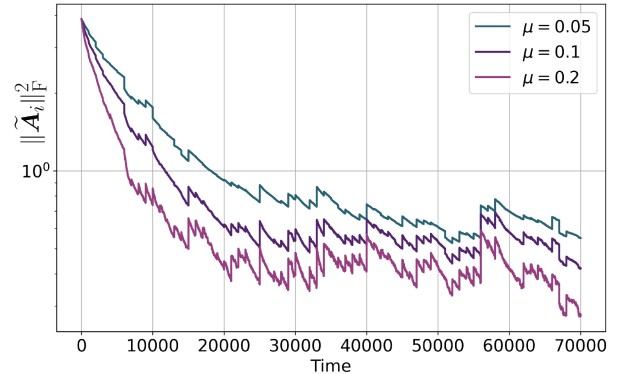


FIGURE 6. Rate of correctly classified truth for different models. Less influential agents denote agents with the same centrality as influential agents, but with smaller KL-divergence between states.

We generate likelihood models according to Appendix E, where less (yet significantly) influential agents have smaller KL-divergence between likelihood models (lines 3 and 4 in the legend). Thus, we illustrate the value of the identified importance measure (30) and the role of “informativeness” (or KL-divergences) of each agent: the presence of agents with high informativeness improves the quality of true state inference.

Finally, we comment on the adaptation abilities of the proposed algorithm that results from its online nature (23). There is a trade-off between the speed of adaptation to changes in A_* or in θ^* and the final MSD. Larger step-size μ leads to faster reaction by the algorithm to changes by the combination matrix, but enlarges the steady-state MSD. However, an important point to note here is that if the topology changes

(a) Algorithm performance when the graph regularly changes its topology (every 1000 iterations). Every element of the adjacency matrix changes its state with probability 0.5%.



(b) Algorithm performance when the true state θ^* changes.

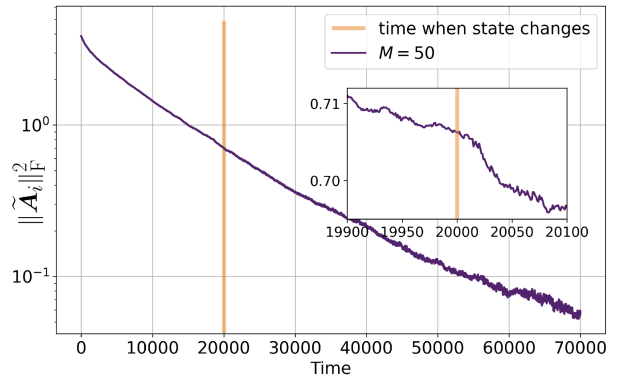


FIGURE 7. Algorithm performance under perturbations.

more frequently than the time needed to reach the steady-state (each change of the true A_* “restarts” the algorithm), then a fairly large step-size μ would be needed. We illustrate this behavior in Fig. 7(a). The ability of the algorithm to adapt to the true state θ^* is also important for its performance due to the fact that we have performance guarantees when Λ_i reaches its steady-state. The adaptation ability is also evident from the social learning update itself (3)–(4). In [6], the authors discuss that small δ leads to an increased confidence of beliefs, but it comes at the cost of an increased adaptation time, on the order of $\approx \log 2/\delta$. Thus, as long as the environment does not change its state faster than the adaptation period, the performance of the algorithm stays close to what is predicted under Theorems 1, 2. We illustrate this scenario in Fig. 7(b).

VII. APPLICATION TO TWITTER DATA

In this section, we apply the graph social learning algorithm introduced in this manuscript to actual datasets. We choose Twitter as a suitable social media platform where we can analyze the posts of users and observe their interactions. In particular, we aim to detect the centrality (i.e., Perron entry) and influence of agents over a network, and identify the most

influential agents. We choose this objective as a more realistic goal for Twitter data, rather than trying to recover the entire combination matrix. This is primarily due to the fact that we can only have access to the true adjacency matrix of the users on Twitter, that is to say, we only know who follows whom, but we do not know the weights (confidence levels) agents assign to each other. In the literature, there have been multiple studies trying to analyze the influence of Twitter accounts on various topics. The work [29] analyzed the propagation of information through Twitter networks, and how different users take part in the effective dissemination of ideas. Moreover, the work [30] analyzes the influential users on Twitter by investigating the linguistic aspects of the tweets of users, such as the grammatical structure and vocabulary of the tweets. However, these studies lack a mathematical foundation to analyze the “influence” of agents, and rely largely on heuristics. In this context, the work [31] aims to leverage ten different attributes of Twitter accounts and their tweets to develop a Twitter-based influence measure. Similarly, the work [32] uses descriptive statistics of the Twitter accounts such as their follower counts, frequency of their posts and the total number of replies, likes and retweets of users. However, what we propose in this study is significantly different. In the proposed algorithm, we do not need to have access to any of these features and the various statistics. In fact, the only input the proposed algorithm needs is the publicly shared tweets of the users. This information is sufficient to learn the centrality of users over a subnetwork of Twitter users, and to identify the most influential user for the formed opinions, using the described mathematical model of social learning.

To identify the influence of agents in a network, we first need to (i) create a subnetwork of Twitter users that is strongly connected, (ii) obtain the tweets (posts) of the users in the created network, and (iii) process the text in the tweets to obtain the log-belief ratios Λ_i over time. After these pre-processing operations, we run the proposed algorithm, and estimate the underlying combination matrix of the network of users and the likelihood models of these users. Then, we compute the Perron eigenvector of the estimated combination matrix. We refer to this vector as the “learned” Perron vector, not to be confused with the “original” Perron vector of the true combination matrix of the network of users. Note that, there is no ground truth regarding the confidence agents place on each other, hence one cannot possibly know the true combination matrix. Therefore, we heuristically form a combination matrix to obtain its Perron vector, through the procedure of Section VII-A. However, as an indisputable ground truth data for a given subnetwork, we know the agents that have the highest number of followers within that subnetwork. In our experiments, these agents happen to coincide with the highest original Perron vector entries.

We run experiments on three different user subnetworks of Twitter users by using Twitter API, where we utilize the Tweepy library to build up our queries in Python. Quantitatively, we show that we can identify the most central user in all three networks, i.e., the largest entries of the learned

Perron vector and the original Perron vector. Note that in these experiments with real data, we do not have access to the likelihood models of the users, hence we do not have “ground truths” for the influence of agents, i.e., $u_k D_{\text{KL}}(L_k(\theta^*) || L_k(\theta))$. However, to obtain a qualitative measurement, we construct the aforementioned three networks (see Fig. 8) around famous public figures (the first network is built around Elon Musk (CEO, 118.5 M followers), the second network is built around Maggie Haberman (journalist, 1.6 M followers), and the third network is built around Ben Shapiro (columnist, 5 M followers)). In all of these networks, we recover these public figures as the most influential users.

A. NETWORK FORMATION

To construct a network, we first select a popular Twitter account A , such as an influential CEO or a journalist whom we expect to be influential among other users. Next, in order to select a strongly connected network, we build a network starting from a less centered Twitter account B that is followed by that popular user A . Starting from account B , we construct a subnetwork of depth 2, i.e. we identify 1 K followers of account B , and 1 K followers of each of those followers. While satisfying these conditions, we filter users who post frequently, so that they provide sufficient data, and who are less centered (they have less than 10 K followers). Among all of these identified follower-following relationships, we construct a network, and verify that the network is strongly connected. Following this procedure, the networks constructed for Elon Musk, Maggie Haberman and Ben Shapiro are of sizes 20, 26 and 28, respectively.

After constructing the network, we obtain its adjacency matrix by finding out who follows whom. Since there is no way in real world experiments to determine the confidence agents assign to each other, there is no ground truth for the combination matrix. Therefore, we assume that agents assign uniform weights to their neighbors (i.e., to the people they follow). This corresponds to applying the averaging rule [74, Chapter 14] to the adjacency matrix.

B. OBTAINING USER POSTS

In order to obtain the tweets of the users, we first build up our query for the Twitter API. This query includes a specific keyword for each network so that we do not fetch unrelated tweets. For the network containing Elon Musk, we choose the keyword to be “coin OR bitcoin OR crypto-currency,” for the tweets between 01.01.2017 and 01.05.2022. In this case, we would like to see the influential agents in that network in shaping the opinion of Twitter users on crypto-currency related matters. For the other networks, we choose the keyword to be “Trump” for the tweets between 01.01.2017 and 31.01.2021, and “Biden” for the tweets between 01.01.2021 and 01.05.2022. Hence, we aim to determine the influence of agents in determining the attitude of their respective networks regarding the issue of “the current president of the United States”.

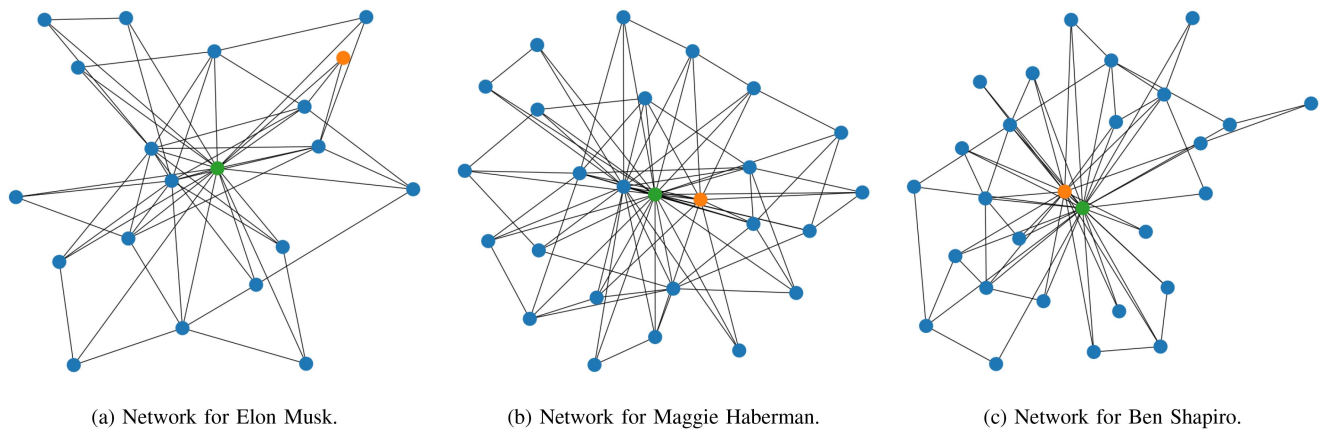


FIGURE 8. Network of agents. Agent A is indicated with color green, and agent B is indicated in color orange according to the description in Section VII-A.

TABLE 1. Examples of Sentiment Analysis and Generated Sentiment Probabilities

Sample Tweets	Positive Sentiment
<i>#Bitcoin for the win</i>	$p = 0.99$
<i>Unless otherwise stated, I am adding Bitcoin every week</i>	$p = 0.82$
<i>Bitcoin falls from \$66K highs, Tesla down 3% after Elon Musk warns he could sell more #Bitcoin</i>	$p = 0.02$

C. OBTAINING THE LOG-RATIO BELIEFS OF USERS

After obtaining the tweets of the users, we need to process the text in those tweets to obtain the intermediate belief vectors. For this purpose, we use a language model based on Roberta, which is trained with around 124 M tweets, and fine tuned with the TweetEval benchmark for the sentiment analysis task [76]. We can feed a text to this model and obtain three different probabilities for three different sentiments of the input text: p_0 for “Negative,” p_1 for “Neutral,” and p_2 for “Positive”. We eliminate the neutral labels and recalculate the probability of the positive sentiment of a text as

$$p \triangleq \frac{p_2}{p_0 + p_2} \quad (46)$$

and accordingly, the probability of the negative sentiment as

$$1 - p = \frac{p_0}{p_0 + p_2} \quad (47)$$

In Table 1, we show sample tweets from the network related to Bitcoins. When p is close to 1, this means that the text asserts a positive attitude towards Bitcoin, and when p is closer to 0, this means that the text asserts a negative attitude.

To construct the log-belief ratio of an agent at a day i , we incorporate the sentiment of all tweets within this day. We denote the number of tweets user k shares at iteration i by $N_{k,i}$. Among those tweets, we denote the positive sentiment

probability of each tweet t by $p_{k,i,t}$, so that $p_{k,i,t}$ approximates belief $\psi_{k,i,t}(\theta_0)$, and $1 - p_{k,i,t}$ as approximates belief $\psi_{k,i,t}(\theta_1)$. Here, θ_0 is the hypothesis that the underlying topic (e.g., Bitcoin, the current president of the United States, etc.) is “good,” and θ_1 is the counter-hypothesis. Then, we construct log-belief of each agent k as:

$$\Lambda_{k,i} = \frac{1}{N_{k,i}} \sum_{t=1}^{N_{k,i}} \log \left(\frac{p_{k,i,t}}{1 - p_{k,i,t}} \right) \quad (48)$$

Note that if $N_{k,i} = 0$ for some agent k at some iteration i , we set $\Lambda_{k,i}$ to previous value of it, i.e. $\Lambda_{k,i} = \Lambda_{k,i-1}$.

D. ADJUSTMENTS TO THE ALGORITHM

We have made some adjustments to the graph social learning algorithm to cope with real life data. In particular, instead of performing gradient updates at each iteration, as in the online stochastic learning algorithm, we use stochastic mini-batches. Namely, we select a window size W , and average the gradients calculated within that window, and then perform the gradient step. In our experiments, we observe that this practice reduces the noise in the gradients due to real life data. Furthermore, we use ℓ_1 regularization to promote sparsity. The motivation for sparsity is to get rid of unnecessary links between different users in the graph. These hyperparameters in the experiments are determined with a grid search on one of the networks (Ben Shapiro), and then the parameters are used in the other two networks. These hyperparameters are: for the stochastic mini-batch size, $W = 30$, for the stepsize $\delta = 0.0001$, for the learning rate $\mu = 0.0003$ and for the ℓ_1 regularization weight $\alpha = 0.006$. The fact that we obtain desirable outcomes in all cases suggests that the algorithm generalizes and performs well across different networks.

E. EXPERIMENTAL RESULTS

We compare the Perron vectors of the learned combination matrix and the original combination matrix (which is constructed by assigning uniform weights to users each user follows). In these comparisons of Perron vectors, we show

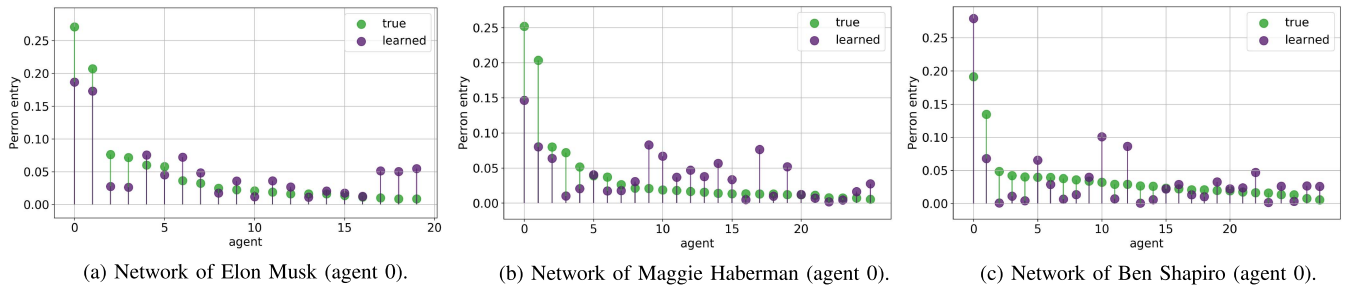


FIGURE 9. Plots of agent centralities. Green points are the entries of the original Perron vector, and purple points belong to the learned Perron vectors. In all networks, we see that the algorithm identifies the most central agent in the graph.

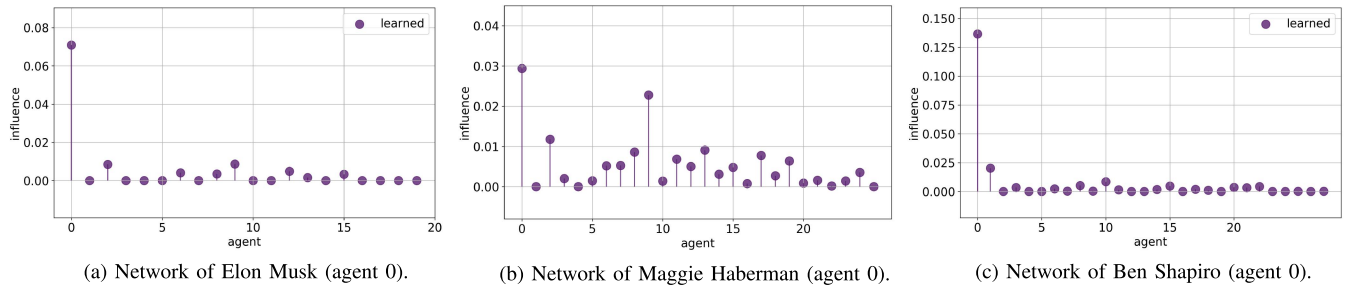


FIGURE 10. Plots of agent influences. In the respective networks containing Elon Musk (CEO – 110 M followers on Twitter), Maggie Haberman (journalist – 1.7 M followers on Twitter) and Ben Shapiro (columnist – 4.5 M followers on Twitter), we can identify these public figures as the most influential agents. For instance, in Fig. 10(a), we can identify Elon Musk as the most influential Twitter user in the corresponding subnetwork.

that we can correctly estimate the most central agents in all three networks. These comparisons can be seen in Fig. 9. Secondly, we show the influence plots, “qualitatively” show that we can indeed identify the popular accounts (Elon Musk, Maggie Haberman and Ben Shapiro) as the most influential agents in their respective networks. For instance, in Fig. 10(a), we can identify Elon Musk as the most influential Twitter user of his network of users.

VIII. CONCLUSION

In this study, we show that a sequence of publicly exchanged beliefs in the adaptive social learning protocol contains rich information about the underlying model. We present an algorithm for learning the agents’ informativeness in terms of KL-divergences between likelihood models, and for identifying a combination graph. We demonstrate that these quantities determine the probability of error of the true hypothesis estimator, and we introduce a notion of a global agent influence, which quantifies the individuals’ contribution to learning. As a result, the suggested approach enables us to determine the most influential agents in the opinion formation process. We also describe how to apply the algorithm to Twitter data. Our experiments on both synthetic data and Twitter data illustrate that we can accurately find global influencers and learn the underlying graph.

APPENDIX A PROOF OF LEMMA 1

From Markov’s inequality, we have

$$\begin{aligned}
 & \mathbb{P} \left(\left\| \frac{1}{M} \sum_{j=i-M}^{i-1} \Lambda_j - \mathbb{E} \Lambda \right\|_F^2 > \varepsilon \right) \\
 & \leq \frac{1}{\varepsilon} \mathbb{E} \left\| \frac{1}{M} \sum_{j=i-M}^{i-1} \Lambda_j - \mathbb{E} \Lambda \right\|_F^2 \\
 & = \frac{1}{M^2 \varepsilon} \mathbb{E} \left\| \sum_{j=i-M}^{i-1} (\Lambda_j - \mathbb{E} \Lambda) \right\|_F^2. \quad (49)
 \end{aligned}$$

The expectation above can be expanded into:

$$\begin{aligned}
 & \mathbb{E} \left\| \sum_{j=i-M}^{i-1} (\Lambda_j - \mathbb{E} \Lambda) \right\|_F^2 = \sum_{j=i-M}^{i-1} \mathbb{E} \left\| \Lambda_j - \mathbb{E} \Lambda \right\|_F^2 \\
 & + 2 \sum_{\substack{j_1, j_2=i-M, \\ j_1 < j_2}}^{i-1} \text{Tr} \left(\mathbb{E} (\Lambda_{j_1} - \mathbb{E} \Lambda) (\Lambda_{j_2} - \mathbb{E} \Lambda)^T \right). \quad (50)
 \end{aligned}$$

Introducing the history

$$\mathcal{F}_i \triangleq \{\xi_{k,j}, j < i, \forall k \in \mathcal{N}\}, \quad (51)$$

which collects all observations up to time i , and conditioning over \mathcal{F}_{j_1+1} , we have

$$\begin{aligned} & \mathbb{E}(\Lambda_{j_1} - \mathbb{E}\Lambda)(\Lambda_{j_2} - \mathbb{E}\Lambda)^\top \\ &= \mathbb{E}\left((\Lambda_{j_1} - \mathbb{E}\Lambda)\mathbb{E}\left(\Lambda_{j_2}^\top - \mathbb{E}\Lambda^\top \middle| \mathcal{F}_{j_1+1}\right)\right) \end{aligned} \quad (52)$$

This equality holds since $\mathbb{E}(\Lambda_{j_1} | \mathcal{F}_{j_1+1}) = \Lambda_{j_1}$. Using the main recursion formula (8), we can represent Λ_{j_2} in terms of Λ_{j_1} and observations \mathcal{L}_t with $j_1 + 1 \leq t \leq j_2$:

$$\begin{aligned} \Lambda_{j_2} &= (1 - \delta)^{j_2-j_1} (A_\star^{j_2-j_1})^\top \Lambda_{j_1} \\ &\quad + \delta \sum_{t=0}^{j_2-j_1-1} (1 - \delta)^t (A_\star^t)^\top \mathcal{L}_{j_2-t} \end{aligned} \quad (53)$$

It follows that

$$\begin{aligned} \mathbb{E}(\Lambda_{j_2} | \mathcal{F}_{j_1+1}) &= (1 - \delta)^{j_2-j_1} (A_\star^{j_2-j_1})^\top \Lambda_{j_1} \\ &\quad + \delta \sum_{t=0}^{j_2-j_1-1} (1 - \delta)^t (A_\star^t)^\top \tilde{\mathcal{L}} \end{aligned} \quad (54)$$

where $\tilde{\mathcal{L}}$ is the expected value of \mathcal{L}_i defined in (16). Using (54), we can rewrite (52) as:

$$\begin{aligned} & \mathbb{E}\left((\Lambda_{j_1} - \mathbb{E}\Lambda)\mathbb{E}\left(\Lambda_{j_2}^\top - \mathbb{E}\Lambda^\top \middle| \mathcal{F}_{j_1+1}\right)\right) \\ &= \mathbb{E}(\Lambda_{j_1} - \mathbb{E}\Lambda)\left((1 - \delta)^{j_2-j_1} (A_\star^{j_2-j_1})^\top \Lambda_{j_1} - \mathbb{E}\Lambda\right)^\top \\ &\quad + \mathbb{E}(\Lambda_{j_1} - \mathbb{E}\Lambda)\left(\delta \sum_{t=0}^{j_2-j_1-1} (1 - \delta)^t (A_\star^t)^\top \tilde{\mathcal{L}}\right)^\top \\ &= \mathbb{E}(\Lambda_{j_1} - \mathbb{E}\Lambda)(\Lambda_{j_1} - \mathbb{E}\Lambda)^\top (1 - \delta)^{j_2-j_1} A_\star^{j_2-j_1} \\ &\quad + \mathbb{E}(\Lambda_{j_1} - \mathbb{E}\Lambda)\mathbb{E}\Lambda^\top \left((1 - \delta)^{j_2-j_1} A_\star^{j_2-j_1} - I\right) \\ &\quad + \mathbb{E}(\Lambda_{j_1} - \mathbb{E}\Lambda)\left(\delta \sum_{t=0}^{j_2-j_1-1} (1 - \delta)^t (A_\star^t)^\top \tilde{\mathcal{L}}\right)^\top \\ &= \mathbb{E}(\Lambda_{j_1} - \mathbb{E}\Lambda)(\Lambda_{j_1} - \mathbb{E}\Lambda)^\top (1 - \delta)^{j_2-j_1} A_\star^{j_2-j_1} \end{aligned} \quad (55)$$

where the last equation holds since in steady state $\mathbb{E}\Lambda_{j_1} = \mathbb{E}\Lambda$. The trace of (55) then becomes²:

$$\begin{aligned} & \text{Tr} \mathbb{E}(\Lambda_{j_1} - \mathbb{E}\Lambda)(\Lambda_{j_1} - \mathbb{E}\Lambda)^\top (1 - \delta)^{j_2-j_1} A_\star^{j_2-j_1} \\ & \leq \mathbb{E}\left(\|\Lambda_{j_1} - \mathbb{E}\Lambda\|_F\right) \\ & \quad \times \left\| (1 - \delta)^{j_2-j_1} (A_\star^{j_2-j_1}) (\Lambda_{j_1} - \mathbb{E}\Lambda) \right\|_F \\ & \leq (1 - \delta)^{j_2-j_1} \|A_\star^{j_2-j_1}\|_F \mathbb{E}\|\Lambda_{j_1} - \mathbb{E}\Lambda\|_F^2 \\ & \leq (1 - \delta)^{j_2-j_1} \sqrt{|\mathcal{M}|} \mathbb{E}\|\Lambda_{j_1} - \mathbb{E}\Lambda\|_F^2, \end{aligned} \quad (56)$$

where the last inequality holds because A_\star and its powers are left-stochastic, i.e. all the entries are non-negative and each column's entries add up to one:

$$\|A_\star\|_F^2 = \sum_k \sum_\ell a_{k\ell}^2 \leq \sum_k \left(\sum_\ell a_{k\ell}\right)^2 = \sum_k 1 = |\mathcal{M}| \quad (57)$$

Next, let us study the following sum:

$$\begin{aligned} & \sum_{\substack{j_1, j_2=i-M, \\ j_1 < j_2}}^{i-1} (1 - \delta)^{j_2-j_1} \\ &= (M - 1)(1 - \delta) + (M - 2)(1 - \delta)^2 + \dots + \\ & \quad + 2(1 - \delta)^{M-2} + (1 - \delta)^{M-1} \\ &= \frac{M - 1 - (1 - \delta)^M}{\delta} - (1 - \delta)\delta^{M-2} = O(M/\delta) \end{aligned} \quad (58)$$

Additionally, under the steady-state condition (13), the following property for Λ holds:

$$\begin{aligned} & \mathbb{E}\|\Lambda - \mathbb{E}\Lambda\|_F^2 \\ &= \mathbb{E} \text{Tr}\left((\Lambda - \mathbb{E}\Lambda)(\Lambda - \mathbb{E}\Lambda)^\top\right) \\ & \stackrel{(13)}{=} \mathbb{E} \text{Tr}\left(\delta \sum_{t_1=0}^{\infty} (1 - \delta)^{t_1} (A_\star^{t_1})^\top (\mathcal{L}_{t_1} - \tilde{\mathcal{L}})\right. \\ & \quad \left. \times \delta \sum_{t_2=0}^{\infty} (1 - \delta)^{t_2} (A_\star^{t_2})^\top (\mathcal{L}_{t_2} - \tilde{\mathcal{L}})\right) \\ &= \delta^2 \mathbb{E} \text{Tr}\left(\sum_{t_1=0}^{\infty} \sum_{t_2=0}^{\infty} (1 - \delta)^{t_1+t_2} (A_\star^{t_1})^\top (\mathcal{L}_{t_1} - \tilde{\mathcal{L}})\right. \\ & \quad \left. \times (\mathcal{L}_{t_2} - \tilde{\mathcal{L}})^\top A_\star^{t_2}\right) \\ &= \delta^2 \sum_{t=0}^{\infty} (1 - \delta)^{2t} \text{Tr}\left((A_\star^t)^\top \mathcal{R}_{\mathcal{L}} A_\star^t\right) = O(\delta) \end{aligned} \quad (59)$$

due to i.i.d. \mathcal{L}_i . Combining the derivations above, expression (49) becomes:

$$\begin{aligned} & \mathbb{P}\left(\left\|\frac{1}{M} \sum_{j=i-M}^{i-1} \Lambda_j - \mathbb{E}\Lambda\right\|_F^2 > \varepsilon\right) \\ & \leq \frac{1}{M^2 \varepsilon} \sum_{j=i-M}^{i-1} \mathbb{E}\|\Lambda_j - \mathbb{E}\Lambda\|_F^2 \\ & \quad + \frac{2}{M^2 \varepsilon} \sum_{j_1 < j_2} (1 - \delta)^{j_2-j_1} \sqrt{|\mathcal{M}|} \mathbb{E}\|\Lambda_{j_1} - \mathbb{E}\Lambda\|_F^2 \end{aligned}$$

²Using $\text{Tr}(AB^\top) \leq \|A\|_F \|B\|_F$.

$$\begin{aligned}
 &= \frac{1}{M\varepsilon} \left(1 + \frac{2\sqrt{[M]}}{M} \sum_{j_1 < j_2} (1-\delta)^{j_2-j_1} \right) \mathbb{E} \|\mathbf{\Lambda} - \mathbb{E}\mathbf{\Lambda}\|_{\mathbb{F}}^2 \\
 &= \frac{1}{M\varepsilon} (1 + O(1/\delta)) O(\delta) \\
 &= O\left(\frac{1}{M\varepsilon}\right) \tag{60}
 \end{aligned}$$

since $\sum_{j_1 < j_2} (1-\delta)^{j_2-j_1} = O(M/\delta)$. Taking $M \rightarrow \infty$, the right-hand side of (60) is zero for any $\varepsilon > 0$. Thus, we establish convergence in probability by definition.

APPENDIX B PROOF OF LEMMA 2

Under some reasonable assumptions on the distribution of the random variables, as required by the dominated convergence theorem in mathematical analysis, it is possible to exchange the expectation and gradient operations [72]. Thus, using (19), the gradient of (21) is given by:

$$\nabla J_i(A) = -(1-\delta)\mathbb{E} \left[\mathbf{\Delta}_{i-1} \left(\mathbf{\Delta}_i^{\top} - (1-\delta)\mathbf{\Delta}_{i-1}^{\top} A \right) \right] \tag{61}$$

We first verify that the risk function $J_i(A)$ has Lipschitz gradients. Note that for any A_1, A_2 :

$$\begin{aligned}
 &\text{Tr} \left((\nabla J_i(A_1) - \nabla J_i(A_2))^{\top} (A_1 - A_2) \right) \\
 &= (1-\delta)^2 \text{Tr} \left(\mathbb{E} \mathbf{\Delta}_{i-1} \mathbf{\Delta}_{i-1}^{\top} (A_1 - A_2) (A_1 - A_2)^{\top} \right) \\
 &\stackrel{(a)}{\leq} (1-\delta)^2 \lambda_{\max} \left(\mathbb{E} \mathbf{\Delta}_{i-1} \mathbf{\Delta}_{i-1}^{\top} \right) \|A_1 - A_2\|_{\mathbb{F}}^2, \tag{62}
 \end{aligned}$$

where step (a) follows from the following considerations. For matrices X and Y of appropriate dimensions, where X is positive definite with eigendecomposition $X = U_X \Lambda_X U_X^{-1}$, the following inequality holds:

$$\begin{aligned}
 &\text{Tr} (X Y Y^{\top}) = \text{Tr} (U_X \Lambda_X U_X^{-1} Y Y^{\top}) \\
 &= \text{Tr} (\Lambda_X U_X^{-1} Y Y^{\top} U_X) \leq \lambda_{\max}(X) \text{Tr} (U_X^{-1} Y Y^{\top} U_X) \\
 &= \lambda_{\max}(X) \|Y\|_{\mathbb{F}}^2. \tag{63}
 \end{aligned}$$

Similarly, we can verify that $J_i(A)$ is strongly convex since

$$\begin{aligned}
 &\text{Tr} \left((\nabla J_i(A_1) - \nabla J_i(A_2))^{\top} (A_1 - A_2) \right) \\
 &\geq (1-\delta)^2 \lambda_{\min} \left(\mathbb{E} \mathbf{\Delta}_{i-1} \mathbf{\Delta}_{i-1}^{\top} \right) \|A_1 - A_2\|_{\mathbb{F}}^{\top}. \tag{64}
 \end{aligned}$$

Next we verify that $\mathbb{E} \mathbf{\Delta}_i \mathbf{\Delta}_i^{\top}$ is positive definite and finite. For this purpose, we refer to [27, Lemma 1], which establishes that $\mathbf{\Lambda}$ given by (13) is element-wise bounded as follows:

$$\begin{aligned}
 |\mathbf{\Lambda}| &\leq \bar{\Lambda} \triangleq \delta b \sum_{t=0}^{\infty} (1-\delta)^t (A_{\star}^t)^{\top} \mathbf{1} \mathbf{1}^{\top} \\
 &= \delta b \left(I - (1-\delta) A_{\star}^{\top} \right)^{-1} \mathbf{1} \mathbf{1}^{\top} \tag{65}
 \end{aligned}$$

Using (65), we can also bound the sample average:

$$\left| \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j \right| \leq \frac{1}{M} \sum_{j=i-M}^{i-1} |\mathbf{\Lambda}_j| = \bar{\Lambda} \tag{66}$$

Then, $\mathbb{E} \mathbf{\Delta}_i \mathbf{\Delta}_i^{\top}$ is also bounded:

$$\begin{aligned}
 &\mathbb{E} \mathbf{\Delta}_i \mathbf{\Delta}_i^{\top} \\
 &= \mathbb{E} \left(\mathbf{\Lambda}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j \right) \left(\mathbf{\Lambda}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j \right)^{\top} \\
 &\preceq \mathbb{E} \left(|\mathbf{\Lambda}_i| + \left| \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j \right| \right) \left(|\mathbf{\Lambda}_i| + \left| \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j \right| \right)^{\top} \\
 &\leq 2\bar{\Lambda} \cdot 2\bar{\Lambda}^{\top} = 4\bar{\Lambda} \bar{\Lambda}^{\top}. \tag{67}
 \end{aligned}$$

From Lemma 1 we know that the following convergence in probability holds:

$$\frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j \xrightarrow{M \rightarrow \infty} \mathbb{E} \mathbf{\Lambda} \tag{68}$$

By definition of convergence in probability, for any $\varepsilon > 0$ and $\pi \in (0, 1)$, there exists M_0 such that for any $M \geq M_0(\pi)$ the probability of the event ω :

$$\omega \triangleq \left\{ \left\| \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j - \mathbb{E} \mathbf{\Lambda} \right\|_{\mathbb{F}}^2 \leq \varepsilon \right\} \tag{69}$$

is bounded as follows:

$$\mathbb{P}(\omega) \geq 1 - \pi. \tag{70}$$

By the law of total expectation and using (70), we can represent $\mathbb{E} \mathbf{\Delta}_i \mathbf{\Delta}_i^{\top}$ as:

$$\begin{aligned}
 &\mathbb{E} \mathbf{\Delta}_i \mathbf{\Delta}_i^{\top} \\
 &= \mathbb{E} \left(\mathbf{\Delta}_i \mathbf{\Delta}_i^{\top} | \omega \right) \mathbb{P}(\omega) + \mathbb{E} \left(\mathbf{\Delta}_i \mathbf{\Delta}_i^{\top} | \bar{\omega} \right) \mathbb{P}(\bar{\omega}) \\
 &\geq (1-\pi) \cdot \mathbb{E} \left(\mathbf{\Delta}_i \mathbf{\Delta}_i^{\top} | \omega \right) + 0 \cdot \mathbb{E} \left(\mathbf{\Delta}_i \mathbf{\Delta}_i^{\top} | \bar{\omega} \right) \tag{71}
 \end{aligned}$$

where $\bar{\omega}$ denotes the complementary event of ω , and $\mathbb{E}(\mathbf{\Delta}_i \mathbf{\Delta}_i^{\top} | \bar{\omega})$ is finite due to finiteness of each $\mathbf{\Lambda}_i$, as established in (65). Next, using definition (20), and the fact that $g(X) = X X^{\top}$ is a continuous bounded function given a bounded input³ we get that:

$$\begin{aligned}
 &\mathbb{E} \left(\mathbf{\Delta}_i \mathbf{\Delta}_i^{\top} | \omega \right) \\
 &= \mathbb{E} \left(\left(\mathbf{\Lambda}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j \right) \left(\mathbf{\Lambda}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_j \right)^{\top} \middle| \omega \right)
 \end{aligned}$$

³If a sequence of random variables $\mathbf{x}_1, \mathbf{x}_2, \dots$ converges in distribution to a random variable \mathbf{x} , then $\mathbb{E} g(\mathbf{x}_i) \xrightarrow{i \rightarrow \infty} \mathbb{E} g(\mathbf{x})$ for a continuous and bounded function $g(\cdot)$.

$$\begin{aligned}
 &\stackrel{(69)}{=} \mathbb{E} (\mathbf{A} - \mathbb{E}\mathbf{A} + O(\sqrt{\varepsilon})) (\mathbf{A} - \mathbb{E}\mathbf{A} + O(\sqrt{\varepsilon}))^T \\
 &= \mathbb{E}(\mathbf{A} - \mathbb{E}\mathbf{A})(\mathbf{A} - \mathbb{E}\mathbf{A})^T + O(\varepsilon) \\
 &= \delta^2 \sum_{t=0}^{\infty} (1 - \delta)^{2t} (A_{\star}^t)^T \mathcal{R}_{\mathcal{L}} A_{\star}^t + O(\varepsilon) \\
 &= \delta^2 \mathcal{R}_{\mathcal{L}} + \sum_{t=1}^{\infty} (1 - \delta)^{2t} (A_{\star}^t)^T \mathcal{R}_{\mathcal{L}} A_{\star}^t + O(\varepsilon) \\
 &\stackrel{(a)}{\geq} \delta^2 \mathcal{R}_{\mathcal{L}} + O(\varepsilon) \geq \tau \delta^2 I + O(\varepsilon) \tag{72}
 \end{aligned}$$

where the last inequality is due to Assumption 2. Thus, (71) becomes:

$$\mathbb{E} \mathbf{\Delta}_i \mathbf{\Delta}_i^T \geq (1 - \pi) (\tau \delta^2 I + O(\varepsilon)) \tag{73}$$

Since we can choose ε as small as possible, let us set $\varepsilon = 1/\sqrt{M}$. Then, using (60), we get that

$$\mathbb{P}(\bar{w}) = O(1/M\varepsilon) = O(1/\sqrt{M}). \tag{74}$$

Therefore, we can take the corresponding π to be on the order of $1/\sqrt{M}$ due to (70). Returning to (73), we get:

$$\begin{aligned}
 \mathbb{E} \mathbf{\Delta}_i \mathbf{\Delta}_i^T &\geq \left(1 + O\left(1/\sqrt{M}\right)\right) (\tau \delta^2 I + O(1/\sqrt{M})) \\
 &= \tau \delta^2 I + O\left(1/\sqrt{M}\right). \tag{75}
 \end{aligned}$$

Thus, $\mathbb{E} \mathbf{\Delta}_i \mathbf{\Delta}_i^T$ is positive-definite for M large enough i.e. as long as $M \gg 1/\delta^4$.

We now examine the relation of the minimizer of $A_{\min,i} \triangleq \min_A J_i(A)$ to the true combination matrix A_{\star} . From strong convexity, the minimizer of $J_i(A)$ is unique and satisfies:

$$\nabla J_i(A_{\min,i}) = 0 \tag{76}$$

Referring to (61), we get

$$-(1 - \delta) \mathbb{E} \mathbf{\Delta}_{i-1} \left(\mathbf{\Delta}_i^T - (1 - \delta) \mathbf{\Delta}_{i-1}^T A_{\min,i} \right) = 0. \tag{77}$$

Therefore,

$$A_{\min,i} = \frac{1}{1 - \delta} \left(\mathbb{E} \mathbf{\Delta}_{i-1} \mathbf{\Delta}_{i-1}^T \right)^{-1} \mathbb{E} \mathbf{\Delta}_{i-1} \mathbf{\Delta}_i^T \tag{78}$$

In turn, using (20), recursion (8) can be modified into

$$\mathbf{\Delta}_i = (1 - \delta) A_{\star}^T \mathbf{\Delta}_{i-1} + \delta \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right) \tag{79}$$

so that

$$\begin{aligned}
 A_{\star} &= \frac{1}{1 - \delta} \left(\mathbb{E} \mathbf{\Delta}_{i-1} \mathbf{\Delta}_{i-1}^T \right)^{-1} \\
 &\quad \times \left(\mathbb{E} \mathbf{\Delta}_{i-1} \mathbf{\Delta}_i^T - \delta \mathbb{E} \mathbf{\Delta}_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right)^T \right) \tag{80}
 \end{aligned}$$

Subtracting (78) and (80) gives:

$$\begin{aligned}
 A_{\min,i} - A_{\star} &= \frac{\delta}{1 - \delta} \left(\mathbb{E} \mathbf{\Delta}_{i-1} \mathbf{\Delta}_{i-1}^T \right)^{-1} \\
 &\quad \times \mathbb{E} \mathbf{\Delta}_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right)^T \tag{81}
 \end{aligned}$$

Iterating (8), we get the following representation for $\mathbf{\Delta}_i$:

$$\mathbf{\Delta}_i = (1 - \delta)^M (A_{\star}^M)^T \mathbf{\Delta}_{i-M} + \delta \sum_{t=0}^{M-1} (1 - \delta)^t (A_{\star}^t)^T \mathcal{L}_{i-t}. \tag{82}$$

Since \mathcal{L}_i are i.i.d and using (81) and the definition of $\mathbf{\Delta}_i$ in (20), we rewrite the expectation in (81) as:

$$\begin{aligned}
 &\mathbb{E} \mathbf{\Delta}_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right)^T \\
 &= \mathbb{E} \left(\mathbf{\Lambda}_{i-1} - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_{j-1} \right) \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right)^T \\
 &\stackrel{(82)}{=} \mathbb{E} \left[\left((1 - \delta)^M (A_{\star}^M)^T \right. \right. \\
 &\quad \left. \left. - \frac{(1 - \delta)^{M-1} (A_{\star}^{M-1})^T + \dots + (1 - \delta) A_{\star}^T + I}{M} \right) \mathbf{\Lambda}_{i-M-1} \right. \\
 &\quad \left. + \delta \left((1 - \delta)^{M-1} (A_{\star}^{M-1})^T \right. \right. \\
 &\quad \left. \left. - \frac{(1 - \delta)^{M-2} (A_{\star}^{M-2})^T + \dots + I}{M} \right) \mathcal{L}_{i-M} \right. \\
 &\quad \left. + \delta \left((1 - \delta)^{M-2} (A_{\star}^{M-2})^T \right. \right. \\
 &\quad \left. \left. - \frac{(1 - \delta)^{M-3} (A_{\star}^{M-3})^T + \dots + I}{M} \right) \mathcal{L}_{i-M+1} \right. \\
 &\quad \left. + \dots \right. \\
 &\quad \left. + \delta \left((1 - \delta) A_{\star}^T - \frac{I}{M} \right) \mathcal{L}_{i-2} \right. \\
 &\quad \left. + \delta \mathcal{L}_{i-1} \right] \times \left[\mathcal{L}_i - \frac{\mathcal{L}_{i-1} + \dots + \mathcal{L}_{i-M}}{M} \right]^T \\
 &= -\frac{\delta}{M} \left(\frac{I}{M} + \frac{2}{M} (1 - \delta) A_{\star}^T + \dots \right. \\
 &\quad \left. + \frac{M-1}{M} (1 - \delta)^{M-2} (A_{\star}^{M-2})^T \right. \\
 &\quad \left. + (1 - \delta)^{M-1} (A_{\star}^T)^{M-1} \right) \mathcal{R}_{\mathcal{L}} \\
 &= O(1/M) I. \tag{83}
 \end{aligned}$$

with

$$\mathcal{R}_{\mathcal{L}} = \mathbb{E} (\mathcal{L}_i - \bar{\mathcal{L}}) (\mathcal{L}_i - \bar{\mathcal{L}})^\top \quad (84)$$

and where we used the fact that

$$\begin{aligned} & \mathbb{E} \left[\Lambda_{i-M-1} \left(\mathcal{L}_i - \frac{\mathcal{L}_{i-1} + \dots + \mathcal{L}_{i-M}}{M} \right)^\top \right] \\ &= \mathbb{E} \Lambda_{i-M-1} \mathbb{E} \left(\mathcal{L}_i - \frac{\mathcal{L}_{i-1} + \dots + \mathcal{L}_{i-M}}{M} \right)^\top \\ &= \mathbb{E} \Lambda_{i-M-1} (\bar{\mathcal{L}} - \bar{\mathcal{L}})^\top = 0. \end{aligned} \quad (85)$$

Thus, combining (75), (81) and (83), we find the following relation between A_\star and A_{\min} :

$$\begin{aligned} & \|A_{\min, i} - A_\star\|_{\mathbb{F}}^2 \\ & \leq \frac{\delta^2}{(1-\delta)^2} \|O(1/M)I\|_{\mathbb{F}}^2 \left\| \left(\mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top \right)^{-1} \right\|_{\mathbb{F}}^2 \\ & \leq \frac{|\mathcal{N}|\delta^2}{\tau^2 \delta^4 (1-\delta)^6} O(1/M^2) = O(1/\delta^2 M^2) \end{aligned} \quad (86)$$

APPENDIX C PROOF OF THEOREM 1

Using recursion (8) for \mathbf{A}_i and the definition of $\widehat{\mathcal{L}}_{i-1}$ in (18), the SGD step (23) can be rewritten as follows:

$$\begin{aligned} \mathbf{A}_i &= \mathbf{A}_{i-1} + \mu(1-\delta)\Delta_{i-1} \\ & \quad \times \left(\Delta_i^\top - (1-\delta)\Delta_{i-1}^\top \mathbf{A}_{i-1} \right) \\ & \stackrel{(79)}{=} \mathbf{A}_{i-1} + \mu(1-\delta)\Delta_{i-1} \\ & \quad \times \left((1-\delta)\Delta_{i-1}^\top (A_\star - \mathbf{A}_{i-1}) \right. \\ & \quad \left. + \delta \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right)^\top \right) \\ &= \mathbf{A}_{i-1} + \mu(1-\delta)^2 \Delta_{i-1} \Delta_{i-1}^\top (A_\star - \mathbf{A}_{i-1}) \\ & \quad + \mu\delta(1-\delta)\Delta_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right)^\top \end{aligned} \quad (87)$$

Recall the deviation from the true combination matrix:

$$\tilde{\mathbf{A}}_i = A_\star - \mathbf{A}_i. \quad (88)$$

Combining (87) with (88), we obtain:

$$\begin{aligned} \tilde{\mathbf{A}}_i &= \left(I - \mu(1-\delta)^2 \Delta_{i-1} \Delta_{i-1}^\top \right) \tilde{\mathbf{A}}_{i-1} \\ & \quad - \mu\delta(1-\delta)\Delta_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right)^\top \\ &= \left(I - \mu(1-\delta)^2 \mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top \right) \tilde{\mathbf{A}}_{i-1} \end{aligned}$$

$$\begin{aligned} & + \mu(1-\delta)^2 \left(\mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top - \Delta_{i-1} \Delta_{i-1}^\top \right) \tilde{\mathbf{A}}_{i-1} \\ & - \mu\delta(1-\delta)\Delta_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right)^\top. \end{aligned} \quad (89)$$

The mean-square deviation is then given by

$$\begin{aligned} & \mathbb{E} \|\tilde{\mathbf{A}}_i\|_{\mathbb{F}}^2 \\ & \leq \rho^2 \left(I - \mu(1-\delta)^2 \mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top \right) \mathbb{E} \|\tilde{\mathbf{A}}_{i-1}\|_{\mathbb{F}}^2 \\ & \quad + \mu^2(1-\delta)^4 \mathbb{E} \left[\left\| \left(\mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top - \Delta_{i-1} \Delta_{i-1}^\top \right) \right\|_{\mathbb{F}}^2 \right. \\ & \quad \left. \times \|\tilde{\mathbf{A}}_{i-1}\|_{\mathbb{F}}^2 \right] \\ & \quad + \mu^2 \delta^2 (1-\delta)^2 \mathbb{E} \left\| \Delta_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right) \right\|_{\mathbb{F}}^2 \\ & \quad - 2\mu\delta(1-\delta) \mathbb{E} \left[\text{Tr} \left(\Delta_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right)^\top \right. \right. \\ & \quad \left. \left. \times \tilde{\mathbf{A}}_{i-1}^\top \left(I - \mu(1-\delta)^2 \Delta_{i-1} \Delta_{i-1}^\top \right) \right) \right] \\ & \quad + 2\mu(1-\delta)^2 \mathbb{E} \left[\text{Tr} \left(\left(\mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top - \Delta_{i-1} \Delta_{i-1}^\top \right) \tilde{\mathbf{A}}_{i-1} \right. \right. \\ & \quad \left. \left. \times \tilde{\mathbf{A}}_{i-1}^\top \left(I - \mu(1-\delta)^2 \mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top \right) \right) \right] \\ &= \rho^2 \left(I - \mu(1-\delta)^2 \mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top \right) \mathbb{E} \|\tilde{\mathbf{A}}_{i-1}\|_{\mathbb{F}}^2 \\ & \quad + \mu^2(1-\delta)^4 \mathbb{E} \left[\left\| \left(\mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top - \Delta_{i-1} \Delta_{i-1}^\top \right) \right\|_{\mathbb{F}}^2 \right. \\ & \quad \left. \times \|\tilde{\mathbf{A}}_{i-1}\|_{\mathbb{F}}^2 \right] \\ & \quad + \mu^2 \delta^2 (1-\delta)^2 \mathbb{E} \left\| \Delta_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right) \right\|_{\mathbb{F}}^2 \\ & \quad + 2\mu^2 \delta (1-\delta)^3 \mathbb{E} \left[\text{Tr} \left(\Delta_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right)^\top \right. \right. \\ & \quad \left. \left. \times \tilde{\mathbf{A}}_{i-1}^\top \Delta_{i-1} \Delta_{i-1}^\top \right) \right] \\ & \quad + 2\mu(1-\delta)^2 \mathbb{E} \left[\text{Tr} \left(\left(\mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top - \Delta_{i-1} \Delta_{i-1}^\top \right) \tilde{\mathbf{A}}_{i-1} \right. \right. \\ & \quad \left. \left. \times \tilde{\mathbf{A}}_{i-1}^\top \left(I - \mu(1-\delta)^2 \mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top \right) \right) \right] \end{aligned}$$

$$-2\mu\delta(1-\delta)\mathbb{E}\left[\text{Tr}\left(\mathbf{\Delta}_{i-1}\left(\mathcal{L}_i-\frac{1}{M}\sum_{j=i-M}^{i-1}\mathcal{L}_j\right)^{\text{T}}\tilde{\mathbf{A}}_{i-1}\right)\right] \quad (90)$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument. Considering the first term:

$$\begin{aligned} & \rho^2\left(I-\mu(1-\delta)^2\mathbb{E}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\right) \\ &= \max\left\{\lambda_{\min}^2\left(I-\mu(1-\delta)^2\mathbb{E}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\right), \right. \\ & \quad \left.\lambda_{\max}^2\left(I-\mu(1-\delta)^2\mathbb{E}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\right)\right\} \\ &= \max\left\{\left(1-\mu(1-\delta)^2\lambda_{\min}\left(\mathbb{E}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\right)\right)^2, \right. \\ & \quad \left.\left(1-\mu(1-\delta)^2\lambda_{\max}\left(\mathbb{E}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\right)\right)^2\right\} \\ &= \max\{(1-\mu\nu)^2, (1-\mu\kappa)^2\} \\ &\leq 1-2\mu\nu+\mu^2\kappa^2=1-2\mu\nu+O(\mu^2), \end{aligned} \quad (91)$$

where the last inequality holds since $\nu\leq\kappa$ by definitions (34) and (35). We omit the time indices in ν_i and κ_i since we are assuming that $\mathbf{\Lambda}_i$ has reached the steady-state distribution (13), and ν and κ are defined as:

$$\begin{aligned} \nu &\triangleq(1-\delta)^2\lambda_{\min}\left(\lim_{i\rightarrow\infty}\mathbb{E}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\right) \\ \kappa &\triangleq(1-\delta)^2\lambda_{\max}\left(\lim_{i\rightarrow\infty}\mathbb{E}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\right) \end{aligned} \quad (92)$$

These constraints are positive by Lemma 2 and exist due to bounded $\mathbf{\Lambda}_i$ in (65). From [27, Lemma 1] it follows that:

$$\begin{aligned} |\mathbf{\Lambda}_i| &\leq\bar{\Lambda}_i\triangleq(1-\delta)^i\left(\mathbf{A}_{\star}^{\text{T}}\right)^i|\Lambda_0| \\ & \quad +\delta b\sum_{t=0}^{i-1}(1-\delta)^t\left(\mathbf{A}_{\star}^{\text{T}}\right)^t\mathbf{1}\mathbf{1}^{\text{T}} \end{aligned} \quad (93)$$

and the upper bound has a finite limit, $\lim_{i\rightarrow\infty}\bar{\Lambda}_i$. Therefore, we can derive another upper bound:

$$\begin{aligned} |\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}| &\leq\frac{1}{M^2}\sum_{j_1=i-M}^{i-1}(\bar{\Lambda}_{i-1}+\bar{\Lambda}_{j_1}) \\ & \quad \times\sum_{j_2=i-M}^{i-1}(\bar{\Lambda}_{i-1}+\bar{\Lambda}_{j_2})^{\text{T}}\triangleq X. \end{aligned} \quad (94)$$

Now, the expectation of the second term in (90) is bounded as follows:

$$\begin{aligned} & \mu^2\mathbb{E}\left(\|(1-\delta)^2\left(\mathbb{E}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}-\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\right)\right\|_{\text{F}}^2 \\ & \quad \times\|\tilde{\mathbf{A}}_{i-1}\|_{\text{F}}^2) \end{aligned}$$

$$\begin{aligned} & \leq^4 2\mu^2\mathbb{E}\left(\|(1-\delta)^2\mathbb{E}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\|_{\text{F}}^2\right. \\ & \quad \left.+\|(1-\delta)^2\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\|_{\text{F}}^2\right) \\ & \quad \times\|\tilde{\mathbf{A}}_{i-1}\|_{\text{F}}^2) \\ & \leq 2\mu^2\left(\|(1-\delta)^2\mathbb{E}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\|_{\text{F}}^2+\|(1-\delta)^2X\|_{\text{F}}^2\right) \\ & \quad \times\mathbb{E}\|\tilde{\mathbf{A}}_{i-1}\|_{\text{F}}^2 \\ & \leq 4\mu^2\|(1-\delta)^2X\|_{\text{F}}^2\mathbb{E}\|\tilde{\mathbf{A}}_{i-1}\|_{\text{F}}^2 \\ & = O(\mu^2)\cdot\mathbb{E}\|\tilde{\mathbf{A}}_{i-1}\|_{\text{F}}^2. \end{aligned} \quad (95)$$

Now, we proceed with the third term in (90). Due to Assumption 1, we have bounded observations \mathcal{L}_i and $\mathbf{\Lambda}_i$ (see (93)), and we conclude that

$$\begin{aligned} & \left|\mathbf{\Delta}_{i-1}\left(\mathcal{L}_i-\frac{1}{M}\sum_{j=i-M}^{i-1}\mathcal{L}_j\right)^{\text{T}}\right| \\ &= \left|\left(\mathbf{\Lambda}_{i-1}-\frac{1}{M}\sum_{j=i-M}^{i-1}\mathbf{\Lambda}_j\right)\left(\mathcal{L}_i-\frac{1}{M}\sum_{j=i-M}^{i-1}\mathcal{L}_j\right)^{\text{T}}\right| \\ &\leq\left(|\mathbf{\Lambda}_{i-1}|+\frac{1}{M}\sum_{j=i-M}^{i-1}|\mathbf{\Lambda}_j|\right)\left(|\mathcal{L}_i|+\frac{1}{M}\sum_{j=i-M}^{i-1}|\mathcal{L}_j|\right)^{\text{T}} \\ &\leq(\bar{\Lambda}+\bar{\Lambda})(b\mathbf{1}\mathbf{1}^{\text{T}}+b\mathbf{1}\mathbf{1}^{\text{T}})^{\text{T}} \\ &= O(1). \end{aligned} \quad (96)$$

Thus,

$$\begin{aligned} & \mu^2\delta^2(1-\delta)^2\mathbb{E}\left\|\mathbf{\Delta}_{i-1}\left(\mathcal{L}_i-\frac{1}{M}\sum_{j=i-M}^{i-1}\mathcal{L}_j\right)^{\text{T}}\right\|_{\text{F}}^2 \\ & \leq\mu^2\delta^2(1-\delta)^2\cdot O(1)=\mu^2\cdot O(\delta^2) \end{aligned} \quad (97)$$

In turn, the fourth term becomes:

$$\begin{aligned} & 2\mu^2\delta(1-\delta)^3\mathbb{E}\left[\text{Tr}\left(\mathbf{\Delta}_{i-1}\left(\mathcal{L}_i-\frac{1}{M}\sum_{j=i-M}^{i-1}\mathcal{L}_j\right)^{\text{T}}\right.\right. \\ & \quad \left.\left.\times\tilde{\mathbf{A}}_{i-1}^{\text{T}}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\right)\right] \\ & \leq^5 \mu^2\delta^2(1-\delta)^6\mathbb{E}\left\|\mathcal{L}_i-\frac{1}{M}\sum_{j=i-M}^{i-1}\mathcal{L}_j\right\|_{\text{F}}^2 \\ & \quad +\mu^2\mathbb{E}\left\|\mathbf{A}_{i-1}^{\text{T}}\mathbf{\Delta}_{i-1}\mathbf{\Delta}_{i-1}^{\text{T}}\mathbf{\Delta}_{i-1}\right\|_{\text{F}}^2 \\ & \stackrel{(a)}{=} \mu^2\delta^2(1-\delta)^6\frac{M+1}{M}\text{Tr}(\mathcal{R}_{\mathcal{L}})+O(\mu^2)\cdot\mathbb{E}\|\tilde{\mathbf{A}}_{i-1}\|_{\text{F}}^2 \\ & = \mu^2O(\delta^2)+O(\mu^2)\cdot\mathbb{E}\|\tilde{\mathbf{A}}_{i-1}\|_{\text{F}}^2 \end{aligned} \quad (98)$$

⁴ $(a-b)^2\leq 2a^2+2b^2$.

⁵ $2\text{Tr}(AB^{\text{T}})\leq\|A\|_{\text{F}}^2+\|B\|_{\text{F}}^2$.

where we use the same steps as in (95), and (a) follows from:

$$\begin{aligned}
 & \mathbb{E} \left\| \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j - \mathcal{L}_i \right\|_F^2 = \mathbb{E} \left\| \frac{1}{M} \sum_{j=i-M}^{i-1} (\mathcal{L}_j - \mathcal{L}_i) \right\|_F^2 \\
 &= \mathbb{E} \operatorname{Tr} \left(\frac{1}{M} \sum_{j=i-M}^{i-1} (\mathcal{L}_j - \mathcal{L}_i) \right) \left(\frac{1}{M} \sum_{j=i-M}^{i-1} (\mathcal{L}_j - \mathcal{L}_i) \right)^\top \\
 &= \frac{1}{M^2} \sum_{j=i-M}^{i-1} \operatorname{Tr} \mathbb{E} \left((\mathcal{L}_j - \mathcal{L}_i) (\mathcal{L}_j - \mathcal{L}_i)^\top \right) \\
 &\quad + \frac{1}{M^2} \sum_{j_1=i-M}^{i-1} \sum_{j_2 \neq j_1}^{i-1} \operatorname{Tr} \mathbb{E} \left((\mathcal{L}_{j_1} - \mathcal{L}_i) (\mathcal{L}_{j_2} - \mathcal{L}_i)^\top \right) \\
 &= \frac{1}{M^2} \sum_{j=i-M}^{i-1} \operatorname{Tr} \left(\mathbb{E} \mathcal{L}_j \mathcal{L}_j^\top - 2\bar{\mathcal{L}} \bar{\mathcal{L}}^\top + \mathbb{E} \mathcal{L}_i \mathcal{L}_i^\top \right) \\
 &\quad + \frac{1}{M^2} \sum_{j_1=i-M}^{i-1} \sum_{j_2 \neq j_1}^{i-1} \operatorname{Tr} \left(\mathbb{E} \mathcal{L}_{j_1} \mathcal{L}_{j_2}^\top - \bar{\mathcal{L}} \bar{\mathcal{L}}^\top \right) \\
 &= \frac{2}{M} \operatorname{Tr}(\mathcal{R}_{\mathcal{L}}) + \frac{M-1}{M} \operatorname{Tr}(\mathcal{R}_{\mathcal{L}}) = \frac{M+1}{M} \operatorname{Tr}(\mathcal{R}_{\mathcal{L}}) \quad (99)
 \end{aligned}$$

To find the expectation of the second trace term in (90), consider first the conditional expectation, and then use Assumption 3:

$$\begin{aligned}
 & \mathbb{E} \left[\operatorname{Tr} \left(\left(\mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top - \Delta_{i-1} \Delta_{i-1}^\top \right) \tilde{\mathbf{A}}_{i-1} \tilde{\mathbf{A}}_{i-1}^\top \right. \right. \\
 &\quad \left. \left. \times \left(I - \mu(1-\delta)^2 \mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top \right) \right) \middle| \mathcal{F}_{i-M-1} \right] \\
 &= \operatorname{Tr} \left(\left(\mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top - \mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top \right) \right. \\
 &\quad \left. \times \mathbb{E} \left[\tilde{\mathbf{A}}_{i-1} \tilde{\mathbf{A}}_{i-1}^\top \middle| \mathcal{F}_{i-M-1} \right] \right. \\
 &\quad \left. \times \left(I - \mu(1-\delta)^2 \mathbb{E} \Delta_{i-1} \Delta_{i-1}^\top \right) \right) \\
 &= 0 \quad (100)
 \end{aligned}$$

where we use the fact that Δ_{i-1} depends only on $\mathbf{A}_{i-M-1}, \dots, \mathbf{A}_{i-1}$. Finally, consider the last term of (90). Using Assumption 3, we obtain:

$$\begin{aligned}
 & -2\mu\delta(1-\delta) \mathbb{E} \left[\operatorname{Tr} \left(\Delta_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right) \tilde{\mathbf{A}}_{i-1} \right) \right] \\
 &= -2\mu\delta(1-\delta) \\
 &\quad \times \mathbb{E} \left[\operatorname{Tr} \left(\mathbb{E} \left(\Delta_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right) \middle| \mathcal{F}_{i-M-1} \right) \right) \right]
 \end{aligned}$$

$$\begin{aligned}
 & \times \mathbb{E} \left(\tilde{\mathbf{A}}_{i-1} \middle| \mathcal{F}_{i-M-1} \right) \Bigg] \\
 &\stackrel{(85)}{=} -2\mu\delta(1-\delta) \mathbb{E} \left[\operatorname{Tr} \left(\mathbb{E} \left(\Delta_{i-1} \left(\mathcal{L}_i - \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j \right) \right) \right) \right. \\
 &\quad \left. \times \mathbb{E} \left(\tilde{\mathbf{A}}_{i-1} \middle| \mathcal{F}_{i-M-1} \right) \right] \\
 &\stackrel{(83)}{=} \mu\delta(1-\delta) \mathbb{E} \left[\operatorname{Tr} \left(O(1/M) \mathbb{E} \left(\tilde{\mathbf{A}}_{i-1} \middle| \mathcal{F}_{i-M-1} \right) \right) \right] \\
 &= \mu\delta(1-\delta) \mathbb{E} \left[\operatorname{Tr} \left(O(1/M) \tilde{\mathbf{A}}_{i-1} \right) \right] \\
 &\leq 6 \mu \frac{(1-\delta)^2}{\delta} O(1/M^2) + \mu\delta^3 \mathbb{E} \|\tilde{\mathbf{A}}_{i-1}\|_F^2 \\
 &= \mu O(1/\delta M^2) + \mu \mathbb{E} \|\tilde{\mathbf{A}}_{i-1}\|_F^2 \cdot O(\delta^3). \quad (101)
 \end{aligned}$$

where in (83) the conditional expectation on \mathcal{F}_{i-M-1} is equal to the full expectation because (85) equals to zero for both full and conditional on \mathcal{F}_{i-M-1} expectations. Summarizing the derivations above, we can transform (90) into:

$$\mathbb{E} \|\tilde{\mathbf{A}}_i\|_F^2 \leq \alpha \mathbb{E} \|\tilde{\mathbf{A}}_{i-1}\|_F^2 + \mu^2 \gamma + \mu c, \quad (102)$$

with $\gamma = O(\delta^2)$, $c = O(1/\delta M^2)$ and $\alpha = 1 - \mu(2\nu + O(\delta^3)) + O(\mu^2)$. Thus,

$$\begin{aligned}
 \mathbb{E} \|\tilde{\mathbf{A}}_i\|_F^2 &\leq \alpha^i \|\tilde{\mathbf{A}}_0\|_F^2 + (\mu^2 \gamma + \mu c) \sum_{t=0}^{i-1} \alpha^t \\
 &= \alpha^i \|\tilde{\mathbf{A}}_0\|_F^2 + (\mu^2 \gamma + \mu c) \frac{1-\alpha^i}{1-\alpha}. \quad (103)
 \end{aligned}$$

For small enough μ , α is strictly less than one. Hence, we obtain the following limiting MSD:

$$\begin{aligned}
 \limsup_{i \rightarrow \infty} \mathbb{E} \|\tilde{\mathbf{A}}_i\|_F^2 &\leq \frac{\mu^2 \gamma + \mu c}{1-\alpha} \\
 &= O(\mu) + O(1/\delta^3 M^2) \quad (104)
 \end{aligned}$$

APPENDIX B PROOF OF THEOREM 2

First, using recursion (8), we rewrite (18) as:

$$\begin{aligned}
 \hat{\mathcal{L}}_{i-1}(\mathbf{A}_i) &= \frac{1}{\delta M} \sum_{j=i-M}^{i-1} \left(\mathbf{A}_j - (1-\delta) \mathbf{A}_i^\top \mathbf{A}_{j-1} \right) \\
 &= \frac{1}{\delta M} \sum_{j=i-M}^{i-1} \left((1-\delta) \tilde{\mathbf{A}}_i^\top \mathbf{A}_{j-1} + \delta \mathcal{L}_j \right) \quad (105)
 \end{aligned}$$

Consider the mean square deviation of $\hat{\mathcal{L}}_i$:

$$\mathbb{E} \|\hat{\mathcal{L}}_{i-1} - \bar{\mathcal{L}}\|_F^2$$

$$\begin{aligned}
 &= \mathbb{E} \left\| \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j - \bar{\mathcal{L}} \right\|_F^2 \\
 &\quad + \frac{(1-\delta)^2}{\delta^2} \mathbb{E} \left\| \tilde{\mathbf{A}}_i^\top \cdot \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_{j-1} \right\|_F^2 \\
 &\quad + 2 \frac{1-\delta}{\delta} \mathbb{E} \operatorname{Tr} \left(\frac{1}{M} \tilde{\mathbf{A}}_i^\top \sum_j \mathbf{\Lambda}_{j-1} \right. \\
 &\quad \left. \times \left(\frac{1}{M} \sum_j \mathcal{L}_j - \bar{\mathcal{L}} \right)^\top \right). \tag{106}
 \end{aligned}$$

Consider the first norm:

$$\begin{aligned}
 &\mathbb{E} \left\| \frac{1}{M} \sum_{j=i-M}^{i-1} \mathcal{L}_j - \bar{\mathcal{L}} \right\|_F^2 = \mathbb{E} \left\| \frac{1}{M} \sum_{j=i-M}^{i-1} (\mathcal{L}_j - \bar{\mathcal{L}}) \right\|_F^2 \\
 &= \mathbb{E} \operatorname{Tr} \left(\frac{1}{M} \sum_{i=j-M}^{i-1} (\mathcal{L}_j - \bar{\mathcal{L}}) \right) \left(\frac{1}{M} \sum_{i=j-M}^{i-1} (\mathcal{L}_j - \bar{\mathcal{L}}) \right)^\top \\
 &= \frac{1}{M^2} \sum_{i=j-M}^{i-1} \operatorname{Tr} \mathbb{E} \left((\mathcal{L}_j - \bar{\mathcal{L}}) (\mathcal{L}_j - \bar{\mathcal{L}})^\top \right) \\
 &= \frac{1}{M} \operatorname{Tr}(\mathcal{R}_{\mathcal{L}}) \tag{107}
 \end{aligned}$$

Let us study the second term of (106) using Assumption 3 and Lemma 1. Conditioning on the history (51), we get

$$\begin{aligned}
 &\mathbb{E} \left(\mathbb{E} \left(\left\| \tilde{\mathbf{A}}_i^\top \cdot \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_{j-1} \right\|_F^2 \middle| \mathcal{F}_{i-M-1} \right) \right) \\
 &\leq \mathbb{E} \left(\mathbb{E} \left(\left\| \tilde{\mathbf{A}}_i^\top \right\|_F^2 \cdot \left\| \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_{j-1} \right\|_F^2 \middle| \mathcal{F}_{i-M-1} \right) \right) \\
 &= \mathbb{E} \left(\mathbb{E} \left(\left\| \tilde{\mathbf{A}}_i^\top \right\|_F^2 \middle| \mathcal{F}_{i-M-1} \right) \right. \\
 &\quad \left. \times \mathbb{E} \left(\left\| \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_{j-1} \right\|_F^2 \middle| \mathcal{F}_{i-M-1} \right) \right) \tag{108}
 \end{aligned}$$

According to (66), we can bound:

$$\mathbb{E} \left(\left\| \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_{j-1} \right\|_F^2 \middle| \mathcal{F}_{i-M-1} \right) \leq \|\bar{\mathbf{\Lambda}}\|_F^2 \tag{109}$$

Therefore, (108) can be bounded as follows:

$$\begin{aligned}
 &\mathbb{E} \left(\mathbb{E} \left(\left\| \tilde{\mathbf{A}}_i^\top \cdot \frac{1}{M} \sum_{j=i-M}^{i-1} \mathbf{\Lambda}_{j-1} \right\|_F^2 \middle| \mathcal{F}_{i-M-1} \right) \right) \\
 &\leq \mathbb{E} \|\tilde{\mathbf{A}}_i\|_F^2 \cdot O(1) \tag{110}
 \end{aligned}$$

Now, consider the trace term in (106):

$$\begin{aligned}
 &\mathbb{E} \operatorname{Tr} \left(\tilde{\mathbf{A}}_i^\top \cdot \frac{1}{M} \sum_j \mathbf{\Lambda}_{j-1} \left(\frac{1}{M} \sum_j \mathcal{L}_j^\top - \bar{\mathcal{L}}^\top \right) \right) \\
 &= \mathbb{E} \operatorname{Tr} \left(\mathbb{E} \left(\tilde{\mathbf{A}}_i^\top \cdot \frac{1}{M} \sum_j \mathbf{\Lambda}_{j-1} \right. \right. \\
 &\quad \left. \left. \times \left(\frac{1}{M} \sum_j \mathcal{L}_j^\top - \bar{\mathcal{L}}^\top \right) \middle| \mathcal{F}_{i-M-1} \right) \right) \\
 &\stackrel{(a)}{=} \mathbb{E} \operatorname{Tr} \left(\mathbb{E} \left(\tilde{\mathbf{A}}_i^\top \middle| \mathcal{F}_{i-M-1} \right) \right. \\
 &\quad \left. \times \mathbb{E} \left(\frac{1}{M} \sum_j \mathbf{\Lambda}_{j-1} \left(\frac{1}{M} \sum_j \mathcal{L}_j^\top - \bar{\mathcal{L}}^\top \right) \middle| \mathcal{F}_{i-M-1} \right) \right) \\
 &\stackrel{(b)}{=} \mathbb{E} \operatorname{Tr} \left(\tilde{\mathbf{A}}_i^\top \cdot O(1/M) I \right) \\
 &\leq^6 \frac{1}{2} \mathbb{E} \|\tilde{\mathbf{A}}_i\|_F^2 + O(1/M^2) \tag{111}
 \end{aligned}$$

where (a) holds due to Assumption 3 and (b) holds due the trick similar to (83). Summarizing the derivations above, we can upper bound the expectation (106) as:

$$\begin{aligned}
 &\mathbb{E} \|\widehat{\mathcal{L}}_{i-1} - \bar{\mathcal{L}}\|_F^2 \\
 &\leq \frac{1}{M} \operatorname{Tr}(\mathcal{R}_{\mathcal{L}}) \\
 &\quad + \frac{(1-\delta)^2}{\delta^2} \mathbb{E} \|\tilde{\mathbf{A}}_i\|_F^2 \cdot O(1) \\
 &\quad + \frac{1-\delta}{\delta} (\mathbb{E} \|\tilde{\mathbf{A}}_i\|_F^2 + O(1/M^2)) \\
 &= \frac{1}{M} \operatorname{Tr}(\mathcal{R}_{\mathcal{L}}) + \frac{1}{\delta^2} \mathbb{E} \|\tilde{\mathbf{A}}_i\|_F^2 \cdot O(1) + O(1/\delta M^2). \tag{112}
 \end{aligned}$$

By Theorem 1, we derive the following MSD in the limit:

$$\begin{aligned}
 &\limsup_{i \rightarrow \infty} \mathbb{E} \|\widehat{\mathcal{L}}_{i-1} - \bar{\mathcal{L}}\|_F^2 \\
 &\leq \frac{1}{M} \operatorname{Tr}(\mathcal{R}_{\mathcal{L}}) + O(\mu/\delta^2) + O(1/\delta^5 M^2) \tag{113}
 \end{aligned}$$

APPENDIX E

In this section, we describe how we generate Bernoulli likelihood model parameters. Under hypothesis θ , each agent k receives observation 0 with probability $p_k(\theta)$, and observation 1 with probability $q_k(\theta) = 1 - p_k(\theta)$. For hypotheses θ_0 , we fix the probabilities as follows:

$$p_k(\theta_0), q_k(\theta_0) = [0.3, 0.7] \tag{114}$$

And for all other $\theta_i \neq \theta_0$, we generate $p_k(\theta_i)$ and $q_k(\theta_i)$ with the following procedure:

$$\varepsilon_{p_k}, \varepsilon_{q_k} \sim \mathcal{G}(0, 1), \tag{115}$$

and

$$p_k(\theta_i), q_k(\theta_i) = \frac{1}{p_k(\theta_0) + \sigma_k^2 \varepsilon_{p_k} + q_k(\theta_0) + \sigma_k^2 \varepsilon_{q_k}} \times [p_k(\theta) + \sigma_k^2 \varepsilon_{p_k}, q_k(\theta_0) + \sigma_k^2 \varepsilon_{q_k}] \quad (116)$$

For “non-influential” agents, we take $\sigma_k^2 = 0.05$, while for “influential” ones we use $\sigma_k^2 = 0.5$ to enlarge the KL-divergences between different states. For additional comparison in Fig. 5, we take $\sigma_k^2 = 0.2$ for “influential” agents with smaller (but yet significant) KL-divergences.

ACKNOWLEDGMENT

We thank our colleague Virginia Bordignon for her valuable suggestions regarding the application to Twitter data.

REFERENCES

- [1] V. Shumovskaia, M. Kayaalp, and A. H. Sayed, “Identifying opinion influencers over social networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.
- [2] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, “Non-Bayesian social learning,” *Games Econ. Behav.*, vol. 76, no. 1, pp. 210–225, 2012.
- [3] A. Nedić, A. Olshevsky, and C. A. Uribe, “Fast convergence rates for distributed non-Bayesian learning,” *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5538–5553, Nov. 2017.
- [4] P. Molavi, A. Tahbaz-Salehi, and A. Jadbabaie, “Foundations of non-Bayesian social learning,” *Columbia Bus. Sch. Res. Paper*, pp. 15–95, 2017.
- [5] P. Molavi, A. Tahbaz-Salehi, and A. Jadbabaie, “A theory of non-Bayesian social learning,” *Econometrica*, vol. 86, no. 2, pp. 445–490, 2018.
- [6] V. Bordignon, V. Matta, and A. H. Sayed, “Adaptive social learning,” *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 6053–6081, Sep. 2021.
- [7] V. Bordignon, V. Matta, and A. H. Sayed, “Social learning with partial information sharing,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 5540–5544.
- [8] A. Lalitha, T. Javidi, and A. D. Sarwate, “Social learning and distributed hypothesis testing,” *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6161–6179, Sep. 2018.
- [9] X. Zhao and A. H. Sayed, “Learning over social networks via diffusion adaptation,” in *Proc. Asilomar Conf. Signals, Syst. Comput.*, 2012, pp. 709–713.
- [10] D. Gale and S. Kariv, “Bayesian learning in social networks,” *Games Econ. Behav.*, vol. 45, no. 2, pp. 329–346, 2003.
- [11] D. Acemoglu, M. A. Dahleh, I. Lobel, and A. Ozdaglar, “Bayesian learning in social networks,” *Rev. Econ. Stud.*, vol. 78, no. 4, pp. 1201–1236, 2011.
- [12] Y. Inan, M. Kayaalp, E. Telatar, and A. H. Sayed, “Social learning under randomized collaborations,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2022, pp. 115–120.
- [13] M. T. Toghiani and C. A. Uribe, “Communication-efficient distributed cooperative learning with compressed beliefs,” in *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 3, pp. 1215–1226, Sep. 2022, doi: [10.1109/TCNS.2022.3198791](https://doi.org/10.1109/TCNS.2022.3198791).
- [14] J. Z. Hare, C. A. Uribe, L. Kaplan, and A. Jadbabaie, “Non-Bayesian social learning with uncertain models,” *IEEE Trans. Signal Process.*, vol. 68, pp. 4178–4193, 2020.
- [15] C. A. Uribe, A. Olshevsky, and A. Nedić, “Nonasymptotic concentration rates in cooperative learning—Part I: Variational non-Bayesian social learning,” *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 3, pp. 1128–1140, Sep. 2022.
- [16] C. A. Uribe, A. Olshevsky, and A. Nedić, “Nonasymptotic concentration rates in cooperative learning—Part II: Inference on compact hypothesis sets,” *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 3, pp. 1141–1153, Sep. 2022.
- [17] M. Kayaalp, V. Bordignon, S. Vlaski, V. Matta, and A. H. Sayed, “Distributed Bayesian learning of dynamic states,” Dec. 2022, *arXiv:2212.02565*.
- [18] J. Hkzla, A. Jadbabaie, E. Mossel, and M. A. Rahimian, “Bayesian decision making in groups is hard,” *Operations Res.*, vol. 69, no. 2, pp. 632–654, 2021.
- [19] M. Kayaalp, Y. Inan, E. Telatar, and A. H. Sayed, “On the arithmetic and geometric fusion of beliefs for distributed inference,” 2022, *arXiv:2204.13741*.
- [20] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, “Interpretable machine learning: Fundamental principles and 10 grand challenges,” *Statist. Surv.*, vol. 16, pp. 1–85, 2022.
- [21] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, “Collective explainable AI: Explaining cooperative strategies and agent contribution in multiagent reinforcement learning with shapley values,” *IEEE Comput. Intell. Mag.*, vol. 17, no. 1, pp. 59–71, Feb. 2022.
- [22] J. J. Ohana, S. Ohana, E. Benhamou, D. Sautiel, and B. Guez, “Explainable AI (XAI) models applied to the multi-agent environment of financial markets,” in *Proc. Workshop Explainable Transparent AI Multi-Agent Syst.*, 2021, pp. 189–207.
- [23] Z. Juozapaitis, A. Koul, A. Fern, M. Erwig, and F. Doshi-Velez, “Explainable reinforcement learning via reward decomposition,” in *Proc. IJCAI/ECACI Workshop Explainable Artif. Intell.*, 2019, pp. 47–53.
- [24] J. Ho and C.-M. Wang, “Explainable and adaptable augmentation in knowledge attention network for multi-agent deep reinforcement learning systems,” in *Proc. IEEE 3rd Int. Conf. Artif. Intell. Knowl. Eng.*, 2020, pp. 157–161.
- [25] M. Brandao, M. Mansouri, A. Mohammed, P. Luff, and A. Coles, “Explainability in multi-agent path/motion planning: User-study-driven taxonomy and requirements,” in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2022, pp. 172–180.
- [26] A. Georgara, J. A. R. Aguilar, and C. Sierra, “Building contrastive explanations for multi-agent team formation,” in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2022, pp. 516–524.
- [27] V. Shumovskaia, K. Ntemos, S. Vlaski, and A. H. Sayed, “Explainability and graph learning from social interactions,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 946–959, 2022.
- [28] V. Shumovskaia, K. Ntemos, S. Vlaski, and A. H. Sayed, “Online graph learning from social interactions,” in *Proc. Asilomar Conf. Signals, Syst., Comput.*, 2021, pp. 1263–1267.
- [29] J. Bisbee, J. Larson, and K. Munger, “#polisci Twitter: A descriptive analysis of how political scientists use Twitter in 2019,” *Perspectives Politics*, vol. 20, no. 3, pp. 879–900, 2022.
- [30] D. Quercia, J. Ellis, L. Capra, and J. Crowcroft, “In the mood for being influential on twitter,” in *Proc. IEEE 3rd Int. Conf. Privacy, Secur., Risk Trust IEEE 3rd Int. Conf. Social Comput.*, 2011, pp. 307–314.
- [31] S. Jain and A. Sinha, “Identification of influential users on twitter: A novel weighted correlated influence measure for Covid-19,” *Chaos, Solitons Fractals*, vol. 139, 2020, Art. no. 110037. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0966077920304355>
- [32] M. Åkerlund, “The importance of influential users in (re) producing Swedish far-right discourse on Twitter,” *Eur. J. Commun.*, vol. 35, no. 6, pp. 613–628, 2020, doi: [10.1177/0267323120940909](https://doi.org/10.1177/0267323120940909).
- [33] M. Pennacchiotti and A.-M. Popescu, “A machine learning approach to Twitter user classification,” in *Proc. Int. AAAI Conf. Web Social Media*, 2011, vol. 5, no. 1, pp. 281–288.
- [34] J. Lin and A. Kolcz, “Large-scale machine learning at Twitter,” in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2012, pp. 793–804.
- [35] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, “Machine learning-based sentiment analysis for Twitter accounts,” *Math. Comput. Appl.*, vol. 23, no. 1, 2018, Art. no. 11.
- [36] S. Zervoudakis, E. Marakakis, H. Kondylakis, and S. Goumas, “Opinionmine: A Bayesian-based framework for opinion mining using twitter data,” *Mach. Learn. Appl.*, vol. 3, 2021, Art. no. 100018.
- [37] D. Camacho, Á. Panizo-LLedot, G. Bello-Orgaz, A. Gonzalez-Pardo, and E. Cambria, “The four dimensions of social network analysis: An overview of research methods, applications, and software tools,” *Inf. Fusion*, vol. 63, pp. 88–120, 2020.
- [38] D. Kempe, J. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network,” in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2003, pp. 137–146.

- [39] D. Kempe, J. Kleinberg, and É. Tardos, "Influential nodes in a diffusion model for social networks," in *Proc. Intern. Colloq. Automata, Lang., Program.*, 2005, pp. 1127–1138.
- [40] Y. Li, J. Fan, Y. Wang, and K.-L. Tan, "Influence maximization on social graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 10, pp. 1852–1872, Oct. 2018.
- [41] M. Gomez-Rodriguez, L. Song, N. Du, H. Zha, and B. Schölkopf, "Influence estimation and maximization in continuous-time diffusion networks," *ACM Trans. Inf. Syst.*, vol. 34, no. 2, pp. 1–33, 2016.
- [42] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "Deepinf: Social influence prediction with deep learning," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2018, pp. 2110–2119.
- [43] W. Karoui, N. Hafiene, and L. Ben Romdhane, "Machine learning-based method to predict influential nodes in dynamic social networks," *Social Netw. Anal. Mining*, vol. 12, no. 1, pp. 1–18, 2022.
- [44] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2001, pp. 57–66.
- [45] K. Saito, R. Nakano, and M. Kimura, "Prediction of information diffusion probabilities for independent cascade model," in *Proc. Int. Conf. Knowl.-Based Intell. Inf. Eng. Syst.*, 2008, pp. 67–75.
- [46] M. Granovetter, "Threshold models of collective behavior," *Amer. J. Sociol.*, vol. 83, no. 6, pp. 1420–1443, 1978.
- [47] W. Chen, W. Lu, and N. Zhang, "Time-critical influence maximization in social networks with time-delayed diffusion process," in *Proc. AAAI Conf. Artif. Intell.*, 2012, pp. 592–598.
- [48] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 235–244.
- [49] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Lett.*, vol. 12, no. 3, pp. 211–223, 2001.
- [50] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Artif. Intell. Statist.*, 2016, pp. 920–929.
- [51] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, Sep. 2017.
- [52] S. Vlaski, H. P. Maretic, R. Nassif, P. Frossard, and A. H. Sayed, "Online graph learning from sequential data," in *Proc. IEEE Data Sci. Workshop*, 2018, pp. 190–194.
- [53] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, May 2019.
- [54] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 3, pp. 481–496, Sep. 2018.
- [55] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 484–499, Sep. 2017.
- [56] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, "Learning sparse graphs under smoothness prior," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 6508–6512.
- [57] R. Shafipour, S. Segarra, A. G. Marques, and G. Mateos, "Network topology inference from non-stationary graph signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 5870–5874.
- [58] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology identification from spectral templates," in *Proc. IEEE Stat. Signal Process. Workshop*, 2016, pp. 1–5.
- [59] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "Graph topology inference based on transform learning," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2016, pp. 356–360.
- [60] H. P. Maretic, D. Thanou, and P. Frossard, "Graph learning under sparsity priors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 6523–6527.
- [61] M. Cirillo, V. Matta, and A. H. Sayed, "Learning Bollobás-Riordan graphs under partial observability," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5360–5364.
- [62] Y. He and H.-T. Wai, "Online inference for mixture model of streaming graph signals with sparse excitation," *IEEE Trans. Signal Process.*, vol. 70, pp. 6419–6433, 2022.
- [63] S. S. Saboksayr and G. Mateos, "Accelerated graph learning from smooth signals," *IEEE Signal Process. Lett.*, vol. 28, pp. 2192–2196, 2021.
- [64] G. Neglia, G. Calbi, D. Towsley, and G. Vardoyan, "The role of network topology for distributed machine learning," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 2350–2358.
- [65] S. Shahrapour, A. Rakhlin, and A. Jadbabaie, "Distributed detection: Finite-time analysis and impact of network topology," *IEEE Trans. Autom. Control*, vol. 61, no. 11, pp. 3256–3268, Nov. 2016.
- [66] P. Hu, V. Bordignon, S. Vlaski, and A. H. Sayed, "Optimal combination policies for adaptive social learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 5842–5846.
- [67] V. Matta, V. Bordignon, A. Santos, and A. H. Sayed, "Interplay between topology and social learning over weak graphs," *IEEE Open J. Signal Process.*, vol. 1, pp. 99–119, 2020.
- [68] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
- [69] V. Matta, A. Santos, and A. H. Sayed, "Graph learning with partial observations: Role of degree concentration," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 1312–1316.
- [70] A. Brovelli, M. Ding, A. Ledberg, Y. Chen, R. Nakamura, and S. L. Bressler, "Beta oscillations in a large-scale sensorimotor cortical network: Directional influences revealed by granger causality," *Proc. Nat. Acad. Sci.*, vol. 101, no. 26, pp. 9849–9854, 2004.
- [71] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY, USA: Cambridge Univ. Press, 2013.
- [72] A. H. Sayed, *Inference and Learning From Data*, vol. 1–3. Cambridge, U.K.: Cambridge Univ. Press, 2022.
- [73] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2008.
- [74] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Found. Trends Mach. Learn.*, vol. 7, no. 4/5, pp. 311–801, 2014. doi:10.1561/22000000051.
- [75] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.
- [76] D. Loureiro, F. Barbieri, L. Neves, L. E. Anke, and J. Camacho-Collados, "TimeLMs: Diachronic language models from Twitter," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics: Syst. Demonstrations*, 2022, pp. 251–260.