Solidago: A Modular Collaborative Scoring Pipeline

Lê Nguyên Hoang\textsuperscript{1,2}, Romain Beylerian\textsuperscript{1}, Bélangère Colbois\textsuperscript{3}, Julien Fageot\textsuperscript{1}, Louis Faucon\textsuperscript{1}, Aidan Jungo\textsuperscript{1}, Alain Le Noac’h\textsuperscript{1}, Adrien Matissart\textsuperscript{1}, and Oscar Villemaud\textsuperscript{3}

\textsuperscript{1}Tournesol Association, Switzerland  
\textsuperscript{2}Calicarpa, Switzerland  
\textsuperscript{3}EPFL, Switzerland

September 27, 2024

Abstract

This paper presents Solidago, an end-to-end modular pipeline to allow any community of users to collaboratively score any number of entities. Solidago proposes a six-module decomposition. First, it uses pretrust and peer-to-peer vouches to assign trust scores to users. Second, based on participation, trust scores are turned into voting rights per user per entity. Third, for each user, a preference model is learned from the user’s evaluation data. Fourth, users’ models are put on a similar scale. Fifth, these models are securely aggregated. Sixth, models are post-processed to yield human-readable global scores. We also propose default implementations of the six modules, including a novel trust propagation algorithm, and adaptations of state-of-the-art scaling and aggregation solutions. Our pipeline has been successfully deployed on the open-source platform tournesol.app. We thereby lay an appealing foundation for the collaborative, effective, scalable, fair, interpretable and secure scoring of any set of entities.

1 Introduction

In 2021, Twitter initiated a vote-based system, now known as the Community Notes, whereby users can not only propose the addition of a contextual note to the posts that are published on the platform, but also weigh in on which of other users’ contextual notes ought to be shown. The governance of the Community Notes is transparent and fully community-driven, without any particular right for Twitter to intervene \cite{50}. While this proposal is inspiring, \cite{12} point out that the note validation process was too slow to prevent an effective reduction of misinformation spread. More importantly, the Community Notes have been argued to be infiltrated by disinformation groups, whose coordinated attacks are endangering the value of the system \cite{17}. Unfortunately, building secure community-driven systems that appropriately moderate and prioritize information is arguably under-researched. As a result, today’s algorithms are mostly designed, managed and governed in a relatively unilateral and opaque manner. As exposed by the Facebook Files \cite{26}, they are benefiting from an alarming lack of accountability. Our paper presents a contribution to the algorithmic governance toolbox, and to the understanding of its challenges. More precisely, we provide an end-to-end modular pipeline, which we instantiate with state-of-the-art algorithms, to allow any community of non-technical users to securely and collaboratively score any number of entities. More specifically, we make the following contributions.

Contributions. Our main contribution is to introduce a modular end-to-end collaborative scoring pipeline called Solidago\textsuperscript{1}. Its six modules are (1) trust propagation, (2) voting rights assignment, (3) preference

\textsuperscript{1}Solidago stands for “Solid Algorithmic Governance”. It is also the name of a flower, also known as “goldenrods”.

1 arXiv:2211.01179v3 [cs.SI] 25 Sep 2024
learning, (4) model scaling, (5) model aggregation and (6) post-process. SOLIDAGO also proposes default implementations for each module, based on the state of the art. Classical solutions did not always meet the needs of our pipeline. Our secondary contribution is to fill missing gaps, by (i) introducing LipschitTrust for secure trust propagation over a directed social network, (ii) presenting new Lipschitz-resilient primitives like QrQtl, (iii) proving their security and consistency guarantees, (iv) adding a presumption of engagement bias in favor of quality entities, (v) proposing a mechanism to favor more consensual entities. Finally, we present an evaluation of the pipeline, both on synthetic and on Tournesol’s data. Better yet, SOLIDAGO has been fully deployed as the engine of the platform Tournesol.app since September 2023, to collaboratively score the recommendability of YouTube videos.

Related work. Social media have raised serious concerns, in terms of disinformation, radicalization, and mental health. Even talks of an existential threat. Conversely, “good” information prioritization could advance important causes, e.g. media literacy, public health and world peace. But who gets to decide what “good” is? WeBuildAI laid an inspiring foundation. This platform essentially allows the stakeholders of a food donation system to collaboratively select the recipients of donations. Similar algorithmic governance systems were proposed by for autonomous car trolley dilemmas, and by for kidney donation. Essentially, these systems rely on a collaborative scoring of different options, which we call entities. The key contribution of our paper is to build a pipeline that generalizes all these systems. Like, , we consider preference elicitation through entity comparisons. While comparisons seem more cognitively demanding, implying lower user retention, they may result in more thoughtful judgments, which have been shown to be less exclusive of outgroup members. Turning comparisons into scores is then a classical problem, with solutions deployed at scale, e.g. to score chess players. SOLIDAGO’s default implementation leverages a recent generalization of the famous Bradley-Terry model. Contrary to, SOLIDAGO does not restrict itself to a predefined list of participants. This exposes us to fake accounts, also known as Sybil attacks. To limit this risk, we leverage a set of pretrusted users and peer-to-peer vouches. Trust is then spread through the vouching network. EigenTrust was proposed to do so, and was applied to file sharing, grid computing, and message routing, and has spurred many variants. However, none provides a guarantee on the maximal impact of a single node. Even in permissioned systems, security issues still arise, as any participant may be malicious, corrupted or hacked. To secure collaborative governance, we draw inspiration from the one person, one unit force fairness principle. More specifically, we adapt MEHESTAN, and provide Lipschitz-resilience guarantees. Additionally, our algorithms account for (genuine) users’ noisy inputs, and output human-friendly scores.
Paper structure. In Section 2, we present the SOLIDAGO inputs and problem formulation. Then, each SOLIDAGO module is explained in a separate section, along with our proposed default implementation. Section 9 evaluates our pipeline, and Section 10 concludes. Our pipeline is represented in Figure 1. Details are provided in the Appendix. The code is in the Supplementary Material.

2 Problem formulation and data

The problem we consider consists of defining a modular, well-behaved and secure pipeline to transform the inputs from \( U \) different users into a human-interpretable score \( \rho_{\text{display}} \in \mathbb{R} \) of every entity \( e \) in a set \( [E] \triangleq \{1, \ldots, E\} \) of entities to assess. Below, we detail SOLIDAGO’s inputs.

Pretrust. To provide Sybil resilience to our system, we leverage pretrust information about users. This may be a list of known users. In the case of Tournesol, pretrust is obtained through the validation of an email, whose domain is consider Sybil-resilient [29]. Typically, @who.int would be considered Sybil-resilient, but @proton.me would not, as anyone can easily create multiple such emails. Formally, we denote \( U_{\text{pre}} \) the set of pretrusted users.

Vouch data. Since validating pretrusted users only is very exclusive, SOLIDAGO also allows users to vouch for one another. The dataset thereby constructed entails entries of the form \((u, v)\), which mean that voucher \( u \) vouched for vouchee \( v \). We denote \( V \) the set of such vouches.

Comparisons. Each user \( u \)’s dataset \( D_u \) is a list of tuples \((e, f, r)\), each saying that contributor \( u \in U \) compared entity \( e \in E \) to entity \( f \in E \), and gave the judgment \( r \in [-R_{\text{max}}, R_{\text{max}}] \). On Tournesol, the maximal comparison is \( R_{\text{max}} \triangleq 10 \) (see Figure 2). A comparison \( r < 0 \) means that user \( u \) prefers \( e \) to \( f \). By considering that comparisons are anti-symmetric (i.e. reporting \( r_{ef} \) is equivalent to reporting \(-r_{fe}\)), we can then represent user \( u \)’s comparisons by an anti-symmetric matrix \( r_{ef} \in \left([-R_{\text{max}}, R_{\text{max}}] \cup \{\bot\}\right) \). The special value \( \bot \) means that the entry is lacking, and is thus undefined.

Privacy Since privacy may be sometimes desirable, especially when retaliation risks are significant, e.g. when upvoting a content that criticizes a government, an employer or a colleague, SOLIDAGO allows each user to assess an entity \( e \) either publicly or privately. However, private comparisons may also be regarded as an attack vector, as such data cannot be audited by third parties. In the sequel, \( E_{\text{private}}^u \) denotes the sets of entities that user \( u \) chose to assess privately. We stress however that SOLIDAGO does not currently provide differential privacy for displayed scores.

Problem formulation. Our pipeline constructs a map SOLIDAGO : \((U_{\text{pre}}^{\text{✓}}, V, D_{1:U}, E_{\text{private}}^{1:U}) \rightarrow \rho_{\text{display}}\). We do so by combining six modular steps, each of which is detailed in the sequel.

3 Trust propagation

Step 1 of the SOLIDAGO pipeline combines pretrusts and vouches to assign trust scores to all users. Our default implementation combines the introduction of a sink vouch and of a novel trust propagation algorithm called LipschiTrust.

3.1 Sink vouch

If vouchers’ vouches were unconstrained, then a malicious user could gain an unbounded influence by vouching for arbitrarily many fake accounts. To secure vouching, it might be tempting to force vouchers’ vouches to be split among their vouchees. However, this disincentivizes vouching, as each new vouch reduces the value of their prior vouches. To incentivize vouching without sacrificing security, we introduce a sink vouch for all
users. Essentially, each user is considered to implicitly vouch for a sink (a non-user), with an intensity equal to $V_{\text{sink}}$ regular vouchees. Tournesol set $V_{\text{sink}} \triangleq 5$. As a result, when a voucher with much fewer than $V_{\text{sink}}$ vouchees vouches for more vouchees, the total vouches given by the voucher grows almost linearly, thereby not penalizing previous vouchees. Vouching is thereby not (too) disincentivized. To integrate sink vouchees formally, we simply split voucher $u$’s vouchees. For each $(u, v) \in V$, this amounts to defining $V_{uv} \triangleq 1 / (\sum_{u} V_{uv} + \{|k| \leq V\})$. Clearly, the sum of weighted vouches given by any voucher to actual users is then at most one. Thus $V$ is row-substochastic.

3.2 LipschiTrust

Trust propagation through the weighted vouch network, with weights $V$, is then performed by LipschiTrust. This novel algorithm is inspired from EigENTRUST [32], whose principle is akin to the celebrated PAGE-RANK [30]. These algorithms have an appealing probabilistic intuition, which we present before providing the formal definition.
Probabilistic intuition. Consider a random walker on the vouch network, initially dropped at a uniformly randomly chosen pretrusted user. Then at each iteration, the random walker first tosses a biased coin. With probability $1 - \beta$ (which will be useful for Sybil resilience), the random walker resets its walk. Otherwise, the random walker randomly selects an outgoing vouch, with a probability equal to the vouch’s weight. If the arc goes to the sink, then the random walk is also reset. The frequency of the random walker visits to any given node eventually converges to an ergodic limit, which defines its trust score. LipschiTrust essentially robustifies the random walk, by frequently preventing the walker from visiting too frequently visited users, thereby bounding their maximal influence.

Defining LipschiTrust. Recall that we are given a set $U^\pre$ of pretrusted users. We define the pretrust vector by $\text{trust}_u^\pre \doteq 1$ if $u \in U^\pre$, and 0 otherwise. We then define $\text{TR}^0 \doteq \text{TR}^\pre \in \mathbb{R}^U$. For all iterations $t$, using vector notations, we consider $\text{TR}^{t+1/2} \doteq \text{TR}^\pre + \beta V^T \text{TR}^t$, where $\beta \in (0, 1)$ is a vouch decay hyperparameter. Tournesol set $\beta \doteq 0.8$. Trusts are then clipped to avoid power concentration.

$$\text{TR}^{t+1} \doteq \min \left\{ \frac{\text{TR}^{t+1/2}}{2}, 1 \right\}.$$  

Proposition 3.1. There is a unique vector $\text{TR}$ such that $\|\text{TR} - \text{TR}\|_1 \leq U^\beta$ for all $t$. Moreover, this vector verifies $\text{TR} = \min \left\{ \text{TR}^\pre + \beta V^T \text{TR}, 1 \right\}$.

Proof sketch. Since $V$ is row-substochastic and $x \mapsto \min \{x, 1\}$ is 1-Lipschitz, $\text{TR}$ is $\beta$-contractive. Hence the convergence. Taking the recursion equation of $\text{TR}$ to the limit concludes the proof.

As a corollary, an error of at most $\varepsilon_{\text{LipschiTrust}} > 0$ can be guaranteed by merely $T(\varepsilon) \doteq \left\lceil \frac{\ln(U/\varepsilon_{\text{LipschiTrust}})}{\ln(1/\beta)} \right\rceil$ iterations. Tournesol uses $\varepsilon_{\text{LipschiTrust}} \doteq 10^{-8}$.

The security guarantee of LipschiTrust. Our novel algorithm LipschiTrust has the following Sybil resilience guarantee that bounds the maximal influence of a user by its received trust.

Theorem 3.2. Denoting $\text{TR}^{-u}$ the trusts computed after removing user $u$’s vouches

$$\|\text{TR} - \text{TR}^{-u}\|_1 \leq \frac{\beta}{1 - \beta} \text{TR}^{-u}.$$  

Proof sketch. The proof considers a similar sequence as $\text{TR}$, but whose initial point is $\text{TR}^{-u}$. We conclude by bounding the difference between the first iteration and the initial point, and by $\beta$-contractiveness. The full proof is given in Appendix A.2.

Other properties of LipschiTrust. We conclude this section with two additional desirable guarantees, whose proofs are provided in Appendix A.3.

Proposition 3.3. Vouching cannot decrease trust, i.e., for all users $u$, $\text{TR} \geq \text{TR}^{-u}$.

Proposition 3.4. A user’s trust score is positive, if and only, if there is a vouch path from a pretrusted user to them.

4 Voting rights

Step 2 of Solidago leverages trust scores and user activity to determine voting rights. As demanded by Tournesol users, our solution values untrusted users, without sacrificing security.
Overview. The computation of users’ voting rights on an entity $e$ proceeds in four steps:

1. Compute the cumulative trusted scores of the users that compared $e$, weighed by $w_{\text{private}}^{\text{penalty}}$.

2. Derive the maximal amount of tolerated overtrust $\text{OverTrust}_e$.

3. Determine a minimal voting right $w_{\text{min,e}}^{\text{public}}$ that yields the maximal tolerated overtrust.

4. Set voting rights as the maximum of the trust score and $w_{\text{min,e}}^{\text{public}}$, weighed by $w_{\text{private}}^{\text{penalty}}$.

On Tournesol, the privacy penalty is set as $w_{\text{private}}^{\text{penalty}} = 0$.

Cumulative trusted score Recall that we are given the sets $E_u^{\text{private}}$ of entities that user $u$ rates privately. From the datasets $D_u$, we also know whether the user has rated an entity. Define $w_{ue}^{\text{penalty}} = 1$ if user $u$ rated entity $e$ publicly, $w_{ue}^{\text{penalty}} = w_{\text{private}}^{\text{penalty}}$ if user $u$ rated entity $e$ privately, and $w_{ue}^{\text{penalty}} = 0$ if they did not rate $e$. The cumulative trusted score is given by

$$\text{trust}_e \triangleq \sum_{u \in [U]} w_{ue}^{\text{penalty}} \text{TRUST}_u$$

Maximal tolerated overtrust The maximal tolerated overtrust must be asymptotically only a fraction $f_\times$ of $\text{trust}_e$, and will be at least $w_{\times,\text{default}}^{\text{total}}$. This principle is formalized by defining

$$\text{OverTrust}_e \triangleq w_{\times,\text{default}}^{\text{total}} + f_\times \cdot \text{trust}_e.$$

Tournesol set $w_{\times,\text{default}}^{\text{total}} \triangleq 2$ and $f_\times \triangleq 0.1$.

Minimal voting right We now guarantee that the minimal voting right given to insufficiently trusted users yields at most an overtrust of $\text{OverTrust}_e$. To do so, define the overtrust when setting a minimal public voting right $w_{\text{min,e}}^{\text{public}}$ by

$$\text{overTrust}_e(w_{\text{min}}) \triangleq \sum_{u \in [U]} w_{ue}^{\text{penalty}} \max \{w_{\text{min}} - \text{TRUST}_u, 0\}$$

Observe that we have $\text{overTrust}_e(0) = 0$ and that $\text{overTrust}_e$ is a strictly increasing function of $w_{\text{min,e}}^{\text{public}}$ as soon as $\text{overTrust}_e(w_{\text{min,e}}^{\text{public}}) > 0$. Thus, given that the maximal overtrust is positive, either $\text{overTrust}_e(1) < \text{OverTrust}_e$ (in which case we set $w_{\text{min,e}}^{\text{public}} \triangleq 1$), or there is a unique $w_{\text{min,e}}^{\text{public}} \in (0, 1]$ for which we have $\text{overTrust}_e(w_{\text{min,e}}^{\text{public}}) = \text{OverTrust}_e$. We propose to compute it with dichotomic search.

Voting rights A user $u$’s voting right on $e$ is then defined as the maximum between their trust and the public minimal voting right for this entity, potentially multiplied by the privacy penalty, i.e.

$$w_{ue} \triangleq w_{ue}^{\text{penalty}} \max \{\text{TRUST}_u, w_{\text{min,e}}^{\text{public}}\}.$$

5 User model inference

Step 3 of SOLIDAGO, which is independent from the two first steps, consists of learning a preference model for each user $u \in [U]$, based on their entity-evaluation dataset $D_u$. Our default implementation transforms quantitative comparisons into scores for compared entities.
Scoring through Generalized Bradley-Terry. To turn a user’s comparisons into scores, we use the recent generalization [20] of [8], which is instantiated with a uniform root law. This yields a negative log-likelihood equal to
\[
L(\theta_u|D_u) \triangleq \sum_{e,f,r \in D_u} \ln \frac{\sinh (\theta_{uef})}{\theta_{uef} + r\theta_{uef}/R_{\text{max}}}.
\] (6)

Including a Gaussian prior yields the following loss, which we minimize:
\[
\theta_{\text{raw}}(r) \triangleq \arg\min_{\theta_u \in \mathbb{R}^{E_u}} \left\{ \frac{\alpha_{\text{user}}}{2} \|\theta_u\|^2 + L(\theta_u|D_u) \right\}.
\] (7)

Tournesol uses $\alpha_{\text{user}} \approx 0.02$. Since [20] guarantees that the loss is strongly convex, the minimum $\theta_{\text{raw}}(r)$ exists, is unique and can be efficiently computed, which we call raw scores. This model also has the following desirable guarantee.

**Proposition 5.1** (Theorem 2 in [20]). If $r_{ef} \leq r'_{ef}$ for all $f$ (i.e., the comparisons are more favorable to $e$ in $r'$ than in $r$) and $\tilde{r}_{gh} = \tilde{r}'_{gh}$ for all $g, h \neq e$, then $\theta_{\text{raw}}(r) \leq \theta_{\text{raw}}(r')$.

Estimating uncertainty. Learning the uncertainty of the scores is important for score aggregation; intuitively, a user should have less influence on the global scores if we have a high uncertainty on their scores. We perform an asymmetric uncertainty estimation, based on increase of the negative log-likelihood term of the loss by one unit (which corresponds to a likelihood $1/e$ times smaller), i.e. we solve for $\Delta \theta_{\text{raw}}^{\text{left}} \geq 0$ and $\Delta \theta_{\text{raw}}^{\text{right}} \geq 0$ such that
\[
L(\theta_u - \Delta \theta_{\text{raw}}^{\text{left}} e_e|D_u) = L(\theta_u + \Delta \theta_{\text{raw}}^{\text{right}} e_e|D_u) = L(\theta_u|D_u) + 1,
\] (8)
where $e_e$ is the $e$-th vector of the canonical basis. Note that such uncertainties may not exist, in which case they are set to $+\infty$.

6 Model scaling

Step 4 of SOLIDAGO performs user model scaling. The default implementation has three steps, namely finding a common scale, accounting for scoring selection bias and standardizing scores.

6.1 Collaborative scaling

As explained by [3], different users express themselves on different implicit scales, which implies scale mismatch. This issue may especially be raised if a user assesses top entities, while another assesses bad entities only. To put all users’ scores on a common scale in a secure manner, we adapted MEHESTAN [5].

Reviewing Mehestan. The original MEHESTAN has three steps:

(i) It individually normalizes all users’ scores.

(ii) Each user’s scores are scaled, by comparing them to other users’ scores for entities that both compared, using a Lipschitz-resilient primitives.

(iii) Third, each users’ scores are appropriately translated, by comparing them to other users’ scores, again using a Lipschitz-resilient primitives.
How we adapted Mehestan. We diverge from [5] in four notable ways:

1. We skip step (i), as comparisons provides a reasonable multiplicative normalization.
2. We rely on a subset of scaling-calibration users (detailed below). Other users’ scales are then made to match scaling-calibration users’ scale.
3. We robustify the estimates of the relative scaling between two users, by using the primitive QRMed instead of an average, and generalizing other primitives to account for asymmetric uncertainties (see Appendix C).
4. We systematically track uncertainties on estimates, which are used when aggregating them. Intuitively, high uncertainty on a scaled user score for an entity implies that it will only have an effect if the aggregated value is farther away from the estimate than the computed uncertainty. We thereby contribute to the growing toolbox of Lipschitz-resilient primitives.

Our adaptation of MEHESTAN is detailed in Appendix D.

Scaling-calibration users. Adaptation (2) is motivated by the “liveness” guarantees of MEHESTAN, which requires users to be sufficiently active. This is far from being the case in general, and in particular in the case of Tournesol. Ignoring inactive users may be reminiscent of [25], who propose to ignore insufficiently active participants to federated learning. Additionally, (2) allows to accelerate MEHESTAN, whose vanilla form is too computationally demanding for frequent score updates, especially given Tournesol’s limited server with a 4-core CPU and no GPU. To make collaborative preference scaling more accurate and faster, we propose to select the $N_{\text{scaling}}$ users who have compared the most entities, among those whose trust score is at least $\text{TRUST}_{\text{min}}^{\text{scaling}}$. The values $N_{\text{scaling}} = 100$ and $\text{TRUST}_{\text{min}}^{\text{scaling}} = 0.1$ are currently used on the platform.

6.2 Presumption of quality of rated entities

In the context of Tournesol, it has been observed that most of the entities that are rated on the platform are of high quality. This raises an issue. Namely, since zero is the default score in absence of comparisons, an unrated entity would be regarded as equally good as an average rated entity. To account for the selection bias among rated entities, we encode a presumption of quality for rated entity, by shifting the scale of the scores of rated entities, so that the $q_{\text{zero shift}}$-quantile of rated entities has a zero score. Tournesol set $q_{\text{zero shift}} \triangleq 15\%$, i.e. they consider that 15% of rated entities are less recommendable than a random YouTube video. More precisely, to do this robustly, we define a new Lipschitz-resilient primitive, which we call the $q$-quadratically regularized quantile (see Section 7). We then subtract to all user scores the $q_{\text{zero shift}} - \text{QRQTL}$ of the users’ scores, given their uncertainties.

6.3 Score standardization to control Lipschitz resilience

Finally, we standardize the scaled scores using a Lipschitz-resilient estimator of their standard deviation. Note that Tournesol’s collaboratively scaled scores are very heavy-tailed, with many values that are over 30 standard deviations away from the mean. This motivated us to use an estimator that intuitively fits $q_{\text{std}} \triangleq 90\%$ of all the scores. Essentially, this amounts to acknowledging that at most 10% of the entities will have scores that are out of scale. More precisely, we consider the $q_{\text{dev}} - \text{QRQTL}$ of the absolute deviations of the scores to the QRMed of the scores. The scaled scores are standardized by dividing them by the resulting estimate $q_{\text{scaled dev}}$ of the standard deviation. The resulting scores are the scaled user models $\theta_{\text{scaled}}$, with uncertainties $\Delta \theta_{\text{scaled}}$.

Each user score is given a voting right inversely proportional to the number of entities the user compared.
7 Model aggregation

Step 5 of SOLIDAGO consists of aggregated users’ scaled models. For each entity $e$, we aggregate the users’ scaled scores $\theta_{ue}^{\text{scaled}}$ using a novel primitive we call the quadratically regularized quantile. Namely, for $\alpha \in (0, 1)$, voting rights $w_u$, scalars $x_u$ and uncertainties $\Delta_u$ for $u \in \mathcal{U}$, we define

$$Q_{rQtl}^{\alpha,L}(w, x, \Delta) \triangleq \arg\min_{m \in \mathbb{R}} \left\{ \frac{m^2}{2L} + \sum_{u:x_u \neq \perp} w_u \text{Huber}_\alpha(m|x_u, \Delta_u) \right\},$$  
(9)

where $\text{Huber}_\alpha$ is an adaptation of the classical Huber loss to asymmetric uncertainties:

$$\text{Huber}_\alpha(m|\mathcal{X}, \Delta) \triangleq \begin{cases} 
\min \left\{ 1, \frac{1-\alpha}{\alpha} \right\} \left( \frac{\Delta_{\text{left}}^2 + (x - m)^2 - \Delta_{\text{left}}}{\Delta_{\text{left}}} \right) & \text{for } m \leq x, \\
\min \left\{ 1, \frac{\alpha}{1-\alpha} \right\} \left( \frac{\Delta_{\text{right}}^2 + (m - x)^2 - \Delta_{\text{right}}}{\Delta_{\text{right}}} \right) & \text{for } m \geq x.
\end{cases}$$  
(10)

Note that, for $\alpha = 0.5$ and $\Delta = 0$, we retrieve the quadratically regularized median as defined by [5]. Crucially, for any $\alpha \in (0, 1)$, $Q_{rQtl}^{\alpha,L}$ remains $L$-Lipschitz resilient,

**Theorem 7.1** (Lipschitz resilience of $Q_{rQtl}$). Let $\alpha \in (0, 1)$. For any $x_{1:U} \in \mathbb{R}^U$, $\Delta_{1:U} \in \mathbb{R}_{\geq 0}^U$, the map $w_{1:U} \mapsto Q_{rQtl}^{\alpha,L}(w_{1:U}, x_{1:U}, \Delta_{1:U})$ is $L$-Lipschitz continuous as a function $(\mathbb{R}^U, \|\cdot\|_1) \rightarrow (\mathbb{R}, |\cdot|)$.

**Proof sketch.** The key to the theorem is the derivative of the term that depends on a user $u$ has an absolute value which is at most $w_u$. See Appendix C.4 for the details.

Appendix C.4 proves several other desirable properties of $Q_{rQtl}$, such as the fact that it asymptotically correctly estimates the quantile, or the fact that the addition of a user with huge uncertainties does not affect the estimate. We then define each global score $\rho_e$ for entity $e$ as

$$\rho_e \triangleq Q_{rQtl}^{\alpha,L}(w_{1:U,e}, \frac{\theta_{ue}^{\text{scaled}}}{\sigma_{ue}^{\text{scaled}}}, \frac{\Delta_{ue}^{\text{scaled}}}{\sigma_{ue}^{\text{scaled}}}).$$  
(11)

Tournesol set $\alpha \triangleq 0.2$ and $L \triangleq 0.1$.

Favoring consensus. By setting $\alpha < 0.5$, Tournesol favors entities that are consensually good, over entities that are excellent according to half of the community, but are very bad according to another half. This measure is aligned, in spirit, with some measures of Twitter’s Community Notes. Appendix G discusses the impact of $\alpha$.

8 Model post-process

Finally, Step 6 of SOLIDAGO responds to the desire for a display of scores on a bounded scale, e.g. from $(-100, 100)$, which may be regarded as more “human-readable”. We do so by squashing the scores using the increasing homeomorphism $\varphi : x \mapsto 100x/\sqrt{1+x^2}$, which transforms $\mathbb{R}$ into $(-100, 100)$. This yields the displayed user scores $\theta_{ue}^{\text{display}}$ and the displayed global scores $\rho_e^{\text{display}}$. These are formally given by $\theta_{ue}^{\text{display}} \triangleq \varphi(\theta_{ue}^{\text{scaled}})$ and $\rho_e^{\text{display}} \triangleq \varphi(\rho_e)$.

9 Evaluation

We perform two evaluations of our pipeline. First we discuss experiments made with a synthetic dataset, before discussing the large-scale deployment on Tournesol.
9.1 Evaluation on synthetic data

User generation. To evaluate SOLIDAGO on synthetic data, we propose a user generation model, where each user may be trustworthy or not, pretrusted or not. Here, we make the simplifying assumption that pretrusted users are trustworthy. Moreover, a user may be more or less engaged in vouching for others, and in comparing entities. They may also be more or less willing to provide numerous comparisons per entity. We draw a singular-value decomposition (SVD) model of users and entities. Namely, we draw a preference $d$-dimensional SVD vector for each trustworthy user from a normal distribution $\mathcal{N}(\rho^{\text{true}}, I_d)$, with $\rho^{\text{true}} \triangleq (3, 0, \ldots, 0)$ (and $d = 2$ in our experiments). Note that $\rho^{\text{true}} \neq 0$ implies that the trustworthy users' preferences are somewhat aligned. Untrustworthy users are given the vector representation $-\rho^{\text{true}}$, i.e. they score items oppositely from the trustworthy users’ approximately consensual preferences. Finally, some users may prefer to evaluate good videos rather than any video. We model this with a random engagement bias for good videos.

Vouch generation. We adapt the [19] model for vouching among trustworthy users, where the probability of vouching is however dependent on the voucher’s activeness in vouching. Similarly, untrustworthy users vouch for one another, according to a similar model.

Entity generation. Each entity is assumed to have a feature SVD vector drawn from $\mathcal{N}(0, I_d)$.

Engagement generation. For each user, we define the engagement bias as the user’s implicit score for an entity scaled by their engagement bias, plus a random standard normal noise. The user then engages with the entities with the best engagement score. Among these entities, the comparisons are made based on an Erdős-Rényi model.

Comparison generation. Each engagement leads to a comparison, made by a 21-nary generalized Bradley Terry model [20] (as Tournesol does), given the true score defined by the scalar product between the user preference’s SVD vector and the entity’s SVD vector.

Reproducibility. The plots of Figure 3 depict the standard deviations after running 100 random seeds. The code is provided in the Supplementary Material.

Resilience evaluation. Figure 3 (left) shows how the correlation between the ground truth scores and the pipeline output increases with the fraction of trustworthy users. For a small Lipschitz constant (which corresponds to a more resilience), this correlation is positive, even given a majority of trustworthy users, thanks to pretrust, vouching and the pipeline resilience. When most users are trustworthy, larger Lipschitz constants improve accuracy.

Evaluation of the correction of engagement bias. The right plot of Figure 3 shows the harm of engagement bias, especially Mehestan’s Lipschitz resilience is too small (which amounts to doing barely any scaling). Overall, the plots highlight the tradeoff between resilience and accuracy.

9.2 Large-scale deployment

SOLIDAGO has been deployed on Tournesol since September 2023, where it allows over 10,000 active users to collaboratively score over 35,000 YouTube videos, by aggregating their over 190,000 comparisons. Figure 4 displays the distribution of the displayed user and global scores on the platform. Appendix F details our hyperparameter selection, by plotting the distributions for other hyperparameters. Note that we heavily exploited numba to accelerate the computations. While no feedback elicitation has been performed, the deployment of SOLIDAGO was well received by several active users, who reckoned a significant improvement of the algorithmic interpretation of their inputs by the platform.
Figure 3: Correlation between the ground truth and the learned global scores, as a function of the fraction of trustworthy users (left), and as a function of engagement bias (right).

10 Conclusion

This paper presented SOLIDAGO, an end-to-end modular pipeline for secure collaborative scoring. The pipeline is composed of 6 modules, for which we introduced state-of-the-art default implementations. We made these modules somewhat independent, to facilitate external contributions to the pipeline. In fact, improving each module is an exciting research problem of its own, involving, e.g., accounting for direct assessments, learning volitions [33], enabling liquid democracy [27], and making user models generalize (see Appendix H). SOLIDAGO is also accompanied with data generation tools, to facilitate pipeline evaluation. Perhaps most remarkably, SOLIDAGO has been deployed at scale on tournesol.app since September 2023.

References

[1] Zoë Abrams, Robert McGrew, and Serge A. Plotkin. A non-manipulable trust system based on eigen-trust. SIGecom Exch., 5(4):21–30, 2005.

[2] Frederick Adams and Alfred R Mele. The intention/volition debate1. Canadian Journal of Philosophy, 22(3):323–337, 1992.

[3] Juan Afanador, Nir Oren, Murilo S. Baptista, and Maria Araujo. From eigentrust to a trust-measuring algorithm in the max-plus algebra. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of Frontiers in Artificial Intelligence and Applications, pages 3–10. IOS Press, 2020.

[4] Amanda Y Agan, Diag Davenport, Jens Ludwig, and Sendhil Mullainathan. Automating automaticity: How the context of human choice affects the extent of algorithmic bias. Technical report, National Bureau of Economic Research, 2023.
Figure 4: Distribution of displayed user scores $\rho^\text{display}$ (top graph) and global $\rho^\text{display}$ (bottom graph). Hyperparameters were selected so that users’ scores have roughly a similar distribution, once they were compared at least 3 times, and so that the global scores of highly compared videos are well spread. Hyperparameter selection is further discussed in Appendix F.

[5] Youssef Allouah, Rachid Guerraoui, Lê-Nguyễn Hoang, and Oscar Villemaud. Robust sparse voting. In International Conference on Artificial Intelligence and Statistics. PMLR, 2024.

[6] Reid Andersen, Christian Borgs, Jennifer T. Chayes, Uriel Feige, Abraham D. Flaxman, Adam Kalai, Vahab S. Mirrokni, and Moshe Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang, editors, Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008, pages 199–208. ACM, 2008.

[7] Mehmet F Bastug, Aziz Douai, and Davut Akca. Exploring the “demand side” of online radicalization: Evidence from the canadian context. Studies in Conflict & Terrorism, 43(7):616–637, 2020.

[8] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika, 39(3/4):324–345, 1952.

[9] Richard P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. The computer journal, 14(4):422–425, 1971.

[10] Lewis Carroll. The principles of parliamentary representation. Harrison and Sons, 1884.

[11] Alice Cheng and Eric Friedman. Sybilproof reputation mechanisms. In Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, pages 128–132, 2005.

[12] Yuwei Chuai, Haoye Tian, Nicolas Prölls, and Gabriele Lenzini. The roll-out of community notes did not reduce engagement with misinformation on twitter. arXiv preprint arXiv:2307.07960, 2023.
[13] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In Proceedings of the Network and Distributed System Security Symposium, NDSS 2009, San Diego, California, USA, 8th February - 11th February 2009. The Internet Society, 2009.

[14] Theodorus Jozef Dekker. Finding a zero by means of successive linear interpolation. Constructive aspects of the fundamental theorem of algebra, pages 37–51, 1969.

[15] John R. Doucerc. The sybil attack. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers, volume 2429 of Lecture Notes in Computer Science, pages 251–260. Springer, 2002.

[16] El-Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, and Lê-Nguyễn Hoang. On the strategy-proofness of the geometric median. CoRR, abs/2106.02394, 2021.

[17] Vittoria Elliott and David Gilbert. Elon musk’s main tool for fighting disinformation on x is making the problem worse, insiders claim. Wired, 2023.

[18] Arpad E Elo and Sam Sloan. The rating of chessplayers: Past and present. (No Title), 1978.

[19] P Erdős and A Renyi. On random graphs i. Publicationes Mathematicae, 6(290-297):18, 1959.

[20] Julien Fageot, Sadegh Farhadkhani, Lê Nguyên Hoang, and Oscar Villemaud. Generalized bradley-terry models for score estimation from paired comparisons. arXiv preprint arXiv:2308.08644, 2023.

[21] Xinxin Fan, Ling Liu, Mingchu Li, and Zhiyuan Su. Eigentrustp++: Attack resilient trust management. In Calton Pu, James Joshi, and Surya Nepal, editors, 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom 2012, Pittsburgh, PA, USA, October 14-17, 2012, pages 416–425. ICST / IEEE, 2012.

[22] Sadegh Farhadkhani, Rachid Guerraoui, Lê Nguyên Hoang, and Oscar Villemaud. An equivalence between data poisoning and byzantine gradient attacks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, volume 162 of Proceedings of Machine Learning Research, pages 6284–6323. PMLR, 2022.

[23] Rachel Freedman, Jana Schaich Borg, Walter Sinnott-Armstrong, John P. Dickerson, and Vincent Conitzer. Adapting a kidney exchange algorithm to align with human values. Artif. Intell., 283:103261, 2020.

[24] Junling Gao, Pinpin Zheng, Yingnan Jia, Hao Chen, Yimeng Mao, Suhong Chen, Yi Wang, Hua Fu, and Junming Dai. Mental health problems and social media exposure during covid-19 outbreak. Plos one, 15(4):e0231924, 2020.

[25] Marie Garin, Theodoros Evgeniou, and Nicolas Vayatis. Weighting schemes for one-shot federated learning. Available at SSRN 4035732, 2022.

[26] Keach Hagey and Jeff Horwitz. Facebook tried to make its platform a healthier place. it got angrier instead. The Wall Street Journal, 2023.

[27] Daniel Halpern, Joseph Y. Halpern, Ali Jadbabaie, Elchanan Mossel, Ariel D. Procaccia, and Manon Revel. In defense of liquid democracy. In Kevin Leyton-Brown, Jason D. Hartline, and Larry Samuelson, editors, Proceedings of the 24th ACM Conference on Economics and Computation, EC 2023, London, United Kingdom, July 9-12, 2023, page 852. ACM, 2023.

[28] Lê Nguyên Hoang. Science communication desperately needs more aligned recommendation algorithms. Frontiers in Communication, 5:598454, 2020.
[29] Lê-Nguyễn Hoang, Louis Faucon, Aidan Jungo, Sergei Volodin, Dalia Papuc, Orfeas Liossatos, Ben Crulis, Mariame Tighanimine, Isabela Constantin, Anastasia Kucherenko, Alexandre Maurer, Felix Grimberg, Vlad Nitu, Chris Vossen, Sébastien Rouault, and El-Mahdi El-Mhamdi. Tournesol: A quest for a large, secure and trustworthy database of reliable human judgments. CoRR, abs/2107.07334, 2021.

[30] Lê-Nguyễn Hoang, Louis Faucon, and El-Mahdi El-Mhamdi. Recommendation algorithms, a neglected opportunity for public health. Revue Médiciné et Philosophie, 4(2), 2021.

[31] Kalonji Kalala, Tao Feng, and Iluju Kiringa. Eigentrust for hierarchically structured chord. In Giovanni Livraga and Chris J. Mitchell, editors, Security and Trust Management - 13th International Workshop, STM 2017, Oslo, Norway, September 14-15, 2017, Proceedings, volume 10547 of Lecture Notes in Computer Science, pages 203–212. Springer, 2017.

[32] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In Gusztáv Hencsey, Bebo White, Yih-Farn Robin Chen, László Kovács, and Steve Lawrence, editors, Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003, pages 640–651. ACM, 2003.

[33] Mohamed Lechiakh and Alexandre Maurer. Volition learning: What would you prefer to prefer? In Helmut Degen and Stavroula Ntoa, editors, Artificial Intelligence in HCI - 4th International Conference, AI-HCI 2023, Held as Part of the 25th HCI International Conference, HCII 2023, Copenhagen, Denmark, July 23-28, 2023, Proceedings, Part I, volume 14050 of Lecture Notes in Computer Science, pages 555–574. Springer, 2023.

[34] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, and Ari D. Procaccia. Webuildai: Participatory framework for algorithmic governance. Proc. ACM Hum. Comput. Interact., 3(CSCW):181:1–181:35, 2019.

[35] Kenli Li, Yan He, Xiaoling Liu, and Ying Wang. Security-driven scheduling algorithms based on eigentrust in grid. In Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2005), 5-8 December 2005, Dalian, China, pages 1068–1072. IEEE Computer Society, 2005.

[36] Herbert Lin. The existential threat from cyber-enabled information warfare. Bulletin of the Atomic Scientists, 75(4):187–196, 2019.

[37] Kun Lu, Junlong Wang, and Mingchu Li. An eigentrust dynamic evolutionary model in P2P file-sharing systems. Peer-to-Peer Netw. Appl., 9(3):599–612, 2016.

[38] Lucas Maystre. Efficient learning from comparisons. Technical report, EPFL, 2018.

[39] Ritesh Noothigattu, Snehalkumar (Neil) S. Gaikwad, Edmond Awad, Sohan Dsouza, Iyad Rahwan, Pradeep Ravikumar, and Ari D. Procaccia. A voting-based system for ethical decision making. In Sheila A. McIlraith and Killian Q. Weinberger, editors, Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 1587–1594. AAAI Press, 2018.

[40] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[41] James A Piazza. Fake news: the effects of social media disinformation on domestic terrorism. Dynamics of Asymmetric Conflict, 15(1):55–77, 2022.

[42] Ouri Poupko, Gal Shahaf, Ehud Shapiro, and Nimrod Talmon. Building a sybil-resilient digital community utilizing trust-graph connectivity. IEEE/ACM Trans. Netw., 29(5):2215–2227, 2021.
[43] José Ramos. Liquid democracy and the futures of governance. *The future internet: Alternative visions*, pages 173–191, 2015.

[44] Mohamed Amine Rguibi and Najem Moussa. Isolating P2P botnet using eigentrust algorithm in P2P file sharing networks. In Mohamed Lazaar, Claude Duvallet, Mohammed Al Achhab, Oussama Mahboub, and Hassan Silkan, editors, *BDIoT’19: The 4th International Conference On Big Data and Internet of Things, Rabat, Morocco, October 23–24, 2019*. ACM, 2019.

[45] Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgílio A. F. Almeida, and Wagner Meira Jr. Auditing radicalization pathways on youtube. In Mireille Hildebrandt, Carlos Castillo, L. Elisa Celis, Salvatore Ruggieri, Linnet Taylor, and Gabriela Zanfir-Fortuna, editors, *FAT* ’20: *Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27–30, 2020*, pages 131–141. ACM, 2020.

[46] Koustitl Saha, Asra Yousuf, Ryan L Boyd, James W Pennebaker, and Mummun De Choudhury. Social media discussions predict mental health consultations on college campuses. *Scientific reports*, 12(1):1–11, 2022.

[47] Surendran Subbaraj and Savarimuthu Prakash. Eigentrust-based non-cooperative game model assisting ACO look-ahead secure routing against selfishness. *EURASIP J. Wirel. Commun. Netw.*, 2014:78, 2014.

[48] Louis L Thurstone. The method of paired comparisons for social values. *The Journal of Abnormal and Social Psychology*, 21(4):384, 1927.

[49] Dinh Nguyen Tran, Bonan Min, Jinyang Li, and Lakshminarayanan Subramanian. Sybil-resilient online content voting. In Jennifer Rexford and Emin Gün Sirer, editors, *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2009, April 22-24, 2009, Boston, MA, USA*, pages 15–28. USENIX Association, 2009.

[50] Twitter. Note ranking algorithm. Twitter Blog, 2023.

[51] Georgia Wells, Jeff Horwitz, and Deepa Seetharaman. Facebook knows instagram is toxic for teen girls, company documents show. *The Wall Street Journal*, 2021.

[52] Samuel Woolley. *The Reality Game: A gripping investigation into deepfake videos, the next wave of fake news and what it means for democracy*. Hachette UK, 2020.

[53] Brandy Zadrozny. ‘carol’s journey’: What facebook knew about how it radicalized users. *NBC News*, 2021.

[54] Jing Zhu. Understanding volition. *Philosophical psychology*, 17(2):247–273, 2004.
Appendix

A Voting rights (algorithms and proofs)

In this section, we provide a pseudocode of LipschiTrust, and we prove Proposition 3.1 and Theorem 3.2.

A.1 Proof of Proposition 3.1

First, let us prove the following useful lemmas.

Lemma A.1. \( x \mapsto \min \{ x, 1 \} \) is 1-Lipschitz.

Proof. Let \( x, y \in \mathbb{R} \). Without loss of generality, assume \( x \geq y \). Then we have three cases to analyze:

- If \( x \geq y \geq 1 \), then \( \min \{ x, 1 \} - \min \{ y, 1 \} = |x - y| \leq |x - y| \).
- If \( x \geq 1 \geq y \), then \( \min \{ x, 1 \} - \min \{ y, 1 \} = |1 - y| \leq |x - y| \).
- If \( 1 \geq x \geq y \), then \( \min \{ x, 1 \} - \min \{ y, 1 \} = |1 - 1| = 0 \leq |x - y| \).

In all three cases, 1-Lipschitzness is guaranteed.

Lemma A.2. Suppose \( M \in \mathbb{R}_{+}^{U \times U} \) is column-substochastic, i.e. \( \sum_{u} M_{uv} \leq 1 \) for all column \( v \in U \) and \( M_{uv} \geq 0 \) for all \( u, v \in U \). Then for any \( x \in \mathbb{R}^{U} \), we have \( \|Mx\|_{1} \leq \|x\|_{1} \).

Proof. \( \|Mx\|_{1} = \sum_{u} |\sum_{v} M_{uv}x_{v}| \leq \sum_{u} \sum_{v} M_{uv} |x_{v}| = \sum_{v} (\sum_{u} M_{uv}) |x_{v}| \leq \sum_{v} |x_{v}| = \|x\|_{1} \).

Since \( V \) is row-substochastic, we thus have \( \|V^T x\|_{1} \leq \|x\|_{1} \). This allows us to prove Proposition 3.1.

Lemma A.3. The sequence \( \text{tr}^{t} \) is \( \beta \)-contractive.

Proof. Denote \( \Delta \text{tr}^{t} \triangleq \text{tr}^{t+1} - \text{tr}^{t} \). Using Lemma A.1 for all \( u \in U \), we have

\[
|\Delta \text{tr}^{t}_{u}| = |\text{tr}^{t+1}_{u} - \text{tr}^{t}_{u}| = \min \left\{ \text{tr}^{t+1/2}_{u}, 1 \right\} - \min \left\{ \text{tr}^{t-1/2}_{u}, 1 \right\} \leq |\text{tr}^{t+1/2}_{u} - \text{tr}^{t-1/2}_{u}| \quad (12)
\]

\[
= \left| \left( \text{trust}^{\text{pre}}_{u} + \beta \sum_{v} V_{vu} \text{tr}^{t}_{v} \right) - \left( \text{trust}^{\text{pre}}_{u} + \beta \sum_{v} V_{vu} \text{tr}^{t-1}_{v} \right) \right| \quad (13)
\]

\[
= \beta \sum_{v \in U} V_{vu} \Delta \text{tr}^{t-1}_{v} \leq \beta \sum_{v \in U} V_{vu} \Delta \text{tr}^{t-1} \leq \beta \sum_{v \in U} V_{vu} \Delta \text{tr}^{t-1} = \beta (V^T)_{u} |\Delta \text{tr}^{t-1}|, \quad (14)
\]

where \( |x| \) of a vector \( x \) is the vector whose coordinates are the absolute values of the coordinates of \( x \). It follows that

\[
\|\Delta \text{tr}^{t}\|_{1} = \|\text{tr}^{t+1} - \text{tr}^{t}\|_{1} = \sum_{u \in U} |\text{tr}^{t+1}_{u} - \text{tr}^{t}_{u}| \leq \beta \|V^T |\Delta \text{tr}^{t-1}|\|_{1}. \quad (15)
\]

By Lemma A.2, we thus have \( \|\text{tr}^{t}\|_{1} \leq \beta \|\Delta \text{tr}^{t-1}\|_{1} \). This amounts to saying that \( \text{tr}^{t} \) is \( \beta \)-contractive.

Proposition 3.1. There is trust such that \( \|\text{tr}^{t} - \text{trust}\|_{1} \leq |U| \beta^t \). Moreover, trust = \( \min \{ \text{trust}^{\text{pre}} + \beta V^T \text{trust}, 1 \} \).

Proof. Lemma A.3 implies its convergence to a limit trust. Taking the recursive defining equation of \( \text{tr}^{t} \) to the limit yields

\[
\text{trust} = \min \{ \text{trust}^{\text{pre}} + \beta V^T \text{trust}, 1 \}.
\]
Using again Lemma A.1 for $t \geq 0$, we then have
\[
\left\| \text{TR}^{t+1} - \text{TRUST} \right\|_1 \leq \beta \left\| (\text{TRUST}^{pre} + \beta V^T \text{TR}) - (\text{TRUST}^{pre} + \beta V^T \text{TRUST}) \right\|_1 \leq \beta \left\| V^T \right\|_1 \left\| \text{TR}^t - \text{TRUST} \right\|_1.
\]
(16)

Using now Lemma A.2 then yields
\[
\left\| \text{TR}^{t+1} - \text{TRUST} \right\|_1 \leq \beta \left\| \text{TR}^t - \text{TRUST} \right\|_1.
\]
(17)

By a straightforward induction, it follows that
\[
\left\| \text{TR}^t - \text{TRUST} \right\|_1 \leq \beta^t \left\| \text{TR}^0 - \text{TRUST} \right\|_1 = \beta^t \left\| \text{TRUST}^{pre} - \text{TRUST} \right\|_1.
\]
(18)

Using triangle inequality yields
\[
\left\| \text{TR}^t - \text{TRUST} \right\|_1 \leq \beta^t \left\| \text{TRUST}^{pre} - \text{TRUST} \right\|_1 = \beta^t \sum_{u \in U} \left| \text{TRUST}^{pre} - \text{TRUST}_u \right| \leq \beta^t |U|,
\]
(19)

given that each term $\text{TR}_u^0 - \text{TRUST}_u$ is necessarily in $[-1, 1]$, and thus has an absolute value at most 1. □

### A.2 Proof of Theorem 3.2

Intuitively the worst case holds when the Byzantine vouches for a large number of fake accounts without certified email or authentic vouches, which themselves each vouch for a large number of fake accounts without certified email or authentic vouches, and so on. Our formal proof instead relies on the study of the sequence $x^t$, which is defined by initialization $x^0 \triangleq \text{TRUST}^{-u}$ (where contributor $u$ is the one whose influence will be studied) and recursion $x^{t+1} \triangleq \min \left\{ \text{TRUST}^{pre} + \beta V^T x^t \right\}$ (the same vector recursion as for TR). This sequence intuitively corresponds to the computation of TRUST, starting from the case where contributor $u$ vouched only for the sink.

**Lemma A.4.** $x^t \rightarrow \text{TRUST}$ as $t \rightarrow \infty$.

**Proof.** Lemma A.3 actually applies to $x^t$ as well, since it only leverages the recursion of TR, which is the same as that of $x^t$. This not only implies that $x^t$ converges, but also that the recursion yields a unique limit. Thus $x^t$ has the same limit as TR, which is TRUST. □

**Lemma A.5.** $\left\| x^1 - x^0 \right\|_1 \leq \beta \text{TRUST}^{-u}$.

**Proof.** Denote $V^{-u}$ the vouch matrix obtained from $V$, but modified to reflect the fact that contributor $u$ only vouches for the sink, i.e.
\[
V^{-u}_{vk} \triangleq \begin{cases} V_{vk} & \text{if } v \neq u, \\ 0 & \text{if } v = u. \end{cases}
\]
(20)

The characteristic equation verified by TRUST$^{-u}$ is then given by
\[
\text{TRUST}^{-u} = \min \left\{ \text{TRUST}^{pre} + \beta (V^{-u})^T \text{TRUST}^{-u}, 1 \right\}.
\]
(21)

But then we can bound the difference between $x^1$ and $x^0 = \text{TRUST}^{-u}$ as (using also Proposition 3.1 and Lemma A.1)
\[
\left\| x^1 - x^0 \right\|_1 = \sum_{v \in U} \left\| \text{min} \left\{ \text{TRUST}^{pre}_v + \beta \sum_{k \in U} V_{vk} \text{TRUST}^{-u}_k, 1 \right\} - \text{min} \left\{ \text{TRUST}^{pre}_v + \beta \sum_{k \in U} V^{-u}_{vk} \text{TRUST}^{-u}_k, 1 \right\} \right\|
\]
(22)
\[
\leq \sum_{v \in U} \left| \left( \text{TRUST}^{pre}_v + \beta \sum_{k \in U} V_{vk} \text{TRUST}^{-u}_k \right) - \left( \text{TRUST}^{pre}_v + \beta \sum_{k \in U} V^{-u}_{vk} \text{TRUST}^{-u}_k \right) \right|
\]
(23)
\[
= \sum_{v \in U} |V_{uv} \text{TRUST}^{-u}_v| = \beta \text{TRUST}^{-u} \sum_{v \in U} V_{uv} \leq \beta \text{TRUST}^{-u},
\]
(24)

which is the lemma. □
Theorem 3.2. Denoting $\text{TRUST}^{-u}$ the trusts computed after removing $u$’s vouches (and thus having $u$ vouching only for the sink),

$$\sum_{v \in U} |\text{TRUST}_v - \text{TRUST}^{-u}_v| \leq \frac{\beta}{1 - \beta} \text{TRUST}^{-u}_u. \quad (25)$$

Proof. Using Lemma A.5 $\beta$-contractiveness and induction, it follows that $\|x^{t+1} - x^t\|_1 \leq \beta^t \|x^1 - x^0\|_1 \leq \beta^{t+1} \text{TRUST}^{-u}_u$. But then,

$$\|x^{t+1} - x^0\|_1 \leq \sum_{s=0}^t \|x^{s+1} - x^s\|_1 \leq \sum_{s=0}^t \beta^{s+1} \text{TRUST}^{-u}_u \leq \text{TRUST}^{-u}_u \sum_{s=0}^\infty \beta^{s+1} = \frac{\beta}{1 - \beta} \text{TRUST}^{-u}_u. \quad (26)$$

Taking this inequality to the limit $t \to \infty$ and invoking Lemma A.4 yields the theorem.

A.3 Other propositions

Proposition 3.3. For all contributors $u$, $\text{TRUST} \geq \text{TRUST}^{-u}$.

Proof. We consider the same notations as in the proof of Lemma A.5. Since $V \geq V^{-u}$, it is clear that $x^1 \geq x^0$. But then, by induction, given $V \geq V^{-u}$ and $x^t \geq x^0$, it follows from the recursion of $x^t$ and the characteristic equation of $\text{TRUST}^{-u}$ that $x^{t+1} \geq x^0$. Taking this to the limit yields the proposition.

Proposition 3.4. A contributor has a positive trust score, if and only if, there is a vouch path from a pretrusted contributor to them.

Proof. Consider a sequence $u_0, u_1, \ldots, u_k$ such that $u_0$ is pretrusted and such that $u_s$ has vouched for $u_{s+1}$ for all $s \in \{0, 1, \ldots, k - 1\}$. By induction over $s$, let us prove that, for all $t \geq s$, $\text{TR}_u^t \geq \beta^s \text{TRU}_{u_0}^{\text{pre}} \prod_{r=0}^{t-1} V_{u_r, u_{r+1}}$.

- For $s = 0$, this trivially follows from the fact that $\text{TR}_u^0 \geq \text{TRU}_{u_0}^{\text{pre}}$.\{TRU}_{u_0}^{\text{pre}}, 1\} = \text{TRU}_{u_0}^{\text{pre}}.

- Now assume that it holds for $s$, and let $t \geq s + 1$. Then $\text{TR}_u^t \geq \min\{V_{u_s, u_{s+1}} \text{TR}_u^{t-1}, 1\}$. Leveraging our induction assumption allows to close the inductive proof.

Now taking the induction guarantee to the limit $t \to \infty$ implies $\text{TR}_u \geq \beta^s \text{TRU}_{u_0}^{\text{pre}} \prod_{r=0}^{t-1} V_{u_r, u_{r+1}} > 0$. Thus, if there is a vouch path from a pretrusted contributor to a contributor, then this latter contributor has a positive trust score. Conversely, let us prove that, in the absence of such a vouch path, the contributor has a zero trust score. More precisely, define $U_{x \times}$ the set of contributors that are not pretrusted, and for which no path exists from pretrusted contributors to them. We prove by induction over $t$ that $\forall u \in U_{x \times}, \text{TR}_u^t = 0$.

- For $t = 0$, this trivially follows from the fact that contributors in $U_{x \times}$ are not pretrusted.

- Assume it holds for iteration $t$. Note that all contributors $u \in U_{x \times}$ are necessarily only vouched by other contributors in $U_{x \times}$, since, otherwise, there would be a path from a pretrusted contributor to them. As a result, if $V_{uu} > 0$, then $v \in U_{x \times}$, and thus, by induction $\text{TR}_v^t = 0$. But then, $(V^T)u \text{TR}_u^t = 0$. Since $\text{TRU}_{u_0}^{\text{pre}} = 0$, computing the recursion then yields $\text{TR}_u^{t+1} = 0$. This concludes the induction.

Taking $\forall u \in U_{x \times}, \text{TR}_u^t = 0$ then yields the other implication of the proposition.
B Generalized Bradley-Terry model

We rely on the generalized Bradley-Terry (GBT) model that has been recently introduced in [20]. In this model, the conditional law of the comparisons \( \tilde{r}_{ef} \) with respect to the score vector \( \theta \) is of the form

\[
P[\tilde{r}_{ef}|\theta] = \frac{\exp(-\theta_{ef} \tilde{r}_{ef})p[\tilde{r}_{ef}|0]}{\int_{-1}^{1} \exp(-\theta_{ef}s)p[s|0]ds}
\]

for some fixed probability law \( p[\tilde{r}|0] \) which corresponds to the probability law of a comparison for null scores. The denominator ensures that \( p[\tilde{r}_{ef}|\theta] \) is a pdf with integral 1. The function \( \theta \mapsto \int_{-\infty}^{\infty} \exp(\theta s)p[s|0]ds \) is the moment generating function of \( \tilde{r}_{ef}|\theta_{ef} = 0 \). Its logarithm is the cumulant generating function given by \( \Phi(\theta) = \log(\int_{-1}^{1} \exp(-\theta s)p[s|0]ds) \). The cumulant generating function \( \Phi \) is known to be strictly convex [20]. Its expression is moreover known for many probability law \( p[s|0] \). In this paper, we consider the case where \( p[s|0] = 1/2 \), i.e. the uniform law. This corresponds to

\[
\Phi(\theta) = \log \left( \frac{1}{2} \int_{-1}^{1} \exp(-\theta s)ds \right) = \log \frac{\sinh(\theta)}{\theta}.
\]

Using this relation, we recover the expression (6).

**The spring analogy.** To provide more insights into our model, a physical analogy may be useful. Interestingly, the loss of Bayesian GBT models corresponds precisely to the total energy of a physical system, where each alternative is a pebble put in a parabolic valley, and where (non-Hookean) springs of length at rest \( t_{ef}^* \) (which verifies \( \tilde{r}_{ef} = \frac{1}{t_{ef}^*} - \coth(t_{ef}^*) \)) are used to attach alternatives \( e \) to \( f \). A pebblenet would then be obtained, where pebbles naturally fall towards the bottom of the valley, but where springs may push them away from the bottom. If pebbles are then forced to lie along a one-dimensional line of the valley, then their equilibrium position along this line would correspond to the scores assigned to the alternatives when minimizing the contributor’s loss function \( L_{\text{score}}(\theta|\tilde{r}) \).

C Primitives and implementation details

To satisfy provable security guarantees, our algorithms repeatedly leverage a set of key Lipschitz-resilient primitives, many of which are adapted from [5]. In particular, we generalize them to account for asymmetric uncertainties on the inputs. But first, let us recall what is precisely meant by \( L \)-Lipschitz resilience.

**Definition C.1 ([5]).** Consider a function \( f: \mathbb{R}_+^U \times Z^U \to X \) that maps a pair \((w, z)\) of contributors’ voting rights \( w_u \) and of their inputs \( z_u \) to outputs \( f(w, z) \) that lie in a metric space \((X, d_X)\). We say that \( f \) is \((L, d_X)\)-Lipschitz resilient if it is \( L \)-Lipschitz continuous in voting rights, under \( \ell_1 \) norm for the voting rights, i.e.

\[
\forall w, w' \in \mathbb{R}_+^U, \forall z \in Z^U, \ d_X(f(w, z), f(w', z)) \leq L \| w - w' \|_1.
\]

In the context of Tournesol, \( z_u \) is typically the comparison tensor \( r_u \) reported by the contributor. But it may also be quantities derived from the tensor, such as scores with uncertainties. Note that this definition clearly implies that the maximal influence of a contributor \( u \) is proportional to its voting right. Indeed, measuring this influence amounts to comparing the effective voting rights \( w \) to the ones obtained by only canceling the contributor \( u \)'s voting right, i.e. \( w'_u \equiv 0 \) and \( w'_v \equiv w_v \) for \( v \neq u \).

C.1 Asymmetric Huber loss

Note that the asymmetric Huber loss is continuous and twice continuously differentiable in \( \mathbb{R} \), with a derivative given by

\[
\frac{d}{dm} \text{Huber}_\alpha(m|\Delta, \Delta) \equiv \begin{cases} \min \left\{ 1, \frac{1-\alpha}{\alpha} \right\} \frac{m-x}{\sqrt{\Delta_{\text{eq}}^2 + (x-m)^2}} \quad & \text{for } m \leq x, \\ \min \left\{ 1, \frac{\alpha}{1-\alpha} \right\} \frac{m-x}{\sqrt{\Delta_{\text{eq}}^2 + (x-m)^2}} \quad & \text{for } m \geq x. \end{cases}
\]
It is also easy to see that the second derivative is nonnegative, which proves that the asymmetric Huber loss is convex. In particular, its unique minimum is reached at $m = x$.

C.2 The quadratically regularized median $\text{QrMed}$ \cite{Mehestan2018}

**Definition.** The key primitive used by Mehestan is the *quadratically regularized median* $\text{QrMed}$, which we generalize to account for asymmetric uncertainties. Namely, given voting rights $w \in \mathbb{R}^U_+$, partial scores $x \in (\mathbb{R} \cup \{\bot\})^U$ and partial score left and right uncertainties $\Delta_{\text{left}}, \Delta_{\text{right}} \in (\mathbb{R} \cup \{\bot\})^U$, we define their $L$-quadratically regularized median by

$$\text{QrMed}_L(w, x, \Delta) \triangleq \arg\min_{m \in \mathbb{R}} \left\{ L_{\text{QrMed}_L}(m|w, x, \Delta) \triangleq \frac{m^2}{2L} + \sum_{u:x_u \neq \bot} w_u \text{Huber}_{1/2}(m | x_u, \Delta_u) \right\}. \quad (31)$$

This operator can be made to ignore some contributors. More precisely, if we want to restrict it to the subset $U_{\ell}$ of email-verified contributors, then we may write it as $\text{QrMed}_L(w_u, x_u, \Delta_u | u \in U_{\ell})$. The key safety property of $\text{QrMed}$ is $L$-Byzantine resilience, which we redefine here.

**Proposition C.2** (adapted from \cite{Mehestan2018}). $\text{QrMed}_L$ is well-defined, $L$-Lipschitz resilient and $1$-Lipschitz continuous in $x$ with respect to the $\ell_\infty$ norm.

**Proof.** Note that \cite{Mehestan2018} discusses $\text{QrMed}$ without uncertainty, which essentially corresponds to $\Delta_u \to 0$. However, their proof relies on controlling the derivative of $L'_{\text{QrMed}_L}(m | w, x, \Delta)$, and in particular on the fact that, if we focus only the term of $L'_{\text{QrMed}_L}(m | w, x, \Delta)$ that corresponds to a contributor $u$, then the derivative is at most $w_u$ in absolute value. These key properties still hold for our generalization of $\text{QrMed}$. Thus their proof applies to our setting as well. \hfill \Box

$\text{QrMed}$ has another desirable property. Namely, for any $\varepsilon > 0$, when the number of contributors with unit voting right is large enough (especially compared to $L$), then $\text{QrMed}$ returns a value that lies between the quantiles $1/2 - \varepsilon$ and $1/2 + \varepsilon$. In the limit of a very large number of contributors, $\text{QrMed}$ thus behaves essentially like a median.

**Computation.** Interestingly, $\text{QrMed}$ can be provably efficiently computed through dichotomic search. In practice, however, we run the Brent-Dekker method \cite{Brent1971} \cite{Press2007}, which provides better empirical performances, despite the lack of theoretical guarantee, by leveraging the smoothness of $L_{\text{QrMed}}$. Below we briefly describe this method, which relies on updating a lower bound $m$ and an upper bound $\overline{m}$, by leveraging the value of the derivative of the loss of Equation (31). This derivative is given by

$$L'_{\text{QrMed}_L}(m | w, x, \Delta) = \frac{m}{L} + \sum_{u:x_u \neq \bot} w_u \frac{m - x_u}{\sqrt{\Delta_{u,\text{left}}^2 1[m \leq x] + \Delta_{u,\text{right}}^2 1[m > x] + (x_u - m)^2}}. \quad (32)$$

The Brent-Dekker algorithm is implemented in the Python library\cite{Scipy} scipy.optimize.brentq, which we adapted (so that we can run it with numba for better performances).

**Proposition C.3.** The initialization step takes $O(\log (1 + |\text{QrMed}_L(w, x, \Delta)|))$ iterations. The dichotomic step takes $O(\log(1/\varepsilon))$ iterations. In both cases, each iteration requires computing $L'_{\text{QrMed}_L}$, which requires $O(|U|)$ steps.

**Subsampling for faster computation.** Note that the computation of $L'_{\text{QrMed}_L}$ may be costly, if there is a very large number of contributors with well-defined inputs. If so, then a random sampling of the contributors may be used instead, especially for early steps of the initialization and dichotomic search. In, investigating how subsampling should be done and what theoretical guarantees it provides is however left for future work.\footnote{For Tournesol, we set $\varepsilon \equiv 0.01$.}
C.3 The Byzantine-robustified mean BrMean [5]

We will use another robust statistics primitive, which successfully returns the mean of bounded inputs, when the amount of allocated voting rights is large enough. We call it BrMean, for Byzantine-Robustified Mean. To define it, we first introduce the Clipped Mean ClipMean centered on center and of radius radius by

\[
\text{ClipMean}(w, x, \Delta|\text{center}, \text{radius}) \triangleq \text{Mean}(w, \text{Clip}(x|\text{center}, \text{radius}))
\]

\[
= \frac{1}{\sum_{u \in U} w_u} \sum_{u \in U} w_u \text{Clip}(x_u|\text{center}, \text{radius})
\]

where Clip(x|center, radius) \( \triangleq \max \{\text{center} - \text{radius}, \min \{\text{center} + \text{radius}, x\}\} \) clips x within the interval [center - radius, center + radius]. Note that, while we made it appear explicitly to conform with other primitives, the uncertainty \( \Delta \) is never used. BrMean is then obtained by executing ClipMean, centered on QrMed, with a radius that grows linearly with the total amounts of votes:

\[
\text{BrMean}_L(w, x, \Delta) \triangleq \text{ClipMean}
\begin{pmatrix}
    w, x, \Delta
\end{pmatrix}
\text{QRMed}_L(w, x, \Delta), \frac{L}{4} \sum_{u \in U} w_u.
\]

Crucially for our purposes, BrMean has the following properties.

**Proposition C.4** [5]. BrMean\(_L\) is L-Byzantine resilient. Moreover, if there exists radius > 0 such that \( L \sum_{u \in U} w_u \geq 8\text{radius} \) and \( x_u \in [-\text{radius}, \text{radius}] \) for all \( u \in U \), then \( \text{BrMean}_L(w, x, \Delta) = \text{Mean}(w, x) \).

We stress, however, that BrMean is oblivious to uncertainty. It should thus only be used when there are enough data, so that the uncertainty can be neglected.

C.4 The quadratically regularized quantile QrQtl

We now introduce a new estimator, called the quadratically regularized quantile QrQtl, which depends on a quantile \( \alpha \) and on a Lipschitz resilience \( L \). We define it by

\[
\text{QrQtl}_{\alpha,L}(w, x, \Delta) \triangleq \arg\min_{m \in \mathbb{R}} \left\{ \frac{m^2}{2L} + \sum_{u: x_u \neq \perp} w_u \text{Huber}_\alpha(m|x_u, \Delta_u) \right\},
\]

**Proposition C.5.** \( \text{QrQtl}_{\alpha,L} \) is L-Lipschitz resilient.

**Proof.** For any \( x, \Delta \), the loss \( \ell(m|x, \Delta) \triangleq \text{Huber}(m|x, \Delta) \) has a derivative \( \ell'(m|x, \Delta) = \frac{d}{dm} \text{Huber}(m|x, \Delta) \), whose absolute value can be bounded by \( |\ell'(m|x, \Delta)| \leq 1 \). Fix \( w_1, w_2 \in \mathbb{R}^+_U \), and denote \( q_1 \triangleq \text{QrQtl}_{\alpha,L}(w_1, x, \Delta) \) and \( q_2 \triangleq \text{QrQtl}_{\alpha,L}(w_2, x, \Delta) \). By optimality of QrQtl, we know that the derivative of the loss must cancel, which implies

\[
q_1 = -L \sum_u w_{1,u} \ell'(q|x_u, \Delta_u) \quad \text{and} \quad q_2 = -L \sum_u w_{2,u} \ell'(q'|x_u, \Delta_u).
\]

Taking the difference and bounding its absolute value then yields

\[
|q_1 - q_2| \leq L \sum_u |w_{1,u} - w_{2,u}| |\ell'(q'|x_u, \Delta_u)| \leq L \|w_1 - w_2\|_1.
\]

This is precisely the needed Lipschitz resilience guarantee.

**Proposition C.6.** Assume \( \Delta = 0 \) and denote \( q_0 \) a weighted \( \beta \)-quantile of \( (w, x) \). If \( \beta \leq \alpha - |q_\beta|/2L \|w\|_1 \), then \( \text{QrQtl}_{\alpha,L}(w, x, \Delta) \geq q_\beta \). Similarly, if \( \beta \geq \alpha + |q_\beta|/2L \|w\|_1 \), then \( \text{QrQtl}_{\alpha,L}(w, x, \Delta) \leq q_\beta \). In particular, in the limit \( L \to \infty \) (no resilience) or \( \|w\|_1 \to \infty \) (high participation), \( \text{QrQtl}_{\alpha + o(1),L} \) becomes a great estimate of \( q_\beta \).

21
Now observe that, on input $u$ and $\Delta$, $Q_{\alpha}(q)$ is to the left of $\beta$ and $Q_{\alpha}^{\prime}(q)$ is to the left of $\beta$ and $\Delta$. More precisely, there exists $A$ such that, if $\Delta'_{u,\text{left}} \geq A$ and $\Delta'_{u,\text{right}} \geq A$, we have $\lim_{\Delta'_{u,\text{left}}, \Delta'_{u,\text{right}} \to \infty} \text{QRTQTL}_{\alpha,L}(u, x, (\Delta'_u, \Delta'_{-u})) = \text{QRTQTL}_{\alpha,L}(0, w_{-u}, x, \Delta)$. 

**Proposition C.7.** Let $u \in [U]$. For any $\alpha \in (0, 1)$, $L \geq 0$ and $w, x, \Delta$, we have

$$\lim_{\Delta'_{u,\text{left}}, \Delta'_{u,\text{right}} \to \infty} \text{QRTQTL}_{\alpha,L}(u, x, (\Delta'_u, \Delta'_{-u})) = \text{QRTQTL}_{\alpha,L}(0, w_{-u}, x, \Delta).$$

Put differently, a user with extremely large left and right uncertainties is asymptotically equivalent to a user with zero voting right.

**Proof.** Let $q = \text{QRTQTL}_{\alpha,L}(0, w_{-u}, x, \Delta)$ and $\epsilon > 0$. Now, since the loss of QRTQTL is a sum of a $\frac{1}{L}$-strongly convex term (the regularization) and of other convex terms, it is itself $\frac{1}{L}$-strongly convex. Thus for $q'$ such that $|q' - q| \geq \epsilon$, the derivative of QRTQTL for inputs $(0, w_{-u}, x, \Delta$ at $q'$ is, in absolute value, at least $\epsilon / L$. Now observe that, on input $w, x, \Delta$, this derivative is the same as above, plus a term $w_{u} \frac{d}{dq} \text{Huber}_{\alpha}(q' | x_{u}, \Delta_{u})$ for user $u$. Note however that, on $[q - \epsilon, q + \epsilon]$ (in fact, on any interval), the term $w_{u} \frac{d}{dq} \text{Huber}_{\alpha}(q' | x_{u}, \Delta_{u})$ converges uniformly to zero as $\Delta'_{u,\text{left}}, \Delta'_{u,\text{right}} \to \infty$. More precisely, there exists $A$ such that, if $\Delta'_{u,\text{left}} \geq A$ and $\Delta'_{u,\text{right}} \geq A$, we have $\lim_{\Delta'_{u,\text{left}}, \Delta'_{u,\text{right}} \to \infty} \text{QRTQTL}_{\alpha,L}(u, x, (\Delta'_u, \Delta'_{-u})) = \text{QRTQTL}_{\alpha,L}(0, w_{-u}, x, \Delta)$. Thus the QRTQTL must be $[q - \epsilon, q + \epsilon]$, which concludes the proof.
C.5 The quadratically regularized deviation \( \text{QrDev} \)

We now propose a Lipschitz-resilient measure of standard deviation, which we call the \textit{Quadratically Regularized Deviation}, or \( \text{QrDev} \). We essentially define it as a \( \text{QRQTL} \) of the deviation to \( \text{QrMED} \). More specifically, given the quadratically regularized median \( m \triangleq \text{QrMED}_L(w, x, \Delta) \) and a default deviation \( \Delta_{default} \), \( \text{QrDev} \) is computed by

\[
\text{QrDEV}_{\alpha,L,\Delta_{default}}(w, x, \Delta) \triangleq \Delta_{default} + \text{QRQTL}_{\alpha,L}(w_u, |x_u - m| - \Delta_{default}, \Delta_u | u \in U).
\]

(50)

Note that, rather than a measure of deviation within the values of \( x_u \), \( \text{QrDEV} \) should rather be regarded as a deviation between these values and \( \text{QrMED} \). It may also be understood as a measure of polarization. More precisely, it would take large values if most contributors disagree with the quadratically regularized median.

\[\text{Proposition \ C.8.} \ \text{QrDEV}_{\alpha,L,\Delta_{default}} \text{ is } 2L\text{-Lipschitz resilient.}\]

\[\text{Proof.} \ \text{Essentially, the proposition results from the Lipschitz resilience of } \text{QrMED} \text{ and } \text{QRQTL, and the 1-Lipschitz continuity of } \text{QRQTL} \text{ with respect to a translation of its second inputs. Let } w, w' \in \mathbb{R}^L. \text{ By Proposition \ C.2 denoting } m \triangleq \text{QrMED}_L(w, x, \Delta) \text{ and } m' \triangleq \text{QrMED}_L(w', x, \Delta), \text{ we know that } |m - m'| \leq \| w - w' \|_1 / L. \text{ Denoting } d \text{ and } d' \text{ the } \text{QrDEV} \text{ for } w \text{ and } w', \text{ we have}
\]

\[
|d - d'| \leq |\text{QRQTL}_{\alpha,L}(w_u, |x_u - m| - \Delta_{default}, \Delta_u) - \text{QRQTL}_{\alpha,L}(w'_u, |x'_u - m'| - \Delta_{default}, \Delta_u)|
\]

\[
+ |\text{QRQTL}_{\alpha,L}(w'_u, |x'_u - m'| - \Delta_{default}, \Delta_u) - \text{QRQTL}_{\alpha,L}(w'_u, |x'_u - m'| - \Delta_{default}, \Delta_u)|
\]

\[
\leq |m - m'| + L \| w - w' \|_1 \leq 2L \| w - w' \|_1,
\]

(51)

which amounts to saying that \( \text{QrDEV}_L \) is \((2L)\)-Byzantine resilient.

\[\]

C.6 The quadratically regularized uncertainty

The quadratically regularized uncertainty aims to provide a meaningful estimate on the estimation of \( \text{QrMED} \). In our default pipeline the quadratically regularized uncertainty is simply made to return \( \text{QrDEV} \). This is a pessimistic assumption, if the data is i.i.d. However, given that we do not expect i.i.d. data, we considered this more conservative estimator of the uncertainty on \( \text{QrMED} \). We leave the problem of deriving more principled uncertainty estimates open.

D Mehestan’s algorithms

This section details the algorithms used by Mehestan.

D.1 Collaborative preference scaling for scaling-calibration contributors

For any contributor \( u \in U \), define the set of clearly ordered pairs of alternatives by \( u \), by

\[\text{EF}_u \triangleq \{(e, f) \in \text{EF}_u \mid \theta_{ue}^{\text{raw}} \geq \theta_{uf}^{\text{raw}} + 2\Delta \theta_{ue,\text{left}}^{\text{raw}} + 2\Delta \theta_{uf,\text{right}}^{\text{raw}} \text{ or } \theta_{uf}^{\text{raw}} \geq \theta_{ue}^{\text{raw}} + 2\Delta \theta_{ue,\text{right}}^{\text{raw}} + 2\Delta \theta_{uf,\text{left}}^{\text{raw}}\}.\]

(53)

We then say that two contributors \( u \) and \( v \) are \textit{comparable}, if and only if, there exists at least one pair of alternatives that both scored sufficiently differently, i.e. \( \text{EF}_u \cap \text{EF}_v \neq \emptyset \). For any two comparable contributors \( u \) and \( v \), each pair \((e, f) \in \text{EF}_u \cap \text{EF}_v\) may then serve as an anchor point to appropriately scale the two contributors’ scores. This leads us to define \( s_{uef} \triangleq \frac{|\theta_{ue}^{\text{raw}} - \theta_{uf}^{\text{raw}}|}{|\theta_{ue}^{\text{raw}} - \theta_{uf}^{\text{raw}}|} \). We also compute their left
uncertainty as

\[ \Delta s_{uvef, \text{left}} \triangleq \left\{ \begin{array}{ll}
\frac{\theta_{\text{raw}} - \theta_{\text{raw}} - \Delta \theta_{\text{raw}}}{\theta_{\text{raw}} - \theta_{\text{raw}}, \theta_{\text{raw}} - \Delta \theta_{\text{raw}}} & \text{if } \theta_{\text{raw}} \geq \theta_{\text{raw}} \text{ and } \theta_{\text{raw}} \geq \theta_{\text{raw}}, \\
\frac{\theta_{\text{raw}} - \theta_{\text{raw}} + \Delta \theta_{\text{raw}}}{\theta_{\text{raw}} - \theta_{\text{raw}}, \theta_{\text{raw}} + \Delta \theta_{\text{raw}}} & \text{if } \theta_{\text{raw}} \leq \theta_{\text{raw}} \text{ and } \theta_{\text{raw}} \leq \theta_{\text{raw}},
\end{array} \right. \]

and similarly for the right uncertainty:

\[ \Delta s_{uvef, \text{right}} \triangleq \left\{ \begin{array}{ll}
\frac{\theta_{\text{raw}} - \theta_{\text{raw}} - \Delta \theta_{\text{raw}}}{\theta_{\text{raw}} - \theta_{\text{raw}}, \theta_{\text{raw}} - \Delta \theta_{\text{raw}}} & \text{if } \theta_{\text{raw}} \geq \theta_{\text{raw}} \text{ and } \theta_{\text{raw}} \geq \theta_{\text{raw}}, \\
\frac{\theta_{\text{raw}} - \theta_{\text{raw}} + \Delta \theta_{\text{raw}}}{\theta_{\text{raw}} - \theta_{\text{raw}}, \theta_{\text{raw}} + \Delta \theta_{\text{raw}}} & \text{if } \theta_{\text{raw}} \leq \theta_{\text{raw}} \text{ and } \theta_{\text{raw}} \leq \theta_{\text{raw}},
\end{array} \right. \]

Assume \( EF_u^* \cap EF_v^* \neq \emptyset \). Now, to mitigate noise, we estimate the relative scaling between two contributors \( u \) and \( v \), using a mean robust estimator. Namely, we consider the quadratically regularized median and the quadratically regularized uncertainty estimator, which yields

\[ s_{uv} \triangleq 1 + \text{QRMED}_{L_{\text{user}}} \left(1, s_{uvef} - 1, \Delta s_{uvef} \mid (e, f) \in EF_u^* \cap EF_v^* \right). \]  \hspace{1cm} (56)

\[ \Delta s_{uv} \triangleq \text{QRUNC}_{L_{\text{user}}, 1} \left(1, s_{uvef} - 1, \Delta s_{uvef} \mid (e, f) \in EF_u^* \cap EF_v^* \right). \]  \hspace{1cm} (57)

Note that these estimates aggregate the influences from a single other contributor \( v \). This is why they only need robustness against noise, not against malicious nodes. Hence the use of a “large” Lipschitz parameter \( L_{\text{user}} = 10 \). To determine the preference re-scaling of a scaling-calibration contributor \( u \), we leverage the relative re-scaling with respect to all comparable scaling-calibration contributors, by using the Byzantine-robustified mean, as

\[ s_u \triangleq 1 + \text{QRMED}_{L} \|s_{\text{raw}}\|_\infty \left( w_{\text{scaling}}^{\text{scaling}}, s_{uv} - 1, \Delta s_{uv} \mid v \in U_{\text{scaling}} \right), \]  \hspace{1cm} (58)

\[ \Delta s_u \triangleq \text{QRUNC}_{L} \|s_{\text{raw}}\|_\infty, 1 \left( w_{\text{scaling}}^{\text{scaling}}, s_{uv} - 1, \Delta s_{uv} \mid v \in U_{\text{scaling}} \right), \]  \hspace{1cm} (59)

where \( U_{\text{scaling}} \triangleq \left\{ v \in U_{\text{scaling}} \mid EF_u^* \cap EF_v^* \neq \emptyset \right\} \) is the set of scaling-calibration contributors comparable to \( u \). Note that the contributor should include themselves, with \( w_{\text{scaling}}^{\text{scaling}} = 1 \), \( s_{uv} = 1 \) and \( \Delta s_{uv} = 0 \). We define similarly the translation normalization \( \tau_{uv} \) of contributor \( u \), compared to contributor \( v \). To do so, let us first denote \( E_{uv} \triangleq \{ \epsilon \in E \mid \exists f \in E, \tau_{uf} \neq \perp \text{ and } \tau_{vf} \neq \perp \} \) the set of alternatives that both contributors \( u \) and \( v \) scored. We then define

\[ \tau_{uv} \triangleq \text{QRMED}_{1} \left(1, s_{\text{raw}} - s_{\text{raw}}, s_{\text{raw}} \Delta s_{\text{raw}} + s_{\text{raw}} \Delta \theta_{\text{raw}} \mid e \in E_{uv} \right), \]  \hspace{1cm} (60)

\[ \Delta \tau_{uv} \triangleq \text{QRUNC}_{1, 1} \left(1, s_{\text{raw}} - s_{\text{raw}}, s_{\text{raw}} \Delta s_{\text{raw}} + s_{\text{raw}} \Delta \theta_{\text{raw}} \mid e \in E_{uv} \right). \]  \hspace{1cm} (61)

To determine the preference translation of a contributor \( u \), we then leverage the relative translation with respect to all comparable contributors, by defining

\[ \tau_u \triangleq \text{QRMED}_{L} \|s_{\text{raw}}\|_\infty \left( w_{\text{scaling}}^{\text{scaling}}, \tau_{uv}, \Delta \tau_{uv} \mid v \in U_{\text{scaling}} \right), \]  \hspace{1cm} (62)

\[ \Delta \tau_u \triangleq \text{QRUNC}_{L} \|s_{\text{raw}}\|_\infty, 1 \left( w_{\text{scaling}}^{\text{scaling}}, \tau_{uv}, \Delta \tau_{uv} \mid v \in U_{\text{scaling}} \right). \]  \hspace{1cm} (63)
Again each user should include themselves, with $\tau_{uu} = 0$ and $\Delta \tau_{uu} = 0$. Scaling-calibration contributor $u$'s Mehestan-scaled score is then given by

$$\theta^{\text{Mehestan}}_{ue} \triangleq s_u \theta^\text{raw}_{ue} + \tau_u. \quad (64)$$

We define the uncertainty on the scaled score by

$$\Delta \theta^{\text{Mehestan}}_{ue} \triangleq s_u \Delta \theta^\text{raw}_{ue} + \Delta \tau_u + \begin{cases} |\theta^\text{raw}_{ue}| (\Delta s_{u,\text{right}}, \Delta s_{u,\text{left}}) & \text{if } \theta^\text{raw}_{ue} \geq 0 \\ \Delta s_{u,\text{right}} & \text{if } \theta^\text{raw}_{ue} < 0 \end{cases} \quad (65)$$

D.2 Scaling of non-scaling-calibration contributors

Nonscalers are then made to fit the scalers' scaled scores ($\theta^{\text{Mehestan}}_u, \Delta \theta^{\text{Mehestan}}_u$), using similar techniques. One slight difference occurs in equations (58) and (62), where $\text{BrMean}$ is used instead of $\text{QrMED}$, and where the scaled user excludes themselves.

E Hyperparameters of the synthetic evaluation

In this section we detail the hyperparameters of the experiments. Note that they are also provided as is in the Supplementary Material, and that experiments can be effortlessly reproduced by specifying their use.

E.1 Resilience

Below, we list the hyperparameters of the resilience experiment.

```json
{
    "title": "Resilience of Solidago to untrustworthy users",
    "ylegend": "Correlation between learned global and ground truth",
    "xlegend": "Fraction of trustworthy users",
    "xparameter": "generative_model.user_model.p_trustworthy",
    "xvalues": [0.01, 0.2, 0.5, 0.8, 1.0],
    "zparameter": "pipeline.aggregation.lipschitz",
    "zvalues": [0.1, 0.3, 1, 3],
    "zlegends": ["Lipschitz = 0.1", "Lipschitz = 0.3", "Lipschitz = 1", "Lipschitz = 3"],
    "n_users": 30,
    "n_entities": 50,
    "n_seeds": 100,
    "generative_model": {
        "user_model": ["NormalUserModel", {
            "p_trustworthy": 0.8,
            "p_pretrusted": 0.2,
            "zipf_vouch": 2.0,
            "zipf_compare": 1.5,
            "poisson_compare": 30.0,
            "n_comparisons_per_entity": 3.0,
            "multiplicator_std_dev": 1.0,
            "svd_mean": [3.0, 0.0],
            "engagement_bias_std_dev": 0.0
        }],
        "vouch_model": ["ErdosRenyiVouchModel"],
        "entity_model": ["NormalEntityModel", {
            "mean": [0.0, 0.0]
        }],
        "engagement_model": ["SimpleEngagementModel", {
```
"p_per_criterion": {"0": 1.0},
"p_private": 0.2
],
"comparison_model": ["KnaryGBT", {
"n_options": 21,
"comparison_max": 10
}]
},
"pipeline": {
"trust_propagation": ["LipschiTrust", {
"pretrust_value": 0.8,
"decay": 0.8,
"sink_vouch": 5.0,
"error": 1e-08
}]
},
"voting_rights": ["AffineOvertrust", {
"privacy_penalty": 0.5,
"min_overtrust": 2.0,
"overtrust_ratio": 0.1
}]
},
"preference_learning": ["UniformGBT", {
"prior_std_dev": 7,
"comparison_max": 10,
"convergence_error": 1e-05,
"cumulant_generating_function_error": 1e-05
}]
},
"scaling": ["ScalingCompose", ["Mehestan", {
"lipschitz": 1,
"min_activity": 1,
"n_scalers_max": 100,
"privacy_penalty": 0.5,
"user_comparison_lipschitz": 10,
"p_norm_for_multiplicative_resilience": 4.0,
"error": 1e-05
}]
],
"aggregation": ["QuantileStandardizedQrMedian", {
"dev_quantile": 0.9,
"lipschitz": 0.1,
"error": 1e-05
}]
},
"post_process": ["Squash", {
"score_max": 100
}]
}
E.2 Engagement bias

Below, we list the hyperparameters of the engagement-bias experiment.

```json
{
    "title": "The role of Mehestan to fix engagement bias",
    "ylegend": "Correlation between learned global and ground truth",
    "xlegend": "Amount of engagement bias",
    "xparameter": "generative_model.user_model.engagement_bias_std_dev",
    "xvalues": [0.01, 1.0, 4.0, 10.0],
    "zparameter": "pipeline.scaling.scalings.0.lipschitz",
    "zvalues": [1.0, 3.0, 10.0, 30.0, 100.0],
    "zlegends": ["Lipschitz = 1", "Lipschitz = 3", "Lipschitz = 10", "Lipschitz = 30", "Lipschitz = 100"],
    "n_users": 30,
    "n_entities": 100,
    "n_seeds": 100,
    "generative_model": {
        "user_model": ["NormalUserModel",
                       {
                           "p_trustworthy": 0.8,
                           "p_pretrusted": 0.5,
                           "zipf_vouch": 1.5,
                           "zipf_compare": 1.5,
                           "poisson_compare": 30.0,
                           "n_comparisons_per_entity": 3.0,
                           "multiplicator_std_dev": 1.0,
                           "svd_mean": [3.0, 0.0],
                           "engagement_bias_std_dev": 10.0
                       }],
        "vouch_model": ["ErdosRenyiVouchModel"],
        "entity_model": ["NormalEntityModel",
                         {
                             "mean": [0.0, 0.0]
                         }],
        "engagement_model": ["SimpleEngagementModel",
                              {
                                  "p_per_criterion": {"0": 1.0},
                                  "p_private": 0.2
                              }],
        "comparison_model": ["KnaryGBT",
                              {
                                  "n_options": 21,
                                  "comparison_max": 10
                              }]
    },
    "pipeline": {
        "trust_propagation": ["LipschiTrust",
                              {
                                  "pretrust_value": 0.8,
                                  "decay": 0.8,
                                  "sink_vouch": 5.0,
                                  "error": 1e-08
                              }],
        "voting_rights": ["AffineOvertrust",
                          {
                              "privacy_penalty": 0.5,
                              "min_overtrust": 2.0,
                              "overtrust_ratio": 0.1
                          }],
        "preference_learning": ["UniformGBT",
                                 {
                                     "prior_std_dev": 7,
                                     "comparison_max": 10
                                 }]
    }
}
```
F Selecting the hyperparameters

In this section, we discuss our hyperparameter selection, by analyzing how the hyperparameters affect the distributions of individual and global squashed scores. In particular, we study how the distributions of the scores vary, depending on the number of comparisons and of contributors the alternatives received. Recall that, for Tournesol, the selected hyperparameters are $\alpha_{\text{user prior}} \approx 0.02$, $q_{\text{zero shift}} \approx 0.15$, $q_{\text{sss dev}} \approx 0.9$ and $L \approx 0.1$. The graphs are all obtained based on Tournesol’s full dataset, which contains private inputs. Note that similar graphs can be obtained using only Tournesol’s public dataset, which can be downloaded from Tournesol’s website at https://tournesol.app/#research.

F.1 The prior weight $\alpha_{\text{user prior}}$

For the relatively large value $\alpha_{\text{user prior}} = 0.2$ (see bottom left graph of Figure 5), we see that alternatives can only reach very high and very low scores if they are compared a large number of times. This is typical of a regularization that is too large. On the other hand, our concern for $\alpha_{\text{user prior}} = 0.002$ was that the noise could be too amplified, i.e., a video may already have an extreme score despite being compared only once.

F.2 The zero-shifting quantile $q_{\text{zero shift}}$

The effect of the zero-shifting quantile $q_{\text{zero shift}}$ is clearly visible in Figure 6. Namely, as the quantile $q_{\text{zero shift}}$ grows (from top to bottom of the figure), the distribution of individual scores is shifted closer to the middle. We selected the middle value $q_{\text{zero shift}}$ as most individual scores were then around 50, leaving enough room for comparisons.
Figure 5: Impact of the prior weight $\alpha_{\text{prior}}$ on the distributions of squashed individual and global scores on the upper end to distinguish top videos, while also marking a clear difference between rated videos on Tournesol and those that were never rated, thereby enforcing a form of presumed non-recommendability for alternatives without evaluation.

Figure 6: Impact of the zero-shifting quantile $q_{\text{zero shift}}$ on the distributions of squashed individual and global scores

F.3 The collaboratively scaled individual score deviation quantile $q_{\text{dev}}^{\text{sss}}$

As explained in the main part of the paper, the distribution of collaboratively scaled scores is very heavy-tailed. This is clearly evidenced by the top-left graph of Figure 7. Indeed, the median deviation to the median (which corresponds to $q_{\text{dev}}^{\text{sss}} = 0.5$) is significantly smaller than the deviations of many scores, which implies that, after standardizing, shifting and squashing using $q_{\text{dev}}^{\text{sss}} = 0.5$, many individual scores take the extreme values $-100$ and 100. We leaned on $q_{\text{dev}}^{\text{sss}} = 0.9$, as we deem it reasonable if at most 10% of the scores take the extreme values $-100$ and 100. Empirically, we found that such extreme values are rarer, which makes this value reasonable.
Figure 7: Impact of the collaboratively scaled individual score deviation quantile $q_{dev}$ on the distributions of squashed individual and global scores

F.4 The Lipschitz resilience $L$

Lipschitz resilience only really matters when aggregating scores into global scores, which is why we only plot them in Figure 9. We see that a smaller choice of $L$ implies that videos must be scored by many contributors to have scores that really deviate from zero. However, we feared that for $L = 0.05$, only videos score by a very large number of users can obtain a noteworthy recommendability score. On the other hand, for $L = 0.2$, the global scores can already exceed 40 with only 3 contributors. Note that Tournesol also uses a recommendability threshold, which we set at 20. In other words, Tournesol will not recommend any video whose global score is below this threshold. Interestingly, given that each contributor $u$ can directly influence the global scores $\rho^{\text{agg}}$ only by $Lw_u$, since the voting rights are always at most unit ($w_u \leq 1$), and since the displayed scores are squashed, for $L = 0.1$, the maximal score of a video with two contributors is $100 \frac{2L}{\sqrt{1 + (2L)^2}} \approx 19.6 < 20$. Put differently, at least three contributors must be involved to make a content recommendable on Tournesol. Evidently, Tournesol may modify the security hyperparameter $L$ as the platform gains in importance.

G Impact of the aggregation quantile

This section presents experiments on the Tournesol public dataset [29], where the aggregation quantile $\alpha$ is changed, from 0.1 to 0.8. Namely, we consider the default pipeline, but with an aggregation whose quantile changes. Moreover, to limit the effect of low numbers of evaluations of a video, we both consider a very large Lipschitz resilience for aggregation ($L \equiv 100$), and we only consider 484 videos with at least 100 comparisons from at least 10 contributors. Figure 10 tracks the ranking of a sample of 100 of these videos, when $\alpha$ is changed. Perhaps unsurprisingly, the videos with the most negative ranking change are politicized, or discuss polarizing topics such as meat consumption.

H Open research challenges

In this section, we briefly list research venues which we believe to be important for any secure and collaborative algorithmic governance system design.
H.1 Accounting for direct assessments

Users issue preferences between alternatives, which are converted into individual scores via the generalized Bradley-Terry model. We also want to make it possible to express assessments (likes or dislikes on the alternatives) which will be taken into account in the estimation of the scores. This presupposes adapting the BT type models to these new data, what we plan for future work.

H.2 Distinguishing instinctive preferences from thoughtful volitions

It has been empirically shown that more reflexive judgments, which are related to what philosophers sometimes call volitions [2, 54], lead to statistically different judgments. Some studies even find that they seem more inclusive to outgroup members [11]. As initiated by [33], it would be worthwhile to construct algorithms capable of making such a distinction, by learning systematic bias between judgments provided with more or less thought. In particular, the Tournesol dataset contains both judgments made only along the main criterion, as well as judgments made while considering all criteria, which area likely to be more thoughtful. Future work should investigate whether this can be leveraged.

H.3 Enabling liquid democracy

One of the key limitations of the platform, is that most humans are not very active on it. To increase (indirect) participation, liquid democracy was proposed [10, 15, 27], where each voter can delegate their vote to any other voter of their choice. A platform like Tournesol seems like a promising venue to further develop, deploy and test such principles, and to adjust them to security and undesirable group dynamic concerns.

H.4 Leveraging (vouched) expertise

Future work should investigate how to certify and leverage contributors’ expertise, e.g. based on the topic of a content and the quality criterion under consideration. In particular, it would be worth partnering with institutions like Amnesty or the IPCC, e.g., to know their members’ top content recommendations. A peer-to-peer and institution-to-peer vouching system may be a promising approach for certification.
H.5 Online updates

Our algorithms are unfortunately slow to run. On Tournesol, they require 40 minutes of computation. This means that they cannot be run whenever a contributors enters a comparison, which prevents near-instantaneous feedbacks. Heuristics have yet to be designed to allow this, which is arguably a key component for interpretability. Moreover, the guarantees of such heuristics would need to be analyzed.

H.6 Distributed computing

Future work should also aim to distribute the computational workload, e.g. by asking contributors to perform operations on their machines. Moreover, distributed computing could allow distributed verifiability, i.e. contributors would no longer have to trust that a central server (like Tournesol’s) correctly runs our algorithms.

H.7 Active learning

Tournesol mostly relies on contributors to select which alternatives to compare. Future work should investigate active learning algorithms to best help contributors to effortlessly provide highly information-valuable comparisons. Keeping contributors motivated is particularly important and challenging.

H.8 Robust raw score generalization.

For security reasons, contributors currently only influence the score of the alternatives they explicitly compared. Using machine learning, contributors’ scores could be generalized, to essentially guess how a given contributor would have likely scored the alternatives that they did not explicitly compare, and to take this into account to compute global scores.
(a) Among a sample of 100 videos.  
(b) Among the 10 videos with the largest ranking changes.  

Figure 10: Changes in ranking

H.9 Byzantine collaborative filtering.

Another approach to score generalize relies on collaborative filtering. However, collaborative filtering has little security guarantees. Future research should investigate the design of Byzantine-resilient collaborative filtering algorithms.

H.10 Bayesian voting.

Unfortunately, we suspect large biases in the population of contributors. To correct these biases, a promising approach would consist of predicting what non-participating humans would have likely voted, e.g. based on what similar contributors have voted. Intuitively, this would amount to assigning more voting rights to contributors who represent large under-represented populations. This raises serious security concerns though, as malicious contributors would be incentivized to claim to be from such under-represented populations to increase their voting rights.

H.11 Measuring Tournesol’s impact.

Since Tournesol’s end goal is to improve information prioritization, it is important to determine the extent to which the daily use of Tournesol modifies users’ content consumption habits. Experiments should be conducted to do so.

H.12 Aligned language models.

Tournesol essentially collaboratively labels which content is desirable to repeat. This could be used to train language models that are more aligned with human preferences.
H.13 Provide end-to-end guarantees

We did not provide end-to-end security guarantees. Doing so without sacrificing accuracy is another of the many exciting future works. But as a result, we expect Tournesol’s algorithms to be regularly updated in the coming years. Nevertheless, we believe that much of the pipeline structure presented in the paper will remain relevant, both for Tournesol and other large-scale secure participatory projects.