Theoretical Computer Science 911 (2022) 92–123

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The power of adaptivity in source identification with time queries on the path

Victor Lecomte[a], Gergely Ódor[b,*], Patrick Thiran[b]

[a] Stanford University, USA
[b] EPFL, Switzerland

ARTICLE INFO

Article history:
Received 20 November 2020
Received in revised form 29 December 2021
Accepted 8 February 2022
Available online 14 February 2022
Communicated by L. Christoph

Keywords:
Graph algorithms
Source location
Noisy information
Lower bounds

ABSTRACT

We study the problem of identifying the source of a stochastic diffusion process spreading on a graph based on the arrival times of the diffusion at a few queried nodes. In a graph $G = (V, E)$, an unknown source node $v^* \in V$ is drawn uniformly at random, and unknown edge weights $w(e)$ for $e \in E$, representing the propagation delays along the edges, are drawn independently from a Gaussian distribution of mean 1 and variance σ^2. An algorithm then attempts to identify v^* by querying nodes $q \in V$ and being told the length of the shortest path between q and v^* in graph G weighted by w. We consider two settings: *non-adaptive*, in which all query nodes must be decided in advance, and *adaptive*, in which each query can depend on the results of the previous ones. Both settings are motivated by an application of the problem to epidemic processes (where the source is called patient zero), which we discuss in detail.

We characterize the query complexity when G is an n-node path. In the non-adaptive setting, $\Theta(n\sigma^2)$ queries are needed for $\sigma^2 \leq 1$, and $\Theta(n)$ for $\sigma^2 \geq 1$. In the adaptive setting, somewhat surprisingly, only $\Theta(\log\log_{1/\sigma} n)$ are needed when $\sigma^2 \leq 1/2$, and $\Theta(\log\log n) + O_\sigma(1)$ when $\sigma^2 \geq 1/2$. This is the first mathematical study of source identification with time queries in a non-deterministic diffusion process.

1. Introduction

When a diffusion process spreads in a network, identifying its *source*, i.e., the first node v^* that started the diffusion, is a difficult and intriguing task. Depending on the application, the diffusion process can model a variety of real-world phenomena, including a worm in a computer network [54], a false-rumor in a social network [43], or an epidemic process [37]. In epidemics, the identification of the source (also called patient zero) can be useful while planning our response as a society, since any information on the disease is crucial in uncertain times [18] (e.g. source identification can aid contact tracing efforts [6,40], or it can give information on how dangerous the outbreak is in the case of a new mutation [19,24]). While often useful, we note that the identification of the source may be undesired in certain cases due to privacy concerns [15,38]; we refer to [13,14] for theoretical work addressing this issue. Since the majority of the recent work in source identification is focused on disease spreading, we adopt the language of epidemics in the introduction.

[*] Corresponding author.
E-mail addresses: vlecomte@stanford.edu (V. Lecomte), gergely.odor@epfl.ch (G. Ódor), patrick.thiran@epfl.ch (P. Thiran).

https://doi.org/10.1016/j.tcs.2022.02.008

If we could observe the entire process of the epidemic propagation and know the precise infection times, identifying its source would be easy. Unfortunately, due to the costs of information collection and the overhead constraints, the data available for source identification is usually very sparse. There are two popular frameworks for source identification that mathematically formalize the data-sparsity constraint: in the setting with binary queries (also called snapshot-based setting), proposed by [43], every node reveals whether they are infected or not at some time t, whereas in the setting with time queries (also called sensor-based setting), proposed by [37], a small subset of nodes, which we call hereafter *queries* or *query nodes*, reveal their infection time after the epidemic has spread to the entire network. The two frameworks are quite different, and in this paper we consider only the formulation with time queries, which is driven by the following three research questions of increasing complexity (as identified by [60]):

(i) given the answers to a fixed set of queries, how can we estimate the source?
(ii) given the maximum number of queries that we can ask, which queries should we choose so that we can solve the estimation problem as accurately as possible?
(iii) if we want to correctly identify the source, what is the minimum number of queries that we need to ask?

We note that in theoretical papers, (ii) and (iii) are difficult to separate, however, in applied papers (ii) is often solved before (iii).

The answers to these three research questions depend on the specific assumptions on the epidemic model. The original paper [37] assumed that the epidemic spreads on a fixed (and known) network following the Susceptible-Infected dynamics [29] (also known as first passage percolation [1]) with a known edge-delay distribution to model the randomness in the spread of the epidemic. Additionally, it is assumed that the epidemic has already infected everyone in the network, and therefore every query node can reveal their precise infection time. Finally, in [37] it is assumed that query nodes also reveal the neighbor from whom they received the infection. This last assumption relies on information that is difficult to obtain in the context of epidemics, and most follow-up works have dropped it. The resulting modified version of [37], which we call S1, is the most popular model in source identification with time queries, and has been the subject of a long list of papers, which address problem (i) [17,26,33,34,44,51,57,62] and problem (ii) (see [32], and the references therein) by algorithms of heuristic nature. Our goal in this paper is to rigorously address problem (iii), albeit on a simpler network model, the path network. The epidemic model in our paper, which we call S2, is exactly the same as S1 with one additional assumption: we assume that the infection time of the source is known. We have several reasons to focus on the S2 model instead of the S1 model, including that S2 is easier to define and it is theoretically more appealing, as pointed out by several papers in the field [58,30], and that there is little difference between the number of queries required in the two models [50]. We further discuss the differences between the two models and how our results can be extended to S1 in Appendix B.

One of the main criticisms of source identification algorithms is that the number of queries required to find the source is large. Although this has not been shown theoretically before our paper, it is widely accepted that source identification is possible only if a constant fraction of the population are queried, which makes the developed algorithms unfit for real-world scenarios. To remedy the situation, a recent research direction suggests to give up the exact identification of the source and to replace it by the computation of confidence sets around it, which can be done with fewer queries (see [4,21] for the binary query and [9] for the time query settings). However, if our goal is to find the source exactly without querying a prohibitively large fraction of the population, the underlying model needs to be changed. A promising approach is to allow the queries to be selected adaptively to previous answers [59,60], which we call *the adaptive setting*. Adaptive strategies have been studied by Spinelli, Celis and Thiran [47,48] by simulations, and they show a large reduction in the number of required queries in real networks. It is important to quantify the magnitude of the reduction, because it is safe to assume that adaptive queries incur more expensive operational costs than the non-adaptive queries, and it is possible that placing $\Theta(\sqrt{n})$ queries adaptively still remains infeasible in practice. To be self-contained, we include simulations in Fig. 1(a), which suggest that in the adaptive setting, the number of required queries grows slowly as a function of the network size, especially on geometric networks. But whether the growth is logarithmic or even lower is difficult to estimate from such plots. In this paper, we show that on the path network, we only need $\Theta(\log\log n)$ queries, which is practically constant in most applications.

We are aware of only one other theoretical work that addresses the role of adaptivity in source identification [30], however, they only consider the case when the propagation delays are deterministic. In this case, if the first infection time is also known (model S2), problem (i) is trivial, problem (ii) is equivalent to finding a *resolving set* in a graph (a set of nodes such that the distance to those nodes is enough to uniquely determine the identity of an unknown node) [58], and problem (iii) is equivalent to the *metric dimension* problem [45]. If the time when the infection starts is unknown (model S1), the corresponding combinatorial notion is the *double metric dimension* [7]. In the past few years, there has been a line of work on the metric and the double metric dimensions in the context of source identification of both of simulation-based [47,49] and of rigorous nature [50]. In fact, the adaptive setting in the context of source identification was also first proposed with the deterministic propagation delay assumption [59]. The adaptive (sequential) version of the metric dimension also exists in the combinatorial literature [41], and the result of Odor and Thiran [30] says that in Erdős-Rényi graphs, the difference between the adaptive and non-adaptive settings is only a constant factor, which suggests that adaptivity plays little role in this setting. On the other hand, Kim et al. [22] finds that the sequential metric dimension is $O(\Delta\log(n))$ on graphs with maximum degree Δ (Theorem 1.2), which is relatively low compared to the metric dimension of most random tree

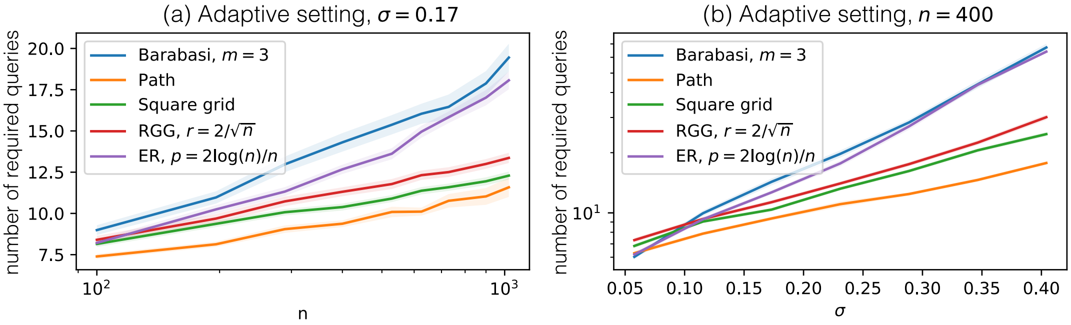

Fig. 1. The number of queries required by the adaptive source identification algorithm called Max-Gain [48] in the S1 source identification model as a function of (a) the network size n, and (b) the standard deviation of the edge-delays σ in the following network models: Barabási-Albert network [2] with average degree $2m$, path graph, square grid, random geometric graphs [35] in the unit square with connection radius r, Erdős-Rényi graphs [11] with connection probability p. The network parameters, controlling the number of edges in the random networks, were chosen slightly above the connectivity threshold, so that all networks in the simulation were connected. Each datapoint is an average of 192 simulations and the confidence intervals are computed using the Student's t distribution-test. See Appendix C for more details about how the Max-Gain was run to produce these simulation results.

distributions, which tends to be $\Theta(n)$ [28,23], suggesting a large role of adaptivity in these settings. Unfortunately, neither Erdős-Rényi graphs, nor tree graphs are good models of real networks, which motivates the analysis of further network models. A recent work by Lichev, Mitsche and Prałat [27] showed that the metric dimension of random geometric graphs of n nodes in the unit square with connectivity range r is $\Omega(\max(1/r^2, n^{2/3}r^{4/3}/\log^{1/3}(n)))$ for $1/\sqrt{n} \ll r \le 1/4$ (Theorem 5.2), which together with the upper bound on the sequential metric dimension by Kim et al. [22], and with the result that the maximum degree of random geometric graphs is $O(nr^2)$ with high probability [35] suggests a large (polynomial in n) role of adaptivity. Simpler geometric graph models without randomness, such as the path graph and the grid graph do not exhibit such a big difference between the adaptive and non-adaptive settings (the metric dimension of the path and the grid is 1 and 2, respectively). It is an interesting question to investigate whether adaptivity plays an important role in these simple geometric graph models if randomness is introduced into the diffusion process (as in the S1 or S2 models) and not in the graph model, especially since in most applications of the source identification problem, the diffusion process is in fact stochastic. Fig. 1(b) suggests that in the adaptive setting, the number of queries required to find the source depend on the stochasticity of the diffusion, as one would expect; the larger the stochasticity, the more queries are needed. Few works study in more detail the role of stochasticity in source identification, even empirically and in the non-adaptive setting alone (see e.g., [49]), and we are not aware of any previous work that has determined the exact dependence of the number of required queries on the randomness of the epidemic, neither in the adaptive nor in the non-adaptive setting.

In this paper, we compute the query complexity of the stochastic version of the source identification problem in the S2 model, on an n-node path, in both the adaptive and the non-adaptive settings. We chose the path network for its simplicity and for the insight that it may offer into the query complexity when the contact graph G has an underlying geometry (see Fig. 1). The choice of the S2 model is motivated by its close resemblance to the S1 model (see Appendix B), which is one of the standard models in the literature, but we mention that both the S1 and the S2 are abstract theoretical models, and they may have several currently unidentified applications. The propagation delays, which we call edge *weights*, are chosen to be i.i.d Gaussian variables with unit mean and variance σ^2, following the model proposed in [37]. One should note that the weights can take negative values, especially when σ is large, in contradiction with the non-negativity of propagation delays. Letting weights take negative values further accounts for the randomness in the incubation and reporting times. It makes source identification more challenging because of the absence of a deterministic, monotone dependence between the time of infection of a query node and its distance to the source. We also discuss how to extend our results to different propagation delay distributions, including ones that only take positive values, in Appendix A.

We find that for a wide range of σ, there is a drastic decrease in the number of required queries in the adaptive setting compared to the non-adaptive setting. For constant σ, which might be the most relevant range for practical purposes, the number of required queries is $\Theta(n)$ in the non-adaptive setting and $\Theta(\log\log n)$ in the adaptive setting. For the more precise dependence of our results on the standard deviation σ, we refer to Section 3.

1.1. Related work in information theory

The role of adaptivity is a central question in several fields in computer science, including property testing [5], information theory [8,25] and learning theory [42]. The most well-known example is perhaps binary search on a line, where being adaptive reduces the number of queries from n to $\log_2(n)$. Such a significant decrease in the query complexity of standard binary search is possible because the queries are very constrained; we can only ask whether the target is to the left or to the right of the queried vertex. If instead we are allowed to query any subset for containment of the target without any noise, then there is no difference between the adaptive and the non-adaptive query complexities (this is the well-known BarKochba or 20 Questions Game between two players, where the first player comes up with an item that the other player must identify by asking (in principle up to 20) yes-no questions). Indeed, $\log_2(n)$ questions are necessary because every answer carries only binary information, and the target can be found by $\log_2(n)$ non-adaptive questions by querying each digit of the binary representation of the index of the target vertex. One way to reintroduce a difference between the adaptive and non-adaptive cases in the 20 Questions Game is to corrupt the answers by a query dependent noise, which was proposed initially by Rényi [39], and has been studied by several follow-up works, including [8,25,61].

The problem setup of [25] has a close resemblance to our setup. In both cases, the search is done on a line, and the answers are corrupted by Gaussian noise, the variance of which depends on how close the query was to finding the target. The notable differences between [25] and our setup are that:

(i) we have more restrictive queries (one query in our setup is a single node (hence there are n possible queries), whereas one query in the setup of [25] is a subset of the nodes (hence there are 2^n possible queries))
(ii) we receive more information (we receive a noisy version of the distance between the queried vertex and the source, whereas in [25] they receive a noisy binary answer for belonging to the query set)
(iii) in our case the noise that corrupts the answers is not independent between queries.

Because of these differences, our proof techniques and our results are also different from [25]. The main tool in [25] for the adaptive upper bound is the *posterior matching scheme*. Roughly speaking, posterior matching produces queries that split the line into two approximately equal-weight subsets weighed by the posterior. In particular, there is no restriction on the queries produced by the posterior matching scheme, and therefore it is not applicable in our case (see (i) above). We also note, that as opposed to our setup, in [25], the geometry of the search space does not play an important role; any subset of the vertices of the line can be a query and the answers are insensitive to the distances between the queried vertices and the target vertex. For this reason, the usual geometry-insensitive information-theoretic notions (such as the entropy of the posterior) that work well in [25], cannot be used in our setup (see Section 6.2.2 for our notion of "progress"). In terms of results, for constant σ, both the non-adaptive and adaptive query complexities are found to be $\Theta(\log(n))$ in [25], which is in sharp contrast with our finding of $\Theta(n)$ and $\Theta(\log\log n)$ in the non-adaptive and adaptive settings, respectively. Finally, we note that the paper [25] features results about the expected query complexity of the search algorithms, whereas we give query complexity bounds that hold with any constant failure (or success) probability.

1.2. Related work in theoretical computer science

Extensions of binary search to graphs have been proposed on numerous occasions [10,16,20,31]. Of these, perhaps [10] has the closest connections with source identification with time queries. In this extension, a target vertex at an unknown position in a general graph is to be identified by adaptively querying vertices. A queried vertex can only respond whether it is the target or not, and if not, it indicates the edge on a shortest path between itself and the target. In the noiseless setting, queries always report the correct answer, whereas in the noisy setting, queries report a correct answer independently with probability $1/2 < p < 1$. In a sense, noisy binary search is an adaptive version of the source identification model proposed by [37] where we would keep the "who infected me" information and drop the time information instead, with the notable difference that in noisy binary search the noise that corrupts the answers is independent between queries. Since the information that a queried node can provide is its distance and/or its direction towards the source, adaptive source identification and noisy binary search on a line can be seen as "duals" of each other, in the sense that the former collects a noisy estimate of the distance whereas the latter collects a noisy estimate of the direction to the source. In the latter case, the adaptive query complexity is found to be $\Theta(\log n)$ for constant p in [10,25]. Comparing this result with our result of $\Theta(\log\log n)$ for the number of required queries in the adaptive case indicates that the distance to the target is far more informative than the direction, at least on the path graph. On different graphs, notably on star graphs, the distance is expected to be less informative than the direction. We limit the study of stochastic source identification in this paper to the path topology because of the complexity of the computations, and we leave the study in other graph topologies for further work.

2. The model

A known graph $G = (V, E)$ is fixed in advance. First, a node $v^* \in V$ is picked uniformly at random, which is called the *source*. Then for each edge $e \in E$, a weight $w(e)$ is drawn independently from some distribution $\mathcal{W}$. Both v^* and the weights $w(e)$ are hidden from the identification algorithm. Once they are drawn, the algorithm will start making queries to identify v^*. To perform a query, the algorithm chooses a query node $q \in V$, and receives an *answer* with value $\mathrm{ans}_w(v^*, q)$: the shortest distance between v^* and q in graph G with edges weighted by w (Fig. 2).

We distinguish between two settings. In the *non-adaptive* setting, the algorithm has to submit all of its queries in one batch, then receives all answers, and has to make a prediction. In the *adaptive* setting, the algorithm can make queries one by one, and adapt the choice of the next query based on previous answers. In both settings, the weights $w(e)$ are only drawn once, at the very beginning, and will not change between queries. As we will see, the difference between these two settings will have a huge impact on the number of queries that the algorithm needs, because in the adaptive setting the algorithm will be able to quickly zero in on the source v^* and receive progressively more refined information.

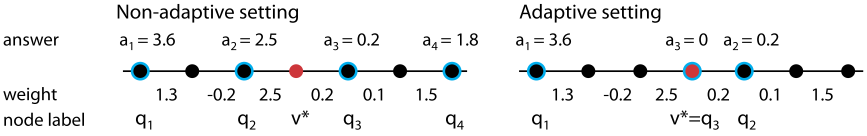

Fig. 2. An illustration of the model. Query nodes are marked blue and the source is marked red. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

In this paper, we treat the case where G is an n-node path, with nodes numbered from 1 to n (so $V = \{1, \dots, n\}$). We will assume $n \geq 3$ for convenience. We will often say that a node u is to the "left" (respectively, "right") of a node v if $u < v$ (resp., $u > v$). For the weight distribution, we choose $\mathcal{W} = \mathcal{N}(1, \sigma^2)$: a normal of mean 1 and variance $\sigma^2 > 0$, where σ is a parameter of the model. We choose a normal distribution because (i) by the Central Limit Theorem, the distances between faraway nodes converge to a normal distribution for most edge-delay distributions $\mathcal{W}$, while close-by nodes can be searched via exhaustive search (we discuss how to extend our results to these other distributions in Appendix A), (ii) there are several properties (e.g. additivity, tight concentration) of the normal distribution that simplify our calculations.

3. Results and discussion

3.1. Non-adaptive setting

We present matching upper and lower bounds for the non-adaptive setting.

Theorem 1. *For any failure probability $0 < \delta < 1/2$, there is a deterministic algorithm for non-adaptive source identification on the n-node path which asks $\min(O(1 + n\sigma^2 \log(1/\delta)), n)$ queries and identifies v^* correctly with probability at least $1 - \delta$, even if v^* is chosen adversarially instead of drawn uniformly at random.*

Theorem 2. *For any success probability $p > 1/n$, any (potentially randomized[1]) algorithm for non-adaptive source identification on the n-node path must ask $\Omega(1 + \min(p^3 n\sigma^2, p^2 n)) = \Omega_p(1 + \min(n\sigma^2, n))$ queries to identify v^* correctly with probability at least p when v^* is drawn uniformly at random.*

We can interpret the results in the following way. Intuitively, answers are "accurate" up to distance roughly $1/\sigma^2$: indeed, for a query node at distance d from the source, the mean of the received answer is d and the variance is $d\sigma^2$, so if $d = \omega(1/\sigma^2)$, the variance becomes $\omega(1)$ and it becomes impossible to deduce the real distance with constant probability. Therefore, instead of thinking of receiving stochastic answers, we can imagine that we receive the exact distance, but only if this distance is $\leq 1/\sigma^2$ (and otherwise they would give no answer at all). That is, we think of the queries as effective within a "limited range" $1/\sigma^2$. In that model, it is clear that $\Theta(1 + \min(n\sigma^2, n))$ queries are necessary and sufficient, which is exactly what we find in Theorems 1 and 2.

Our proofs also build on the intuition of query nodes with limited range. In the proof of Theorem 1, we show that if the query nodes are spaced equally (and deterministically), every v^* is in the "range" of the closest two query nodes, meaning that once rounded to the closest integer, they give the correct answer with high probability. This probability is based only on the randomness of the edge weights $w(\cdot)$, and is high no matter where v^* ends up, hence we can identify v^* even without assuming any prior on its distribution. This extra guarantee was not required by the model, but it comes naturally without any additional cost.

In contrast, in Theorem 2 we show that any algorithm that succeeds with constant probability must use $\Omega(1 + \min(n\sigma^2, n))$ queries even if the algorithm is allowed to take advantage of the assumption that v^* is uniformly distributed over the nodes V. The proof works by showing that if one uses fewer queries, then most of the nodes are so far away from the closest query node that they are indistinguishable from other close-by nodes.

While in this paper we only consider the path graph, these results indicate that limited range query nodes might be a good proxy for non-adaptive source identification in other graphs as well, which has not been thoroughly explored in the source identification literature.

3.2. Adaptive setting

The adaptive setting is more complex and more interesting than the non-adaptive case. For instance, it is not obvious anymore how the queries should be selected. We may consider an algorithm that, at each decision, selects a query node based on the posterior probabilities that the source is at some node: we call *posterior* at a node v the probability that the

[1] Since the distribution for the identity of the source is fixed, rather than adversarial, randomness in the algorithm is not useful (as long as we are not considering running time): the algorithm should simply choose the set of queries that maximizes the probability of finding the source, and output the likeliest source given the answers it receives.

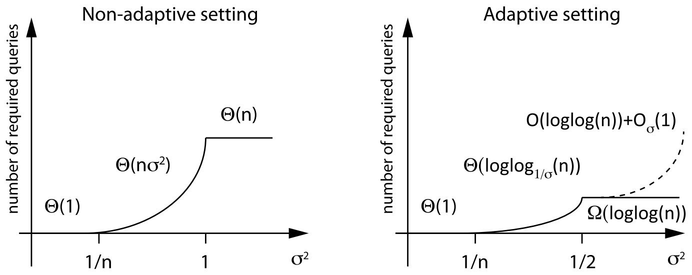

Fig. 3. A sketch of a linear-log plot of the optimal number of queries in the non-adaptive and adaptive cases as a function of σ.

source is v, conditioned on the answers made so far. However, those posteriors might be hard to compute, and might not be well-behaved as a function of v (for example, they might not be unimodal). Fortunately, as long as the variance of the edge-delays is relatively low, the answers that we see are concentrated around their expected value and we can form a fairly good idea about what the posteriors might look like. This inspires the following algorithm, which (for intuition) can be seen as a procedure that computes at each step the posteriors approximately, and selects the next query node close to the node with the highest posterior (the node that is most likely to be the source).

Theorem 3. *For any failure probability $0 < \delta < 1/2$, there is a deterministic algorithm for adaptive source identification on the n-node path which uses*

$$\begin{cases} O\left(\log(1+\log_{1/\sigma} n) + \operatorname{polylog}(1/\delta)\right) \text{ queries if } \sigma^2 \le 1/2 \\ O\left(\log\log n + \sigma^2 \cdot \operatorname{polylog}(\sigma, 1/\delta)\right) \text{ queries if } \sigma^2 \ge 1/2 \end{cases}$$

and identifies v^* correctly with probability at least $1-\delta$, even if v^* is chosen adversarially instead of drawn uniformly at random.

To show optimality, as we did in the non-adaptive case, we show that no algorithm can succeed without asking a large number of queries, even under the assumption that v^* is uniformly distributed over V.

Theorem 4. *For any success probability $p > 1/n$ and any $n \ge \Theta_p(\max(\sigma^3, 1))$, any (potentially randomized) algorithm for adaptive source identification on the n-node path must use*

$$\begin{cases} \Omega_p(1 + \log(1+\log_{1/\sigma} n)) \text{ queries if } \sigma^2 \le 1/2 \\ \Omega_p(\log\log n) \text{ queries if } \sigma^2 \ge 1/2 \end{cases}$$

to identify v^* correctly with probability at least p when v^* is drawn uniformly at random.

The proof of Theorem 4 is the most challenging proof we present. Since the algorithm is allowed to adapt each query based on the results of the previous ones, we have to somehow quantify the progress it has made towards identifying v^*. However, this is made delicate by the fact that the edge weights $w(e)$ are drawn only once at the onset, which means that the algorithm does not only accumulate information about v^*, but also about the edge weights $w(e)$. In particular, proof approaches that try to “fool” the algorithm by giving it answers from a modified distribution tend to fail because the algorithm can test for consistency across queries. Instead, the proof that we present builds on a detailed understanding what the posteriors can look like in each step.

The upper and lower bounds in Theorems 3 and 4 match for σ up to $\tilde{\Theta}(\log\log n)$ (Fig. 3). For $\sigma \gg \log\log(n)$, our upper and lower bounds are separated by a $\sigma^2 \cdot \operatorname{polylog}(\sigma)$ term, which comes from the final steps, when the algorithm has gotten so close to the source that the variance of the edge-delays is about as big as the expected value of the answers, at which point the algorithm simply queries every node.

This $\sigma \gg \log\log(n)$ regime is a difficult regime to analyse, because for larger σ we lose the concentration of the answers, and cannot control the shape of the posteriors anymore. Moreover, we believe that in the high-σ regime, any asymptotically tight results for the Gaussian case would not carry over to other edge-delay distributions $\mathcal{W}$, because the $O_{\sigma,\delta}(1)$ term becomes sensitive to the specific $\mathcal{W}$ we pick. As an example, consider $\mathcal{W}$ to be a Gaussian distribution with a very large σ (say, larger than n), truncated at 0 (to prevent negative edge weights w). With this $\mathcal{W}$, we can always figure out which direction the source is from node v by simply querying a neighbor of v, and checking which answer is larger. Hence we can find the source with binary search in $\log_2(n)$ rounds, however, if $\mathcal{W}$ is a non-truncated Gaussian random variable and σ is large enough, we clearly have to query every node to find the source.

Finally, as an additional strength of our results, we note that the lower bounds (Theorems 2 and 4) continue to apply even if the algorithm were to also receive direction information (i.e. whether the source is on the left or on the right of the query node), whereas the upper bounds (Theorems 1 and 3) work well even without using this information.

4. Preliminaries

We denote a normal distribution with mean μ and variance σ^2 as $\mathcal{N}(\mu,\sigma^2)$. We occasionally call variables distributed according to this a normal distribution "Gaussians". We will often use the following basic facts about the normal distribution.

Fact 1. If $X \sim \mathcal{N}(\mu_1,\sigma_1^2), Y \sim \mathcal{N}(\mu_2,\sigma_2^2)$ are independent Gaussians, then $X+Y \sim \mathcal{N}(\mu_1+\mu_2,\sigma_1^2+\sigma_2^2)$.

Fact 2. If $X \sim \mathcal{N}(\mu,\sigma^2)$, then $\Pr[X \notin \mu \pm a] \le e^{-\frac{a^2}{2\sigma^2}}$.

Proof. First, using the probability density function of the normal distribution and the change of variables $z = \frac{x-\mu}{\sigma}$ we have

$$\Pr[X \notin \mu \pm a] = 2 \times \int_{\mu+a}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx = \sqrt{\frac{2}{\pi}} \int_{a/\sigma}^{\infty} e^{-z^2/2} dz.$$

It only remains is to prove that for all $b \ge 0$, $\sqrt{\frac{2}{\pi}} \int_b^\infty e^{-z^2/2} dz \le e^{-b^2/2}$.

We separate into two cases.

- If $b \ge 1$, then we have

$$\begin{aligned} \sqrt{\frac{2}{\pi}} \int_b^\infty e^{-z^2/2} dz &\le \sqrt{\frac{2}{\pi}} \int_b^\infty \frac{z}{b} e^{-z^2/2} dz && (z \ge b) \\ &= \sqrt{\frac{2}{\pi}} \frac{1}{b} e^{-b^2/2} && \left(\tfrac{d}{dz} e^{-z^2/2} = -z e^{z^2/2}\right) \\ &\le e^{-b^2/2}. && \left(b \ge 1 \ge \sqrt{\tfrac{2}{\pi}}\right) \end{aligned}$$

- One can easily check that on interval [0, 1], $\sqrt{\frac{2}{\pi}} \int_b^\infty e^{-z^2/2} dz$ is convex while $\sqrt{\frac{2}{\pi}} \int_b^\infty e^{-z^2/2} dz$ is concave (by computing their second derivatives), and in addition we have

$$\sqrt{\frac{2}{\pi}} \int_0^\infty e^{-z^2/2} dz = 1 = e^{-0^2/2} \quad \text{and} \quad \sqrt{\frac{2}{\pi}} \int_1^\infty e^{-z^2/2} dz < 0.318 < 0.606 < e^{-1^2/2}.$$

  Therefore, $\sqrt{\frac{2}{\pi}} \int_b^\infty e^{-z^2/2} dz \le e^{-b^2/2}$ on the whole interval. □

5. Proofs for the non-adaptive setting

5.1. Upper bound

Proof of Theorem 1. First of all, observe that one can always find the source with probability 1 if one is willing to use n queries: just query each node. The query at node v^* will produce answer 0, while the other queries will almost surely produce nonzero answers.

Now it suffices to show that the source can be identified with $O(1 + n\sigma^2 \log(1/\delta))$ queries. The strategy is the one that is naturally expected: query $\sim n/d$ nodes along the path at fixed intervals of some length d, where d is small enough to ensure that the query nodes nearest to the source v^* return an answer that is very close to the real distance (and in particular, that will be exactly equal to it once rounded to the nearest integer).

What makes things a bit more complex is that:

(a) even if all the weights are positive, it may not be easy to determine between which two query nodes v^* is identified;
(b) since the weights may be negative, it is possible that a query node q_1 gives a smaller answer than a query node q_2 even though q_2 lies between q_1 and v^* (in particular, the answers do not necessarily form a unimodal sequence when read from left to right).

Concretely, the algorithm will query nodes $1, d+1, 2d+1, \dots, 1 + \lfloor \frac{n-1}{d} \rfloor d$. It will then find the query node with the smallest answer, which we call q_{smallest}, and the next query node to its left $q_{\text{left}} := q_{\text{smallest}} - d$ (let's assume for now that $q_{\text{smallest}} \ne 1$, so that q_{left} exists). Let $a_{\text{smallest}} := \text{ans}_w(v^*, q_{\text{smallest}})$ and $a_{\text{left}} := \text{ans}_w(v^*, q_{\text{left}})$ be the corresponding answers.

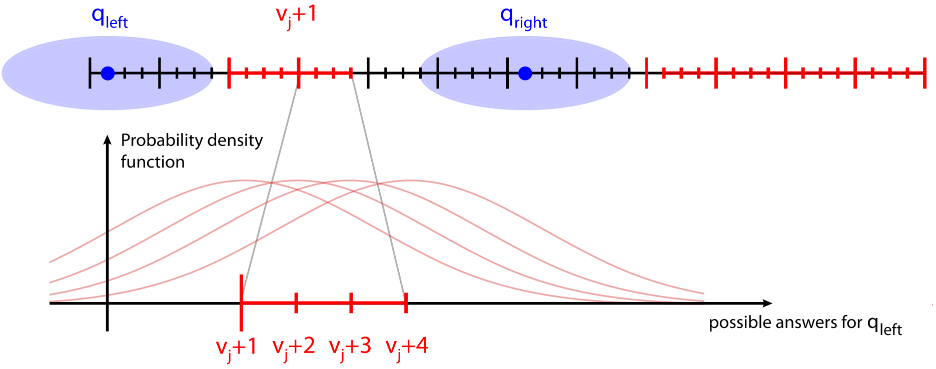

Fig. 4. An illustration for the proof of Theorem 2 with $d = 6$ and $k = 4$. At the top of the figure, the graph G is shown with the set Q (the query nodes) and the set $C(d)$ (the "covered" nodes) marked blue, and the set $K(d,k)$ marked red. At the bottom of the figure, the probability density functions of the answers recorded by q_{left} are shown for each candidate source in the highlighted segment. Intuitively, the union of the areas under the red curves corresponds to the probability of success of the optimal source identification algorithm (this is made concrete in Equation (2), although we need to also consider the answer from q_{right}, so instead of a single integral we get a double integral). In the proof, we show that if too few nodes are queried, then the red segments will be far from the closest query node, and therefore the red curves will have a large overlap, and their union will be small.

Then the algorithm just assumes that both of them are correct (equal to the real distance) once rounded to the nearest integer (that is, $\lfloor a_{\text{smallest}} \rceil = |v^* - q_{\text{smallest}}|$ and $\lfloor a_{\text{left}} \rceil = |v^* - q_{\text{left}}|$), and computes v^* as[2]

$$\begin{cases} q_{\text{smallest}} + \lfloor a_{\text{smallest}} \rceil \text{ if } \lfloor a_{\text{left}} \rceil \geq d \\ q_{\text{smallest}} - \lfloor a_{\text{smallest}} \rceil \text{ otherwise.} \end{cases}$$

For this strategy to work, it is enough if the following statements hold simultaneously:

(a) among the query nodes located at or to the left of v^*, the closest one is the one with the smallest answer;
(b) among the query nodes located at or to the right of v^*, the closest one is the one with the smallest answer;
(c) the two closest query nodes to v^* on its left side and the closest query node on its right side all give a correct answer once rounded to the nearest integer.

Indeed, if this is true, then q_{smallest} will be the closest query node to v^* on either its left or right side, and thus both q_{smallest} and q_{left} will be among the three query nodes that are guaranteed by point (c) to give the correct result once rounded.

The following claim, which is purely technical and easily obtained from concentration bounds, is proved in Appendix D.

Claim 1. *For some $d = \Omega\left(\frac{1}{\sigma^2 \log(1/\delta)}\right)$, all of (a), (b), (c) hold simultaneously with probability $\geq 1 - \delta$.*

This means that the number of queries used is $\left\lfloor \frac{n-1}{d} \right\rfloor + 1 = O(1 + n/d) = O(1 + n\sigma^2 \log(1/\delta))$. □

5.2. Lower bound

Proof of Theorem 2. Since $p > 1/n$, it is clear that at least one query is necessary (otherwise one could not do better than randomly guessing the source, which gives $p = 1/n$). In the rest of the proof, we show that one needs $\Omega(\min(p^3 n\sigma^2, p^2 n))$ queries to identify the source.

Let us introduce some notation. Let Q be the set of query nodes that the algorithm chooses, let $\text{ans}_w(v^*, Q) := \{\text{ans}_w(v^*, q)\}_{q \in Q}$ be the answers it receives from each query node, and let f be the function that takes in these answers and returns a prediction for v^*. Since we are not bounding the running time of the algorithm, we can assume that both Q and f are deterministic (the algorithm can simply choose the values of Q and f that give the best chance of finding the source), while the answers $\text{ans}_w(v^*, q)$ are random variables depending on both v^* and w (recall that v^* is drawn uniformly in $V = [n]$). The overall success probability of the algorithm is given by $p = \Pr_{v^*, w}[f(\text{ans}_w(v^*, Q)) = v^*]$. For a fixed node v, let $p(v) := \Pr_w[f(\text{ans}_w(v, Q)) = v]$ be the probability that the algorithm will output v conditioned on $v^* = v$. Clearly, $p = \frac{1}{n}\sum_{v \in V} p(v)$.

As discussed before in Section 3, our proofs in the non-adaptive case build on the intuition of query nodes with limited range: roughly speaking, we will show that most nodes are further than a distance $1/\sigma^2$ away from the closest query node, and therefore they will be hard to distinguish from their neighbors. Fig. 4 sketches some of the key points used in the proof.

[2] If q_{left} does not exist, which happens only when $q_{\text{smallest}} = 1$, then the algorithm can simply compute v^* as $1 + \lfloor a_{\text{smallest}} \rceil$, again assuming that a_{smallest} is correct once rounded to the nearest integer.

Let $d > 0$ be an integer which we will fix later in equation (6), representing the "range" of the query nodes. Intuitively, nodes outside the range d of any query node (the "uncovered" nodes) might be hard to distinguish, contrary to nodes within the range of a query node (the "covered" nodes). Let $C(d) \subset V$ be the set of nodes that are within distance d of some query node in Q.

In addition, let us subdivide the first $k\lfloor n/k \rfloor$ nodes of V into $\lfloor n/k \rfloor$ segments of length k, where $k > 0$ is an integer that we will fix later, and let $K(d,k) \subseteq V \setminus C(d)$ be the set of nodes contained in the segments that are entirely included in $V \setminus C(d)$ (we will call such segments "uncovered"). Our goal in defining these segments is to show that there are few "covered" segments, and that the source identification problem is hard to solve on "uncovered" segments.

More precisely, to demonstrate that Q needs to be large, we will split the probability of success $p = \frac{1}{n}\sum_{v \in V} p(v)$ into two parts:

- the part due to $v \in V \setminus K(d,k)$ (the "covered" segments), which will be small whenever Q, d and k are small (simply because the set $V \setminus K(d,k)$ will be small);
- the part due to $v \in K(d,k)$ (the "uncovered" segments), which will be small ($\leq p/2$) whenever d and k are large enough.

Concretely,

$$\begin{aligned}
pn &= \sum_{v \in V} p(v) \\
&= \sum_{v \in V \setminus K(d,k)} p(v) + \sum_{v \in K(d,k)} p(v) \\
&\leq |V \setminus K(d,k)| + \sum_{v \in K(d,k)} p(v). && (p(v) \text{ is a probability, so } p(v) \leq 1) \\
&\leq (2d + 2k - 1)|Q| + (k-1) + \sum_{v \in K(d,k)} p(v) && (1)
\end{aligned}$$

where the factor $(2d + 2k - 1)$ is because each query node covers $\leq 2d + 1$ nodes directly, and can affect $\leq 2(k-1)$ more nodes by touching their segment; also, the $+(k-1)$ comes from the $< k$ nodes that were not within the first $k\lfloor n/k \rfloor$ nodes and therefore are not in a segment.

Let us now prove that the sum $\sum_{v \in K(d,k)} p(v)$ is small when d and k are large. This makes intuitive sense: the nodes in $K(d,k)$ are far from the closest query node, so they will be hard to distinguish from each other. To do this, we will use the following lemma.

Claim 2. *Let $\{v+1, \dots v+k\}$ be a set of k adjacent nodes. Let $q_{left} \in Q$ be the closest query node at or to the left of $v+1$, and let $q_{right} \in Q$ be the closest query node at or to the right of $v+k$. Assume that there are no query nodes between q_{left} and q_{right}, and that $q_{left} \leq v - d$ and $q_{right} \geq v + k + d$. Then*

$$\sum_{i=1}^{k} p(v+i) < \frac{2(d+k)e^{\frac{k^2}{2(d+k)\sigma^2}}}{d}.$$

Proof. Let us consider a scenario where the source is sampled uniformly from $\{v+1, \dots, v+k\}$, and let $f' := \arg\max_f \sum_{i=1}^{k} \Pr_w[f(\text{ans}_w(v+i, Q)) = v+i]$ be the algorithm that maximizes the success probability in this scenario. Let $p'(v+i) := \Pr_w[f'(\text{ans}_w(v+i, Q)) = v+i]$, then clearly $\sum_{i=1}^{k} p(v+i) \leq \sum_{i=1}^{k} p'(v+i)$ by definition of f'. Also observe that f' will only depend on the answers at q_{left} and q_{right}, since the algorithm already knows that $v^* \in [q_{\text{left}}, q_{\text{right}}]$, and the other query nodes outside $[q_{\text{left}}, q_{\text{right}}]$ do not carry relevant information. Indeed, any answer outside of q_{left} and q_{right} is just the sum of the answer at q_{left} or q_{right} plus some extra term that does not say anything about the identity of v^*. More formally, one could recreate the other answers from just the answers at q_{left} and q_{right} in a way that exactly replicates the original distribution, so we can transform any algorithm that uses all the answers into an algorithm that uses only the answers at q_{left} and q_{right} with the same performance.

For similar reasons, we can assume that $q_{\text{left}} = v - d$ and $q_{\text{right}} = v + k + d$. If the query nodes were any further, that would be more difficult for the algorithm f', because we can simulate that case using the answers at $v - d$ and $v + k + d$.

Since f' maximizes the success probability of estimating the hidden parameter v^* with both the likelihood function and the prior distribution being completely known, the optimal f' computes the posterior distribution using Bayes rule, and picks the v^* that maximizes it (this is called Maximum A Posteriori or MAP estimation) [37]. In our case we have a uniform prior, which implies that f' is simply the Maximum Likelihood Estimator, i.e., for any answer $(a_{\text{left}}, a_{\text{right}})$,

$$f'(a_{\text{left}}, a_{\text{right}}) = \arg\max_{i\in\{1,\dots,k\}}(g_{v+i}(a_{\text{left}}, a_{\text{right}})),$$

where $g_{v+i}(x, y)$ denotes the probability density function of $\mathcal{W}_{v+i} = (\mathcal{N}(d+i, (d+i)\sigma^2), \mathcal{N}(d+k-i, (d+k-i)\sigma^2)$, the distribution of the answers at q_{left} and q_{right} (note that $\mathcal{W}_{v+i}$ is a pair of *independent* normal distributions). Consequently,

$$\sum_{i=1}^{k} p'(v+i) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \max_{i\in\{1,\dots,k\}}(g_{v+i}(x, y))\mathrm{d}x\mathrm{d}y. \tag{2}$$

Next, we provide the following upper bound to g_{v+i} for every $i \in \{1, \dots, k\}$:

$$\begin{aligned} g_{v+i}(x, y) &= \frac{1}{2\pi\sigma^2\sqrt{(d+i)(d+k-i)}}\exp\left(-\frac{(x-(d+i))^2}{2(d+i)\sigma^2} - \frac{(y-(d+k-i))^2}{2(d+k-i)\sigma^2}\right) \\ &< \frac{1}{2\pi d\sigma^2}\exp\left(-\frac{(x-(d+i))^2 + (y-(d+k-i))^2}{2(d+k)\sigma^2}\right) \end{aligned} \tag{3}$$

Notice that if we consider a triangle ABC with $A = (x, y)$, $B = (d+i, d+k-i)$ and $C = (d, d)$, and we denote the side lengths opposite of each point by a, b and c, then the numerator of the exponent in equation (3) equals c^2. The following lower bound holds for c^2 based on the law of cosines and elementary algebra:

$$c^2 = a^2 - 2ab\cos(\measuredangle ACB) + b^2 \ge a^2 - 2ab + b^2 \ge \frac{a^2}{2} - b^2.$$

The last inequality can be confirmed if we move all terms to the left side and find the expression $(a/\sqrt{2} - \sqrt{2}b)^2 \ge 0$. After substituting back into a, b and c, since the maximum distance between points $(d+i, d+k-i)$ and (d, d) is k for any $i \in \{1, \dots, k\}$, we get

$$(x-(d+i))^2 + (y-(d+k-i))^2 \ge \frac{1}{2}((x-d)^2 + (y-d)^2) - k^2. \tag{4}$$

Substituting equation (4) back into equation (3) yields

$$\begin{aligned} g_{v+i}(x, y) &< \frac{1}{2\pi d\sigma^2}\exp\left(-\frac{(x-d)^2 + (y-d)^2 - 2k^2}{4(d+k)\sigma^2}\right) \\ &= \frac{2(d+k)e^{\frac{k^2}{(d+k)2\sigma^2}}}{d} \cdot \frac{1}{2\pi(d+k)2\sigma^2}\exp\left(-\frac{(x-d)^2 + (y-d)^2}{2(d+k)2\sigma^2}\right). \end{aligned} \tag{5}$$

Notice that the last line of equation (5) can be written as

$$\frac{2(d+k)e^{\frac{k^2}{2(d+k)\sigma^2}}}{d} g(x, y),$$

where $g(x, y)$ is the probability density function of two independent copies of $\mathcal{N}(d, (d+k)2\sigma^2)$, so its double integral must sum to 1. Thus, plugging this upper bound into equation (2), we get

$$\sum_{i=1}^{k} p'(v+i) < \frac{2(d+k)e^{\frac{k^2}{2(d+k)\sigma^2}}}{d}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x, y)\mathrm{d}x\mathrm{d}y = \frac{2(d+k)e^{\frac{k^2}{2(d+k)\sigma^2}}}{d}.$$

Since $\sum_{i=1}^{k} p'(v+i)$ is an upper bound on $\sum_{i=1}^{k} p(v+i)$, the proof is completed. □

Since each segment in $K(d, k)$ contains k consecutive nodes that are all a distance d away from the closest query node, we can apply Claim 2 to each of them, and get

$$\begin{aligned} \sum_{v\in K(d,k)} p(v) &\le \frac{|K(d,k)|}{k} \cdot \frac{2(d+k)e^{\frac{k^2}{2(d+k)\sigma^2}}}{d} \\ &\le \frac{n}{k} \cdot \frac{2(d+k)e^{\frac{k^2}{2(d+k)\sigma^2}}}{d}. \end{aligned}$$

In order to make this $\le pn/2$, let us set

$$k := \lceil 16e/p \rceil \qquad \text{and} \qquad d := \max\left(\left\lceil \frac{k^2}{2\sigma^2 \ln(pk/8)} \right\rceil, k\right) \tag{6}$$

(this value of k is chosen so that the $\ln(\cdot)$ in the definition of d is positive[3]). Indeed, these choices give

$$\begin{aligned}
\sum_{v \in K(d,k)} p(v) &\le \frac{n}{k} \cdot \frac{2(d+k)e^{\frac{k^2}{2(d+k)\sigma^2}}}{d} \\
&\le \frac{n}{k} \cdot \frac{4d}{d} e^{\frac{k^2}{2d\sigma^2}} && (k \le d \text{ by } (6)) \\
&\le \frac{n}{k} \cdot 4e^{\ln(pk/8)} && \left(d \ge \tfrac{k^2}{2\sigma^2 \ln(pk/8)} \text{ by } (6)\right) \\
&= pn/2. && (7)
\end{aligned}$$

Combining (1) with (7), we finally get

$$pn \le (2d + 2k - 1)|Q| + (k-1) + pn/2$$

which further implies

$$\begin{aligned}
|Q| &\ge \frac{pn/2 - k}{2d + 2k} \\
&\ge \frac{pn - 2k}{8d}. && (k \le d)
\end{aligned}$$

Let us assume $k \le pn/4$ (otherwise we have $p^2 n < 4pk \le 4p \left\lceil \frac{16e}{p} \right\rceil = O(1)$, which means the $\Omega(\min(p^3 n\sigma^2, p^2 n))$ bound we are trying to prove becomes a trivial $\Omega(1)$). Then we get

$$\begin{aligned}
|Q| &\ge \frac{pn - 2k}{8d} \\
&\ge \frac{pn}{16d} && (k \le pn/4) \\
&\overset{(6)}{=} \frac{pn}{16 \max\left(\left\lceil \frac{k^2}{2\sigma^2 \ln(pk/8)} \right\rceil, k\right)} && (\text{by definition of } d) \\
&\overset{(6)}{=} \frac{pn}{O\left(\max\left(\frac{1}{p^2\sigma^2}, \frac{1}{p}\right)\right)} && (\text{by definition of } k) \\
&= \Omega(\min(p^3 n\sigma^2, p^2 n)). \quad \square
\end{aligned}$$

6. Proofs for the adaptive setting

6.1. Upper bound

Proof of Theorem 3. The algorithm crucially uses the following result on the concentration of the answers at large distances. We prove it in Appendix E.

Lemma 1. *For any probability $0 < \delta < 1/2$, there is some constant $C(\delta) = O\left(\sqrt{\log(1/\delta)}\right)$ such that for any n, σ and any source $v^* \in V$, we have*

$$\Pr_w[\forall q \in V,\ \text{ans}_w(v^*, q) \in |v^* - q| \pm C(\delta) \cdot \sigma\sqrt{|v^* - q|} \ln(1 + |v^* - q|)] \ge 1 - \delta. \tag{8}$$

[3] This value of k also makes sense intuitively: we are showing that no algorithm can solve the source identification problem with probability better than $p/2$ in an uncovered interval, so we definitely need at least $k \ge 2/p$ to rule out random guessing.

That is, the concentration bound $|v^* - q| \pm C(\delta) \cdot \sigma\sqrt{|v^* - q|}\ln(1 + |v^* - q|)$ holds simultaneously for all nodes q with probability at least $1 - \delta$ over the choice of the weights w.

With this concentration result in hand, the algorithm follows a natural “iterative refining” strategy: start by obtaining a rough estimate of the identity of v^*, then progressively refine it by querying nodes closer and closer to v^*. After k steps (where k is defined by Claim 3), only a few possible candidate sources will remain, and the algorithm will switch to testing them one by one.

Concretely, let us assume that Lemma 1 holds with the desired probability of failure δ. Then the algorithm will maintain a shrinking interval $[l_i, r_i]$ which contains v^*. Initially $l_0 = 1$ and $r_0 = n$. At each step, the algorithm will query node l_i. Let d_i be equal to $r_i - l_i$. v^* has to be to the right of l_i and at distance at most d_i from l_i, so

$$\begin{aligned}\text{ans}_w(v^*, l_i) &\in |v^* - l_i| \pm C(\delta) \cdot \sigma\sqrt{|v^* - l_i|}\ln(1 + |v^* - l_i|), && \text{(by (8))}\\ &\subseteq v^* - l_i \pm C(\delta) \cdot \sigma\sqrt{d_i}\ln(1 + d_i) && (v^* \geq l_i \text{ and } |v^* - l_i| \leq d_i)\end{aligned}$$

and thus given answer $\text{ans}_w(v^*, l_i)$ the algorithm knows that v^* must be in interval

$$l_i + \text{ans}_w(v^*, l_i) \pm C(\delta) \cdot \sigma\sqrt{d_i}\ln(1 + d_i). \tag{9}$$

Therefore, it shrinks its interval as follows:

$$\begin{cases} l_{i+1} = \max(l_i, l_i + \lceil \text{ans}_w(v^*, l_i) - C(\delta) \cdot \sigma\sqrt{d_i}\ln(1 + d_i)\rceil) \\ r_{i+1} = \min(r_i, l_i + \lfloor \text{ans}_w(v^*, l_i) + C(\delta) \cdot \sigma\sqrt{d_i}\ln(1 + d_i)\rfloor) \end{cases}$$

The resulting interval has length $d_{i+1} = r_{i+1} - l_{i+1} \leq 2C(\delta) \cdot \sigma\sqrt{d_i}\ln(1 + d_i)$.

Now what remains to do is to figure out how fast this interval shrinks, and when we should switch to testing the remaining candidates one by one. To get a rough initial intuition of the speed at which it shrinks, let us imagine that $d_{i+1} = \sigma\sqrt{d_i}$. Then, the sequence would decrease very fast at the onset, when d_i is still large, then decrease slower and slower. We would observe that $\log(d_{i+1}/\sigma^2) = \log(\sqrt{d_i}/\sigma) = \frac{1}{2}\log(d_i/\sigma^2)$: the *logarithm* of the ratio of d_i to σ^2 is divided by 2 at each step. So it would be reasonable to assume that d_i will approach σ^2 in a doubly-logarithmic number of steps. This is made rigorous in the following claim, which is proved in Appendix F.

Claim 3. *Assume $d_0 \leq n$, and $d_{i+1} \leq C \cdot \sigma\sqrt{d_i}\ln(1 + d_i)$ for some value $C > 0$. Then*

- *if $\sigma^2 \leq 1/2$, there exists $k = O(\log\log_{1/\sigma} n)$ such that $d_k = \text{poly}(C)$;*
- *if $\sigma^2 \geq 1/2$, there exists $k = O(\log\log n)$ such that $d_k = \sigma^2 \cdot \text{poly}(C, \log(1 + \sigma^2))$.*

We instantiate Claim 3 with $C := 2C(\delta)$. After the first k steps, we simply go through the $d_k + 1$ remaining possible positions for node v^* in $[l_k, r_k]$, and check them all with one query each.[4] With probability 1, v^* will be the only one to give 0 as an answer. Thus, overall, this algorithm will succeed with probability at least $1 - \delta$. The total number of queries used is $k + d_k + 1$, which by Claim 3 gives the desired bounds in both the $\sigma^2 \leq 1/2$ case and the $\sigma^2 \geq 1/2$ case. □

6.2. Lower bound

For your reading convenience, here is a quick reference of the notations that are used throughout the proof.

[4] As we explain in Appendix A, this theorem can be extended to apply to many other edge weight distributions. If the distribution's support is positive, a binary search can be used instead.

Notations cheatsheet (not exhaustive)

- p: the desired probability of identifying the true source.
- R: the internal randomness of the algorithm (see Definition 1).
- q_j: the j^{th} node queried by the algorithm.
- a_j: the j^{th} answer the algorithm receives (it will take the value $\text{ans}_w(v^*, q_j)$).
- T: shorthand for $\text{Typical}_{p/2}(v^*, w)$, i.e. the event that the concentration bounds from Definition 3 hold.
- C: shorthand for $C(p/2)$, a constant (in n and σ) factor involved in the concentration bounds of Definition 3 (see Definition 12 for its precise value).
- D: shorthand for $D(\sigma, p/2)$, the minimum distance at which the concentration bounds of Definition 3 hold (see Definition 12 for its precise value).
- l_j, r_j: the step counters at which the closest query nodes to v^* have been placed so far at step j; i.e. by the time the j^{th} query has been asked, v^* lies between query nodes q_{l_j} and q_{r_j}.
- μ_j: the minimum of the answers to query nodes q_{l_j} and q_{r_j}. Intuitively, it is a proxy for the smallest answer made so far (after asking j queries).
- $\text{reduce}_{n,\sigma}(x)$: a function $\mathbb{R} \to \mathbb{R}$ that models the fastest decrease of μ_j an algorithm can hope for: most of the time $\mu_{j+1} \geq \text{reduce}_{n,\sigma}(\mu_j)$ (see Definition 6).
- λ_i: a lower bound on μ_j with high probability, and therefore a limit on the progress that the algorithm can make (see Definition 7).
- K_j: a random variable representing all the information that the algorithm has at its disposal after asking the first j queries (see Definition 8).
- A_j: the event that $\mu_j \geq \lambda_j$; intuitively, the event that at step j, the algorithm has not queried any nodes very close to v^*.
- B_j: informally, the event that even based on everything the algorithm knows at step j, no node is particularly likely to be the source (see Definition 10).
- j_{stop} (also, $j_{\min}$): a lower bound on the number of steps that an algorithm needs to find the source with probability p (see Definition 11).

Definition 1 *(R).* Let R be a random variable denoting the internal randomness of the algorithm. One can for example think of R as drawn uniformly from interval $[0, 1]$, as this puts no limitation on the amount of randomness the algorithm can use.

Definition 2 *(q_j, a_j).* Let q_j be the j^{th} query node selected by the algorithm, and let a_j be the answer that it gets to query q_j (i.e. $a_j := \text{ans}_w(v^*, q_j)$). Both q_j and a_j are random variables that can depend on v^*, w and the internal randomness of the algorithm.

Note that under the hypotheses of Theorem 4, any algorithm needs to query at least one node: otherwise it would succeed with probability at most $1/n < p$.

To simplify the proof, we make the following adaptations to the model, which only give the algorithm more power to identify the source, and therefore hold without loss of generality:

(a) Before the algorithm starts, two initial query nodes $q_{-1} = 1$ and $q_0 = n$ are already selected, resulting in answers a_{-1} and a_0 at no cost to the algorithm. The first query that is actually chosen by the algorithm is q_1.
(b) When querying node q_j, in addition to the answer a_j, the algorithm is told on which side of q_j the source v^* is identified.[5]
(c) Once the algorithm is ready to guess the position of v^*, it should query it.[6] If at any point the algorithm queries v^*, it immediately terminates and the identification is considered successful. More precisely, the number of queries that the algorithm uses is defined as the first positive integer j such that $q_j = v^*$.

The details of the proof are at times technically heavy, so we first give a general outline to provide the gist of the proof. It proceeds in the following 8 steps. We will cover each of them in detail in the next 8 subsections (Section 6.2.*x* corresponds to step *x*).

1. We define a random event T (over v^* and w) which has probability $\geq 1 - p/2$, guarantees that v^* is not too close to the ends of the path, and gives some concentration bounds for the answers when the query node is at least some

[5] Note that this gives the algorithm the ability to perform a binary search, which is not necessarily easy when the weight distribution is not positive.

[6] This extra query does not affect the asymptotics because as noted in the previous paragraph the algorithm always needs to query at least one node anyway.

distance D away from v^* (D will be defined in Definition 4). T represents a "typical situation": the role of this event is to exclude some extreme cases (e.g. $v^* = 1$ or $v^* = n$) that would derail the proof.

2. We define a sequence of random variables μ_j that describe how close the algorithm is to find v^* after it has asked j queries. μ_j is (roughly) the distance between v^* and the query node closest to v^*, and tends to decrease as j increases. We also define a corresponding *deterministic* sequence $\lambda_0 > \lambda_1 > \cdots$ where for each j, $\mu_j \geq \lambda_j$ with high probability.
3. We define the following events which (basically) imply each other in alternation (i.e. $A_j \Rightarrow B_j \Rightarrow A_{j+1}$), and will help us bound the progress of the algorithm:
   - A_j is the event that $\mu_j \geq \lambda_j$; it intuitively means "none of the j first query nodes are too close to v^*";
   - B_j will be defined later, and intuitively means "even after asking j queries, the algorithm has only a vague idea where v^* is", or a bit more precisely, "even conditioned on all the answers gathered by the algorithm during the first j steps, none of the nodes have a high probability of being the source".
4. We define j_{stop} be the largest j such that $\lambda_j \geq D$ (recall that D is the distance above which event T gives concentration bounds on answers). Our goal will be to prove that with high probability, the algorithm needs to ask at least j_{stop} queries.
5. We prove two key lemmas, which show that in most cases, $A_j \Rightarrow B_j$ and $B_j \Rightarrow A_{j+1}$. They state that for $j < j_{\text{stop}}$,
   - A_j implies B_j (Lemma 3);
   - with probability $1 - \frac{1}{\log n}$, $T \wedge A_j \wedge B_j$ implies A_{j+1} (Lemma 4).

   This is the core technical part of the proof.
6. We chain the above lemmas by induction and use the fact that $\Pr[\neg T] \leq p/2$ to obtain $\Pr[A_{j_{\text{stop}}}] \geq 1 - p$.
7. We prove that $j_{\text{stop}} = \Omega(\log\log_{\max(1/\sigma,2)} n)$, the desired lower bound.
8. We observe that event $A_{j_{\text{stop}}}$ implies that the algorithm has not found v^* after asking j_{stop} queries, which using 6 and 7 completes the proof.

6.2.1. *Typical instances: event T*

In our model, there are no hard guarantees on how far away the answer $\text{ans}_w(v^*, q)$ might be from the real distance $|v^* - q|$. For example, $\text{ans}_w(v^*, v^* + 1) \sim \mathcal{N}(1, \sigma^2)$ might be as large as 1000, even if $\sigma = 1$ (though with very low probability). While such extreme events are intuitively disadvantageous for the algorithm, they also make it harder to prove lower bounds. Therefore, we need to make basic assumptions on the range of $\text{ans}_w(v^*, q)$ at high distances.

To do this, we will need to use the notion of a "typical" instance: a choice of v^* and w for which some reasonable concentration results hold. Note that part (i) below is very similar to Lemma 1, which we used for the upper bound. Let $C(\delta)$ and $D(\sigma, \delta)$ be two values (defined later in Definition 12) such that

$$\max(\sigma^2, e^2) \leq D(\sigma, \delta) = o_\delta(\max(\sigma^2 \log \sigma, 1)). \tag{10}$$

Definition 3 *(*$\text{Typical}_\delta(v^*, w)$*).* For any probability $\delta > 0$, let $\text{Typical}_\delta(v^*, w)$ be the event that the following holds:

(a) $\min(\text{ans}_w(v^*, 1), \text{ans}_w(v^*, n)) \geq \frac{n}{C(\delta)}$;
(b) for all q with $d_q := |v^* - q| \geq D(\sigma, \delta)$,
  (i) $\text{ans}_w(v^*, q) \in d_q \pm \sigma\sqrt{d_q}\ln d_q$
  (ii) $\text{ans}_w(v^*, q) \in d_q \pm d_q/4 = [\frac{3}{4}d_q, \frac{5}{4}d_q]$.

Part (a) means that the two answers from the query nodes at either end of the path are not too much smaller than their expectation $\Omega(n)$, and part (b) means that above a certain distance threshold $D(\sigma, \delta)$, all answers are concentrated around their mean.

As the name indicates, most instances are typical (the proof is given in Appendix G).

Lemma 2. *For any probability $\delta > 0$ and any $n \geq \Theta_\delta(\max(\sigma^2 \ln \sigma, 1))$, $\Pr_{v^*,w}[\text{Typical}_\delta(v^*, w)] \geq 1 - \delta$.*

We will apply Lemma 2 with $\delta := p/2$. We will use the following shorthands.

Definition 4 *(T, C, D).* Let $T := \text{Typical}_{p/2}(v^*, w)$, $C := C(p/2)$ and $D := D(\sigma, p/2)$.

Corollary 1. $\Pr[T] \geq 1 - p/2$.

6.2.2. *Measure of progress μ_j and benchmark λ_j*

It turns out that the right metric of progress to look at is (roughly speaking) the *smallest answer value seen so far*. More precisely, suppose that the algorithm has asked j queries so far (and hence is at step j). Then we define the quantity μ_j as follows.

Definition 5 *(l_j, r_j, μ_j).* Let $l_j := \arg\max_{i\le j, q_i \le v^*}(q_i)$ and $r_j := \operatorname{argmin}_{i\le j, q_i\ge v^*}(q_i)$, which means that q_{l_j} (resp. q_{r_j}) is the closest query node at or to the left (resp. right) of v^* placed so far. Then $\mu_j := \min(a_{l_j}, a_{r_j})$, the smaller of the corresponding answers.

Note in passing that by simplifying assumption (b) in the beginning of Section 6.2, the algorithm knows l_j and r_j. Also, if $\mu_j > 0$, then the algorithm has not found v^* yet (otherwise we would have $q_{l_j} = q_{r_j} = v^*$ and thus $a_{l_j} = a_{r_j} = 0$).

We want to show that, with high probability, μ_j cannot decrease too fast with j. To make it formal, we define an analogous *deterministic* sequence λ_j, which we will show is a lower bound for μ_j with high probability. We call λ_j a "benchmark" because it is a point of comparison to determine whether the algorithm is making fast progress or not. It decreases with j according to the following function.

Definition 6. Let $\text{reduce}_{n,\sigma}(x) = \frac{\sigma\sqrt{x}}{400\ln x \log n}$.

Definition 7. Let $\lambda_0 := n/C$ and $\lambda_{j+1} := \text{reduce}_{n,\sigma}(\lambda_j)$.[7]

Observe that by point (a) in Definition 3, T implies $\mu_0 \ge n/C = \lambda_0$. Our goal will be to prove that $\mu_j \ge \lambda_j$ will likely continue to hold as j increases.

6.2.3. Events A_j and B_j

Informally, at step j, A_j is the event that the algorithm has not queried any nodes very close to v^*, and B_j is the event that the algorithm has only a vague idea of where v^* is (or more precisely, that even conditioned on all the answers so far, no node has a high probability of being the source). As we will see in Section 6.2.5), intuitively,

- A_j implies B_j because if the algorithm does not have any query nodes close to v^*, then the answers it got are all very noisy, and thus its confidence interval for v^* is wide (Lemma 3);
- B_j implies A_{j+1} because if all nodes are very unlikely to be the source v^*, then wherever it decides to query the next node, it is unlikely to be very close to v^* (Lemma 4).

As we will see, both events depend only on information that is available to the algorithm at step j. For convenience, we define random variable K_j, which describes all the knowledge of the algorithm up to step j.

Definition 8 *(K_j).* Let $\{q_i\}_{\le j} := (q_{-1}, \dots, q_j)$ and $\{a_i\}_{\le j} := (a_{-1}, \dots, a_j)$ be the query nodes and answers available at step j. Then let $K_j = (\{q_i\}_{\le j}, \{a_i\}_{\le j}, l_j, r_j)$. This encodes the locations of all query nodes, the answers received from them, as well as the identity of the two query nodes between which v^* lies.

A_j is the event that μ_j is greater than the benchmark λ_j.

Definition 9. Let A_j be the event that $\mu_j \ge \lambda_j$.

B_j is the event that the posterior of $v^* = v$ given K_j is "diluted".

Definition 10. Let B_j be the event that for all nodes $v \in V$,

$$\Pr[T \wedge (v^* = v) \mid K_j] \le \frac{1}{\left(\frac{8}{3}\lambda_{j+1} + 1\right)\log n}.$$

Note that $\Pr[T \wedge (v^* = v) \mid K_j]$ itself is a random variable since it depends on K_j, so B_j is still a random event even though it is a statement about a probability. An equivalent way to define B_j is to first define random variable

$$P_j := \max_{v\in V} \Pr[T \wedge (v^* = v) \mid K_j],$$

then to let B_j be the event that $P_j \le \frac{1}{\left(\frac{8}{3}\lambda_{j+1}+1\right)\log n}$.

[7] If $\lambda_j \le 0$, we define $\lambda_{j+1} := 0$. However, we will never use such values.

6.2.4. Stopping step j_{stop}

Our goal is to show that for a high value of j, we have $\mu_j > 0$ with high probability, and therefore the algorithm has failed to find v^* using only j queries. We now define that value of j.

Definition 11 *($j_{\min}$, j_{stop}).* Let $j_{\min}$ be the smallest integer $j \geq 0$ such that $\lambda_j < D$. Then

$$j_{\text{stop}} := \min\left(j_{\min} - 1, \left\lfloor \frac{p \log n}{2} \right\rfloor\right).$$

This means that at step $j \leq j_{\text{stop}}$, $\lambda_j \geq D$ is still big enough for the concentration bounds of event T to hold. The second argument of the $\min(\cdot,\cdot)$ is just for convenience of the proof, and will not matter if n is large enough. We will also use the following easily believable fact, proved in Appendix H.

Fact 3. $\lambda_0 > \lambda_1 > \cdots > \lambda_{j_{\text{stop}}} > \lambda_{j_{\text{stop}}+1}$.

6.2.5. Key lemmas

We now state our two main lemmas, which constitute the core technical part of the proof. The proof of Lemma 3 is very technical and not particularly enlightening, so it is deferred to Appendix I. The proof of Lemma 4, on the other hand, is much more straightforward, and we include it here.

Lemma 3. *If $j < j_{\text{stop}}$, then $A_j \Rightarrow B_j$.*

Lemma 4. *If $j \leq j_{\text{stop}}$, then $\Pr[\neg T \vee \neg A_j \vee \neg B_j \vee A_{j+1}] \geq 1 - \frac{1}{\log n}$.*

Note that "$\neg T \vee \neg A_j \vee \neg B_j \vee A_{j+1}$" is logically equivalent to "$(T \wedge A_j \wedge B_j) \Rightarrow A_{j+1}$". Intuitively, if B_j holds, then the probability of $v^* = v$ (conditioned on the answers so far) is low for any v, which means that whatever the algorithm picks as its next query q_{j+1}, the probability that q_{j+1} is within some distance d of v^* is upper bounded by the sum of those probabilities over $v \in [q_{j+1} - d, q_{j+1} + d]$. Therefore, with high probability, a_{j+1} will not be too small, and the same holds for μ_{j+1}.

Proof of Lemma 4. We will show equivalently that $\Pr[T \wedge A_j \wedge B_j \wedge \neg A_{j+1}] \leq \frac{1}{\log n}$.

At step j, the algorithm queries node q_{j+1} based on the information K_j it has so far and its internal randomness R, then receives answer a_{j+1}. The only way for both A_j and $\neg A_{j+1}$ to hold is for the new answer a_{j+1} to be smaller than λ_{j+1}.[8]

Let $d := |v^* - q_{j+1}|$. If we had $d \geq \frac{4}{3}\lambda_{j+1} \geq D$, then if T occurs, by concentration bound (ii) we would have $a_{j+1} \geq \frac{3}{4}d \geq \lambda_{j+1}$. Let $I := V \cap (q_{j+1} \pm \frac{4}{3}\lambda_{j+1})$ (I also depends on (K_j, R)). Then the only way to have $a_{j+1} < \lambda_{j+1}$ is for v^* to be in I, which implies

$$\Pr[T \wedge A_j \wedge B_j \wedge \neg A_{j+1}] \leq \Pr[T \wedge v^* \in I \wedge B_j]. \tag{11}$$

Now, for any assignment (k_j, r) of random variables (K_j, R), we have

$$\begin{aligned}
&\Pr[T \wedge v^* \in I \wedge B_j \mid K_j = k_j \wedge R = r] \\
&\quad = \sum_{v \in I} \Pr[T \wedge (v^* = v) \wedge B_j \mid K_j = k_j \wedge R = r] && (I \text{ is fixed by } (K_j, R)) \\
&\quad = \sum_{v \in I} \Pr[T \wedge (v^* = v) \wedge B_j \mid K_j = k_j]. && (T, v^* \text{ are independent from } R\text{, and } B_j \text{ is fixed by } K_j)
\end{aligned}$$

If B_j is false given $K_j = k_j$, then the above sum has probability 0. If on the other hand B_j is true given $K_j = k_j$, then by definition of B_j,

$$\begin{aligned}
\sum_{v \in I} \Pr[T \wedge (v^* = v) \wedge B_j \mid K_j = k_j] &= \sum_{v \in I(k_j, r)} \Pr[T \wedge (v^* = v) \mid K_j = k_j] \\
&\leq \frac{|I|}{\left(\frac{8}{3}\lambda_{j+1} + 1\right)\log n} \leq \frac{1}{\log n}.
\end{aligned}$$

Therefore, in either case, $\Pr[T \wedge v^* \in I \wedge B_j] \leq \frac{1}{\log n}$, which, combined with (11), completes the proof. □

[8] Formally, if $A_j \wedge \neg A_{j+1}$, then using Fact 3 we have $\mu_{j+1} < \lambda_{j+1} < \lambda_j \leq \mu_j$. Therefore, $\mu_{j+1} = a_{j+1}$, and thus $a_{j+1} < \lambda_{j+1}$.

6.2.6. *Induction on j*

Lemmas 3 and 4 can now be chained to obtain the following result.

Lemma 5. $\Pr[A_{j_{\text{stop}}}] \geq 1 - p$.

Proof. First, as already noted at the end of Section 6.2.2, T implies $\mu_0 \geq \lambda_0$, which means that $T \Rightarrow A_0$ (by definition of A_0). Also, by Lemma 3, $A_j \Rightarrow B_j$ for $0 \leq j < j_{\text{stop}}$. In addition, by Corollary 1, we have $\Pr[T] \geq 1 - p/2$, and by Lemma 4, for $0 \leq j < j_{\text{stop}}$, we have $\Pr[\neg T \vee \neg A_j \vee \neg B_j \vee A_{j+1}] \geq 1 - \frac{1}{\log n}$. Therefore by a union bound, both T and "$\neg T \vee \neg A_j \vee \neg B_j \vee A_{j+1}$ for $0 \leq j < j_{\text{stop}}$" simultaneously hold with probability at least

$$1 - p/2 - \frac{j_{\text{stop}}}{\log n} \geq 1 - p/2 - \frac{\left\lfloor \frac{p \log n}{2} \right\rfloor}{\log n} \geq 1 - p. \qquad \text{(by Definition 11)}$$

If they do hold, then the following logical statements are all true: "T", "$T \Rightarrow A_0$", "$A_j \Rightarrow B_j$" ($\forall j < j_{\text{stop}}$), and "$(T \wedge A_j \wedge B_j) \Rightarrow A_{j+1}$" ($\forall j < j_{\text{stop}}$). It is easy to see that, chained together, they imply $A_{j_{\text{stop}}}$. □

6.2.7. *Asymptotics of j_{stop}*

The following lemma gives us an asymptotic lower bound on j_{stop}. We prove it in Appendix J.

Lemma 6. *For $n \geq \Theta_p(\max(\sigma^3, 1))$, we have*

$$j_{\text{stop}} + 1 = \begin{cases} \Omega_p(1 + \log(1 + \log_{1/\sigma} n)) \text{ if } \sigma^2 \leq 1/2 \\ \Omega_p(\log \log n) \text{ if } \sigma^2 \geq 1/2. \end{cases}$$

6.2.8. *Proof of Theorem 4*

All that is left to do is to conclude.

Proof of Theorem 4. If $A_{j_{\text{stop}}}$ holds, then $\mu_{j_{\text{stop}}} > 0$, which means the algorithm has not found v^* after asking j_{stop} queries (recall our assumption from the beginning of Section 6.2 that, without loss of generality, the algorithm must query the source in order to make its guess). By Lemma 5, this happens with probability at least $1 - p$. Therefore, any algorithm that finds v^* with probability at least p must use at least $j_{\text{stop}} + 1$ queries. The theorem then follows from Lemma 6. □

7. Conclusion and future work

We presented the first mathematical study of source identification with time queries in a non-deterministic diffusion process. We considered both the setting when the queries are selected adaptively and non-adaptively. We found that when the edge-delay distribution has constant variance, the number of required queries is $\Theta(\log \log n)$ in the adaptive setting, and $\Theta(n)$ in the non-adaptive setting. Our results are in sharp contrast with similar problems, such as measurement dependent noisy search on a line [25], or probabilistic binary search in graphs [10], where the query complexities were found to be $\Theta(\log n)$ in both cases.

The main open question is of course what happens in other graphs. Extending our results to certain classes of trees might be feasible with the methods presented in this paper, however, an extension to graphs with cycles seems very challenging. Still, we hope that our results can inspire some, potentially more heuristic, ideas for treating graphs with cycles as well.

While we do not consider this scenario, given the sensitive nature of health information, it would be interesting to study source identification with time queries in the context of privacy preserving learning. In a scenario where an adversary is watching our queries, but not the responses, a recent line of work characterized the tradeoff between query complexity and privacy in adaptive binary search on a line [52,56]. The model has been extended to the case when the answers we receive are noisy in a follow-up work by [55]. It would be interesting to combine the methods presented in the current paper with the methods of [52,56,55] for new results in privacy preserving source identification.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work presented in this paper was supported in part by the Swiss National Science Foundation under grant number 200021-182407. Most of Victor Lecomte's work was done while visiting EPFL on a Summer@EPFL fellowship.

Appendix A. Extending to other edge-delay distributions

Throughout the paper, we assumed that the edge-delay distribution was Gaussian, however due to the Central Limit Theorem, it is natural to expect that our result generalizes to other distributions as well. However, there definitely are edge-delay distributions for which our result cannot generalize. Consider an edge-delay distribution $\mathcal{W}$ supported on two values: 1 and π. Since π is irrational, a single query node at one end of the path can determine the identity of the source with absolute certainty. Moreover, our results are not likely to generalize to heavy tailed $\mathcal{W}$ due to the lack of concentration in the answers.

We sketch how our proofs could be generalized to continuous sub-gaussian random variables. In the proofs of our main results, we exploit two types of properties of the edge-delay distribution; we are using the tight concentration of their sum in the non-adaptive upper bound and the adaptive upper and lower bounds, and we are using an anti-concentration result on their sum in the non-adaptive and adaptive lower bounds.

All of the concentration bounds are derived from Fact 2. This tail-bound result is easily extendable to sub-gaussian random variables (see Proposition 5.10 of [53]). The only difference in the results would be that σ would be replaced by the sub-gaussian norm

$$\|\mathcal{W}\|_{\psi_2} = \text{supp}_{p\geq 1} p^{-1/2}(\mathrm{E}\,|\mathcal{W}|^p)^{1/p}.$$

In the adaptive lower bound proof, when we make the anti-concentration arguments, our proof uses the density function of the Gaussian distribution. Therefore, we need that the density function of $\sum \mathcal{W}_i$ is pointwise close to the density function of the corresponding Gaussian distribution. Such statements are called local limit theorems for sums of independent random variables. In a sense, we are asking for much more than a tail-bound, but we can also be much looser than an exponential decay. In Lemma 8 we need that the probability mass function of the posterior is bounded above by $\log(\mu_j)/(\sigma\sqrt{\mu_j})$. We prove this by writing the probability mass function (as a function of potential source v') explicitly using Bayes rule and the density of $\sum_{i=1}^{d'_l} \mathcal{W}_i$ and $\sum_{i=1}^{d'_r} \mathcal{W}_i$, where $d'_l = v' - l_j$ and $d'_r = r_j - v'$ are the distances between a node v' and the closest query nodes to the left and the right. We need these densities to be pointwise $o(\log(\mu_j)/(\sigma\sqrt{\mu_j}))$ close to the densities in the Gaussian case. Since in Claim 6 we prove that $d'_l \in [1/2, 2]a_{l_j}$ and $d'_r \in [1/2, 2]a_{r_j}$, and by the definition $\mu_j = \min(a_{l_j}, a_{r_j})$, it is enough to show that the density of $\sum_{i=1}^{d'_l} \mathcal{W}_i$ is pointwise $o(\log(d'_l)/(\sigma\sqrt{d'_l}))$ close to the density in the Gaussian case (and we need the symmetric statement for d'_r). Such results are readily available for continuous distributions $\mathcal{W}$ with finite third moment (see Theorem 7.15 in [36]). We also point out, that similar results exist for discrete distributions $\mathcal{W}$ satisfying a certain lattice condition that can be used to rule out distributions like the one supported on 1 and π that we used as a counterexample in the beginning of the section (see Theorem 7.6 in [36]).

For the anti-concentration result in the non-adaptive lower bound, we proved that the hypothesis testing problem cannot be solved between k neighboring nodes at distance d or more away from the query nodes. For this we upper bounded the union of the area under the probability density functions of the answers under each of the k hypotheses by another function, which we could easily integrate. For general edge-delay distributions, again we aim to approximate the probability density functions of the answers by the probability density function of Gaussian random variables. Since this time, instead of small l_∞ distance, we need small l_1 distance between the densities, a Berry-Esseen type theorem [3,12] suffices instead of a local limit theorem.

We note that only the concentration arguments required the sub-gaussianity of the edge-delay distribution, the anti-concentration results held for a much more general class of distributions (finite third moment and continuity or lattice condition). We believe that with more advanced proof techniques the sub-gaussianity condition can also be relaxed.

Appendix B. The difference between S1 and S2

The only difference between two models S1 and S2 defined in Section 1 is that the starting time of the epidemics is unknown in S1 and known in S2. We already mentioned that S2 is theoretically more appealing, and that there is little difference between the number of queries required in the two models. The main consequence of the difference between the source identification algorithms in the two models is that in S1, the answers that they can use are the relative differences between time measurements at different pairs of query nodes, whereas in S2 the answers they can use are the absolute differences between the (known) starting time of the epidemics and the time measurement at each query node. Since S1 is more restrictive than S2, our lower bounds on the number of required queries in S2 clearly also hold in S1. We comment on how the upper bounds can be extended in Remarks 1 and 2.

Additionally, we argue that while S2 has a simpler mathematical definition than S1, on the path network, proving lower bounds for S2 raises important challenges that would not have appeared in S1. Indeed, in the path network, the pair of query nodes that surround the source provide two independent answers about it in S2 (one from each direction between each query node and the source), but only one in S1 (because only the time difference between the measurements is meaningful). As a result, the analysis of the required number of queries is more challenging in S2 than in S1 because of the richer set of independent answers. Incorporating several independent measurements will be the main difficulty for the analysis of the number of queries needed to identify the source in more complex network models, such as bounded-degree trees. By focusing on S2 in the path network, our paper therefore paves the way towards the analysis of more complex network models.

Remark 1. If the time of the first infection is not known (model S1), we can model the answers by adding an unknown constant T_{start} to all of them. Then, Claim 1 (a) and (b) hold without modification and we can prove a version of (c) where the differences of the distances equal the differences of the answers rounded to the nearest integer (we just need a slightly tighter concentration result). Let us also define $q_{\text{right}} := q_{\text{smallest}} + d$, and a_{right} as the corresponding answer. Then, by Claim 1, if $\lfloor a_{\text{smallest}} - a_{\text{left}} \rceil = d$ then v^* is between q_{smallest} and q_{right} and we can find v^* by computing

$$\frac{\lfloor a_{\text{smallest}} - a_{\text{right}} \rceil + q_{\text{smallest}} + q_{\text{right}}}{2} = \frac{(v^* - q_{\text{smallest}}) - (q_{\text{right}} - v^*) + q_{\text{smallest}} + q_{\text{right}}}{2} = v^*$$

Otherwise, if $\lfloor a_{\text{smallest}} - a_{\text{left}} \rceil < d$ then v^* is between q_{left} and q_{smallest}, and v^* can be found analogously.

Remark 2. If the time of the first infection is not known (model S1), and we model the answers by adding an unknown constant T_{start} to all of them, then a version of Lemma 1 shifted by T_{start} still holds. In this case, a slightly modified version of the algorithm finds the source. At each step the algorithm will query two nodes: one at l_i and r_i, with $l_0 = 1$ and $r_0 = n$. Then, we have a similar equation as (9) for the difference of the answers

$$\text{ans}_w(v^*, l_i) - \text{ans}_w(v^*, r_i) \in (v^* - l_i) - (r_i - v^*) \pm 2C(\delta) \cdot \sigma\sqrt{d_i}\ln(1 + d_i),$$

where $d_i := r_i - l_i$, which means that we can keep track of a shrinking interval

$$\begin{cases} l_{i+1} = \max\left(l_i, \left\lceil \frac{\text{ans}_w(v^*,l_i) - \text{ans}_w(v^*,r_i) + l_i + r_i}{2} - C(\delta) \cdot \sigma\sqrt{d_i}\ln(1 + d_i) \right\rceil\right) \\ r_{i+1} = \min\left(r_i, \left\lfloor \frac{\text{ans}_w(v^*,l_i) - \text{ans}_w(v^*,r_i) + l_i + r_i}{2} + C(\delta) \cdot \sigma\sqrt{d_i}\ln(1 + d_i) \right\rfloor\right). \end{cases}$$

The rest of the proof can be written similarly to the case when the time of the first infection is known, and the only change in the final result is that we used twice as many queries to identify the source (which does not affect the asymptotic results).

Appendix C. Simulation details for Fig. 1

The simulation results were generated in the S1 source identification model with the Python toolbox [46], which has been published in [48]. The underlying diffusion process was a Susceptible-Infected process (also called First Passage percolation) with uniform edge-delay distribution supported on the interval $[1 - \sqrt{3}\sigma, 1 + \sqrt{3}\sigma]$ (with mean 1 and standard deviation σ). Thereafter, a uniformly random query node was picked, and all further queries were selected by the Max-Gain algorithm as implemented in [46]. The algorithm was stopped when the candidate set reduced to a single node, which always had to be the source, since the Max-Gain algorithm always finds the correct source if enough queries are provided and the edge-delay distribution is bounded in some interval $[1 - \epsilon, 1 + \epsilon]$ for $\epsilon \in (0, 1)$ (see Theorem 2 of [48]). The number of queries plotted in Fig. 1 is the number of queries used by the Max-Gain until it was stopped, averaged over 192 simulations.

Appendix D. Proof of Claim 1

Claim 1. *For some $d = \Omega\left(\frac{1}{\sigma^2 \log(1/\delta)}\right)$, all of the following hold simultaneously with probability $\geq 1 - \delta$:*

(a) among the query nodes located at or to the left of v^*, the closest one is the one with the smallest answer;
(b) among the query nodes located at or to the right of v^*, the closest one is the one with the smallest answer;
(c) the two closest query nodes to v^* on its left side and the closest query node on its right side all give a correct answer once rounded to the nearest integer.

Proof. In this proof, we will assume that

$$\sigma^2 \leq \frac{1}{16\ln(6/\delta)} \leq \frac{1}{2\ln(12/\delta)}. \tag{D.1}$$

If this is not the case, then $\sigma^2 = \Omega(1/\log(1/\delta))$, so we can simply query every node, which gives $d = 1 = \Omega\left(\frac{1}{\sigma^2 \log(1/\delta)}\right)$.

We need to choose d such that wherever v^* is identified, (a), (b), (c) simultaneously hold with probability $\geq 1 - \delta$ over the choice of the weights $w(\cdot)$. Let us first study point (b) (point (a) is analogous). Let q be the closest query node at or to the right of v^*. Then (b) is true iff

- the sum of the weights of the edges between q and $q+d$ is positive;
- the sum of the weights of the edges between q and $q+2d$ is positive;
- ...
- the sum of the weights of the edges between q and $1+\left\lfloor \frac{n-1}{d} \right\rfloor d$ is positive.

More formally, (b) is true iff for all integers $i>0$ such that $q+id \le n$, the sum of the weights of the edges between q and $q+id$ is positive.

A sufficient condition for this to hold is: for all positive integers x, the sum of the weights of the edges between nodes q and $q+x$ is positive. By Fact 1, each of these sums is distributed as a Gaussian $\mathcal{N}(x, x\sigma^2)$, so it is positive except with probability

$$\begin{aligned}\Pr_{X\sim\mathcal{N}(x,x\sigma^2)}[X<0] &\le \Pr_{X\sim\mathcal{N}(x,x\sigma^2)}[X\notin x\pm x]\\ &\le e^{-\frac{x^2}{2x\sigma^2}} && \text{(Fact 2)}\\ &= e^{-\frac{x}{2\sigma^2}}.\end{aligned}$$

Therefore, by a union bound, (b) holds except with probability at most

$$\begin{aligned}\sum_{x=1}^{\infty}\left(e^{-\frac{1}{2\sigma^2}}\right)^x &= \frac{e^{-\frac{1}{2\sigma^2}}}{1-e^{-\frac{1}{2\sigma^2}}}\\ &< 3e^{-\frac{1}{2\sigma^2}} && \text{(because } \sigma^2\le 1 \Rightarrow e^{-\frac{1}{\sigma^2}} < 2/3\text{)}\end{aligned}$$

which, assuming $\sigma^2 \le \frac{1}{2\ln(12/\delta)}$ (equation (D.1)), is at most $\delta/4$.

Finally, we study the probability that (c) holds. Let d_1, d_2, d_3 be the distances of those three query nodes to v^*. They are all at most $2d$ away from v^*. For $i=1,2,3$, the corresponding answer is distributed as $X\sim\mathcal{N}(d_i, d_i\sigma^2)$, and is correct after rounding iff $X\in(d_i-1/2, d_i+1/2)$. Therefore, (c) holds except with probability

$$\begin{aligned}\sum_{i=1}^{3}\Pr_{X\sim\mathcal{N}(d_i,d_i\sigma^2)}[X\notin d_i\pm 1/2] &\le \sum_{i=1}^{3} e^{-\frac{1}{8d_i\sigma^2}} && \text{(Fact 2)}\\ &\le 3e^{-\frac{1}{16d\sigma^2}}\end{aligned}$$

which, assuming $d\le\frac{1}{16\sigma^2\ln(6/\delta)}$, is at most $\delta/2$. Therefore, we set $d := \left\lfloor \frac{1}{16\sigma^2\ln(6/\delta)} \right\rfloor$. By (D.1), $\frac{1}{16\sigma^2\ln(6/\delta)} \ge 1$, so $d \ge (1/2)\cdot \frac{1}{16\sigma^2\ln(6/\delta)} = \Omega\left(\frac{1}{\sigma^2\log(1/\delta)}\right)$, as required.

Finally, by one more union bound, for our chosen value of d, all of (a), (b), (c) hold except with probability at most $\delta/4+\delta/4+\delta/2=\delta$. □

Appendix E. Proof of Lemma 1

Lemma 1. *For any probability* $0<\delta<1/2$*, there is some constant* $C(\delta)=O\left(\sqrt{\log(1/\delta)}\right)$ *such that for any* n, σ *and any source* $v^*\in V$*, we have*

$$\Pr_w[\forall q\in V,\ \mathrm{ans}_w(v^*,q)\in|v^*-q|\pm C(\delta)\cdot\sigma\sqrt{|v^*-q|}\ln(1+|v^*-q|)]\ge 1-\delta. \tag{E.1}$$

Proof. We will use the quantity $|v^*-q|$ many times in this proof, so to simplify notation, let $d_q := |v^*-q|$. We will fix $C(\delta)$ later, but for the moment assume $C(\delta)\ge 2$ (this is clearly the case for the value we set it to in (E.3)).

First of all, for $q=v^*$, (E.1) holds trivially. For any other q, by Fact 1, $\mathrm{ans}_w(v^*,q)$ is distributed according to Gaussian $\mathcal{N}(d_q, d_q\sigma^2)$ (though not independently). Then, by Fact 2, for any $q\ne v^*$,

$$\begin{aligned}&\Pr[\mathrm{ans}_w(v^*,q)\notin d_q\pm C(\delta)\sigma\sqrt{d_q}\ln(1+d_q)]\\ &\le e^{-\frac{(C(\delta)\sigma\sqrt{d_q}\ln(1+d_q))^2}{2d_q\sigma^2}}\\ &= e^{-\frac{C(\delta)^2(\ln(1+d_q))^2}{2}}\end{aligned}$$

$$
\begin{aligned}
&= \left(\frac{1}{1+d_q}\right)^{\frac{C(\delta)^2\ln(1+d_q)}{2}} \\
&= \left(\frac{1}{1+d_q}\right)^{\frac{(C(\delta)^2-1)\ln(1+d_q)+\ln(1+d_q)}{2}} \\
&= \left(\frac{1}{1+d_q}\right)^{\frac{(C(\delta)^2-1)\ln(1+d_q)}{2}} \left(\frac{1}{1+d_q}\right)^{\frac{\ln(1+d_q)}{2}} \\
&\le \left(\frac{1}{1+d_q}\right)^{\frac{C(\delta)^2}{4}} \left(\frac{1}{1+d_q}\right)^{\frac{\ln(1+d_q)}{2}} && (C(\delta)\ge 2 \text{ so } (C(\delta)^2-1)\ln(2)\ge C(\delta)^2/2) \\
&\le \left(\frac{1}{2}\right)^{\frac{C(\delta)^2}{4}} \left(\frac{1}{1+d_q}\right)^{\frac{\ln(1+d_q)}{2}}.
\end{aligned}
$$

By a union bound over all $q \neq v^*$, this implies that

$$
\begin{aligned}
&\Pr[\exists q \in V,\ \mathrm{ans}_w(v^*,q) \notin d_q \pm C(\delta)\sigma\sqrt{d_q}\ln(1+d_q)] \\
&\le 2\sum_{d=1}^{\infty}\left(\frac{1}{2}\right)^{\frac{C^2}{4}}\left(\frac{1}{1+d}\right)^{\frac{\ln(1+d)}{2}} \\
&= 2\left(\frac{1}{2}\right)^{\frac{C^2}{4}}\sum_{d=1}^{\infty}\left(\frac{1}{1+d}\right)^{\frac{\ln(1+d)}{2}} \\
&= 2\left(\frac{1}{2}\right)^{\frac{C^2}{4}}\left(\sum_{d=1}^{\lceil e^4+1\rceil-1}\left(\frac{1}{1+d}\right)^{\frac{\ln(1+d)}{2}} + \sum_{d=\lceil e^4+1\rceil}^{\infty}\left(\frac{1}{1+d}\right)^{\frac{\ln(1+d)}{2}}\right) \\
&\le 2\left(\frac{1}{2}\right)^{\frac{C^2}{4}}\left(\left\lceil e^4+1\right\rceil - 1 + \sum_{d=\lceil e^4+1\rceil}^{\infty}\left(\frac{1}{1+d}\right)^{2}\right) && (d\ge e^4-1 \text{ implies } \ln(1+d)/2\ge 2) \\
&\le 2\left(\frac{1}{2}\right)^{\frac{C^2}{4}}\left(\left\lceil e^4\right\rceil + \sum_{d=1}^{\infty}\frac{1}{d^2}\right) \\
&= 2\left(\frac{1}{2}\right)^{\frac{C^2}{4}}\left(\left\lceil e^4\right\rceil + \frac{\pi^2}{6}\right) \\
&\le 114\left(\frac{1}{2}\right)^{\frac{C^2}{4}}
\end{aligned}
\tag{E.2}
$$

Setting

$$
C(\delta) := \sqrt{4\log_2(114/\delta)} = O\left(\sqrt{\log(1/\delta)}\right), \tag{E.3}
$$

(E.2) becomes $\le \delta$, and we are done. □

Appendix F. Proof of Claim 3

Claim 3. *Assume $d_0 \le n$, and $d_{i+1} \le C\cdot\sigma\sqrt{d_i}\ln(1+d_i)$ for some value $C > 0$. Then*

- *if $\sigma^2 \le 1/2$, there exists $k = O(\log\log_{1/\sigma} n)$ such that $d_k = \mathrm{poly}(C)$;*
- *if $\sigma^2 \ge 1/2$, there exists $k = O(\log\log n)$ such that $d_k = \sigma^2\cdot\mathrm{poly}(C, \log(1+\sigma^2))$.*

Proof. We track the value of d_i/σ^2 as i increases. First, as long as

$$
d_i \ge \sigma^2(C\ln(1+d_i))^6 \tag{F.1}
$$

we have

$$\frac{d_{i+1}}{\sigma^2} \le \frac{C\sigma\sqrt{d_i}\ln(1+d_i)}{\sigma^2} = C\sqrt{\frac{d_i}{\sigma^2}}\ln(1+d_i) \overset{(F.1)}{\le} \left(\frac{d_i}{\sigma^2}\right)^{2/3}$$

(the last inequality can be deduced by dividing both sides by $\sqrt{\frac{d_i}{\sigma^2}}$ then raising both sides to the sixth power). Thus, by induction, as long as (F.1) holds, we have

$$\frac{d_i}{\sigma^2} \le \left(\frac{d_0}{\sigma^2}\right)^{(2/3)^i} \Rightarrow d_i \le \sigma^2\left(\frac{n}{\sigma^2}\right)^{(2/3)^i}.$$

Let $d_{\min}$ be the smallest value greater than $\max(2\sigma^2, 1)$ that we can assign to d_i such that (F.1) holds. Let k be the smallest integer for which $d_k \le d_{\min}$. Then we have

$$\sigma^2\left(\frac{n}{\sigma^2}\right)^{(2/3)^{k-1}} \ge d_{\min} \Leftrightarrow k \le 1 + \log_{3/2}\left(\frac{\log\left(\frac{n}{\sigma^2}\right)}{\log\left(\frac{d_{\min}}{\sigma^2}\right)}\right) \le 1 + \log_{3/2}\left(\frac{\log\left(\frac{n}{\sigma^2}\right)}{\log(\max(1/\sigma^2, 2))}\right).$$

- If $\sigma^2 \le 1/2$, then it is easy to verify that $d_{\min} = O((C\log C)^6) = \text{poly}(C)$. Therefore, for $k = O(1+\log(1+\log_{1/\sigma} n))$, we have $d_k \le d_{\min} = \text{poly}(C)$.
- If $\sigma^2 \ge 1/2$, then it is easy to verify that $d_{\min} = O(\sigma^2 C^6 \log(1+\sigma^2 C^6)^6) = \sigma^2 \cdot \text{poly}(C, \log(1+\sigma^2))$. Therefore, for $k = O(\log\log n)$, we have $d_k \le d_{\min} = \sigma^2 \cdot \text{poly}(C, \log(1+\sigma^2))$. □

Appendix G. Proof of Lemma 2

Lemma 2. *For any probability $\delta > 0$ and any $n \ge \Theta_\delta(\max(\sigma^2 \ln\sigma, 1))$, $\Pr_{v^*,w}[\text{Typical}_\delta(v^*, w)] \ge 1-\delta$.*

Before proving Lemma 2, we first prove two claims.

Claim 4. *For any probability $\delta_1 > 0$, there exists $D_1(\delta_1) > 0$ such that for any n, σ and any source $v^* \in V$,*

$$\Pr_w\left[\textit{for all } q \textit{ such that } d_q := |v^* - q| \ge D_1(\delta_1),\ \text{ans}_w(v^*, q) \in d_q \pm \sigma\sqrt{d_q}\ln d_q\right] \ge 1-\delta_1.$$

Proof. At first, let us consider only the case $q \ge v^*$. That is, consider node $q = v^* + d$ for some distance $d \ge e^2$. Then $\text{ans}_w(v^*, q) \sim \mathcal{N}(d, d\sigma^2)$, so

$$\begin{aligned}\Pr_w\left[\text{ans}_w(v^*, q) \in d \pm \sigma\sqrt{d}\ln d\right] &\ge 1 - e^{\frac{(\sigma\sqrt{d}\ln d)^2}{2d\sigma^2}} && \text{(from Fact 2)}\\ &= 1 - e^{-\frac{(\ln d)^2}{2}}\\ &= 1 - \frac{1}{d^{(\ln d)/2}}.\end{aligned}$$

Now, for any integer $D_1 \ge e$, by a union bound, this will hold for all $q \ge v^* + D_1$ with probability at least

$$1 - \sum_{d=D_1}^{\infty} \frac{1}{d^{(\ln d)/2}}.$$

Note that this sum converges, because $(\ln d)/2 > 1$ for large enough d. Thus the sequence of sums $(\sum_{d=k}^{\infty} 1/d^{(\ln d)/2})_{k\ge 3}$ converges to 0 and we can define $D_1(\delta_1) := \min\{k \ge 3 \mid \sum_{d=k}^{\infty} 1/d^{(\ln d)/2} \le \delta_1/2\}$.

Therefore, by going through the same reasoning for $q \le v^*$ and taking a union bound, we get that

$$\text{ans}_w(v^*, q) \in d_q \pm \sigma\sqrt{d_q}\ln d_q$$

will hold for all q at distance $d_q := |v^* - q| \ge D_1(\delta_1)$, except with probability at most $\delta_1/2 + \delta_1/2 = \delta_1$. □

Claim 5. *For any probability $\delta_2 > 0$ and any $\epsilon \in (0, 1)$, there exists $D_2(\delta_2, \epsilon, \sigma) > 0$ such that for any n, σ and any source $v^* \in V$,*

$$\Pr_w\left[\textit{for all } q \textit{ such that } d_q := |v^* - q| \ge D_2(\delta_2, \epsilon, \sigma),\ \text{ans}_w(v^*, q) \in (1 \pm \epsilon)d_q\right] \ge 1-\delta_2,$$

and $D_2(\delta_2, \epsilon, \sigma) = O_{\delta_2,\epsilon}(\max(\sigma^2 \log\sigma, 1))$.

Proof. At first, let us consider only the case $q \geq v^*$. That is, consider node $q = v^* + d$ for some distance d. Then $\mathrm{ans}_w(v^*, q) \sim \mathcal{N}(d, d\sigma^2)$, so

$$\Pr_w\left[\mathrm{ans}_w(v^*, q) \in (1 \pm \epsilon)d\right] \geq 1 - e^{\frac{(\epsilon d)^2}{2d\sigma^2}} \qquad \text{(from Fact 2)}$$

$$= 1 - (e^{-\frac{\epsilon^2}{2\sigma^2}})^d.$$

Now, for any integer D_2, by a union bound, this will hold for all $q \geq v^* + D_2$ except with probability at most

$$\sum_{d=D_2}^{\infty} \left(e^{-\frac{\epsilon^2}{2\sigma^2}}\right)^d = e^{-\frac{\epsilon^2}{2\sigma^2}D_2} \sum_{d=0}^{\infty} \left(e^{-\frac{\epsilon^2}{2\sigma^2}}\right)^d$$

$$= \frac{e^{-\frac{\epsilon^2}{2\sigma^2}D_2}}{1 - e^{-\frac{\epsilon^2}{2\sigma^2}}}, \qquad \text{(G.1)}$$

where the last step uses the fact that this is a geometric series.

- If $\frac{\epsilon^2}{2\sigma^2} \geq 1$, then (G.1) $\leq \frac{e^{-D_2}}{1-1/e}$, so if we set $D_2(\delta_2, \epsilon, \sigma) := \ln\left(\frac{2}{\delta_2(1-1/e)}\right)$, then (G.1) $\leq \delta_2/2$.
- If $\frac{\epsilon^2}{2\sigma^2} \leq 1$, then we can use $e^{-x} \leq 1 - x/2$ on $[0, 1]$ to obtain that

  $$\text{(G.1)} \leq \frac{2e^{-\frac{\epsilon^2}{2\sigma^2}D_2}}{\frac{\epsilon^2}{2\sigma^2}},$$

  so if we set $D_2(\delta_2, \epsilon, \sigma) := \frac{2\sigma^2}{\epsilon^2}\ln\left(\frac{4\sigma^2}{\epsilon^2\delta_2}\right)$, then (G.1) $\leq \delta_2/2$.

It is easy to check that both these values are $O_{\delta_2,\epsilon}(\max(\sigma^2 \ln\sigma, 1))$.

Finally, by going through the same reasoning for $q \leq v^*$ and taking a union bound, we get that

$$\mathrm{ans}_w(v^*, q) \in (1 \pm \epsilon)d_q$$

will hold for all q at distance $d_q := |v^* - q| \geq D_2(\delta_2, \epsilon, \sigma)$, except with probability at most $\delta_2/2 + \delta_2/2 = \delta_2$. □

Definition 12. Let $C(\delta) := 8/\delta$ and $D(\sigma, \delta) := \max(D_1(\delta/3), D_2(\delta/3, 1/4, \sigma), \sigma^2, e^2)$.

Let us verify that this definition of $D(\sigma, \delta)$ satisfies the bounds claimed in equation (10). The lower bound of $\max(\sigma^2, e^2)$ is trivial. The upper bound of $O_\delta(\max(\sigma^2 \log\sigma, 1))$ comes from the fact that $D_2(\delta_2, \epsilon, \sigma) = O_{\delta_2,\epsilon}(\max(\sigma^2 \log\sigma, 1))$.

We can now prove Lemma 2.

Proof of Lemma 2. Apply Claim 4 with $\delta_1 := \delta/3$, and Claim 5 with $\delta_2 := \delta/3$ and $\epsilon := 1/4$. Assume $n \geq 12/\delta$, and let $C' := 6/\delta$. Since v^* is uniformly distributed over $V = [n]$, we have

$$\Pr[\min(|v^* - 1|, |v^* - n|) < n/C'] \leq \frac{2 + n/C'}{n} \leq \frac{2}{n} + \frac{1}{C'} \leq \delta/3.$$

By a union bound, the concentration bounds of both Claim 4 and Claim 5 as well as inequality $\min(|v^* - 1|, |v^* - n|) \geq n/C'$ will all hold with probability at least $1 - \delta/3 - \delta/3 - \delta/3 = 1 - \delta$.

Furthermore, by the concentration bound of Claim 5, if $\min(|v^* - 1|, |v^* - n|) \geq n/C' \geq D_2(\delta/3, 1/4)$, then

$$\min(\mathrm{ans}_w(v^*, 1), \mathrm{ans}_w(v^*, n)) \geq (n/C')(1 - 1/4) = \frac{3n}{4C'} = \frac{n}{C(\delta)}. \qquad \text{(G.2)}$$

Then for $n \geq \max(12/\delta, C'D_2(\delta/3, 1/4, \sigma)) = O_\delta(\max(\sigma^2 \ln\sigma, 1))$, with probability at least $1 - \delta$, we have

- $\min(\mathrm{ans}_w(v^*, 1), \mathrm{ans}_w(v^*, n)) \geq \frac{n}{C(\delta)}$ (from (G.2));
- for all q with $d_q := |v^* - q| \geq D(\sigma, \delta)$,
  - $\mathrm{ans}_w(v^*, q) \in d_q \pm \sigma\sqrt{d_q}\ln d_q$ (from Claim 4)
  - $\mathrm{ans}_w(v^*, q) \in d_q(1 \pm 1/4)$ (from Claim 4). □

Appendix H. Proof of Fact 3

Fact 3. For $0 \le j \le j_{\text{stop}}$, $\lambda_{j+1} < \lambda_j$.

Proof of Fact 3. Since $j \le j_{\text{stop}}$, by definition of j_{stop}, $\lambda_j \ge D$. Also, by equation (10), $D \ge \max(\sigma^2, e^2)$. Therefore,

$$\begin{aligned}
\lambda_{j+1} &= \text{reduce}_{n,\sigma}(\lambda_j) && \text{(Definition 7)}\\
&= \frac{\sigma\sqrt{\lambda_j}}{400 \ln \lambda_j \log n} && \text{(Definition 6)}\\
&< \sigma\sqrt{\lambda_j} && (n \ge 3,\ \lambda_j \ge e^2)\\
&\le \lambda_j. \quad \square && (\lambda_j \ge \sigma^2)
\end{aligned}$$

Appendix I. Proof of Lemma 3

Lemma 3. *If $j < j_{\text{stop}}$, then $A_j \Rightarrow B_j$.*

The first step in proving Lemma 3 is to prove that only a small part of the information contained in K_j will actually influence the posterior of v^* given K_j: only the closest query nodes to the source q_{l_j}, q_{r_j} and the corresponding answers a_{l_j}, a_{r_j} will have an influence (they are introduced in Definition 5).

Definition 13 *($E_{l,r,x,y}$).* For any l, r, x, y, let $E_{l,r,x,y}$ be the event that $v^* \in [l, r]$, $\text{ans}_w(v^*, l) = x$, and $\text{ans}_w(v^*, r) = y$.

Note that event $E_{l,r,x,y}$ depends purely on v^* and w, not on the actions of the algorithm.

Lemma 7. *Recall that R is the internal randomness of the algorithm (see Definition 1). For any node $v \in V$,*

$$\Pr_{v^*,w,R}[v^* = v \mid K_j] = \Pr_{v^*,w}[v^* = v \mid E_{q_{l_j},q_{r_j},a_{l_j},a_{r_j}}].$$

Before proving this lemma, we need to show a simple property of independence and conditional probability.

Fact 4. Let X, Y be independent random variables. Let $E(X), F(X)$ be Boolean functions depending only on X, and let $G(F(X), Y)$ be a Boolean function depending only on $F(X)$ and Y. For simplicity, let us use $E(X)$ to denote the event $E(X) = 1$ (and similarly for F, G). Then we have

$$\Pr[E(X) \mid F(X) \wedge G(F(X), Y)] = \Pr[E(X) \mid F(X)].$$

Proof. Intuitively, the reason this is true is that as G depends only on F (which is already provided in the conditioning) and Y (which is independent from X), adding G to the conditioning does not bring more information towards figuring out whether E will happen or not. Formally, let $G'(Y) = G(1, Y)$ (1 represents "true"). Then

$$\begin{aligned}
\Pr[E(X) \mid F(X) \wedge G(F(X), Y)] &= \Pr[E(X) \mid F(X) \wedge G'(Y)]\\
&= \frac{\Pr[E(X) \wedge F(X) \wedge G'(Y)]}{\Pr[F(X) \wedge G'(Y)]}\\
&= \frac{\Pr[E(X) \wedge F(X)] \cdot \Pr[G'(Y)]}{\Pr[F(X)] \cdot \Pr[G'(Y)]} && \text{(independence of } X \text{ and } Y)\\
&= \frac{\Pr[E(X) \wedge F(X)]}{\Pr[F(X)]}\\
&= \Pr[E(X) \mid F(X)] \quad \square
\end{aligned}$$

We will notation $\mathbb{1}[\cdots]$ to denote the indicator Boolean function corresponding to some expression.

Proof of Lemma 7. Fix any possible assignment k_j for K_j. We observe that event "$K_j = k_j$" is entirely determined by

(a) R (the internal randomness of the algorithm);
(b) the values of $\text{ans}_w(v^*, q_i)$ for each $i \le j$ (the answers to the first j queries);

(c) whether $v^* \in [q_{l_j}, q_{r_j}]$.

Conversely, (b) and (c) are entirely determined by K_j. In addition, (b) and (c) depend only on v^* and w, which are independent from R.

By the above, we can use Fact 4, plugging in $E((v^*, w)) := \mathbb{1}[v^* = v]$, $F((v^*, w)) := \mathbb{1}[v^* \in [q_{l_j}, q_{r_j}] \wedge \text{ans}_w(v^*, q_i) = a_i \, \forall i \geq j]$ and $G(F((v^*, w)), R) := \mathbb{1}[K_j = k_j]$. This gives

$$\begin{aligned}
&\Pr_{v^*,w,R}[v^* = v \mid K_j = k_j] \\
&\quad = \Pr_{v^*,w,R}[v^* = v \mid \underbrace{v^* \in [q_{l_j}, q_{r_j}]}_{\text{(c)}} \wedge \underbrace{\text{ans}_w(v^*, q_i) = a_i \, \forall i \geq j}_{\text{(b)}} \wedge K_j = k_j] \\
&\quad = \Pr_{v^*,w,R}[\underbrace{v^* = v}_{E} \mid \underbrace{v^* \in [q_{l_j}, q_{r_j}] \wedge \text{ans}_w(v^*, q_i) = a_i \, \forall i \geq j}_{F} \wedge \underbrace{K_j = k_j}_{G}] \\
&\quad = \Pr_{v^*,w}[\underbrace{v^* = v}_{E} \mid \underbrace{v^* \in [q_{l_j}, q_{r_j}] \wedge \text{ans}_w(v^*, q_i) = a_i \, \forall i \geq j}_{F}]
\end{aligned} \tag{I.1}$$

Now, let us conceptually split w into two parts: w_{in}, which contains the weights of only the edges between nodes q_{l_j} and q_{r_j}, and w_{out}, which contains all the other weights. Since weights are distributed independently, (v^*, w_{in}) is independent from w_{out}.

Let $E((v^*, w_{\text{in}})) := \mathbb{1}[v^* = v]$, $F((v^*, w_{\text{in}})) := \mathbb{1}[v^* \in [q_{l_j}, q_{r_j}] \wedge \text{ans}_w(v^*, q_{l_j}) = a_{l_j} \wedge \text{ans}_w(v^*, q_{r_j}) = a_{r_j}]$, and $G(F((v^*, w_{\text{in}})), w_{\text{out}}) := \mathbb{1}[\text{ans}_w(v^*, q_i) = a_i \, \forall i \geq j]$. It is easy to see why E and F depend only on v^* and w_{in}. For G, we note that all other answers can be deduced just from the fact that $v^* \in [q_{l_j}, q_{r_j}]$, the values of $\text{ans}_w(v^*, q_{l_j})$ and $\text{ans}_w(v^*, q_{r_j})$, and w_{out}. Therefore, G depends only on F and w_{out}. Thus we can again apply Fact 4, to obtain

$$\begin{aligned}
&\Pr_{v^*,w}[v^* = v \mid v^* \in [q_{l_j}, q_{r_j}] \wedge \text{ans}_w(v^*, q_i) = a_i \, \forall i \geq j] \\
&\quad = \Pr_{v^*,w}[\underbrace{v^* = v}_{E} \mid \underbrace{v^* \in [q_{l_j}, q_{r_j}] \wedge \text{ans}_w(v^*, q_{l_j}) = a_{l_j} \wedge \text{ans}_w(v^*, q_{r_j}) = a_{r_j}}_{F} \wedge \underbrace{\text{ans}_w(v^*, q_i) = a_i \, \forall i \geq j}_{G}] \\
&\quad = \Pr_{v^*,w}[\underbrace{v^* = v}_{E} \mid \underbrace{v^* \in [q_{l_j}, q_{r_j}] \wedge \text{ans}_w(v^*, q_{l_j}) = a_{l_j} \wedge \text{ans}_w(v^*, q_{r_j}) = a_{r_j}}_{F}] \\
&\quad = \Pr_{v^*,w}[v^* = v \mid E_{q_{l_j}, q_{r_j}, a_{l_j}, a_{r_j}}].
\end{aligned} \tag{I.2}$$

The result follows from combining (I.1) with (I.2). □

For the remainder of this section, to make the notation lighter, we will use the following shorthands.

$$l := q_{l_j} \quad r := q_{r_j} \quad a_l := a_{l_j} \quad a_r := a_{r_j} \quad \mu := \mu_j = \min(a_l, a_r)$$

Definition 14 *(μ').* Let $\mu' := \frac{\sigma\sqrt{\mu}}{400 \ln \mu}$.

Definition 15 *(I).* Let I be the interval of all sources v^* that are consistent with event T and E_{l,r,a_l,a_r}: more precisely,

$$I := \{v \mid \Pr[T \wedge (v^* = v) \mid E_{l,r,a_l,a_r}] > 0\}.$$

Lemma 8. *Assume $\mu, \mu' \geq D$. Then for any node $v \in I$,*

$$\Pr_{v^*,w}[v^* = v \mid E_{l,r,a_l,a_r}] \leq \frac{100 \ln \mu}{\sigma\sqrt{\mu}}.$$

Lemma 8 has long and complicated proof, but its meaning is intuitive: it states is that when μ is large, the posteriors of v^* are not very concentrated at any point of segment I. The expression of this posterior is too complex to work with directly, so we will need the help of some facts and claims to prove what we want.

Before we prove Lemma 8, we start with Fact 5, which gives us a couple of useful inequalities, and Fact 6, a simple calculus result that we will use in this section and the next.

Fact 5. Assume $\mu, \mu' \geq D$. Then

$$\mu \geq 800^2 \cdot \max(\mu', D, \sigma^2, 1). \tag{I.3}$$

In particular, this implies

$$\begin{aligned} \mu &\geq (5/4)D && \text{(I.4)} \\ \sigma\sqrt{\mu} &\leq \mu/5 && \text{(I.5)} \\ \sigma\sqrt{\mu} &\leq \sigma\sqrt{(4/5)\mu}\ \ln((4/5)\mu) && \text{(I.6)} \\ (3/5)\mu &\geq 6000. && \text{(I.7)} \end{aligned}$$

Proof. Recall from equation (10) that $D \geq \sigma^2, e^2$. Since $\mu \geq D \geq e^2$, we have

$$400\ln\mu \geq 400\ln e^2 = 800.$$

This implies

$$\sigma^2 \leq D \leq \mu' = \frac{\sigma\sqrt{\mu}}{400\ln\mu} \leq \frac{\sigma\sqrt{\mu}}{800},$$

from which we get $\sigma \leq \sqrt{\mu}/800$ and $\mu' \leq \frac{\sigma\sqrt{\mu}}{800}$. Combining those, we get

$$\mu' \leq \frac{\sigma\sqrt{\mu}}{800} \leq \frac{\mu}{800^2},$$

which proves $\mu \geq 800^2\mu'$. The other three parts then follow directly from $\mu' \geq D \geq \sigma^2, e^2$.

Among (I.4)–(I.7), all are trivial from (I.3), except for (I.6) which can be rewritten as $5/4 \leq \ln((4/5)\mu)$. This clearly holds for $\mu \geq 10^4$. □

Fact 6. Let $f(d) := \frac{\ln d}{\sqrt{d}}$ and $g(d) := \frac{\sqrt{d}}{\ln d}$. On $[e^2, \infty)$, f is decreasing and g is increasing.

Proof. The derivative of f is $\frac{2-\ln d}{2d\sqrt{d}}$. □

Proof of Lemma 8. Since we have $\mu, \mu' \geq D$, Fact 5 applies here.

Assume I is not empty (otherwise, the lemma holds vacuously). Let I' be I extended by $\sigma\sqrt{\mu}$ on both sides. Note that, by definition, the length of I' is at least $2\sigma\sqrt{\mu}$.

Claim 6. *For any $v' \in I'$, let $d'_l := v' - l$ and $d'_r := r - v'$. Then*

$$\begin{aligned} d'_l &\in [1/2, 2]a_l && \text{(I.8)} \\ d'_r &\in [1/2, 2]a_r && \text{(I.9)} \\ |a_l - d'_l| &\leq 3\sigma\sqrt{d'_l}\ln d'_l && \text{(I.10)} \\ |a_r - d'_r| &\leq 3\sigma\sqrt{d'_r}\ln d'_r. && \text{(I.11)} \end{aligned}$$

Proof. We will only prove (I.8) and (I.10); the proof of the other two is analogous.

Let us first take a look at the properties of source candidates in I. Take some candidate $v \in I$ and consider the (true) distance between v and the left side of the interval, $d_l := v - l$. First, let us show that $d_l \geq D$. Indeed, if $d_l < D$, then point $v - D$ would be strictly to the left of l. Thus we would have

$$\text{ans}_w(v, v-D) > \text{ans}_w(v, l) = a_l \geq \mu \overset{\text{(I.4)}}{\geq} (5/4)D,$$

which contradicts part (ii) in Definition 3 for $q := v - D$.

Now that we have $d_l \geq D$, we can apply part (ii) in Definition 3 again to obtain that

$$a_l \in [3/4, 5/4]d_l \Rightarrow d_l \in [4/5, 4/3]a_l \tag{I.12}$$

and

$$|a_l - d_l| \in \sigma\sqrt{d_l}\ln d_l. \tag{I.13}$$

Let us now extend these results to I'. Any point $v' \in I'$ is at most $\sigma\sqrt{\mu}$ away from some point $v \in I$. Therefore, if we continue using notation $d_l := v - l$, we have $d'_l \in d_l \pm \sigma\sqrt{\mu}$.

From (I.12), we get

$$d'_l \in d_l \pm \sigma\sqrt{\mu} \subseteq [4/5, 4/3]a_l \pm \sigma\sqrt{\mu}.$$

Besides, from (I.5), $\sigma\sqrt{\mu} \le \mu/5 \le a_l/5$, so we have

$$d'_l \in [4/5, 4/3]a_l \pm a_l/5 = [3/5, 23/15]a_l,$$

which proves (I.8).

In addition, we note that

$$d'_l \ge d_l - \sigma\sqrt{\mu} \ge d_l - a_l/5 \overset{(I.12)}{\ge} d_l - d_l/4 = (3/4)d_l. \tag{I.14}$$

Therefore,

$$\begin{aligned} |a_l - d'_l| &\le |a_l - d_l| + \sigma\sqrt{\mu} && (\text{definition of } I') \\ &\le \sigma\sqrt{d_l}\ln d_l + \sigma\sqrt{\mu} && (\text{from (I.13)}) \\ &\le \sigma\sqrt{d_l}\ln d_l + \sigma\sqrt{(4/5)\mu}\ln((4/5)\mu) && (\text{from (I.6)}) \\ &\le 2\sigma\sqrt{d_l}\ln d_l && (d_l \ge (4/5)a_l \ge (4/5)\mu) \\ &\le 2\sigma\sqrt{4d'_l/3}\ln(4d'_l/3) && (\text{from (I.14)}) \\ &\le 3\sigma\sqrt{d'_l}\ln d'_l, \end{aligned}$$

where the last step holds because $d'_l \ge (3/5)a_l \ge (3/5)\mu \overset{(I.7)}{\ge} 6000$, which is big enough. This proves (I.10). □

Note that Claim 6 implies in particular that $I' \subseteq [l, r]$. Let us study the ratios of the posterior probabilities of $v^* = v'$ between different values of $v' \in I'$, conditioned on a_l, a_r (but regardless of whether T holds). We will use the shorthand

$$p(v') := \Pr[v^* = v' \mid v^* \in I' \wedge \text{ans}_W(v^*, l) = a_l \wedge \text{ans}_W(v^*, r) = a_r]. \tag{I.15}$$

Given that v^* is initially distributed uniformly, $p(v')$ is proportional to

$$\begin{aligned} &\Pr[\text{ans}_W(v^*, l) = a_l \wedge \text{ans}_W(v^*, r) = a_r \mid v^* = v'] \\ &\quad = \Pr[\text{ans}_W(v^*, l) = a_l \mid v^* = v']\Pr[\text{ans}_W(v^*, r) = a_r \mid v^* = v'], \end{aligned} \tag{I.16}$$

where the independence comes from the fact that $\text{ans}_W(v^*, l)$ and $\text{ans}_W(v^*, r)$ depend on completely separate weights.

Because we assumed that all weights are independently distributed from $\mathcal{N}(1, \sigma^2)$, both factors in (I.16) follow a normal distribution. Those distributions are $\mathcal{N}(d'_l, \sigma^2 d'_l)$ and $\mathcal{N}(d'_r, \sigma^2 d'_r)$, where we continue notations $d'_l := v' - l$ and $d'_r := r - v'$. This means that $p(v')$ is proportional to

$$\frac{1}{\sqrt{2\pi\sigma^2 d'_l}} e^{-\frac{(a_l - d'_l)^2}{2\sigma^2 d'_l}} \times \frac{1}{\sqrt{2\pi\sigma^2 d'_r}} e^{-\frac{(a_r - d'_r)^2}{2\sigma^2 d'_r}}.$$

Note that in the above expression, a_l, a_r are fixed by the conditioning, while d'_l and d'_r depend on v'. From now on, we will denote them as $d'_l(v')$ and $d'_r(v')$ to make this clear.

Of course, the constant $\frac{1}{2\pi\sigma^2}$ does not matter. Besides, we know that $d'_l(v') \in [1/2, 2]a_l$ and $d'_r(v') \in [1/2, 2]a_r$ with a_l, a_r fixed, so the factor $\frac{1}{\sqrt{d'_l(v')d'_r(v')}}$ will vary only by a factor 4. Thus we can conclude that $p(v')$ is also proportional to

$$F(v') := e^{-\frac{1}{2\sigma^2}\left(\frac{(a_l - d'_l(v'))^2}{d'_l(v')} + \frac{(a_r - d'_r(v'))^2}{d'_r(v')}\right)},$$

up to a factor 4 of error. More precisely, we know that there exists some $k > 0$ such that for all $v' \in I'$,

$$p(v') \in [1, 4]kF(v'). \tag{I.17}$$

Let

$$G(v') := \frac{1}{2\sigma^2}\left(\frac{(a_l - d'_l(v'))^2}{d'_l(v')} + \frac{(a_r - d'_r(v'))^2}{d'_r(v')}\right)$$

so that $F(v') = e^{-G(v')}$, and let v'_{peak} be the value of $v' \in I'$ that maximizes the expression $F(v')$. We will show the existence of a relatively large interval $J \subseteq I'$ centered around v'_{peak} such that for all $v' \in J$, the value $F(v')$ is not much smaller than $F(v'_{\text{peak}})$.

Claim 7. *There is some interval $J \subseteq I'$ of length at least $\frac{\sigma\sqrt{\mu}}{6\sqrt{2}\ln\mu}$, such that for all $v' \in J$,*

$$F(v') \geq \frac{F(v'_{\text{peak}})}{e}.$$

Proof. The derivative of $F(v')$ is $G'(v')e^{G(v')} = G'(v')F(v')$, where

$$G'(v') = \frac{1}{2\sigma^2}\left(-\frac{a_l - d'_l(v')}{d'_l(v')} - \left(\frac{a_l - d'_l(v')}{d'_l(v')}\right)^2 + \frac{a_r - d'_r(v')}{d'_r(v')} + \left(\frac{a_r - d'_r(v')}{d'_r(v')}\right)^2\right).$$

First, note that by (I.8) and (I.9), we have

$$\left|\frac{a_l - d'_l(v')}{d'_l(v')}\right| \leq 1 \text{ and } \left|\frac{a_r - d'_r(v')}{d'_r(v')}\right| \leq 1.$$

Therefore,

$$|G'(v')| \leq \frac{1}{\sigma^2}\left(\left|\frac{a_l - d'_l(v')}{d'_l(v')}\right| + \left|\frac{a_r - d'_r(v')}{d'_r(v')}\right|\right).$$

And then we can use equations (I.10) and (I.11) to bound this further:

$$\begin{aligned}
|G'(v')| &\leq \frac{1}{\sigma^2}\left(\frac{3\sigma\sqrt{d'_l(v')}\ln d'_l(v')}{d'_l(v')} + \frac{3\sigma\sqrt{d'_r(v')}\ln d'_r(v')}{d'_r(v')}\right)\\
&\leq \frac{3}{\sigma}\left(\frac{\ln d'_l(v')}{\sqrt{d'_l(v')}} + \frac{\ln d'_r(v')}{\sqrt{d'_r(v')}}\right)\\
&\leq \frac{3}{\sigma}\left(\frac{\ln(\mu/2)}{\sqrt{\mu/2}} + \frac{\ln(\mu/2)}{\sqrt{\mu/2}}\right) && (d'_l \geq a_l/2 \geq \mu/2 \geq e^2 \text{ and Fact 6; same for } d'_r)\\
&\leq \frac{6\sqrt{2}\ln\mu}{\sigma\sqrt{\mu}}.
\end{aligned}$$

We now have

$$|F'(v')| \leq \frac{6\sqrt{2}\ln\mu}{\sigma\sqrt{\mu}}F(v'),$$

for all $v' \in I'$, which from $F(v'_{\text{peak}}) > 0$ and Grönwall's Lemma for ordinary differential inequalities can be seen to imply that for all $v' \in I'$,

$$F(v') \geq e^{-\left|v' - v'_{\text{peak}}\right|\frac{6\sqrt{2}\ln\mu}{\sigma\sqrt{\mu}}}F(v'_{\text{peak}}).$$

This means that for any $v' \in I'$ within distance at most $\frac{\sigma\sqrt{\mu}}{6\sqrt{2}\ln\mu}$ of v'_{peak},

$$F(v') \geq e^{-1}F(v'_{\text{peak}}) \geq \frac{F(v'_{\text{peak}})}{e}.$$

We can then set

$$J := I' \cap \left(v'_{\text{peak}} \pm \frac{\sigma\sqrt{\mu}}{6\sqrt{2}\ln\mu}\right).$$

Defined this way, J will clearly have length at least $\frac{\sigma\sqrt{\mu}}{6\sqrt{2}\ln\mu}$. Indeed v'_{peak} is in I', and the length of I' is at least $2\sigma\sqrt{\mu} \geq 2 \times \frac{\sigma\sqrt{\mu}}{6\sqrt{2}\ln\mu}$. □

Now, recall from (I.17) that $p(v') \in [1,4]kF(v')$. Thus for any $v' \in I'$,

$$p(v') \leq 4kF(v') \leq 4kF(v'_{\text{peak}}). \tag{I.18}$$

But in particular, by Claim 7, for any $v' \in J$,

$$p(v') \geq kF(v') \geq \frac{kF(v'_{\text{peak}})}{e}. \tag{I.19}$$

Besides, being a probability distribution, $p(v')$ must sum up to 1, so we have

$$1 = \sum_{v' \in I'} p(v') \geq \sum_{v' \in J} p(v') \overset{\text{(I.19)}}{\geq} |J| \frac{kF(v'_{\text{peak}})}{e},$$

thus $kF(v'_{\text{peak}}) \leq \frac{e}{|J|}$. Therefore, for any $v' \in I'$,

$$p(v') \overset{\text{(I.18)}}{\leq} 4kF(v'_{\text{peak}}) \leq \frac{4e}{|J|} \leq \frac{24e\sqrt{2}\ln\mu}{\sigma\sqrt{\mu}} \leq \frac{100\ln\mu}{\sigma\sqrt{\mu}}. \tag{I.20}$$

We are finally ready to prove the lemma. For $v \in I$, we can observe that

$$\begin{aligned}
&\Pr[v^* = v \mid E_{l,r,a_l,a_r}] \\
&\quad = \Pr[v^* = v \mid v^* \in [l,r] \wedge \text{ans}_w(v^*,l) = a_l \wedge \text{ans}_w(v^*,r) = a_r] \\
&\quad \leq \Pr[v^* = v \mid v^* \in I' \wedge \text{ans}_w(v^*,l) = a_l \wedge \text{ans}_w(v^*,r) = a_r] && \text{(strengthen the condition)} \\
&\quad = p(v) && \text{(defined in (I.15))} \\
&\quad \leq \frac{100\ln\mu}{\sigma\sqrt{\mu}}. \quad \square && (v \in I \subset I' \text{ and (I.20)})
\end{aligned}$$

With Lemma 8 in hand, we are finally ready to prove Lemma 3.

Proof of Lemma 3. First, we show that $j < j_{\text{stop}}$ and A_j imply that $\mu, \mu' \geq D$. Indeed, $j < j_{\text{stop}}$ implies $\lambda_j, \lambda_{j+1} \geq D$. If in addition A_j holds, then $\mu = \mu_j \geq \lambda_j \geq D$, and

$$\mu' = \frac{\sigma\sqrt{\mu}}{400\ln\mu} \geq \frac{\sigma\sqrt{\lambda_j}}{400\ln\lambda_j} > \frac{\sigma\sqrt{\lambda_j}}{400\ln\lambda_j\log n} = \lambda_{j+1} \geq D.$$

Therefore, the assumptions of Lemma 8 hold.

We need to prove that for any $v \in V$,

$$\Pr[T \wedge (v^* = v) \mid K_j] \leq \frac{1}{\left(\frac{8}{3}\lambda_{j+1} + 1\right)\log n}.$$

If $v \notin I$, then by definition of I, $\Pr[T \wedge (v^* = v) \mid K_j] = 0$, so the inequality holds trivially. On the other hand, if $v \in I$,

$$\begin{aligned}
&\Pr[T \wedge (v^* = v) \mid K_j] \\
&\qquad \leq \Pr[v^* = v \mid K_j] \\
&\qquad = \Pr[v^* = v \mid E_{l,r,a_l,a_r}] && \text{(by Lemma 7)} \\
&\qquad \leq \frac{100\ln\mu}{\sigma\sqrt{\mu}} && \text{(by Lemma 8)} \\
&\qquad \leq \frac{100\ln\lambda_j}{\sigma\sqrt{\lambda_j}} \\
&\qquad = \frac{1}{4\lambda_{j+1}\log n} \\
&\qquad \leq \frac{1}{\left(\frac{8}{3}\lambda_{j+1} + 1\right)\log n}. \quad \square && (\lambda_{j+1} \geq D \geq e^2)
\end{aligned}$$

Appendix J. Proof of Lemma 6

Lemma 6. *For* $n \geq \Theta_p(\max(\sigma^3, 1))$, *we have*

$$j_{\text{stop}} + 1 = \begin{cases} \Omega_p(1 + \log(1 + \log_{1/\sigma} n)) \textit{ if } \sigma^2 \leq 1/2 \\ \Omega_p(\log\log n) \textit{ if } \sigma^2 \geq 1/2. \end{cases}$$

This proof is very similar in spirit to the proof of Claim 3 (Appendix F).

Proof of Lemma 6. We track the value of λ_j/σ^2 as j increases. First, as long as

$$\frac{\lambda_j}{(\ln \lambda_j)^6} \geq \sigma^2 (400 \log n)^6, \tag{J.1}$$

we have

$$\frac{\lambda_{j+1}}{\sigma^2} = \frac{\text{reduce}_{n,\sigma}(\lambda_{j+1})}{\sigma^2} = \frac{\sigma\sqrt{\lambda_j}}{400\sigma^2 \ln \lambda_j \log n} = \sqrt{\frac{\lambda_j}{\sigma^2}} \overset{\text{(J.1)}}{\geq} \left(\frac{\lambda_j}{\sigma^2}\right)^{1/3}.$$

Let $\lambda_{\min}$ be the smallest possible value for λ_j at least as large as D such that (J.1) holds. Then, by induction, as long as $\lambda_j \geq \lambda_{\min}$, we have

$$\frac{\lambda_j}{\sigma^2} \geq \left(\frac{\lambda_0}{\sigma^2}\right)^{1/3^j} \Rightarrow \lambda_j \geq \sigma^2 \left(\frac{n}{C\sigma^2}\right)^{1/3^j}. \tag{J.2}$$

Recall that $j_{\min}$ is the smallest integer $j \geq 0$ such that $\lambda_j < D$. Let j^* be the smallest integer $j \geq 0$ such that $\lambda_j < \lambda_{\min}$. Then $j^* \leq j_{\min}$, and applying (J.2) to j^* we get

$$\sigma^2 \left(\frac{n}{C\sigma^2}\right)^{1/3^{j^*}} < D \Rightarrow j_{\min} \geq j^* > \log_3 \left(\frac{\log\left(\frac{n}{C\sigma^2}\right)}{\log\left(\frac{\lambda_{\min}}{\sigma^2}\right)}\right). \tag{J.3}$$

Since $\lambda_0 = n/C$, for $n \geq CD = O_p(\sigma^2 \ln \sigma)$, we have $j_{\min} \geq 1$. We separate into cases to obtain more lower bounds of $j_{\min}$. First, if $\sigma^2 \leq 1/2$, it is easy to verify that $\lambda_{\min} = \max(D, (\log n)^{O(1)}) = (\log n)^{O_p(1)}$. Therefore, by (J.3), there exists $n_1 = O_p(1)$ such that for $n \geq n_1$, we have

$$j_{\min} \geq \Omega_p(\log(1 + \log_{1/\sigma} n)).$$

Second, if $\sigma^2 \geq 1/2$, it is easy to verify that $\lambda_{\min} = \max(D, \sigma^2(\log n)^{O(1)}) = \sigma^2(\log n)^{O_p(1)}$. Therefore, by (J.3), there exists $n_2 = O_p(\sigma^3)$ such that for $n \geq n_2$, we have

$$j_{\min} \geq \Omega_p(\log\log n).$$

In summary, there exists $n_3 = \max(n_1, n_2) = O_p(\max(\sigma^3, 1))$ such that for $n \geq n_3$,

$$j_{\min} = \begin{cases} \Omega_p(1 + \log(1 + \log_{1/\sigma} n)) \text{ if } \sigma^2 \leq 1/2 \\ \Omega_p(\log\log n) \text{ if } \sigma^2 \geq 1/2, \end{cases}$$

which is exactly the bounds we want for $j_{\text{stop}} + 1$. We can then conclude by observing that

$$j_{\text{stop}} + 1 = \min\left(j_{\min}, 1 + \left\lfloor \frac{p \log n}{2} \right\rfloor\right)$$

and that $1 + \left\lfloor \frac{p \log n}{2} \right\rfloor$ is larger than the claimed lower bounds for n large enough (which is covered in the $\Theta_p(\max(\sigma^3, 1))$ lower bound on n in the statement of the lemma). □

References

[1] Antonio Auffinger, Michael Damron, Jack Hanson, 50 years of first passage percolation, preprint, arXiv:1511.03262, 2015.

[2] Albert-László Barabási, Réka Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.

[3] Andrew C. Berry, The accuracy of the gaussian approximation to the sum of independent variates, Trans. Am. Math. Soc. 49 (1) (1941) 122–136.

[4] Sébastien Bubeck, Luc Devroye, Gábor Lugosi, Finding adam in random growing trees, Random Struct. Algorithms 50 (2) (2017) 158–172.

[5] Clément L. Canonne, Tom Gur, An adaptivity hierarchy theorem for property testing, Comput. Complex. 27 (4) (2018) 671–716.

[6] Fabrizio Carinci, Covid-19: Preparedness, Decentralisation, and the Hunt for Patient Zero, 2020.

[7] Xujin Chen, Changjun Wang, Approximability of the minimum weighted doubly resolving set problem, in: International Computing and Combinatorics Conference, Springer, 2014, pp. 357–368.

[8] Sung-En Chiu, Tara Javidi, Sequential measurement-dependent noisy search, in: 2016 IEEE Information Theory Workshop (ITW), IEEE, 2016, pp. 221–225.

[9] Quinlan Dawkins, Tianxi Li, Haifeng Xu, Diffusion source identification on networks with statistical confidence, in: Proceedings of the 38th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 2500–2509.

[10] Ehsan Emamjomeh-Zadeh, David Kempe, Vikrant Singhal, Deterministic and probabilistic binary search in graphs, in: Proceedings STOC16, 2016, pp. 519–532.

[11] Pál Erdős, Alfréd Rényi, On random graphs, Publ. Math. (Debr.) (1959).

[12] Carl-Gustav Esseen, et al., Fourier analysis of distribution functions. a mathematical study of the laplace-gaussian law, Acta Math. 77 (1945) 1–125.

[13] Giulia Fanti, Peter Kairouz, Sewoong Oh, Kannan Ramchandran, Pramod Viswanath, Metadata-conscious anonymous messaging, in: International Conference on Machine Learning, 2016, pp. 108–116.

[14] Giulia Fanti, Peter Kairouz, Sewoong Oh, Pramod Viswanath, Spy vs. Spy: Rumor Source Obfuscation, ACM SIGMETRICS Performance Evaluation Review, vol. 43, ACM, 2015, pp. 271–284.

[15] Giulia Fanti, Pramod Viswanath, Deanonymization in the bitcoin p2p network, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 1364–1373.

[16] Uriel Feige, Prabhakar Raghavan, David Peleg, Eli Upfal, Computing with noisy information, SIAM J. Comput. 23 (5) (1994) 1001–1018.

[17] Zhao-Long Hu, Zhesi Shen, Chang-Bing Tang, Bin-Bin Xie, Jian-Feng Lu, Localization of diffusion sources in complex networks with sparse observations, Phys. Lett. A 382 (14) (2018) 931–937.

[18] Nicholas E. Ingraham, David H. Ingbar, The omicron variant of sars-cov-2: understanding the known and living with unknowns, Clin. Transl. Med. 11 (12) (2021) e685.

[19] Mahmoud Kandeel, Maged Elsayed Mohamed Mohamed, Hany M. Abd El-Lateef, Katharigatta N. Venugopala, Hossam S. El-Beltagi, Omicron variant genome evolution and phylogenetics, J. Med. Virol. (2021).

[20] Richard M. Karp, Robert Kleinberg, Noisy binary search and its applications, in: Proceedings SODA, 2007, pp. 881–890.

[21] Justin Khim, Po-Ling Loh, Confidence sets for the source of a diffusion in regular trees, IEEE Trans. Netw. Sci. Eng. 4 (1) (2016) 27–40.

[22] Younjin Kim, Mohit Kumbhat, Zoltán Lóránt Nagy, Balázs Patkós, Alexey Pokrovskiy, Máté Vizer, Identifying codes and searching with balls in graphs, Discrete Appl. Math. 193 (2015) 39–47.

[23] Júlia Komjáthy, Gergely Ódor, Metric dimension of critical galton–watson trees and linear preferential attachment trees, Eur. J. Comb. 95 (2021) 103317.

[24] Kai Kupferschmidt, Where Did 'weird'omicron Come from?, 2021.

[25] Anusha Lalitha, Nancy Ronquillo, Tara Javidi, Measurement dependent noisy search: the gaussian case, in: 2017 IEEE International Symposium on Information Theory (ISIT), IEEE, 2017, pp. 3090–3094.

[26] Xiang Li, Xiaojie Wang, Chengli Zhao, Xue Zhang, Dongyun Yi, Locating the source of diffusion in complex networks via gaussian-based localization and deduction, Appl. Sci. 9 (18) (2019) 3758.

[27] Lyuben Lichev, Dieter Mitsche, Pawel Pralat, Localization game for random geometric graphs, preprint, arXiv:2102.10352, 2021.

[28] Dieter Mitsche, Juanjo Rué, On the limiting distribution of the metric dimension for random forests, Eur. J. Comb. 49 (2015) 68–89.

[29] Mark Newman, Networks, Oxford University Press, 2018.

[30] Gergely Odor, Patrick Thiran, Sequential metric dimension for random graphs, J. Appl. Probab. 58 (4) (2021) 909–951, https://doi.org/10.1017/jpr.2021.16.

[31] Krzysztof Onak, Pawel Parys, Generalization of binary search: searching in trees and forest-like partial orders, in: Proceedings FOCS'06, IEEE, 2006, pp. 379–388.

[32] Robert Paluch, Łukasz G. Gajewski, Janusz A. Hołyst, Boleslaw K. Szymanski, Optimizing sensors placement in complex networks for localization of hidden signal source: a review, Future Gener. Comput. Syst. 112 (2020) 1070–1092.

[33] Robert Paluch, Xiaoyan Lu, Krzysztof Suchecki, Bolesław K. Szymański, Janusz A. Hołyst, Fast and accurate detection of spread source in large complex networks, Sci. Rep. 8 (1) (2018) 1–10.

[34] Robert Paluch, Krzysztof Suchecki, Janusz A. Hołyst, Locating the source of interacting signal in complex networks, preprint, arXiv:2012.02039, 2020.

[35] Mathew Penrose, et al., Random Geometric Graphs, Vol. 5, Oxford University Press, 2003.

[36] Valentin Vladimirovich Petrov, Sums of Independent Random Variables, Vol. 82, Springer Science & Business Media, 2012.

[37] P. Pinto, P. Thiran, M. Vetterli, Locating the source of diffusion in large-scale networks, Phys. Rev. Lett. 109 (2012).

[38] Shobhit Kumar Prasad, Aarti Karahda, Priti Singh, Rajiv Gupta, Role of mental health professionals in dealing with the stigma attached to covid-19, Gen. Psychiatry 33 (5) (2020).

[39] Alfréd Rényi, On a problem of information theory, MTA Matematikai Kutato Intezet Kozlemeny B 6 (1961) 505–516.

[40] Lucia Russo, Cleo Anastassopoulou, Athanasios Tsakris, Gennaro Nicola Bifulco, Emilio Fortunato Campana, Gerardo Toraldo, Constantinos Siettos, Tracing day-zero and forecasting the covid-19 outbreak in lombardy, italy: a compartmental modelling and numerical optimization approach, PLoS ONE 15 (10) (2020) e0240649.

[41] Suzanne M. Seager, A sequential locating game on graphs, Ars Comb. 110 (2013).

[42] Burr Settles, Active learning literature survey, Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[43] Devavrat Shah, Tauhid Zaman, Rumors in a network: who's the culprit?, IEEE Trans. Inf. Theory 57 (8) (2011) 5163–5181.

[44] Zhesi Shen, Shinan Cao, Wen-Xu Wang, Zengru Di, H. Eugene Stanley, Locating the source of diffusion in complex networks by time-reversal backward spreading, Phys. Rev. E 93 (3) (2016) 032301.

[45] Peter J. Slater, Leaves of trees, Congr. Numer. 14 (549–559) (1975) 37.

[46] Brunella Spinelli, Code for the paper: back to the source: an online approach forsensor placement and source localization, https://github.com/bmspinelli/back_to_the_source, 2018. (Accessed 21 December 2021).

[47] Brunella Spinelli, Elisa Celis, Patrick Thiran, A general framework for sensor placement in source localization, IEEE Trans. Netw. Sci. Eng. (2017).

[48] Brunella Spinelli, L. Elisa Celis, Patrick Thiran, Back to the source: an online approach for sensor placement and source localization, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 1151–1160.

[49] Brunella Spinelli, L. Elisa Celis, Patrick Thiran, The effect of transmission variance on observer placement for source-localization, Appl. Netw. Sci. 2 (1) (2017) 20.

[50] Brunella Spinelli, L. Elisa Celis, Patrick Thiran, How many sensors to localize the source? The double metric dimension of random networks, in: 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, 2018, pp. 1036–1043.
[51] Wenchang Tang, Feng Ji, Wee Peng Tay, Estimating infection sources in networks using partial timestamps, IEEE Trans. Inf. Forensics Secur. 13 (12) (2018) 3035–3049.
[52] John Tsitsiklis, Kuang Xu, Zhi Xu, Private sequential learning, in: Conference on Learning Theory, 2018, pp. 721–727.
[53] Roman Vershynin, Introduction to the Non-asymptotic Analysis of Random Matrices, Cambridge University Press, 2012.
[54] Yinglian Xie, Vyas Sekar, David A. Maltz, Michael K. Reiter, Hui Zhang, Worm origin identification using random moonwalks, in: 2005 IEEE Symposium on Security and Privacy (S&P'05), IEEE, 2005, pp. 242–256.
[55] Jiaming Xu, Kuang Xu, Dana Yang, Optimal query complexity for private sequential learning against eavesdropping, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 2296–2304.
[56] Kuang Xu, Query complexity of bayesian private learning, in: Advances in Neural Information Processing Systems, 2018, pp. 2431–2440.
[57] Shuaishuai Xu, Cong Teng, Yinzuo Zhou, Junhao Peng, Yicheng Zhang, Zi-Ke Zhang, Identifying the diffusion source in complex networks with limited observers, Phys. A, Stat. Mech. Appl. 527 (2019) 121267.
[58] Sabina Zejnilovic, Joao Gomes, Bruno Sinopoli, Network observability and localization of the source of diffusion based on a subset of nodes, in: Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on, IEEE, 2013, pp. 847–852.
[59] Sabina Zejnilović, João Gomes, Bruno Sinopoli, Sequential observer selection for source localization, in: Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on, IEEE, 2015, pp. 1220–1224.
[60] Sabina Zejnilović, Joño Gomes, Bruno Sinopoli, Sequential source localization on graphs: a case study of cholera outbreak, in: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE, 2017, pp. 1010–1014.
[61] Lin Zhou, Alfred O. Hero, Resolution limits for the noisy non-adaptive 20 questions problem, IEEE Trans. Inf. Theory 67 (4) (2021) 2055–2073.
[62] Kai Zhu, Zhen Chen, Lei Ying, Locating the contagion source in networks with partial timestamps, Data Min. Knowl. Discov. 30 (5) (2016) 1217–1248.