MIT Open Access Articles

Ant-inspired density estimation via random walks

The MIT Faculty has made this article openly available. ***Please share*** how this access benefits you. Your story matters.

Citation: Musco, Cameron et al. “Ant-Inspired Density Estimation via Random Walks.” Proceedings of the National Academy of Sciences 114, 40 (September 2017): 10534–10541 © 2017 National Academy of Sciences

As Published: http://dx.doi.org/10.1073/PNAS.1706439114

Publisher: National Academy of Sciences (U.S.)

Persistent URL: http://hdl.handle.net/1721.1/114568

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.

Ant-inspired density estimation via random walks

Cameron Musco[a,1], Hsin-Hao Su[a,1], and Nancy A. Lynch[a,1,2]

[a]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139

This contribution is part of the special series of Inaugural Articles by members of the National Academy of Sciences elected in 2016.

Contributed by Nancy A. Lynch, August 14, 2017 (sent for review April 18, 2017; reviewed by Yehuda Afek, Ziv Bar-Joseph, and Amos Korman)

Many ant species use distributed population density estimation in applications ranging from quorum sensing, to task allocation, to appraisal of enemy colony strength. It has been shown that ants estimate local population density by tracking encounter rates: The higher the density, the more often the ants bump into each other. We study distributed density estimation from a theoretical perspective. We prove that a group of anonymous agents randomly walking on a grid are able to estimate their density within a small multiplicative error in few steps by measuring their rates of encounter with other agents. Despite dependencies inherent in the fact that nearby agents may collide repeatedly (and, worse, cannot recognize when this happens), our bound nearly matches what would be required to estimate density by independently sampling grid locations. From a biological perspective, our work helps shed light on how ants and other social insects can obtain relatively accurate density estimates via encounter rates. From a technical perspective, our analysis provides tools for understanding complex dependencies in the collision probabilities of multiple random walks. We bound the strength of these dependencies using local mixing properties of the underlying graph. Our results extend beyond the grid to more general graphs, and we discuss applications to size estimation for social networks, density estimation for robot swarms, and random walk-based sampling for sensor networks.

population density estimation | random walk sampling | network exploration | ant colony algorithms | biological distributed algorithms

The ability to sense local population density is an important tool used by many ant species. When a colony of *Temnothorax* ants must relocate to a new nest, scouts search for potential nest sites, assess their quality, and recruit other scouts to high-quality locations. A high enough density of scouts at a potential new nest (a quorum threshold) triggers those ants to decide on the site and transport the rest of the colony there (2). When neighboring colonies of *Azteca* ants compete for territory, a high relative density of a colony's ants in a contested area will cause those ants to attack enemies in the area, while a low relative density will cause the colony to retreat (3). Varying densities of harvester ants successfully performing certain tasks such as foraging or brood care can trigger other ants to switch tasks, maintaining proper worker allocation in the colony (4, 5).

It has been shown that ants estimate density in a distributed manner, by measuring encounter rates (2, 6). As ants randomly walk around an area, if they bump into a larger number of other ants, this indicates a higher population density. By tracking encounters with specific types of ants, for example, successful foragers or enemies, ants can estimate more specific densities. This strategy allows each ant to obtain an accurate density estimate and requires very little communication: Ants must simply detect when they collide and do not need to perform any higher-level data aggregation.

Density Estimation on a Grid

We study distributed density estimation from a theoretical perspective. We model a colony of ants as a set of anonymous agents randomly placed on a 2D grid. Computation proceeds in rounds, with each agent stepping in a random direction in each round. A collision occurs when two agents reach the same position in the same round, and encounter rate is measured as the number of collisions an agent is involved in during a sequence of rounds divided by the number of rounds. Aside from collision detection, the agents have no other means of communication.

The intuition that encounter rate tracks density is clear. It is easy to show that, for a set of randomly walking agents, the expected encounter rate measured by each agent is exactly the density d—the number of agents divided by the grid size (see *Lemma 2*). However, it is unclear if encounter rate actually gives a good density estimate, that is, if the estimate is close to its expectation with high probability.

Consider agents positioned not on the grid but on a complete graph. In each round, each agent steps to a uniformly random position, and, in expectation, the number of other agents it collides with in this step is d. Since each agent chooses its new location uniformly at random in each step, collisions are essentially independent between rounds. The agents are effectively taking independent Bernoulli samples with success probability d, and, by a standard Chernoff bound, within $O\left(\log(1/\delta)/d\epsilon^2\right)$ rounds, each obtains a $(1 \pm \epsilon)$ multiplicative approximation to d with probability $1-\delta$.

On the grid graph, the picture is significantly more complex. If two agents are initially located near each other, they are more likely to collide via random walking. After a first collision, due to their proximity, they are likely to collide repeatedly in future rounds. Since the agents are anonymous, they cannot recognize repeat collisions, and, even if they could, it is unclear that it would help. On average, compared with the complete graph,

Significance

Highly complex distributed algorithms are ubiquitous in nature: from the behavior of social insect colonies and bird flocks, to cellular differentiation in embryonic development, to neural information processing. In our research, we study biological computation theoretically, combining a scientific perspective, which seeks to better understand the systems being studied, with an engineering perspective, which takes inspiration from these systems to improve algorithm design. In this work, we focus on the problem of population density estimation in ant colonies, demonstrating that extremely simple algorithms, similar to those used by ants, solve the problem with strong theoretical guarantees and have a number of interesting computational applications.

An extended abstract of this work has been previously published (1).

Author contributions: C.M., H.-H.S., and N.A.L. designed research; C.M., H.-H.S., and N.A.L. performed research; C.M. and H.-H.S. contributed new reagents/analytic tools; and C.M., H.-H.S., and N.A.L. wrote the paper.

Reviewers: Y.A., Tel-Aviv University; Z.B.-J., Carnegie Mellon University; and A.K., French National Center for Scientific Research.

The authors declare no conflict of interest.

See QnAs on page 10512.

[1]C.M., H.S., and N.L. contributed equally to this work.

[2]To whom correspondence should be addressed. Email: lynch@csail.mit.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1706439114/-/DCSupplemental.

agents collide with fewer individuals and collide multiple times with those individuals that they do encounter, making encounter rates a less reliable estimate of population density.

Mathematically speaking, on a graph with a fast mixing time (7), like the complete graph, each agent's location is only weakly correlated with its previous locations. This ensures that collisions are also weakly correlated between rounds, and encounter rate serves as a very accurate estimate of density. The grid graph, on the other hand, is slow mixing: Agent positions and hence collisions are highly correlated between rounds, lowering the accuracy of encounter-rate-based estimation.

Results

Surprisingly, despite the high correlation between collisions, we show that encounter rate-based density estimation on the grid is nearly as accurate as on the complete graph. After just $O\big(\log(1/\delta)\log\log(1/\delta)\log(1/d\epsilon)/d\epsilon^2\big)$ rounds, each agent's encounter rate is a $(1 \pm \epsilon)$ approximation to d with probability $1 - \delta$ (*Theorem 1*). This matches performance on the complete graph up to a $\log\log(1/\delta)\log(1/d\epsilon)$ factor.

Technically, to bound accuracy on the grid, we obtain moment bounds on the number of times that two randomly walking agents collide over a set of rounds (*Lemma 5*). These bounds also apply to the number of equalizations (returns to origin) of a single walk. While expected random walk hitting times, return times, and collision rates are well studied for many graphs, including grid graphs (7–9), higher moment bounds and high probability results are much less common.

Our moment bounds show that, while the grid graph is slow mixing, it has strong local mixing. That is, random walks tend to spread quickly over a local area and not repeatedly cover the same nodes, making random walk-based density estimation accurate. Significant work has focused on showing that random walk sampling is nearly as good as independent sampling for fast-mixing expander graphs (10, 11). We extend this type of analysis to slowly mixing graphs, showing that strong local mixing is sufficient in many applications.

The key to the local mixing property of the grid is an upper bound on the probability that two random walks starting from the same position recollide (or that a single random walk equalizes) after a certain number of steps (*Lemma 3*). We show that recollision probability bounds imply collision moment bounds on general graphs, and apply this technique to extend our results to d-dimensional grids, regular expanders, and hypercubes. We discuss applications of our bounds to the task of estimating the size of a social network using random walks (12), obtaining improvements over prior work for networks with relatively slow global mixing times but strong local mixing. We also discuss connections to density estimation by robot swarms and random walk-based sensor network sampling (13, 14).

Theoretical Model for Density Estimation

We consider a set of agents populating a 2D torus with A nodes (dimensions $\sqrt{A} \times \sqrt{A}$). At each time step, each agent has an associated ordered pair $position$, which gives its coordinates on the torus. We assume that A is large—larger than the area agents traverse over the runtimes of our algorithms. We believe the torus model successfully captures the dynamics of density estimation on a surface, while avoiding complicating factors of boundary behavior on a finite grid.

Initially, each agent is placed independently at a uniform random node in the torus. Computation proceeds in discrete, synchronous rounds. Each agent updates its position with a step chosen uniformly at random from $\{(0,1),(0,-1),(1,0),(-1,0)\}$ in each round. Of course, in reality, ants do not move via pure random walk; observed encounter rates seem to actually be lower than predicted by a pure random walk model (6, 15). However, we feel that our model sufficiently captures the highly random movement of ants while remaining tractable to analysis and applicable to ant-inspired random walk-based algorithms. Extending our work to more realistic models of ant movement would be an interesting next direction.

Aside from the ability to move in each round, agents can sense the number of agents other than themselves at their position at the end of each round, formally through the function $count(position)$. We say that two agents collide in round r if they have the same position at the end of the round. Outside of collision counting, agents have no means of communication. They are anonymous (cannot uniquely identify each other) and execute identical density estimation routines. A basic illustration of our model is depicted in Fig. 1.

The Density Estimation Problem

Let $(n+1)$ be the number of agents, and define population density as $d \overset{\text{def}}{=} n/A$. Each agent's goal is to estimate d to $(1 \pm \epsilon)$ accuracy with probability at least $1 - \delta$ for $\epsilon, \delta \in (0,1)$, that is, to return an estimate $\tilde{d}$ with $\mathbb{P}\left[\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d]\right] \geq 1 - \delta$. As a technicality, with $n+1$ agents, we define $d = n/A$ instead of $d = (n+1)/A$, for convenience of calculation. In the natural case, when n is large, the distinction is unimportant.

Local vs. Global Density. The problem described above requires estimating the global population density. We assume that agents are initially distributed uniformly at random on the torus, which is critical for fast global density estimation: When agents are uniformly distributed, the local density in a small radius around their starting position reflects the global density with good probability. Of course, in nature, ants are not typically uniformly distributed in the nest or surrounding areas. Additionally, they are often interested in estimating local population densities, e.g., in a new nest site when house-hunting (2) or around a nest entrance

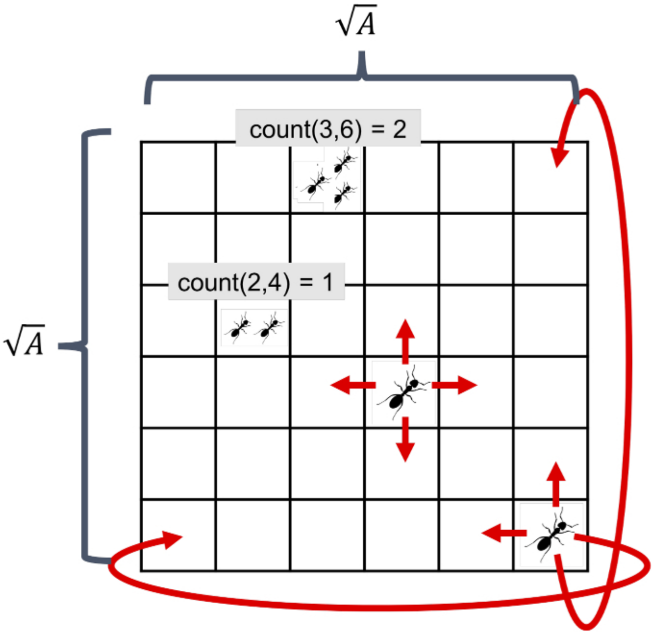

Fig. 1. A basic illustration of our computational model. Each agent (ant) may move to a random adjacent position on the 2D torus in each round (illustrated by the red arrows). A collision occurs when two or more agents are located at the same position. The agents detect collisions through the *count*(*position*) function, which returns the number of other agents at their current position. In this illustration, *position* is given as the (*x*, *y*) position, with the bottom left corner corresponding to (1, 1). However, the precise convention used is unimportant.

when estimating the number of successful foragers for task allocation (4).

We view our work as a first step toward a theoretical understanding of density estimation, and we focus on the global density for simplicity. Removing our assumption of uniformly distributed agents and understanding local density estimation are important directions for future work.

Random Walk-Based Density Estimation on the 2D Torus

As discussed, the challenge in analyzing random walk-based density estimation on the torus arises from correlations between collisions of nearby agents. If we do not restrict agents to random walking, and instead allow each agent to take an arbitrary step in each round, they can avoid collision correlations by splitting into "stationary" and "mobile" groups and counting collisions only between members of different groups. This allows them to essentially simulate independent sampling of grid locations to estimate density. This method is simple to analyze (*SI Appendix*, section S1), but it is not "natural" in a biological sense or useful for the applications we present. Further, independent sampling is unnecessary! *Algorithm 1* describes a simple random walk-based approach that gives a nearly matching bound.

Algorithm 1 Random-Walk-Based Density Estimation

```
Each agent independently executes:
  $c := 0$
  for $r = 1, ..., t$ do
      $step := rand\{(0,1), (0,-1), (1,0), (-1,0)\}$
      $position := position + step$
      $c := c + count(position)$          ▷ Update collision count.
  return $\tilde{d} = \frac{c}{t}$
```

Our main theoretical result follows; its proof appears at the end of this section, after a number of preliminary lemmas. Throughout our analysis, we take the viewpoint of a single agent executing *Algorithm 1*.

Theorem 1 (Random Walk Sampling Accuracy Bound). *After running for t rounds, assuming $t \leq A$, an agent executing* Algorithm 1 *returns $\tilde{d}$ such that, for any $\delta > 0$, with a probability of $\geq 1 - \delta$, $\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d]$ for $\epsilon = \Theta\left(\sqrt{\log(1/\delta)\log(2t)/td}\right)$. In other words, for any $\epsilon, \delta \in (0,1)$ if $t = \Theta\big(\log(1/\delta)\log\log(1/\delta)\log(1/d\epsilon)/d\epsilon^2\big)$, $\tilde{d}$ is a $(1 \pm \epsilon)$ multiplicative estimate of d with a probability of $\geq 1 - \delta$.*

Theorem 1 focuses on the density estimate of a single agent executing *Algorithm 1*. However, we note that, if we set $\delta = \delta'/n$, then, by a union bound, all n agents will have $\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d]$ with probability δ'. The required running time t will depend just logarithmically on δ' and n.

Correctness of Encounter Rate in Expectation. The first step in proving *Theorem 1* is to show that the encounter rate $\tilde{d}$ is an unbiased estimator of d. This result, in fact, holds for any ants randomly walking on any regular graph.

Lemma 2 (Unbiased Estimator). $\mathbb{E}\tilde{d} = d$.

Proof. We can decompose the collision bound c maintained by each agent in *Algorithm 1* as the sum of collisions with different agents over different rounds. Specifically, give the n other agents arbitrary labels $1, 2, ..., n$ and let $c_j(r)$ equal 1 if the agent collides with agent j in round r, and 0 otherwise. By linearity of expectation, $\mathbb{E}c = \sum_{j=1}^{n}\sum_{r=1}^{t}\mathbb{E}c_j(r)$.

Since each agent is initially at a uniform random location and, after any number of steps, is still at a uniform random location, for all j, r, $\mathbb{E}c_j(r) = 1/A$. Thus, $\mathbb{E}c = nt/A = dt$ and $\mathbb{E}\tilde{d} = \mathbb{E}c/t = d$.

We note that the torus is bipartite, and hence two agents initially located an odd number of steps away from each other will never meet via random walking. However, this fact does not change the expectation of $\tilde{d}$ computed above and, in fact, does not affect any of our following proofs.

With *Lemma 2* in place, it remains to show that the encounter rate is close to its expectation with high probability and so provides a good estimate of density. To do this, we must bound the strength of correlations between collisions of nearby agents in successive rounds, which can decrease the accuracy of the encounter rate-based estimate.

A Recollision Probability Bound. The key to bounding collision correlations is bounding the probability of a recollision between two agents in round $r + m$, assuming a collision in round r, which we do in this section.

Let $c_j = \sum_{r=1}^{t} c_j(r)$ be the total number of collisions with agent j. Due to the initial uniform distribution of the agents, the c_j are all independent and identically distributed.

Each c_j is the sum of highly correlated random variables; due to the slow mixing of the grid, if two agents collide at round r, they are much more likely to collide in successive rounds. However, by bounding this recollision probability, we are able to give strong moment bounds for the distribution of each c_j. We bound not only its variance but all higher moments. This allows us to show that the average $\tilde{d} = 1/t\sum_{j=1}^{n} c_j$ falls close to its expectation d with high probability, giving *Theorem 1*.

Lemma 3 (Recollision Probability Bound). *Consider two agents a_1 and a_2 randomly walking on a 2D torus of dimensions $\sqrt{A} \times \sqrt{A}$. If a_1 and a_2 collide in round r, for any $m \geq 0$, the probability that a_1 and a_2 collide again in round $r + m$ is $\Theta(1/m + 1) + O(1/A)$.*

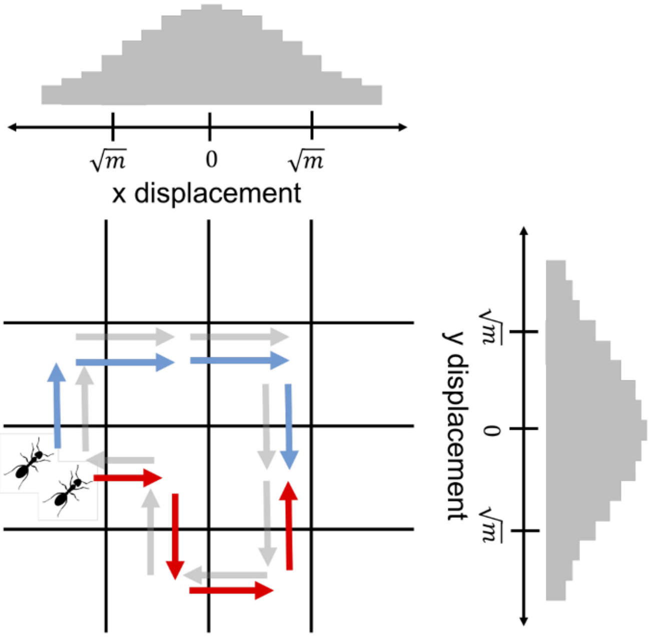

Fig. 2. A schematic of the proof of *Lemma 3*. We argue that the recollision probability of two agents after *m* steps (shown in red and blue) is equivalent to the probability that a length 2*m* random walk (shown in gray) returns to its origin. We then argue that the random walk is likely to take roughly *m* steps in both the *x* and *y* directions and hence has zero displacement in each direction with probability $\Theta(1/\sqrt{m})$.

Proof sketch. The full proof of *Lemma 3* is given in *SI Appendix*, section S2. We sketch the main ideas here and illustrate them in Fig. 2.

We first show that the probability that a_1 and a_2 recollide in round $r+m$ is identical to the probability that a single $2m$-step random walk ends at its starting position.

The recollision probability is the probability that a_1 and a_2 have identical displacements after taking m steps each. By symmetry of the random walk steps, this is equal to the probability that a_1's displacement vector is equal to the negative of a_2's. Furthermore, this is just the probability that their $2m$ total random walk steps have 0 overall displacement, which is the probability that a $2m$-step random walk ends at its origin.

One idea might be to bound this "equalization probability" using the global mixing time of the torus (7). After $\Theta(A \log A)$ steps, a random walk is nearly as likely to be at any node in the graph, including its origin. Thus, the equalization probability is bounded by $O(1/A)$ for $2m = \Omega(A \log A)$. Unfortunately, such a bound says nothing about this probability for small m.

Thus, we must take a different approach. We first assume, for simplicity, that the walk is on an infinite grid, and so there is no possibility of returning to its origin by "wrapping around" the torus. We later show that this only affects the equalization probability by an $O(1/A)$ factor.

Considering a walk on the infinite grid, we condition on the walk taking roughly m steps in both the x and y directions, which occurs with high probability. We separately bound the probability of zero displacement in each direction.

It is well known that an m-step random walk on the line has roughly equal probability of ending at any point within radius $\Theta(\sqrt{m})$ of its origin. It thus has probability $\Theta(1/\sqrt{m})$ of ending at its origin. Fixing the number of steps in each direction, the walk's x and y displacements are independent. So, we can multiply the probabilities for each direction, giving the final bound of $\Theta(1/m+1)$ (we write $m+1$ in the denominator instead of m so that the formula holds for $m=0$.)

Since it may be of independent interest, in *Corollary 15* in *SI Appendix*, section S3, we restate the result of *Lemma 3* explicitly in terms of a bound on the probability that a single random walk returns to its origin (equalizes) after m steps.

Collision Moment Bound. With *Lemma 3* in hand, we can prove our collision moment bound, which we use to show that the number of collisions an agent sees concentrates strongly around its expectation. We first show that any agent is likely to collide with many other agents during the execution of *Algorithm 1*, rather than repeatedly colliding with just a few other agents. That is, the probability that an agent collides at least once with any given other agent is not too low.

Lemma 4 (First Collision Probability). *Assuming* $t \le A$, *for all* $j \in [1, \ldots, n]$, $\mathbb{P}[c_j \ge 1] = \Theta(t/A \log 2t)$.

Proof sketch. By *Lemma 3* and the assumption that $t \le A$, in t rounds, an agent expects to recollide with any agent it encounters $\sum_{m=0}^{t-1} \Theta(1/m+1) = \Theta(\log 2t)$ times. By *Lemma 2*, an agent expects to be involved in $dt = nt/A$ total collisions. So, accounting for recollisions, it expects to collide with $\Theta(nt/A \log 2t)$ unique individuals. By symmetry, its collision probability with any single individual is thus $\Theta(t/A \log 2t)$. A formal proof is given in *SI Appendix*, section S2.

Lemma 4 used that, by *Lemma 3*, an agent expects to collide $O(\log 2t)$ times with any other agent it encounters. We can, in fact, show that this bound is not just in expectation but extends to the higher moments of the collision distribution.

Lemma 5 (Collision Moment Bound). *For* $j \in [1, \ldots, n]$, *let* $\bar{c}_j \overset{\text{def}}{=} c_j - \mathbb{E}c_j$ *and assume* $t \le A$. *There is some fixed constant* w *such that, for any integer* $k \ge 2$,

$$\mathbb{E}\left[\bar{c}_j^k\right] \le \frac{tw^k}{A} \cdot k! \log^{k-1}(2t).$$

When $k=2$, *Lemma 5* gives a bound on the variance of c_j, which can be used to show that c_j falls close to its mean with good probability. By bounding the kth moment $\mathbb{E}[\bar{c}_j^k]$ for all k, we are able to show even stronger concentration results.

Proof sketch. Very roughly, we separately consider the simple case when $c_j = 0$ and the case when $c_j \ge 1$, whose probability is bounded in *Lemma 4*. In the latter case, we split c_j over rounds as $c_j = \sum_{r=1}^{t} c_j(r)$ and expand out,

$$\mathbb{E}[c_j^k] = \sum_{r_1=1}^{t} \sum_{r_2=1}^{t} \ldots \sum_{r_k=1}^{t} \mathbb{E}\left[c_j(r_1)c_j(r_2)\ldots c_j(r_k)\right]. \qquad \textbf{[1]}$$

$\mathbb{E}\left[c_j(r_1)c_j(r_2)\ldots c_j(r_k)\right]$ is just the probability that two agents collide in each of rounds $r_1, r_2, \ldots, r_k$. Assuming that $r_1 \le r_2 \le \ldots \le r_k$ and that there is a collision in round r_1, we can apply *Lemma 3* to bound this probability as $\le w^k/(r_2 - r_1 + 1)\ldots(r_k - r_{k-1} + 1)$ for some constant w.

Obtaining the theorem requires combining this bound with Eq. **1** and applying a number of careful rearrangements. However, the bound on $\mathbb{E}\left[c_j(r_1)c_j(r_2)\ldots c_j(r_k)\right]$ is the crux of the analysis. A full proof is in *SI Appendix*, section S2.

As with *Lemma 3*, the techniques used in *Lemma 5* can be applied to bounding the moments of the number of equalizations of a single random walk. See *Corollaries 16* and *17* in *SI Appendix*, section S3.

Correctness of Encounter Rate with High Probability. Armed with *Lemma 5*, we can finally show that $\sum_{j=1}^{n} c_j$ concentrates strongly about its expectation. Since $\tilde{d} = 1/t \sum_{j=1}^{n} c_j$, this is enough to prove the accuracy of encounter rate-based density estimation (*Algorithm 1*). We first restate *Lemma 5* using a standard "Bernstein condition" on the sum $\sum_{j=1}^{n} c_j$.

Corollary 6 (Bernstein Condition). *Assuming* $t \le A$,

$$\mathbb{E}\left[\left(\sum_{j=1}^{n} c_j - \mathbb{E}\left[\sum_{j=1}^{n} c_j\right]\right)^k\right] \le \frac{1}{2} k! \sigma^2 b^{k-2}$$

for all $k \ge 2$ *and some* $b = \Theta(\log 2t)$ *and* $\sigma^2 = \Theta(td \log 2t)$.

Proof. By *Lemma 5*, there exists some constant w such that, for $\sigma^2 = wt \log 2t/A$ and $b = w \log 2t$, $\bar{c}_j \overset{\text{def}}{=} c_j - \mathbb{E}c_j$ satisfies

$$\mathbb{E}\left[\bar{c}_j^k\right] \le \frac{1}{2} k! \sigma^2 b^{k-2}.$$

Since each c_j is independent,

$$\mathbb{E}\left[\left(\sum_{j=1}^{n} c_j - \mathbb{E}\left[\sum_{j=1}^{n} c_j\right]\right)^k\right] = \mathbb{E}\left[\left(\sum_{j=1}^{n} \bar{c}_j\right)^k\right]$$
$$= \sum_{j=1}^{n} \mathbb{E}[\bar{c}_j^k] \le \frac{n \cdot k! \sigma^2 b^{k-2}}{2}.$$

The lemma follows after replacing σ^2 with $n\sigma^2 = \Theta(td \log 2t)$.

We use the following concentration bound for random variables satisfying such a Bernstein condition.

Lemma 7. *Suppose that* X *satisfies* $\mathbb{E}[(X - \mathbb{E}X)^k] \le 1/2k!\sigma^2 b^{k-2}$ *for all* $k \ge 3$. *Then, for any* $\Delta \ge 0$, $\mathbb{P}[|X - \mathbb{E}X| \ge \Delta] \le 2e^{-\Delta^2/2(\sigma^2 + b\Delta)}$.

We conclude this section by proving our main theorem on the accuracy of random walk-based density estimation.

Proof of Theorem 1. In *Algorithm 1*, $\tilde{d}$ is set to $1/t\sum_{j=1}^{n} c_j$. So the probability that $\tilde{d}$ falls within an ϵ multiplicative factor of its mean is the same as the probability that $\sum_{j=1}^{n} c_j$ falls within an ϵ multiplicative factor of its mean, which is equal to $t\mathbb{E}\tilde{d} = td$ by *Lemma 2*. By *Corollary 6* and *Lemma 7*,

$$\delta \overset{\text{def}}{=} \mathbb{P}\left[\left|\sum_{j=1}^{n} c_j - \mathbb{E}\left[\sum_{j=1}^{n} c_j\right]\right| \geq \epsilon\mathbb{E}\left[\sum_{j=1}^{n} c_j\right]\right]$$

$$= \mathbb{P}\left[\left|\sum_{j=1}^{n} c_j - td\right| \geq \epsilon td\right] \leq 2e^{\Theta\left(-\frac{\epsilon^2 t^2 d^2}{2(td\log 2t + \epsilon td \log 2t)}\right)}.$$

Restricting $\epsilon \leq 1$ and rearranging gives $\epsilon^2 td/\log 2t = \Theta\left(\log(1/\delta)\right)$ and so $\epsilon = \Theta\left(\sqrt{\log(1/\delta)\log 2t/td}\right)$, yielding the theorem.

Extensions to Other Topologies

We now discuss extensions of our results to a broader set of graph topologies, demonstrating the generality of our local mixing analysis. We illustrate divergence between local and global mixing properties, which can have significant effects on random walk-based algorithms. Full proofs for all results in this section are deferred to *SI Appendix*, section S4.

From Recollision Bounds to Accurate Density Estimation. Our proofs for the 2D torus are largely independent of graph structure, using just a recollision probability bound (*Lemma 3*) and the regularity (uniform node degrees) of the grid, so agents remain uniformly distributed on the nodes in each round (see, for example, *Lemma 2*). Hence, extending our results to other regular graphs primarily involves obtaining recollision probability bounds for these graphs.

We consider agents on a graph with A nodes that execute analogously to *Algorithm 1*, stepping to a random neighbor in each round. Again, we focus on the multiagent case, but similar bounds (resembling *Corollaries 16* and *17* in *SI Appendix*, section S3) hold for a single random walk. We start with a lemma which gives density estimation accuracy in terms of recollision probability. This is a direct generalization of our grid analysis.

Lemma 8 (Recollision Probability to Density Estimation Accuracy). Consider a regular graph with A nodes such that, if two randomly walking agents a_1 and a_2 collide in round r, for any $0 \leq m \leq t$, the probability that they collide again in round $r+m$ is $\Theta\left(\beta(m)\right)$ for some nonincreasing function $\beta(m)$. Let $B(t) \overset{\text{def}}{=} \sum_{m=0}^{t} \beta(m)$. After running for $t \leq A$ steps, *Algorithm 1* returns $\tilde{d}$ such that, for any $\delta > 0$, with a probability of $\geq 1-\delta$, $\tilde{d} \in [(1-\epsilon)d, (1+\epsilon)d]$ for $\epsilon = O\left(\sqrt{\log(1/\delta)B(t)/td}\right)$.

Note that, in the special case of the 2D torus, by *Lemma 3*, we can set $\beta(m) = 1/(m+1)$ and hence $B(t) = \Theta(\log 2t)$, recovering *Theorem 1*.

Density Estimation on *k*-Dimensional Tori. We first consider k-dimensional tori for general k. As k increases, local mixing becomes stronger, fewer recollisions occur, and density estimation becomes easier. In fact, for constant $k \geq 3$, although the torus still mixes slowly, density estimation is as accurate as on the complete graph! Throughout this section, we assume that k is a small constant and so hide multiplicative factors in $f(k)$ for any function f in our asymptotic notation. We subscript the notation with k to make this clear. We begin with the case of $k = 1$.

Lemma 9 (Recollision Probability Bound: Ring). *If two randomly walking agents a_1 and a_2 are located on a 1D torus (a ring) with A nodes, and collide in round r, for any $m \geq 0$, the probability that a_1 and a_2 collide again in round $r+m$ for $k \geq 1$ is $\Theta\left(1/\sqrt{m+1}\right) + O\left(1/A\right)$.*

Proof sketch. This bound can be shown similarly to *Lemma 3* (and, in fact, its proof is fully contained in the proof of *Lemma 3*.) A $2m$-step random walk on a line ends at its origin with probability $\Theta(1/\sqrt{m+1})$. On a ring with A nodes, the slightly weaker bound of $\Theta\left(1/\sqrt{m+1}\right) + O\left(1/A\right)$ holds.

For $m \leq A$, the $O\left(1/A\right)$ term is absorbed into the $\Theta\left(1/\sqrt{m+1}\right)$, and one can show that $\sum_{m=0}^{t} 1/\sqrt{m+1} = \Theta(\sqrt{t})$. Plugging into *Lemma 8*, on a ring, random walk-based density estimation gives $\epsilon = O\left(\sqrt{\log(1/\delta)\sqrt{t}/td}\right) = O\left(\sqrt{\log(1/\delta)/\sqrt{t}d}\right)$. Rearranging, $t = \Theta\left(\left(\log(1/\delta)/\epsilon^2 d\right)^2\right)$ rounds are necessary to obtain a $1 \pm \epsilon$ approximation with a probability of $\geq 1-\delta$ for any $\epsilon, \delta \in (0,1)$. Local mixing on the ring is much worse than on the torus. Hence, density estimation is much more difficult, requiring t to be quadratic rather than linear in $1/d$ and $1/\epsilon^2$.

We now cover $k \geq 3$. While global mixing time is on the order of $A^{2/k}$, local mixing is so strong that our accuracy bounds nearly match those of independent sampling.

Lemma 10 (Recollision Probability Bound: High-Dimensional Torus). *If two randomly walking agents a_1 and a_2 are located on a k-dimensional torus with A nodes, and collide in round r, for any constant $k \geq 3$, $m \geq 0$, the probability that a_1 and a_2 collide in round $r+m$ is $\Theta_k\left(1/(m+1)^{k/2}\right) + O\left(1/A\right)$.*

Proof sketch. The proof is similar to that of *Lemma 3*. To collide in round $r+m$, the agents must have identical displacements in each of the k dimensions after m steps. Since k is a small constant, with high probability, the agents take $\Theta(m/k)$ steps in each dimension. After conditioning on the step counts, the k collisions are independent, each occurring with probability $\Theta\left(1/\sqrt{m/k}\right)$ via the argument of *Lemma 3*. The result follows by multiplying these k probabilities together, noting that k dependence is hidden in the asymptotic notation.

To convert the above bound to a density estimation accuracy, we can use a slightly modified version of *Lemma 8*, which applies to the case when our collision probability is $O(\beta(m))$ but not neccesarily $\Theta(\beta(m))$. For $t \leq A$ and $k \geq 3$, $\sum_{m=0}^{t}(1/(m+1)^{k/2} + 1/A) < 1 + \sum_{m=0}^{\infty} 1/(m+1)^{k/2} = O(1)$. So we can set $B(t) = O_k(1)$ and have $\epsilon = O_k\left(\sqrt{\log(1/\delta)/td}\right)$. Rearranging, we require $t = \Theta_k\left(\log(1/\delta)/\epsilon^2 d\right)$. This matches independent sampling up to constants and multiplicative factors in k.

Density Estimation on Regular Expanders. When a graph does mix well globally, it mixes well locally. An obvious example is the complete graph, on which random walk-based and independent sampling-based density estimations are equivalent. We extend this intuition to any regular expander. An expander is a graph whose random walk matrix has its second eigenvalue bounded away from 1, and so on which random walks mix quickly. Expanders are "well-connected" graphs with many applications, including in the design of robust communication networks (16) and efficient sampling schemes (10).

Lemma 11 (Recollision Probability Bound: Regular Expander). *Let G be a k-regular expander with A nodes and adjacency matrix $\mathbf{M}$. Let $\mathbf{W} = 1/k \cdot \mathbf{M}$ be its random walk matrix, with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_A$. Let $\lambda = \max\{|\lambda_2|, |\lambda_A|\} < 1$. If two randomly walking agents a_1 and a_2 collide in round r, for any $m \geq 0$, the probability that they collide again in round $r+m$ is, at most, $\lambda^m + 2/A$.*

Proof sketch. The bound follows from noting that the stable distribution on a regular expander is uniform, and the location distribution of any agent after m steps converges exponentially quickly to this distribution, with rate λ.

Again, we bound density estimation accuracy via a modification of *Lemma 8*, which applies when we have collision probability $O(\beta(m))$ but not necessarily $\Theta(\beta(m))$. This modified lemma gives a $B(t)^2$ dependence. $B(t) = \sum_{m=0}^{t} \beta(m) \leq 1/1 - \lambda + 2t/A$. Assuming $t = O(A)$, $\epsilon = O\left(\sqrt{\log(1/\delta)/td(1-\lambda)^2}\right)$. Rearranging, $t = \Theta\left(\log(1/\delta)/\epsilon^2 d(1-\lambda)^2\right)$, matching independent sampling up to a factor of $O(1/(1-\lambda)^2)$.

Density Estimation *k*-Dimensional Hypercubes. Finally, we give bounds for a k-dimensional hypercube. Such a graph has $A = 2^k$ vertices mapped to the elements of $\{\pm 1\}^k$, with an edge between any two vertices that differ by Hamming distance 1. The hypercube is relatively fast mixing. Its adjacency matrix eigenvalues are $[-k, -k+2, \ldots, k-2, k]$. Since it is bipartite, we can ignore the negative eigenvalues: To return to its origin, a random walk must take an even number of steps, so we need only need to consider the squared walk matrix $\mathbf{W}^2$, which has all positive eigenvalues. Applying *Lemma 11* with $\lambda = \Theta(1 - 2/k) = \Theta(1 - 1/\log A)$ gives $t = \Theta\left(\log(1/\delta)\log^2(A)/\epsilon^2 d\right)$. However, it is possible to remove the dependence on A via a more refined analysis: While the global mixing time of the graph increases as A grows, local mixing becomes stronger!

Lemma 12 (Recollision Probability Bound: *k*-Dimensional Hypercube). *If two randomly walking agents a_1 and a_2 are located on a k-dimensional hypercube with $A = 2^k$ vertices and collide in round r, for any $m \geq 0$, the probability that a_1 and a_2 collide in round $r + m$ is $O\left((7/10)^m + 1/\sqrt{A}\right)$.*

Converting to a density estimation bound, we have $B(t) = \sum_{m=0}^{t} \beta(m) \leq 10/3 + t/\sqrt{A}$. If we assume $t = O(\sqrt{A})$, this gives $\epsilon = O\left(\sqrt{\log(1/\delta)/td}\right)$ and so $t = \Theta\left(\log(1/\delta)/\epsilon^2 d\right)$, matching independent sampling.

Applications

We conclude by discussing algorithmic applications of our ant-inspired density estimation algorithm (*Algorithm 1*), variations on this algorithm, and the analysis techniques we have developed.

Social Network Size Estimation. Random walk-based density estimation is closely related to work on estimating the size of social networks and other massive graphs using random walks (12, 17–19). In these applications, one does not have access to the full graph (so cannot exactly count the nodes) but can simulate random walks by following links between nodes (20, 21). One approach is to run a single random walk and count repeat node visits (17, 18). Alternatively, ref. 12 proposes running multiple random walks and counting their collisions, which gives an estimate of the walk's density. Since the number of walks is known, this yields an estimate for network size.

This approach can be significantly more efficient, since the dominant cost is typically in link queries to the network. With multiple, shorter random walks, this cost can be trivially distributed to multiple servers simulating walks independently. Visit information can then be aggregated, and the collision count can be computed in a centralized manner.

Random walk-based algorithm for network size estimation. Consider an undirected, connected, nonbipartite graph $G = (V, E)$. Let S be the set of vertices of G that are "known." Initially, $S = \{v\}$, where v is a seed vertex. We can access G by looking up the neighborhood $\Gamma(v_i)$ of any vertex $v_i \in S$ and adding $\Gamma(v_i)$ to S.

To compute the network size $|V|$, we could scan S, looking up the neighbors of each vertex and adding them to the set. Repeating this process until no new nodes are added ensures that $S = V$ and we know the network size. However, this method requires $|V|$ neighborhood queries. The goal is to use significantly fewer queries using random walk-based sampling.

A number of challenges are introduced by this application. While we can simulate many random walks on G, we can no longer assume these random walks start at randomly chosen nodes, as we do not have the ability to uniformly sample nodes from the network. Instead, we must allow the random walks to run for a burn-in phase of length proportional to the mixing time of G. After this phase, the walks are distributed approximately according to the stable distribution of G.

Further, in general, G is not regular. In the stable distribution, a random walk is located at a vertex with probability proportional to its degree. Hence, collisions tend to occur more at higher-degree vertices. To correct for this bias, we count a collision at vertex v_i with weight $1/\deg(v_i)$.

Our results depend on a natural generalization of recollision probability. For any i, j, let $p(v_i, v_j, m)$ be the probability that an m-step random walk starting at v_i ends at v_j. Define

$$\beta(m) \overset{\text{def}}{=} \frac{\max_{i,j} p(v_i, v_j, m)}{\deg(v_j)}.$$

Intuitively, $\beta(m)$ is the maximum m-step collision probability, weighted by degree since higher-degree vertices are visited more in the stable distribution. Let $B(t) = \sum_{m=1}^{t} \beta(m)$. Note that this weighted $B(t)$ is trivially upper bounded by the unweighted measure used in *Lemma 8*.

For simplicity, we initially ignore burn-in and assume that our walks start distributed exactly by the stable distribution of G. A walk starts at vertex v_i with probability $p_i \overset{\text{def}}{=} \deg(v_i)/\sum_i \deg(v_i) = \deg(v_i)/2|E|$, and initial locations are independent. We also assume knowledge of the average degree $\overline{\deg} = 2|E|/|V|$. See *SI Appendix*, section S5 for a rigorous analysis of burn-in and average degree estimation.

Algorithm 2 Random-Walk-Based Network Size Estimation

```
input: step count $t$, average degree $\overline{\deg}$, $n$ random starting locations $[w_1, \ldots, w_n]$ distributed independently according to the network's stable distribution
  $[c_1, \ldots, c_n] := [0, 0, \ldots, 0]$
  for $r = 1, \ldots, t$ do
      $\forall j$, set $w_j := randomElement(\Gamma(w_j))$  ▷ $\Gamma(w_j)$ denotes the neighborhood of $w_j$.
      $\forall j$, set $c_j := c_j + \frac{count(w_j)}{\deg(w_j)}$  ▷ $count(w_j)$ returns the number of other walkers currently at $w_j$.
  $C := \frac{\overline{\deg}\sum c_j}{n(n-1)t}$
  return $\tilde{A} = 1/C$
```

Note that there are many ways to implement the $count(\cdot)$ function used in *Algorithm 2*. One possibility is to simulate the random walks in parallel, recording their paths, and then to perform centralized postprocessing to count collisions. As queries to the network are considered to dominate time cost, this collision counting step is relatively inexpensive.

We prove the following theorem in *SI Appendix*, section S5.

Theorem 13. *If* Algorithm 2 *is run using n random walks for t steps, as long as $n^2 t = \Theta\left(B(t)\overline{\deg} + 1/\epsilon^2\delta \cdot |V|\right)$, then, with probability at least $1 - \delta$, it returns $\tilde{A} \in [(1-\epsilon)|V|, (1+\epsilon)|V|]$.*

Proof sketch. The proof is similar to that of *Theorem 1*. It is not hard to see that, due to our reweighting of each collision by $1/\deg(w_j)$, $\mathbb{E}C = 1/|V|$. The challenge is showing that C concentrates around its expectation and hence $\tilde{A} = 1/C$ is close to $|V|$. Due to the complicating factors of nonuniform degree, we are unable to compute a general moment bound for each c_j as done in *Lemma 5*. However, we can give a variance bound on C, and bound its deviation via Chebyshev's inequality. This gives a worse dependence on the failure probability: $1/\delta$ instead of $\log(1/\delta)$. We note that this can be improved by running the algorithm $\log(1/\delta)$ times, each with success probability $1/3$, and taking the median of the results.

Overall runtime and comparison to previous work. Let M denote the burn-in time required before running *Algorithm 2* (see *SI Appendix,* section S5 for details). To obtain a $(1 \pm \epsilon)$ estimate of network size with probability $1 - \delta$, we must run n random walks for $M + t$ steps, making $n(M + t)$ link queries, where, by *Theorem 13* and our analysis of average degree estimation in *SI Appendix,* section S5, we have

$$n = \Theta\left(\max\left\{\frac{\overline{\deg}}{\deg_{\min}\epsilon^2\delta}, \sqrt{\frac{|V| \cdot (B(t)\,\overline{\deg} + 1)}{t \cdot \epsilon^2\delta}}\right\}\right). \quad [2]$$

Typically, the second term dominates, since $\overline{\deg} << |V|$. Hence, by increasing t, we are able to use fewer random walks, significantly decreasing the number of link queries if M is large.

Ref. 12 uses a different approach, halting random walks and counting collisions immediately after burn-in. For reasonable node degrees, they require $n = \Theta\left(|V| \cdot \overline{\deg}/\epsilon^2\delta \cdot \sqrt{\sum \deg(v_i)^2}\right)$. Assuming that $\sqrt{\sum \deg(v_i)^2} < n$, and setting $t = 1$, this is somewhat smaller than our bound, as $\sum \deg(v_i)^2 \geq |V| \cdot \overline{\deg}$. However, Eq. **2** gives an important tradeoff: By increasing t, we can increase the number of steps in our random walks, decreasing the total number of walks.

As an illustrative example, consider a k-dimensional torus graph for $k \geq 3$ [for $k = 2$, mixing time is $\Theta(|V|)$, so we might as well census the full graph]. The mixing time required for *Algorithm 2* (see *SI Appendix,* section S5 for details) is $M = \Theta(\log(|V|/\delta)|V|^{2/k})$. All nodes have degree $2k$, and using the bounds above, to obtain a $(1 \pm \epsilon)$ estimate of $|V|$, the algorithm of ref. 12 requires $M \cdot n = \Theta\left(\log(|V|/\delta)/\epsilon\sqrt{d} \cdot |V|^{2/k+1/2}\right)$ link queries to obtain a size estimate. In contrast, assuming $|V|$ is large, we require $n = \Theta\left(\sqrt{|V|/t \cdot \epsilon^2\delta}\right)$, since, by *Lemma 10*, $B(t) = O(1/k)$ and $\overline{\deg} = \deg_{\min} = k$. If we set $t = \Theta(M)$, the total number of link queries needed is $n(M + t) = O\left(\sqrt{\log(|V|/\delta)}/\epsilon\sqrt{d} \cdot |V|^{(k+1)/2k}\right)$. This beats ref. 12 by improving dependence on $|V|$ and the logarithmic burn-in term. Ignoring error dependences, if $k = 3$, ref. 12 requires $\Theta(n^{7/6})$ queries, which is more expensive than fully censusing the graph. We require $O(n^{2/3})$ queries, which is sublinear in the graph size.

We leave open comparing our bounds with those of ref. 12 on more natural classes of graphs. It would be interesting to determine typical values of $B(t)$ in real work networks or popular graph models, such as preferential attachment models and others with power-law degree distributions.

Distributed Density Estimation by Robot Swarms. *Algorithm 1* can be directly applied as a simple and robust density estimation algorithm for robot swarms moving on a 2D plane modeled as a grid. Additionally, the algorithm can be used to estimate the frequency of certain properties within the swarm. Let d be the overall population density and d_P be the density of agents with some property P. Let $f_P = d_P/d$ be the relative frequency of P.

Assuming that agents with property P are distributed uniformly in population and that agents can detect this property (through direct communication or some other signal), then they can separately track encounters with these agents. They can compute an estimate $\tilde{d}$ of d and $\tilde{d}_P$ of d_P. By *Theorem 1*, after running for $t = \Theta\left(\log(1/\delta)\log\log(1/\delta)\log(1/d\epsilon)/d_P\epsilon^2\right)$ steps, with probability $1 - 2\delta$, $\tilde{d}_P/\tilde{d} \in [(1 - \epsilon/1 + \epsilon) f_P, (1 + \epsilon/1 - \epsilon) f_P] = [(1 - O(\epsilon))f_P, (1 + O(\epsilon))f_P]$ for small ϵ.

In an ant colony, properties may include whether an ant has recently completed a successful foraging trip (4), or if an ant is a nestmate or enemy (3). In a robotics setting, properties may include whether a robot is part of a certain task group, whether it has completed a certain task, or whether it has detected a certain event or environmental property.

Random Walk-Based Sensor Network Sampling. Finally, we believe our moment bounds for a single random walk (*Corollaries 16* and *17* in *SI Appendix,* section S3) can be applied to random walk-based distributed algorithms for sensor network sampling. We leave obtaining rigorous bounds in this domain to future work.

Random walk-based sensor network sampling (13, 14) is a technique in which a query message (a "token") is initially sent by a base station to some sensor. The token is relayed randomly between sensors, which are connected via a grid network, and its value is updated appropriately at each step to give an answer to the query. This scheme is robust and efficient; it easily adapts to node failures and does not require setting up or storing spanning tree communication structures.

Random walk-based sampling could be used, for example, to estimate the percentage of sensors that have recorded a specific condition, or the average value of some measurement at each sensor. However, as in density estimation, unless an effort is made to record which sensors have been previously visited, additional error is added due to repeat visits. Recording previous visits introduces computational burden: Either the token message size must increase or nodes themselves must remember which tokens they have seen. We are hopeful that our moment bounds can be used to show that this is unnecessary: Due to strong local mixing, the number of repeat sensor visits will be low, and the performance reduction limited.

We remark that estimating the percentage of sensors in a network or the density of robots in a swarm with a property that is uniformly distributed is a special case of a more general data aggregation problem: Each agent or sensor holds a value v_i drawn independently from some distribution $\mathcal{D}$. The goal is to estimate some statistic of $\mathcal{D}$, such as its expectation. In the case of density estimation, v_i is simply an indicator random variable which is 1 with probability d and 0 otherwise. Extending our results to more general data aggregation problems and showing that random walk sampling matches independent sampling in some cases is an interesting future direction.

Discussion and Future Work

We have presented a theoretical analysis of random walk-based density estimation by agents moving synchronously on a 2D torus graph. We have also presented applications of our techniques to density estimation on other simple graph topologies and to the problems of social network size estimation and density estimation on robot swarms.

Aside from using our bounds to study sensor network sampling and giving improved theoretical and empirical understanding of our social network size estimation algorithm, our work leaves open a number of questions related to modeling random walk-based density estimation in ant colonies.

We feel that our simple computational model well reflects the behavior of ants estimating density via collision rates while moving around a 2D surface. However, extending our results to more realistic models, e.g., with continuous movement along a surface which is either bounded or extends out indefinitely, is an interesting future direction.

As discussed, understanding how close actual ant movements are to random walks, and how nonrandom behavior influences density estimation via collision detection, is also important. In conjunction with this issue, removing our uniform density assumption and understanding how ants may estimate local population densities which may vary throughout the nest or surrounding area is an important direction.

Finally, we note that the accuracy bound of *Theorem 1* depends on the density d. In many applications, such as in quorum sensing, ants only need to detect when d is above some fixed threshold. In this case, better bounds, where t can be determined independently of the density, may be possible.

ACKNOWLEDGMENTS. We thank Amartya Shankha Biswas, Christopher Musco, and Mira Radeva for useful discussions. Research was supported by National Science Foundation (NSF) Grants BIO-1455983 and CCF-1461559, NSF Center for Science of Information Grant CCF-0939370, and Air Force Office of Scientific Research Grant FA9550-13-1-0042. C.M. is partially supported by an NSF graduate student fellowship.

1. Musco C, Su H-H, Lynch NA (2016) Ant-inspired density estimation via random walks. *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing* (Association for Computing Machinery, New York), pp 469–478.
2. Pratt SC (2005) Quorum sensing by encounter rates in the ant *Temnothorax albipennis*. *Behav Ecol* 16:488–496.
3. Adams ES (1990) Boundary disputes in the territorial ant *Azteca trigona*: Effects of asymmetries in colony size. *Anim Behav* 39:321–328.
4. Gordon DM (1999) Interaction patterns and task allocation in ant colonies. *Information Processing in Social Insects* (Springer, New York), pp 51–67.
5. Schafer RJ, Holmes S, Gordon DM (2006) Forager activation and food availability in harvester ants. *Anim Behav* 71:815–822.
6. Gordon DM, Paul RE, Thorpe K (1993) What is the function of encounter patterns in ant colonies? *Anim Behav* 45:1083–1100.
7. Lovász L (1993) Random walks on graphs: A survey. *Combinatorics, Paul Erdos Is Eighty* (Janos Bolyai Math Soc, Budapest), Vol 2, pp 1–46.
8. Elsässer R, Sauerwald T (2009) Tight bounds for the cover time of multiple random walks. *Automata, Languages and Programming* (Springer, New York), pp 415–426.
9. Kanade V, Mallmann-Trenn F, Sauerwald T (2016) On coalescence time in graphs–when is coalescing as fast as meeting? arXiv:161102460.
10. Gillman D (1998) A Chernoff bound for random walks on expander graphs. *SIAM J Comput* 27:1203–1220.
11. Chung KM, Lam H, Liu Z, Mitzenmacher M (2012) Chernoff-Hoeffding bounds for Markov chains: Generalized and simplified. arXiv:12010559.
12. Katzir L, Liberty E, Somekh O, Cosma IA (2014) Estimating sizes of social networks via biased sampling. *Internet Math* 10:335–359.
13. Avin C, Brito C (2004) Efficient and robust query processing in dynamic environments using random walk techniques. *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks* (Association for Computing Machinery, New York), pp 277–286.
14. Lima L, Barros J (2007) Random Walks on Sensor Networks. *5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007* (Inst Electr Electron Eng, New York), pp 1–5.
15. Nicolis SC, Theraulaz G, Deneubourg JL (2005) The effect of aggregates on interaction rate in ant colonies. *Anim Behav* 69:535–540.
16. Bassalygo L, Pinsker M (1973) The complexity of an optimal non-blocking commutation scheme without reorganization. *Problemy Peredaci Informacii* 9: 84–87.
17. Kurant M, Butts CT, Markopoulou A (2012) Graph size estimation. arXiv:1210.0460.
18. Lu J, Li D (2012) Sampling online social networks by random walk. *Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research* (Association for Computing Machinery, New York), pp 33–40.
19. Lu J, Wang H (2014) Variance reduction in large graph sampling. *Inform Process Manage* 50:476–491.
20. Mislove A, Marcon M, Gummadi KP, Druschel P, Bhattacharjee B (2007) Measurement and analysis of online social networks. *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement* (Association for Computing Machinery, New York), pp 29–42.
21. Gjoka M, Kurant M, Butts CT, Markopoulou A (2009) A walk in Facebook: Uniform sampling of users in online social networks. arXiv:09060060.