Corrected pair correlation functions for environments with obstacles

Stuart T. Johnston[1,2,*] and Edmund J. Crampin[1,2,3]

[1]*Systems Biology Laboratory, School of Mathematics and Statistics, and Department of Biomedical Engineering, University of Melbourne, Parkville, Victoria 3010, Australia*

[2]*ARC Centre of Excellence in Convergent Bio-Nano Science and Technology, Melbourne School of Engineering, University of Melbourne, Parkville, Victoria 3010, Australia*

[3]*School of Medicine, Faculty of Medicine Dentistry and Health Sciences, University of Melbourne, Parkville, Victoria 3010, Australia*

(Received 19 November 2018; published 21 March 2019)

Environments with immobile obstacles or void regions that inhibit and alter the motion of individuals within that environment are ubiquitous. Correlation in the location of individuals within such environments arises as a combination of the mechanisms governing individual behavior and the heterogeneous structure of the environment. Measures of spatial structure and correlation have been successfully implemented to elucidate the roles of the mechanisms underpinning the behavior of individuals. In particular, the pair correlation function has been used across biology, ecology, and physics to obtain quantitative insight into a variety of processes. However, naively applying standard pair correlation functions in the presence of obstacles may fail to detect correlation, or suggest false correlations, due to a reliance on a distance metric that does not account for obstacles. To overcome this problem, here we present an analytic expression for calculating a corrected pair correlation function for lattice-based domains containing obstacles. We demonstrate that this obstacle pair correlation function is necessary for isolating the correlation associated with the behavior of individuals, rather than the structure of the environment. Using simulations that mimic cell migration and proliferation we demonstrate that the obstacle pair correlation function recovers the short-range correlation known to be present in this process, independent of the heterogeneous structure of the environment. Further, we show that the analytic calculation of the obstacle pair correlation function derived here is significantly faster to implement than the corresponding numerical approach.

DOI: 10.1103/PhysRevE.99.032124

I. INTRODUCTION

Environments that contain obstacles are of interest in a wide variety of fields [1–12]. In biology, it is well known that the motion of macromolecules and proteins in the cytosol is restricted by the densely crowded nature of the interior of cells [6,9]. The meshlike structure of the enteric nervous system, highlighted in Fig. 1(a), contains clusters of glial cells connected by nerve strands, as well as regions that are inaccessible to the enteric glial cells [3,10]. Hence the location and movement of glial cells is constricted by these inaccessible regions [3]. In the context of pedestrian dynamics, successful navigation around an obstacle without jamming is a key criteria for the design of safe egress routes [2,5,7,12,13]. Similarly, predicting how pedestrians will react to path-blocking obstacles is a key question in computer vision, as developing algorithms for robots to reliably avoid collisions with pedestrians is crucial [8,14].

Within such environments individuals can undergo self-organization and form highly spatially structured populations, such as the aforementioned clusters of glial cells [3,10] or pedestrian lanes [7,15]. Quantifying the amount of spatial structure present within an environment provides insight into the mechanisms by which the individuals are governed. Therefore, measures that can be applied to experimental data to obtain estimates of the spatial structure within the data are critical [16]. Various methods for quantifying spatial structure have been proposed previously (for example, see the review by Perry *et al.* [16], and references therein). Here we focus on the use of pair correlation functions (PCFs), which are a powerful and versatile tool for analyzing spatial structure and spatial correlation [17–27]. PCFs have been successfully employed in astrophysics [18], particle physics [23], ecology [27–29], and cell biology [19,25], among others. Briefly, the pair correlation for a given distance m, $P(m)$, can be defined as

$$P(m) = \frac{C(m)}{E[C(m)]},$$

where $C(m)$ is the number of pairs of individuals separated by a distance m observed in the data, and the normalization term, $E[C(m)]$, is the expected number of individuals separated by distance m if the individuals are located randomly throughout the experimental domain. If there are more individuals separated by a particular distance than expected for randomly located individuals, then $P(m) > 1$, and hence there is spatial structure corresponding to correlation at that distance. Similarly, if fewer individuals are separated by a particular distance than expected, then $P(m) < 1$, which suggests that there is spatial structure corresponding to anticorrelation present at distance m [19].

*stuart.johnston@unimelb.edu.au

 ©2019 American Physical Society

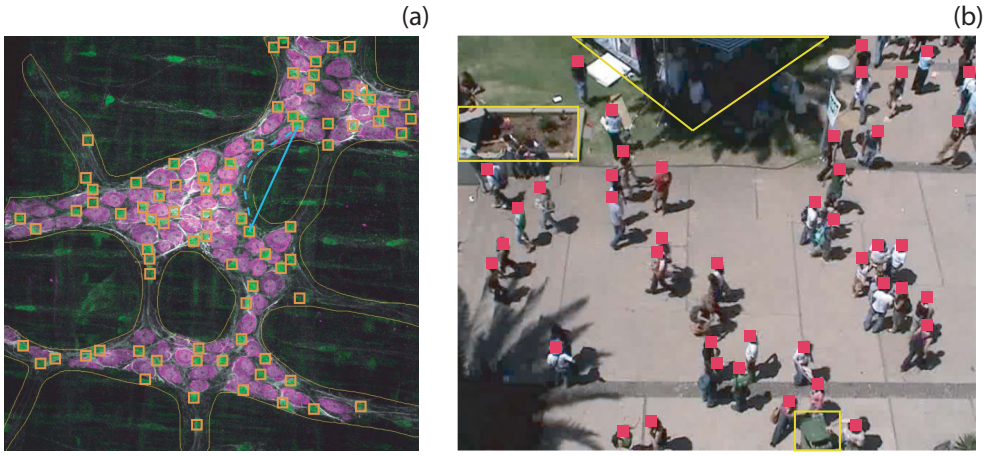

FIG. 1. (a) Experimental image of the nervous system within the mouse colon, containing neurons (magenta), glial cells (green), and glial processes (white). Glial cells within clusters (known as ganglia) are highlighted with orange squares. Yellow lines indicate inaccessible regions. An example of path distance and Cartesian distance between cells are highlighted in cyan (dashed and solid, respectively). (b) Experimental image of pedestrian locations (red squares) in the presence of obstacles (yellow lines). Image is obtained from the freely available data set provided by the authors of Ref. [1].

As it is unlikely that any two pairs of individuals are separated by exactly the same distance, m is typically divided into bins [20]. This can take the form of considering the environment as continuous space and binning the measured distance between pairs of individuals, or by mapping individuals onto a discrete domain such that there is a finite number of possible distances between pairs of individuals [19,20,24,25]. There has been significant recent focus on PCFs for discrete, or lattice-based, domains [17,19,20,24–26], and deriving analytic expressions for the normalization term under various distance metrics [19,24]. In particular, Binder and Simpson [19] present a normalization term for rectilinear distance in x and y, illustrated in Figs. 2(a) and 2(b), which corresponds to the distance separating two lattice sites in x and y, respectively. More recently, Gavagnin *et al.* [24] derive a normalization term under the taxicab and square uniform distance metrics, illustrated schematically in Figs. 2(c) and 2(d), respectively. Under the taxicab distance metric and the square uniform distance metric, the distance between two lattice sites can be thought of as the minimum number of "jumps" between the two sites under movement occurring in a von Neumann neighborhood (four nearest neighbors) and a Moore neighborhood (eight nearest neighbors), respectively [24].

However, while these PCFs have proven useful in a range of applications, they are unsuitable for analyzing environments that contain inaccessible regions, due to either "holes" in the domain or the presence of obstacles. In these environments, Cartesian distance measures do not adequately describe the distance between two individuals. For example, for the glial cells presented in Fig. 1(a), certain cells are separated by a path distance that is significantly longer than the Cartesian distance between the cells. Therefore, naively calibrating a standard PCF to these data may result in a lack of identification of spatial correlation between cells or in spurious correlations being reported. Here we propose an analytic method for calculating a corrected PCF for lattice domains containing obstacles or inaccessible regions, which we refer to as an obstacle PCF (oPCF). In Sec. II we construct the oPCF in a systematic manner, first considering a single inaccessible site, and subsequently increasing the number of sites within an inaccessible region, as well as increasing the number of inaccessible regions. Through comparison with path-finding algorithms, we show that the derived normalization term is exact. In Sec. III we demonstrate that this oPCF is required to isolate the correlation associated with the mechanisms governing the behavior of individuals from correlation associated with the structure of the environment.

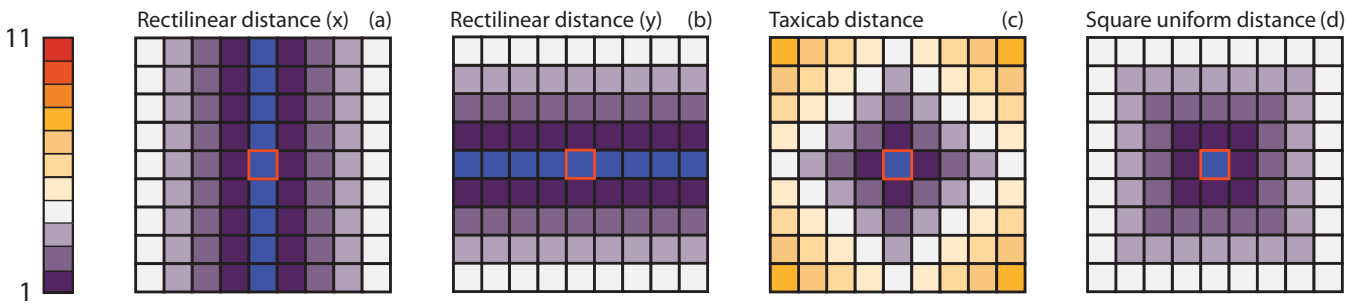

FIG. 2. Distance between the center lattice site (highlighted in red) and other lattice sites under the (a) rectilinear x, (b) rectilinear y, (c) taxicab, and (d) square uniform distance metrics. Note that blue sites correspond to a distance of zero.

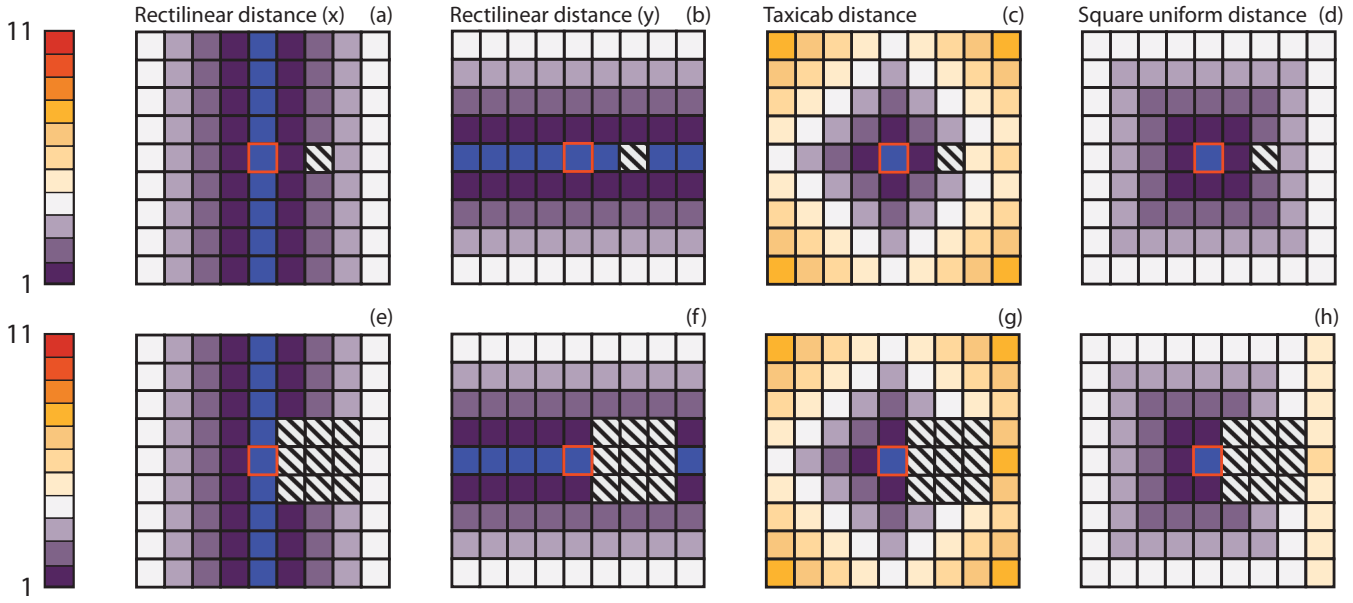

FIG. 3. Distance between the center lattice site (highlighted in red) and other lattice sites under (a), (e) rectilinear x, (b), (f) rectilinear y, (c), (g) taxicab, and (d), (h) square uniform distance metrics in the presence of (a)–(d) one or (e)–(h) nine inaccessible sites (cross-hatched). Note that blue sites correspond to a distance of zero.

Further, we show that analysis with the exact normalization term is significantly less computationally intensive to perform, compared to using a path-finding algorithm, and we discuss environments where an approximation to the normalization term can be used effectively. Finally, in Sec. IV we discuss our results and suggest potential avenues for future research.

The code used to generate the results in this paper can be found at Ref. [30].

II. DERIVATION

First we illustrate domains where a different distance metric is required. In Figs. 3(a)–3(d) we introduce a single inaccessible lattice site and examine how the distance metrics change, compared to the domain in Fig. 2. We note that neither of the rectilinear distances change, and hence all sites remain the same distance from the center site, excluding the inaccessible sites. As such, the counts of pair distances are reduced only by the reduction in the number of lattice sites. The taxicab distance is more significantly impacted by the introduction of the inaccessible site, because to travel from the center of the domain to the rightmost side now requires that the inaccessible site is avoided. Hence the taxicab distance between the center site and sites on the opposite side of the inaccessible site, with respect to the center site, increases by two [Fig. 3(c)]. The square uniform distance is also unaffected by the inaccessible site, because diagonal "jumps" count the same as either horizontal or vertical "jumps." Therefore the inaccessible site can be avoided by two diagonal "jumps," rather than the two horizontal "jumps," and the distance does not change [Fig. 3(d)]. If we introduce a larger inaccessible region, as presented in Figs. 3(e)–3(h), we again see that both rectilinear distances are not influenced. As before, the taxicab distance is influenced because the inaccessible region must be avoided to travel between the center site and sites on the right boundary [Fig. 3(g)]. In contrast to the small inaccessible region, the square uniform distance is now affected by the presence of the larger inaccessible region, because the larger region cannot be avoided through diagonal movement [Fig. 3(h)]. We note that any inaccessible region aside from a single site will influence the square uniform distance. As the size of a lattice site typically corresponds to the size of an individual, a distance metric that is not impacted by the presence of obstacles of that size is not appropriate. Furthermore, the majority of models that are implemented on a lattice with obstacles typically allow movement to only one of four nearest neighbors [13,31–36], and hence the taxicab distance metric is implicitly applied. As such, in the remainder of this work, we consider only the taxicab distance metric.

A. Standard pair correlation functions

To obtain the oPCF for an environment with obstacles under the taxicab distance metric, we first introduce the counts of pair distances for an environment without obstacles. For a domain containing L_x sites in the x direction and L_y sites in the y direction with no-flux boundary conditions, the maximum pair distance is $L_x + L_y - 2$. As experimental images are typically captured such that the influence of boundary effects are minimized, no-flux boundary conditions are perhaps the most relevant boundary conditions [25]. An alternative choice is periodic boundary conditions, which are particularly relevant if the experimental image captured is a small region, yet representative of a larger experimental domain. However, in this work, we focus on no-flux boundary conditions. Recently, Gavagnin *et al.* [24] derived the counts of pair distances for a domain without obstacles, $D_{\mathrm{NO}}(m)$, for $m < \min(L_x, L_y)$:

$$D_{\mathrm{NO}}(m) = 2mL_xL_y - (L_x + L_y)m^2 + \frac{m^3 - m}{3}.$$

Introducing inaccessible sites into the domain increases the distances between pairs of sites (Fig. 3), and hence we require an expression for the counts of pair distances for $m \geqslant \min(L_x, L_y)$. For an arbitrary L_x by L_y domain with no obstacles, the counts of pair distances are (see Appendix A for the derivation)

$$D_{\mathrm{NO}}(m) = \begin{cases} 2mL_xL_y - (L_x + L_y)m^2 + \frac{m^3-m}{3}, & 1 \leqslant m \leqslant \min(L_x, L_y) \\ D_{\mathrm{NO}}(\min(L_x, L_y)) - \min(L_x, L_y)^2[m - \min(L_x, L_y)], & \min(L_x, L_y) < m < \max(L_x, L_y) \\ \frac{k(k+1)(k+2)}{3}, \text{ where } k = L_x + L_y - 1 - m, & \max(L_x, L_y) \leqslant m \leqslant L_x + L_y - 2 \end{cases}, \tag{1}$$

The PCF is calculated by evaluating the counts of pair distances between occupied sites, $C(m)$, and normalizing by the expected number of pair distances obtained from $D(m)$ and the average occupancy of the domain. If there are z occupied sites and $n_a = L_xL_y - n_h$ accessible sites, where n_h is the number of inaccessible sites, then the expected counts of pair distances are [19]

$$E[C(m)] = \frac{z(z-1)}{n_a(n_a-1)}D(m). \tag{2}$$

Picking two accessible sites at random, z/n_a is the probability that the first selected site is occupied, and $(z-1)/(n_a-1)$ is the probability that the second site is occupied, given that an occupied site has been selected previously.

There are two counts that must be obtained from the data: the counts of pair distances between occupied sites, $C(m)$, and the counts of pair distances between accessible sites, $D(m)$. While $C(m)$ may have to be obtained via a path-finding algorithm, the number of occupied sites is typically small compared to the total number of sites. As such, calculating $C(m)$ will require significantly fewer iterations of the path-finding algorithm, because the number of iterations required scales with the square of occupied sites for $C(m)$ or the square of accessible sites for $D(m)$ [24]. Hence, even if calculating $D(m)$ via a path-finding algorithm is prohibitively computationally intensive, $C(m)$ should be able to be calculated rapidly.

B. Corrected pair correlation functions

Next, we focus on obtaining an expression for $D(m)$ for domains containing inaccessible sites, by adjusting the counts of pair distances for a domain with no obstacles, D_{NO}, to account for inaccessible sites. This takes the form of several additional terms:

(1) *Accessible-inaccessible pairs*, denoted $A(m)$, which are pairs of sites in the domain that consist of an accessible site and an inaccessible site. As these pairs are counted in $D_{\mathrm{NO}}(m)$, we require that $A(m)$ is accounted for via the removal of these pairs.

(2) *Inaccessible-inaccessible pairs*, denoted $I(m)$, which are pairs of inaccessible sites. Again, these sites are counted in $D_{\mathrm{NO}}(m)$ and must be removed to obtain $D(m)$.

(3) *Shifted pairs*, denoted $S(m)$, which are pairs of accessible sites where the path distance between the sites is different to the taxicab distance due to the presence of inaccessible sites. Shifted pairs consist of *lost pairs*, $L(m)$, which are pairs of sites in $D_{\mathrm{NO}}(m)$ that are no longer present due to inaccessible sites altering the distance [Figs. 5(b) and 5(d) below] and *gained pairs*, $G(m)$, which are pairs of sites that are not in $D_{\mathrm{NO}}(m)$ but are now present due to the introduction of inaccessible sites [Figs. 5(c) and 5(e)].

C. Single inaccessible site

We will derive an expression for each of the adjustment terms by systematically considering different configurations of inaccessible sites. We first consider a single inaccessible site, such as presented in Fig. 4, and use the subscript s to denote the special case of a single inaccessible site. The coloring on each site in Fig. 4 highlights the distance between that site and the inaccessible site. For a single inaccessible site, the number of sites with a specific color therefore corresponds to the accessible-inaccessible pairs, $A_s(m)$. We note that $A_s(m)$ is a function of both the domain size, L_x and L_y, and the location of the inaccessible site, (H_x, H_y). However, for notational convenience, we do not explicitly denote this dependence. In the absence of boundaries, there are $4m$ accessible-inaccessible pairs for a distance m.

Intuitively, the boundaries reduce the number of pairs of sites separated by larger values of m. To calculate which of the $4m$ pairs lie outside the boundary, we introduce eight values, which represent the distance between the inaccessible site and the boundaries and corners of the domain. The distance to the boundary, b_i, where $i \in \{L, R, D, U\}$ for the boundary in the left, right, down, and up directions, respectively, is defined as

$$b_L = H_x, \quad b_R = L_x - H_x + 1, \quad b_D = H_y, \quad b_U = L_y - H_y + 1,$$

where H_x and H_y correspond to the x and y location of the inaccessible site. Similarly, the distance to the corner, $c_{j,k}$ where $j \in \{D, U\}$ and $k \in \{L, R\}$ for the corner in the down-left, down-right, up-left, and up-right directions, respectively, is defined as

$$c_{D,L} = b_D + b_L, \quad c_{D,R} = b_D + b_R, \quad c_{U,L} = b_U + b_L, \quad c_{U,R} = b_U + b_R.$$

These values allow us to define a function for the number of pairs containing a site that is located outside of the boundaries at a distance m, $\alpha(m)$, referred to as out-of-domain pairs, and hence at the corresponding distance m:

$$A_s(m) = 4m - \alpha(m). \tag{3}$$

The accessible site belonging to an accessible-inaccessible pair can either be located in the same row or column as the inaccessible site [Figs. 4(a)–4(c)] or not located in the same row and not in the same column as the inaccessible site [Figs. 4(d)–4(f)]. For sites in either the same row or column as the inaccessible site, there is at most one site at a distance m in each direction [Figs. 4(a)–4(c)]. Further, the sites will be

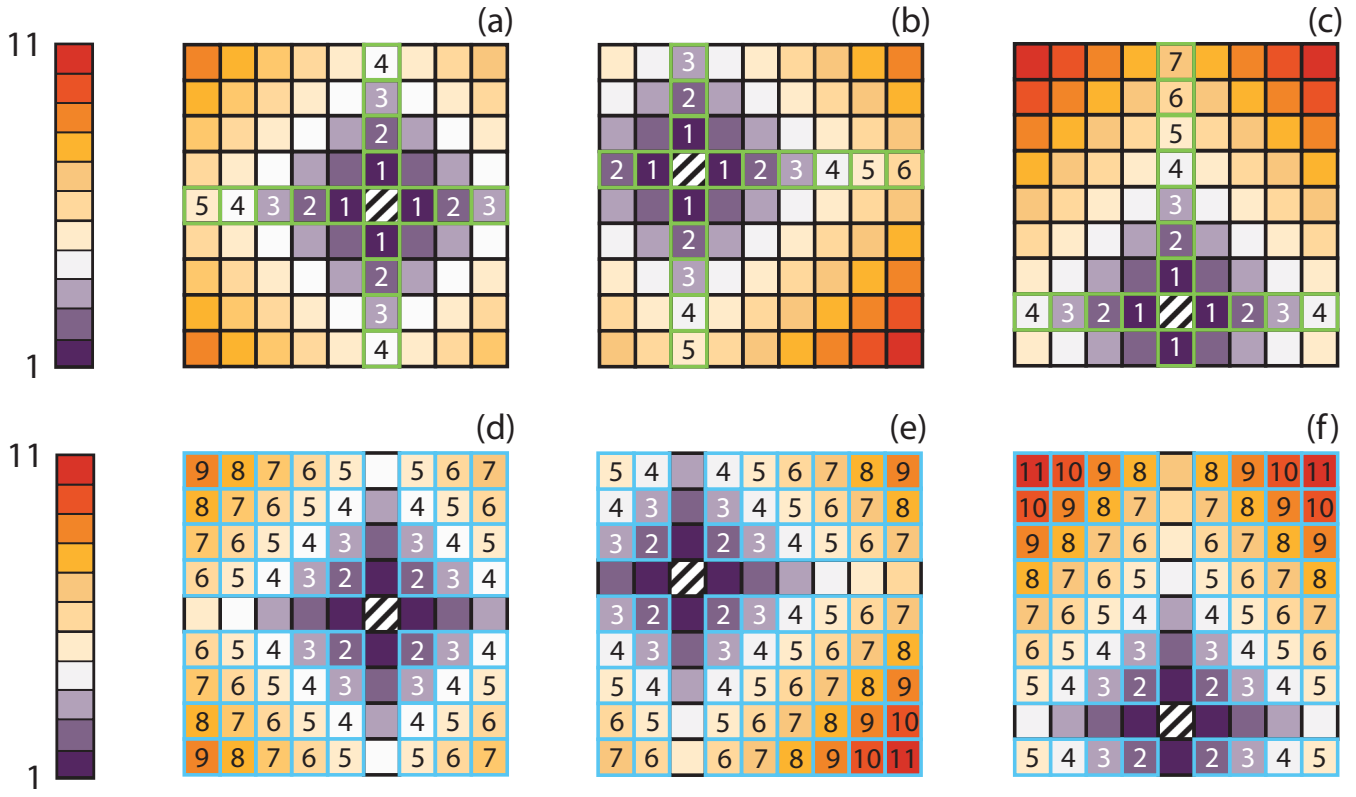

FIG. 4. Example domains with a single inaccessible site (cross-hatched). The color of individual sites corresponds to the distance between that site and the inaccessible site. Accessible-inaccessible pairs are highlighted in (a)–(c) green or (d)–(f) cyan, depending on whether the accessible site is within the same row or column as the inaccessible site.

in the domain only for distances less than the distance to the boundary. We therefore introduce the function

$$N_i(m) = \begin{cases} 0, & \text{for } m < b_i \\ 1, & \text{for } m \geqslant b_i \end{cases},$$

which represents the number of out-of-domain pairs of distance m with respect to the boundary b_i, for all $i \in \{L, R, D, U\}$. For the number of out-of-domain pairs of distance m in diagonal directions, intuitively there will be no out-of-domain pairs if the distance is less than the minimum distance to a boundary of interest. We note that diagonal refers to a pair of sites that have neither a row or column in common. For all distances greater than the minimum distance to a boundary, but less than the distance to the other boundary of interest, we observe that there are the same number of sites in the domain, for each of these distances. For example, in Fig. 4(e), in the down-left direction, we observe only two sites highlighted in cyan for $m = 3, 4, 5, 6$. In comparison, in the down-right direction, where $m = 6$ is less than the minimum distance to a boundary, we observe two, three, four, and five sites highlighted in cyan for $m = 3, 4, 5, 6$, respectively. Hence, the number of out-of-domain pairs increases exactly with distance for distances greater than the minimum distance to a boundary, but less than the distance to the other boundary of interest. For distances greater than the maximum distance to a boundary of interest, we observe that the number of sites highlighted in cyan decreases exactly with distance. Hence the number of out-of-domain pairs increases by two for an increase in distance of one. Finally, for distances greater than the distance to the corner site, there will be no pairs inside the domain, and the number of out-of-domain pairs must be $m - 1$. We note that in all cases the sum of number of the pairs outside and inside the domain is $m - 1$ for a distance m in a particular diagonal direction. Combining these observations, we introduce the function

$$M_{j,k}(m) = \begin{cases} 0, & \text{for } m \leqslant \min(b_j, b_k) \\ m - \min(b_j, b_k), & \text{for } \min(b_j, b_k) < m \leqslant \max(b_j, b_k) \\ 2m - c_{j,k}, & \text{for } \max(b_j, b_k) < m \leqslant c_{j,k} - 2 \\ m - 1, & \text{for } m > c_{j,k} - 2 \end{cases},$$

which represents the number of out-of-domain pairs in each diagonal direction. Note that both $N_i(m)$ and $M_{j,k}(m)$ do not need to be evaluated for pair distances greater than the largest distance between the inaccessible site and the boundary or

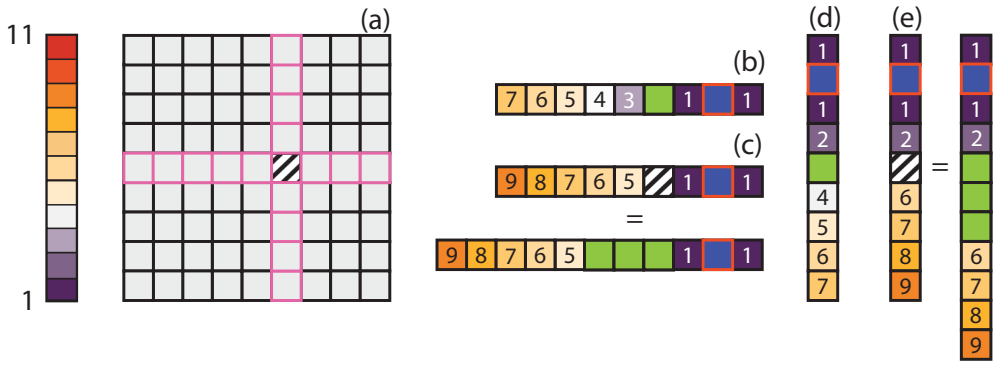

FIG. 5. (a) Example domain with a single inaccessible site (cross-hatched) and sites contributing to shifted pairs highlighted in pink. Shifted pairs consist of a pair of pink sites, where there is one site on either side of the inaccessible site. (b)–(e) Distance between an example lattice site (highlighted in red) and other lattice sites for (b)–(c) the row of pink sites and (d)–(e) the column of pink sites if the inaccessible site (b), (d) does not have to be avoided (green) or (c), (e) must be avoided. (c), (e) Note the equivalence between a subdomain with a single inaccessible site that must be avoided and a subdomain with three inaccessible sites that do not need to be avoided, with two additional sites in the subdomain.

corner, respectively. The number of out-of-domain pairs is therefore

$$\alpha(m) = \sum_i N_i(m) + \sum_j \sum_k M_{j,k}(m).$$

For a single inaccessible site there are no inaccessible-inaccessible pairs. Therefore, we next consider the shifted pairs for a single inaccessible site. In this case, pairs of sites that are located on either side of the inaccessible site are shifted pairs, as highlighted in Fig. 5. The relevant row and column are presented in Figs. 5(b)–5(e), with an example site highlighted to demonstrate the difference between the pair distances in the domain if the inaccessible site is included [Figs. 5(c) and 5(e)] or not [Figs. 5(b) and 5(d)]. Intuitively, we observe that if both sites in the pair are on the same side of the inaccessible site, then the presence of the inaccessible site is irrelevant. If the sites in the pair are on opposite sides of the inaccessible site, then the inaccessible site increases the pair distance by two, which accounts for a path that avoids the inaccessible site. As we initially consider the counts of pair distances for the domain without inaccessible sites, we can incorporate the shifted pairs by removing the pairs present in Figs. 5(b) and 5(d) and including the pairs present in Figs. 5(c) and 5(e). We note that these are the lost pairs and gained pairs, respectively.

We therefore require an expression for the counts of pair distances within the subdomains in Figs. 5(b)–5(e) for pairs with an inaccessible site on both sides of the inaccessible site. Two observations are useful here: the total number of pairs is conserved, and avoiding the inaccessible site is equivalent to extending the subdomain by two sites, adding two more inaccessible sites, and calculating the pair distances as if the inaccessible site can be passed through. The minimum pair distance for pairs that are located on separate sides of an inaccessible site is one greater than the number of inaccessible sites between them. Hence $L(m)$ is defined on $2 \leqslant m \leqslant L_y - 1$ and $2 \leqslant m \leqslant L_y - 1$, respectively, for the subdomains in Figs. 5(b) and 5(d). The maximum number of pairs separated by a given distance is restricted by the requirement that sites are located on both sides of the inaccessible site and hence has an upper bound of $d_H = \min(b_L, b_R) - 1$ in the horizontal direction and $d_V = \min(b_U, b_D) - 1$ in the vertical direction. These values represent the minimum of the number of sites on either side of the inaccessible site. Further, there is only one possible pair of sites separated by a distance of two and $L_x - 1$ (or $L_y - 1$), two possible pairs of sites for a distance of three and $L_x - 2$ (or $L_y - 2$), and so forth. The one possible pair of sites separated by a distance of two, recalling that we require that the sites are located on both sides of the inaccessible site, are the two sites located immediately next to the inaccessible site. Similarly, for a distance of three, there are two options: a site located immediately next to the inaccessible site, and a site located a distance of two from the inaccessible site. As the site located immediately next to the inaccessible site can be on either side of the inaccessible site, this gives the two possible pairs of sites.

Combining this with the aforementioned upper bound we obtain an expression for $L(m)$ for a single inaccessible site:

$$L_s(m) = \min\left(-\left|m - \frac{L_y+1}{2}\right| + \frac{L_y-1}{2}, d_V\right) + \min\left(-\left|m - \frac{L_x+1}{2}\right| + \frac{L_x-1}{2}, d_H\right). \quad (4)$$

As noted previously, the number of shifted pairs is conserved. As the pair distance increases by two in the presence of a single inaccessible site, here

$$G_s(m) = \min\left(-\left|m - \frac{L_y+5}{2}\right| + \frac{L_y-1}{2}, d_V\right) + \min\left(-\left|m - \frac{L_x+5}{2}\right| + \frac{L_x-1}{2}, d_H\right). \quad (5)$$

This corresponds to an increase of two in both the number of inaccessible sites and the subdomain length. We have now considered all the adjustment terms for a single inaccessible site, and therefore by combining (1), (3), (4), and (5), we obtain the expression for the count of pair distances for a domain with a single inaccessible site, noting that $I(m) = 0$

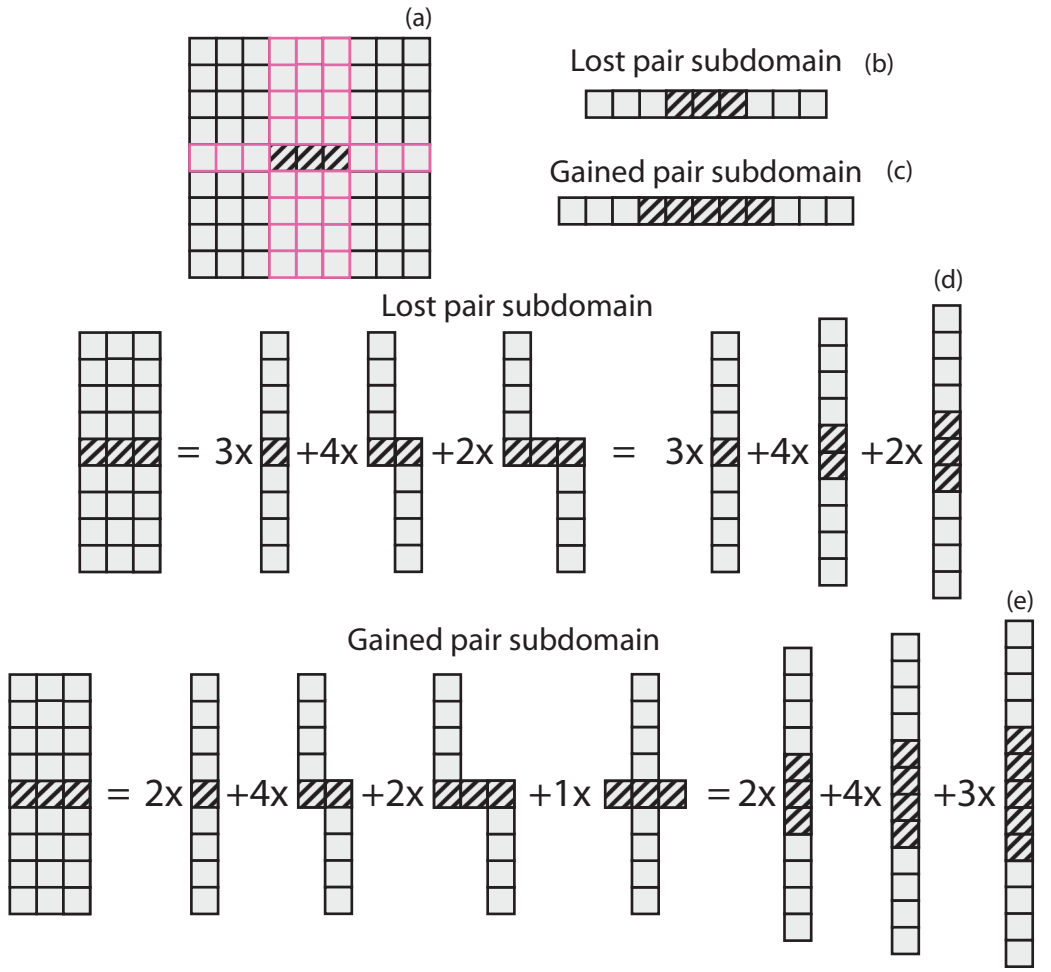

FIG. 6. (a) Example domain with a cluster of inaccessible sites (cross-hatched) and sites contributing to shifted pairs highlighted in pink. (b) Subdomain that contains all lost pairs for the row of pink sites in panel (a); that is, all pairs in this subdomain are contained within $A(m)$ but must be removed to incorporate the influence of the inaccessible site. (c) Subdomain that contains all gained pairs for the row of pink sites in panel (a); that is, all pairs in this subdomain are not contained within $A(m)$ but need to be included to incorporate the influence of the inaccessible site. (d) Subdomain, and associated transformation to multiple subdomains, which contain all lost pairs for the three columns of pink sites in panel (a). (e) Subdomain, and associated transformation to multiple subdomains, which contain all gained pairs for the three columns of pink sites in panel (a).

for a single inaccessible site:

$$D_s(m) = D_{\mathrm{NO}}(m) - A_s(m) - L_s(m) + G_s(m). \tag{6}$$

D. Clusters of inaccessible sites

We now consider a domain with several inaccessible sites that form a C_x by 1 cluster of inaccessible sites, as presented in Fig. 6. Introducing the two additional sites in this example, compared to the example in Fig. 5, results in an increase in the number of accessible-inaccessible pairs. The total number of accessible-inaccessible pairs, $A_s(m)$, is given by

$$A(m) = \sum_{H} A_s(m), \tag{7}$$

where H is the set of inaccessible sites in the domain and $A_s(m)$ is defined in (3). Including the additional inaccessible sites introduces inaccessible-inaccessible pairs. The calculation of the number of inaccessible-inaccessible pairs, $I(m)$, is straightforward, because the distance between such pairs is simply the taxicab distance between the two relevant sites:

$$I(m) = \sum_{i=1}^{n_h} \sum_{j=i+1}^{n_h} \mathbf{1}_m[d_{\mathrm{taxicab}}(\mathbf{h}_i - \mathbf{h}_j)], \tag{8}$$

where $\mathbf{1}_m(x)$ is the indicator function, one when $x = m$ and zero otherwise, n_h is the number of inaccessible sites, and $d_{\mathrm{taxicab}}(\mathbf{h}_i - \mathbf{h}_j)$ is the taxicab distance between two sites located at sites $\mathbf{h}_i = (H_x^i, H_y^i)$ and $\mathbf{h}_j = (H_x^j, H_y^j)$. We observe that $I(m) = 0$ for $n = 1$, as discussed previously.

To calculate lost and gained pairs for a cluster of inaccessible sites, we consider the example domain in Fig. 6. The lost pairs and gained pairs associated with the horizontal direction are relatively straightforward and can be calculated from the subdomains presented in Figs. 6(b) and 6(c). We introduce the general counts of pair distances for a one-dimensional subdomain $K(m, n, X, d)$, where X is the total number of sites in the subdomain, n is the number of inaccessible sites in the

subdomain, and d is the relevant d_V or d_H value, as defined previously. The function is

$$K(m,n,X,d) = \min\left(-\left|m - \frac{X+n}{2}\right| + \frac{X-n}{2}, d\right). \quad (9)$$

We note that this generalizes Eqs. (4) and (5) by allowing for an arbitrary number of inaccessible sites. By using this generalized definition, we can consider transformations of the cluster of inaccessible sites to multiple one-dimensional subdomains with varying X and n, and calculating the K value for each. For example, for the horizontal subdomains in Fig. 6(b), the lost pair subdomain uses the X, n, and d values obtained from the original domain, whereas the gained pair subdomain in Fig. 6(c) uses $X+2$, $n+2$, and d. This is consistent with the previous observation of a longer pair distance corresponding to the path around the inaccessible sites. We reiterate that this implies that the sites around the inaccessible sites are accessible. The vertical subdomains in Fig. 6(a) are more complicated due to the increase in the number of columns where pairs of sites can be located such that the distance between them is influenced by the inaccessible sites. These columns are highlighted in pink in Fig. 6(a). As the accessible sites must be located on either side of the inaccessible sites, there are a total of nine combinations of columns. Hence, the transformation of the vertical subdomain around the cluster of inaccessible sites results in nine one-dimensional subdomains. For the lost pairs, there can be one, two, or three inaccessible sites between the pair of accessible sites. The number of each of these possibilities is highlighted in Fig. 6(d). Note that an increase in the number of inaccessible sites corresponds to an increase in both X and n. For the gained pairs, there can again be one, two, or three inaccessible sites between the pair of accessible sites. However, the magnitude of the increase in pair distance due to the inaccessible sites depends on which columns the pair belongs to. If either one of the accessible sites is in an outermost column, then the increase in pair distance is two. However, if both accessible sites are in the middle column, then the increase in pair distance is four, as the path from one site to the other must avoid all of the inaccessible sites. In general, the increase in distance is two times the minimum distance between the columns (rows) that the accessible sites belong to and the columns (rows) on either side of the inaccessible sites. We illustrate the transformation of the vertical subdomain into one-dimensional subdomains for gained pairs in Fig. 6(e), as well as the relevant number of each subdomain. In general, an C_x by 1 cluster of inaccessible sites results in

$$L(m) = K(m, C_x, L_x, d_H) + C_x K(m, 1, L_y, d_V) + \sum_{i=2}^{C_x} 2(C_x - i + 1) K(m, i, L_y + i - 1, d_V) \quad (10)$$

and

$$G(m) = K(m, C_x + 2, L_x + 2, d_H) + C_x K(m, C_x + 2, L_y + C_x + 1, d_V) + \sum_{i=1}^{C_x - 1} 2iK(m, i + 2, L_y + i + 1, d_V). \quad (11)$$

The first term in both equations corresponds to the horizontal subdomain, and the second and third terms correspond to the $(C_x)^2$ vertical subdomains. A similar function for the lost and gained pairs would describe a 1 by C_y cluster of inaccessible sites, because this is simply a rotation of the C_x by 1 cluster of inaccessible sites.

It is now relatively straightforward to extend the lost pairs and gained pairs functions [(10) and (11)] to apply to an C_x by C_y cluster of inaccessible sites, as presented in Fig. 7. We note that the vertical subdomain transformation is similar to the previous C_x by 1 cluster, albeit with an increase in n. Now that the cluster has $C_y > 1$, the horizontal subdomain transformation also results in multiple one-dimensional subdomains. We note that the transformation is the same as for the vertical subdomain except for rotation, and hence it is straightforward to obtain the lost pairs function for an arbitrary C_x by C_y cluster of inaccessible sites,

$$L_c(m, C_x, C_y) = L_c^h(m, C_x, C_y, L_y) + L_c^v(m, C_x, C_y, L_y), \quad (12)$$

where

$$L_c^h(m, C_x, C_y, L_y) = C_y K(m, C_x, L_x, d_H) + \sum_{i=2}^{C_y} 2(C_y - i + 1) \times K(m, i + C_y - 1, L_x + i - 1, d_H)$$

is the horizontal contribution to the lost pair distances and

$$L_c^v(m, C_x, C_y, L_y) = C_x K(m, C_y, L_y, d_V) + \sum_{i=2}^{C_x} 2(C_x - i + 1) \times K(m, i + C_x - 1, L_y + i - 1, d_V)$$

is the vertical contribution to the lost pair distances. Similarly, the gained pairs function for an arbitrary C_x by C_y cluster of inaccessible sites can be obtained and is given by

$$G_c(m, C_x, C_y) = G_c^h(m, C_x, C_y, L_x) + G_c^v(m, C_x, C_y, L_x), \quad (13)$$

where

$$G_c^h(m, C_x, C_y, L_x) = C_y K(m, C_x + C_y + 1, L_x + C_y + 1, d_H) + \sum_{i=1}^{C_y - 1} 2iK(m, i + C_x + 1, L_x + i + C_x, d_H)$$

is the horizontal contribution to the gained pair distances and

$$G_c^v(m, C_x, C_y, L_x) = C_x K(m, C_y + C_x + 1, L_y + C_x + 1, d_V) + \sum_{i=1}^{C_x - 1} 2iK(m, i + C_y + 1, L_y + i + C_y, d_V)$$

is the vertical contribution to the gained pair distances.

Combining (7), (8), (12), and (13) we obtain the number of pair distances for a cluster of inaccessible sites:

$$D_c(m) = P_{NO}(m) - A(m) + I(m) - L_c(m) + G_c(m). \quad (14)$$

Note that the $I(m)$ term is positive because the inaccessible-inaccessible pairs are counted twice and removed in $A(m)$.

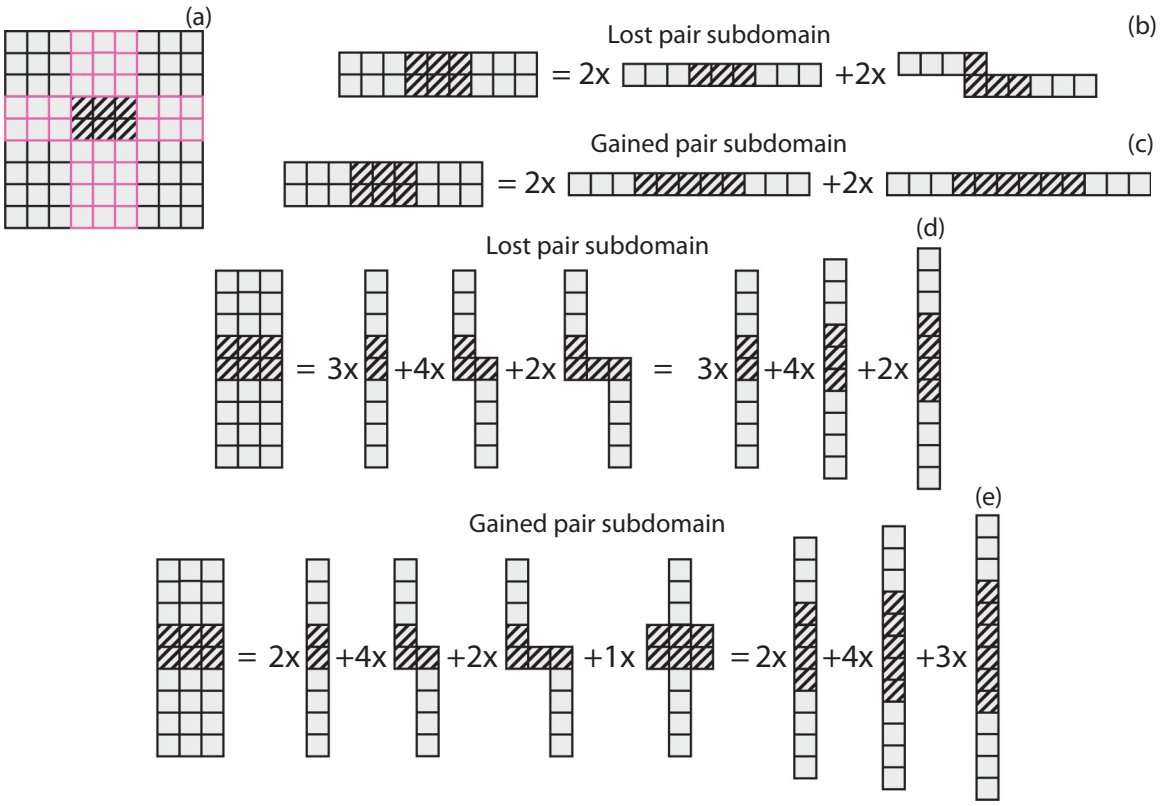

FIG. 7. (a) Example domain with a cluster of inaccessible sites (cross-hatched) and sites contributing to shifted pairs highlighted in pink. (b)–(c) Subdomain, and associated transformation to multiple subdomains, which contain all lost and gained pairs, respectively, for the two rows of pink sites in panel (a). (d)–(e) Subdomain, and associated transformation to multiple subdomains, which contain all lost and gained pairs, respectively, for the three columns of pink sites in panel (a).

E. Multiple clusters of inaccessible sites

Thus far we have considered only a single cluster of inaccessible sites. We now consider the generalization to multiple clusters of inaccessible sites, as illustrated in Fig. 8(a). In this example, we consider a domain with 1 by 1 clusters of inaccessible sites. However, we note that the approach generalizes to rectangular clusters of any size provided that the clusters are arranged such that entirely accessible rows and columns exist on either side of all clusters. We also note that the previous definitions of the accessible-inaccessible pairs and inaccessible-inaccessible pairs [(7) and (8)] are valid here. Similar to the approach to obtain the lost and gained pairs for

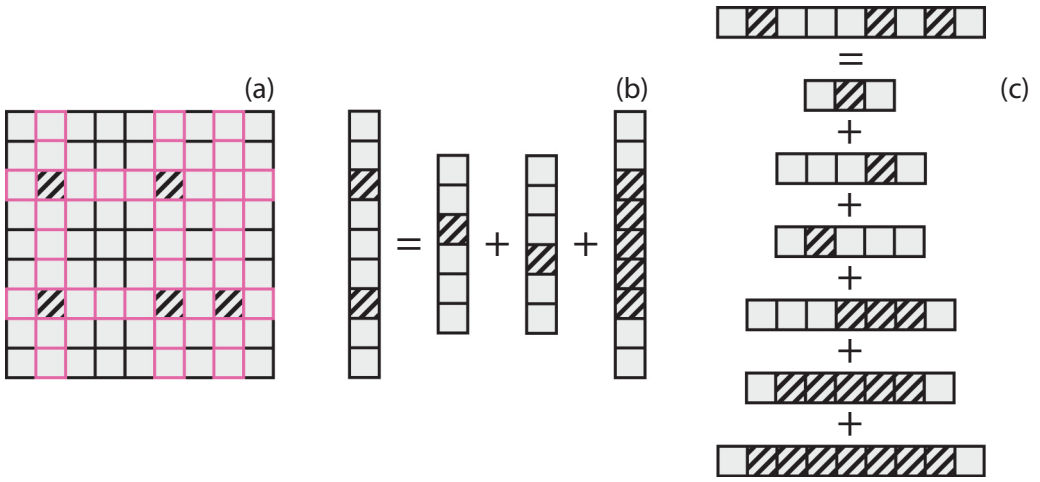

FIG. 8. (a) Example domain with multiple inaccessible sites (cross-hatched) and sites contributing to shifted pairs highlighted in pink. (b) Subdomain, and associated transformation to multiple subdomains, for the leftmost column of pink sites. (c) Subdomain, and associated transformation to multiple subdomains, for the bottommost row of pink sites. Note that each transformed subdomain has both lost and gained pairs as described previously and that the gained pair subdomains are not shown.

a C_x by C_y cluster, a subdomain is isolated and transformed into $B^i(B^i+1)/2$ one-dimensional subdomains, where B^i is the number of clusters within a subdomain. Consider the column subdomain presented in Fig. 8(b). The transformation isolates all possible combinations of inaccessible sites, and the lost pairs and gained pairs functions are calculated for each one-dimensional subdomain. The transformation of the subdomain in Fig. 8(b) results in three subdomains. The first two subdomains contain a single inaccessible site and X total sites, where X corresponds to the maximum number of sites from the original subdomain that are contiguous and still contain only that single inaccessible site. The third subdomain contains both inaccessible sites and renders all of the intervening sites inaccessible. Here X corresponds to the length of the original subdomain. As we are able to calculate the lost and gained pair functions on a subdomain with a single cluster of inaccessible sites, transforming the subdomain with multiple separate inaccessible sites provides a straightforward method for this calculation. We note that the single inaccessible site subdomains provide pairs that contain sites in the uppermost (or bottommost) region of the original subdomain as well as in the middle of the original subdomain. The subdomain with additional inaccessible sites provides pairs that are located in both the uppermost and bottommost region of the original subdomain. In Fig. 8(c) we consider the row with three single inaccessible sites and the associated transformation. As before, we consider all possible combinations of inaccessible sites: three subdomains with a single inaccessible site, two subdomains with inaccessible regions bounded by two inaccessible sites, and finally a subdomain containing all three inaccessible sites, and the sites between them treated as inaccessible.

We now generalize this approach to B distinct clusters within a row (column) of clusters of inaccessible sites, where $\mathbf{s}_\mathrm{h}$ ($\mathbf{s}_\mathrm{v}$) is the set of coordinates of the leftmost (uppermost) site in inaccessible clusters and $\mathbf{f}_\mathrm{h}$ ($\mathbf{f}_\mathrm{v}$) is the set of coordinates of the rightmost (bottommost) site in inaccessible clusters:

$$L(m)=\sum_{i=1}^{B}\sum_{j=1}^{B-i+1} L_c^h\big(m, \mathbf{f}_\mathrm{h}^{\,i+j-1}-\mathbf{s}_\mathrm{h}^{\,j}-1, C_y, \mathbf{s}_\mathrm{h}^{\,i+j}-\mathbf{f}_\mathrm{h}^{\,j-1}-1\big). \tag{15}$$

The first summation represents the number of clusters of inaccessible sites in the domain, and the second summation represents the number of combinations of i neighboring clusters. The $\mathbf{f}_\mathrm{h}^{\,i+j-1}-\mathbf{s}_\mathrm{h}^{\,j}-1$ value corresponds to the number of sites in the cluster in the horizontal direction, and the $\mathbf{s}_\mathrm{h}^{\,i+j}-\mathbf{f}_\mathrm{h}^{\,j-1}-1$ value corresponds to the length of the subdomain. Repeating this processes over all distinct rows and columns of clusters of inaccessible sites, we obtain

$$\begin{aligned} L(m) = &\sum_{i=1}^{n_\mathrm{rows}}\sum_{j=1}^{B^i}\sum_{k=1}^{B^i-j+1} L_c^h\big(m, \mathbf{f}_{\mathrm{h},i}^{\,j+k-1}-\mathbf{s}_{\mathrm{h},i}^{\,k}-1, C_y, \mathbf{s}_{\mathrm{h},i}^{\,j+k}-\mathbf{f}_{\mathrm{h},i}^{\,k-1}-1\big)\\ &+\sum_{i=1}^{n_\mathrm{columns}}\sum_{j=1}^{B^i}\sum_{k=1}^{B^i-j+1} L_c^v\big(m, C_x, \mathbf{f}_{\mathrm{v},i}^{\,j+k-1}-\mathbf{s}_{\mathrm{v},i}^{\,k}-1, \mathbf{s}_{\mathrm{v},i}^{\,j+k}-\mathbf{f}_{\mathrm{v},i}^{\,k-1}-1\big) \end{aligned} \tag{16}$$

for the lost pair distances and, following similar arguments,

$$\begin{aligned} G(m) = &\sum_{i=1}^{n_\mathrm{rows}}\sum_{j=1}^{B^i}\sum_{k=1}^{B^i-j+1} G_c^h\big(m, \mathbf{f}_{\mathrm{h},i}^{\,j+k-1}-\mathbf{s}_{\mathrm{h},i}^{\,k}-1, C_y, \mathbf{s}_{\mathrm{h},i}^{\,j+k}-\mathbf{f}_{\mathrm{h},i}^{\,k-1}-1\big)\\ &+\sum_{i=1}^{n_\mathrm{columns}}\sum_{j=1}^{B^i}\sum_{k=1}^{B^i-j+1} G_c^v\big(m, C_x, \mathbf{f}_{\mathrm{v},i}^{\,j+k-1}-\mathbf{s}_{\mathrm{v},i}^{\,k}-1, \mathbf{s}_{\mathrm{v},i}^{\,j+k}-\mathbf{f}_{\mathrm{v},i}^{\,k-1}-1\big) \end{aligned} \tag{17}$$

for the gained pair distances, where n_rows (n_columns) is the number of rows (columns) that contain distinct clusters of inaccessible sites. A cluster that has $C_x > 1$ would contribute only once to n_columns rather than C_x times, and we note that the domain in Fig. 8(a) has $n_\mathrm{columns}=3$ and $n_\mathrm{rows}=2$.

Therefore, the expression for the counts of pair distances for a domain with obstacles is

$$D(m) = D_\mathrm{NO}(m) - A(m) + I(m) - L(m) + G(m), \tag{18}$$

where $D_\mathrm{NO}(m)$, $A(m)$, $I(m)$, $L(m)$, and $G(m)$ are defined in (1), (7), (8), (16), and (17), respectively. Again, we note that this expression is exact provided that the obstacles are arranged such that each obstacle has an entirely vacant row or column on all sides of the obstacle.

III. RESULTS

We first verify that the analytic expression (18) exactly calculates the counts of pair distances for a domain with obstacles. In Figs. 9(a), 9(c), 9(e), and 9(g) we present four different domains with inaccessible sites highlighted in black. For each domain we calculate the counts of pair distances numerically using Matlab's `graphshortestpath`, which calculates the shortest distance between any two points on a graph, given the adjacency matrix of the graph. This is the approach suggested by Gavagnin *et al.* [24] for calculating general PCFs, who note that computational cost is $O[L^3(2L-2)]$ for a square domain with L sites in each direction, which can become computationally infeasible even for modest L values. We then evaluate the analytic expression (18) and present the count of pair

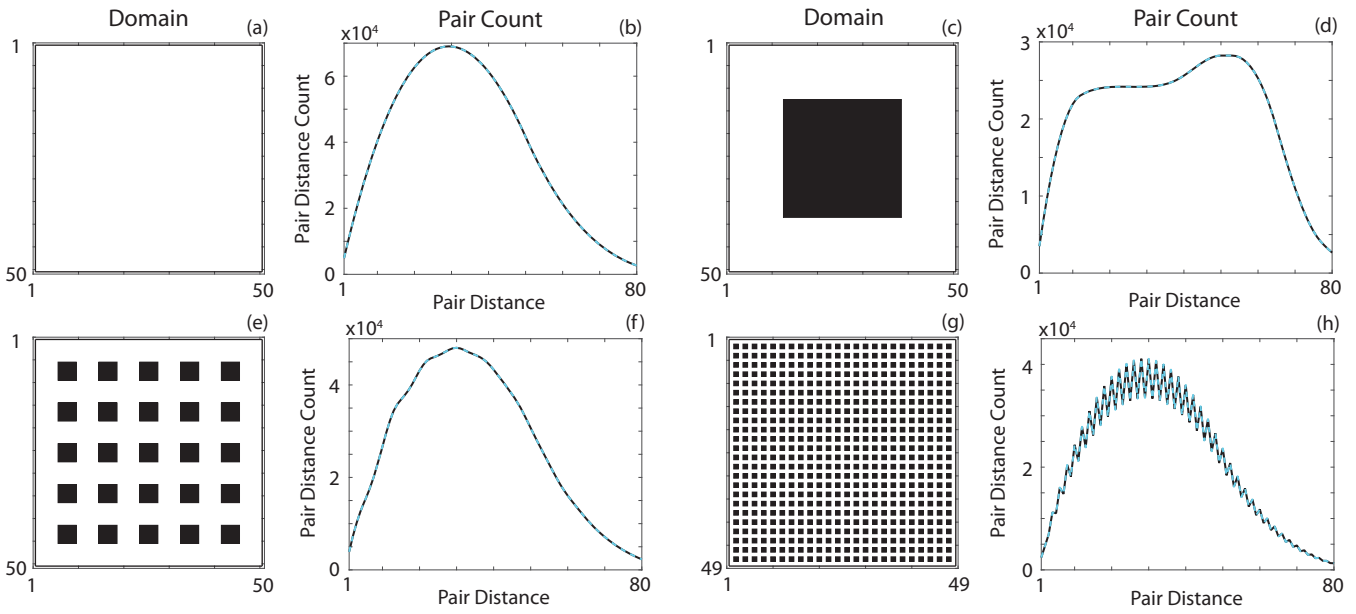

FIG. 9. (a), (c), (e), (g) Domains containing various configurations of inaccessible sites (black) and (b), (d), (f), (h) the corresponding count of pair distances obtained from the analytic expression (cyan) or numerical approach (black, dashed). The domain in panel (a) has zero inaccessible sites. The domain in panel (c) has one cluster of inaccessible sites, with 676 sites per cluster. The domain in panel (e) has 25 clusters of inaccessible sites, with 16 sites per cluster. The domain in panel (g) has 576 clusters of inaccessible sites, with one site per cluster.

distances in Figs. 9(b), 9(d), 9(f), and 9(h). We observe that the analytic and numerical counts are the same in each case.

We next examine the differences between the PCF and the oPCF for a range of domains that contain inaccessible sites. In Fig. 10 we present four domains, containing 0, 1, 25, and 576 clusters of inaccessible sites. The remaining sites are populated with agents (red) at random such that the average occupancy of accessible sites is 20%. We calculate both the PCF and the oPCF for 100 identically prepared realizations and present the functions in Figs. 10(b), 10(e), 10(h), 10(k), 10(c), 10(f), 10(i), and 10(l) for the PCF and the oPCF, respectively. As the accessible sites are populated randomly, there should be no pair correlation present, and hence $P(m) = 1$ for all m. Due to the small number of pairs of sites separated by a pair distance $m > 80$, and the subsequent higher variance in the calculated PCF and oPCF for these pair distances, we present the PCF and oPCF for $1 \leqslant m \leqslant 80$ (see Appendix B for further details). We note that this is a standard choice [19,24]. For the domain with zero inaccessible sites [Fig. 10(a)], we expect that the PCF and oPCF will be identical, because the oPCF reduces to the PCF in this case. As expected, we observe that the correlation functions in Figs. 10(b) and 10(c) are indistinguishable. For all three domains with inaccessible sites, the PCF incorrectly suggests that pair correlation is present within the images for a range of pair distances. For the domain in Fig. 10(d), the PCF implies that there is a mechanism that results in both short- and long-range aggregation, because the correlation value is greater than one for $m < 10$ and $m > 40$. Further, there appears to be a mechanism which inhibits clustering at intermediary distances. A similar trend is observed for the third domain [Fig. 10(g)], as well as regular oscillations in correlation for $m < 40$. For the fourth domain [Fig. 10(j)], these oscillations dominate the PCF. In contrast, the oPCF is approximately one for all pair distances and correctly indicates that there is no mechanism influencing clustering present in the locations of the agents. Hence, the correlation observed in Figs. 10(e), 10(h), and 10(k) is an artifact associated with the inaccessible sites.

We next compare the PCF and the oPCF for the same domains as analyzed previously, but for agents that follow a birth-movement exclusion-based random walk process. Such processes are discussed in detail elsewhere and have been used to mimic the behavior of a cell population [25,37]. Briefly, we populate accessible sites with z agents at random such that the average occupancy is 1%. Agents undergo birth and movement events with probabilities P_b and P_m per time step, respectively. During a time step, z agents are selected randomly with replacement and undergo birth events, where a daughter agent is placed at one of four randomly selected nearest-neighbor sites [37]. This birth event is successful if the selected nearest-neighbor site does not contain an agent, and the selected site is not inaccessible. After the birth events have been attempted, z agents are selected randomly with replacement to undergo movement events. During a movement event, one of four nearest-neighbor sites is selected at random, and an agent attempts to move to that lattice site [37]. Similar to the birth events, this event is successful if the selected site does not contain an agent and the selected site is not inaccessible. This random walk process is used because it is known to result in short-range correlation between agents due to the birth mechanism [25], since birth events inherently cause agents to be located at neighboring sites. This clustering tendency is countered by the movement mechanism, which acts in a diffusive manner. Hence, for higher ratios of P_b to P_m, we expect to see more short-range correlation, and for $P_m \gg P_b$, we expect to see no correlation.

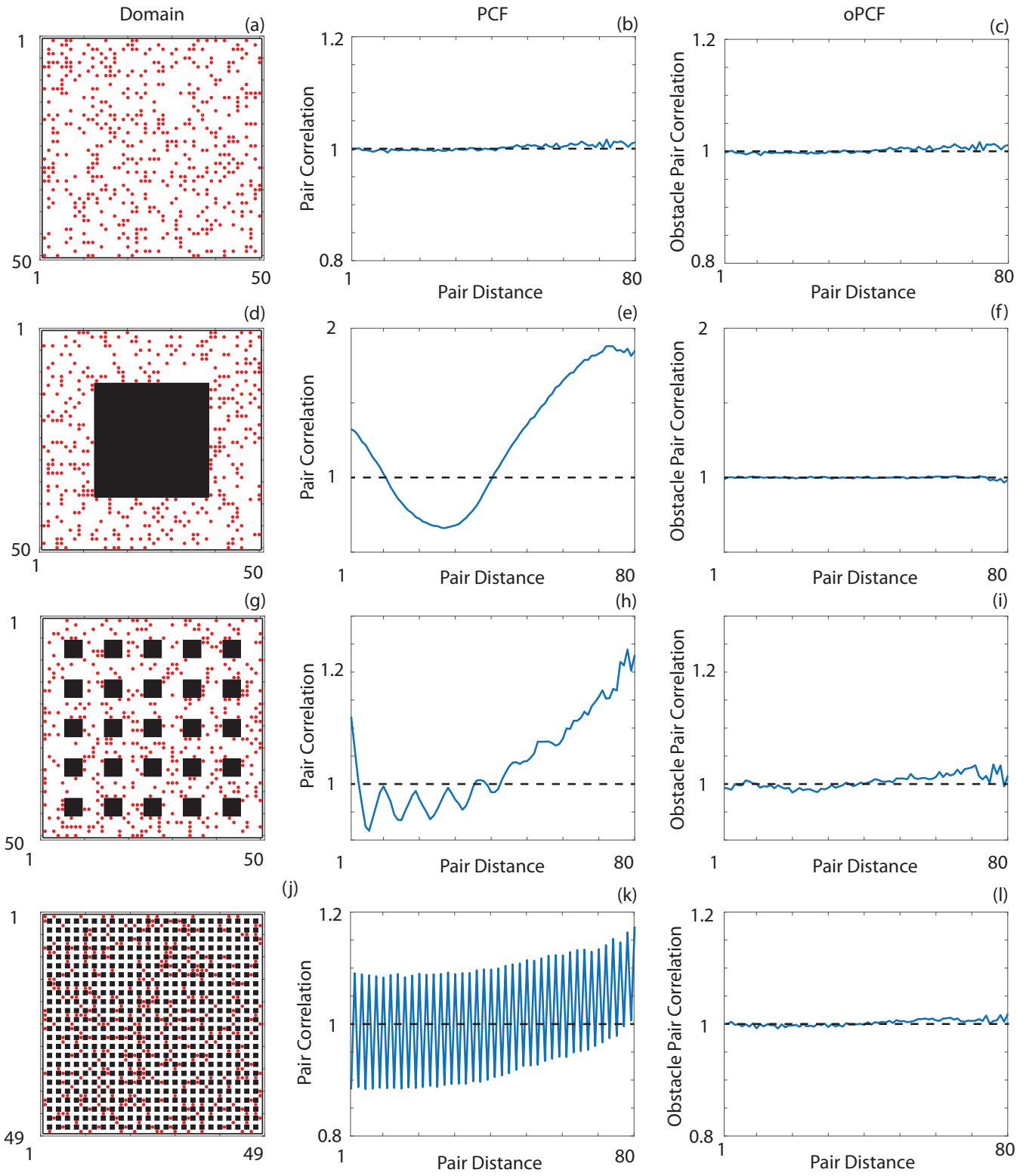

FIG. 10. (a), (d), (g), (j) Domains containing various configurations of inaccessible sites (black) with agents (red) randomly placed on accessible sites with the corresponding (b), (e), (h), (k) standard PCF, that is, the pair correlation calculated ignoring inaccessible sites, and (c), (f), (i), (l) oPCF. The dashed black line corresponds to no correlation. All PCFs are the average of 100 identically prepared domains.

Four representative snapshots of domain occupancy for agents following the birth-movement random walk process are shown in Figs. 11(a), 11(d), 11(g), and 11(j). In each simulation we use a final time that is weighted by the chance of successfully undergoing movement or birth, because the location and number of the inaccessible sites can influence this chance, and hence comparisons between simulations may have an effectively different final time. For the domain in

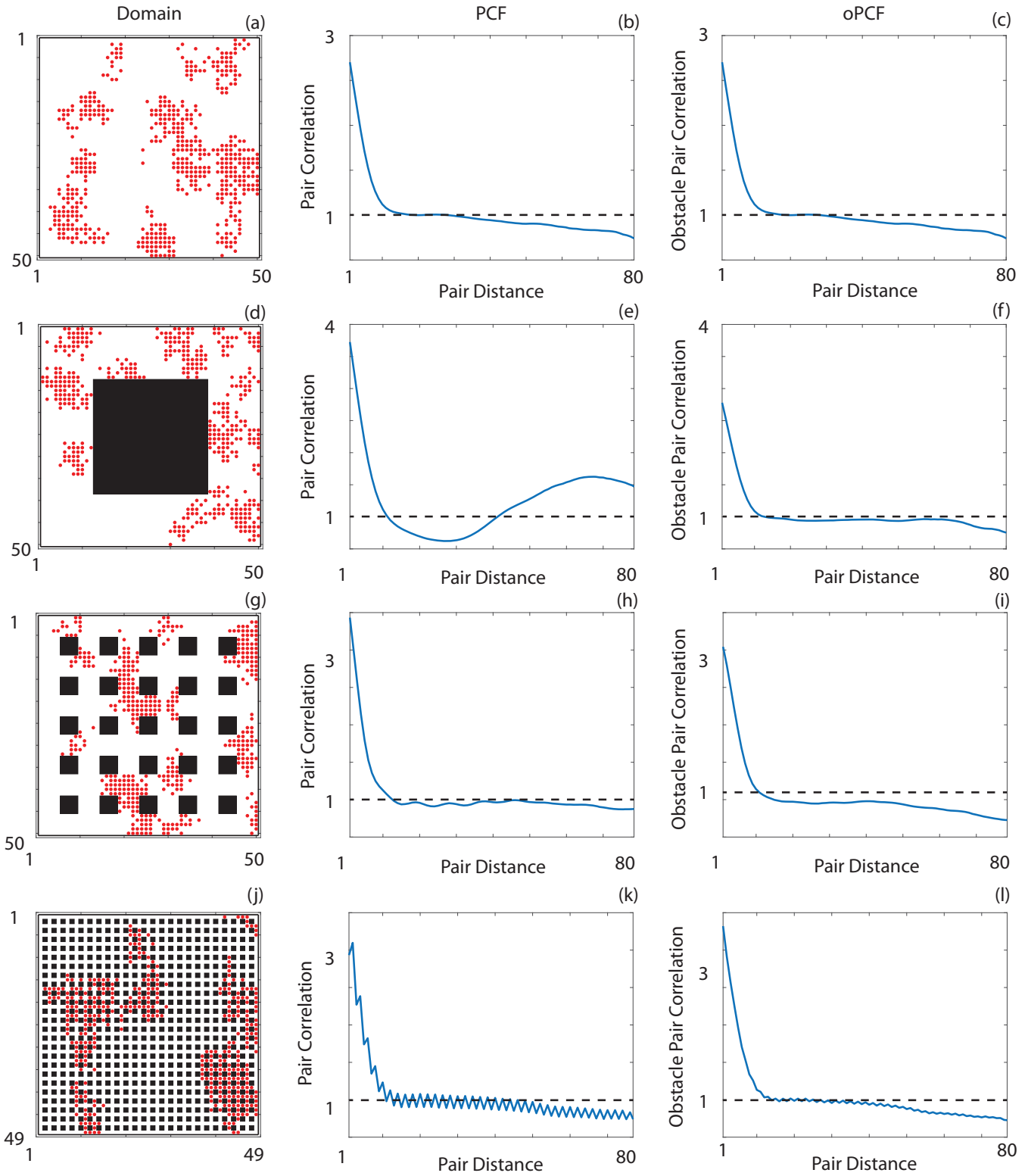

FIG. 11. (a), (d), (g), (j) Domains containing various configurations of inaccessible sites (black) with agents (red) located at accessible sites after undergoing a birth-movement random walk with the corresponding (b), (e), (h), (k) standard PCF, that is, the pair correlation calculated ignoring inaccessible sites, and (c), (f), (i), (l) oPCF. The dashed black line corresponds to no correlation. All PCFs are the average of 100 identically prepared domains and the subsequent realization of the random walk process.

Fig. 11(j), for example, there are many sites that have only two accessible neighbor sites. This can be compensated for by scaling the final time by the ratio of the number of accessible neighbor sites if all neighbor sites are accessible to the actual number of accessible neighbor sites in the domain. Hence, for Fig. 11(j), we scale the final time by approximately 1.48, as

TABLE I. Average time taken to evaluate the oPCF for a randomly occupied domain at 20% density with the specified number of clusters of inaccessible sites for the analytic expression and the numerical path-finding algorithm. Times reported are the average time taken for 100 randomly generated domains.

Domain	50 × 50	50 × 50	100 × 100	100 × 100	150 × 150	150 × 150
Number of clusters	25	100	25	100	25	100
Analytic time (s)	2.18	2.36	55.61	30.35	211.64	194.83
Numerical time (s)	6.48	6.41	310.85	75.58	738.05	573.98

there are 5196 out of a possible 7696 accessible neighbor sites. For all simulations $P_m = P_b = 0.1$ and $t_{\text{end}} = 70$, before scaling. Compared to the randomly occupied domains presented in Figs. 10(a), 10(d), and 10(g), 10(j) we immediately observe that the agents are located in clusters. When we calculate the PCFs for these domains, we therefore expect to observe pair correlation values greater than one for short distances. We present the PCF for all four domains in Figs. 11(b), 11(e), 11(h), and 11(k). While we observe that the pair correlation is greater than one at short distances in all four cases, the correlation at longer pair distances varies. For the domain with no inaccessible sites [Fig. 11(a)], the pair correlation is below one for $m > 20$ and above one for $m < 20$, as expected. In the second domain, the correlation is below one for intermediate pair distances and above one for $m > 40$. The correlation for the third domain is approximately one for intermediary and large m values. For the fourth domain, the pair correlation is only above one for short distances, and below one otherwise. However, there is an oscillatory pattern between odd and even pair distances. As the proliferation mechanism in the random walk process only explicitly produces correlation at a pair distance of one, and we expect this correlation to decay with pair distance due to the random movement mechanism, it is unlikely that these oscillations of this magnitude arise from the random walk process. To examine whether these correlations are indeed present due to the random walk mechanisms, we present the oPCF in Figs. 11(c), 11(f), 11(i), and 11(l) for the four domains. Again, for the domain with no inaccessible sites, the PCF and the oPCF are the same. In all cases, we observe the expected high correlation at short distance associated with birth events. For the domain in Fig. 11(d) the correlation is approximately one for the remainder of the pair distances. For the domains in Figs. 11(g), and 11(j), however, it appears that the correlation decreases with pair distance. This suggests that the restricted geometry of the domain may influence the spreading of the agents, even while scaling the final time. Interestingly, this decrease is also present in the PCF, for the domain in Fig. 11(j), albeit in the presence of significant oscillations. Comparing the results in Fig. 11 to Fig. 10, we observe less qualitative difference between the standard PCF and the oPCF. This suggests that qualitative differences between the standard PCF and the oPCF may be more pronounced for populations containing less aggregation. However, for populations displaying aggregation, obtaining robust quantitative measures of the pair correlation is critical. For all four domains the oPCF is relatively consistent, compared to the PCF, which is strongly domain dependent. As such, the oPCF provides a meaningful measure of the correlation present in the domain, because it is able to isolate the agent-agent correlation from spurious correlations arising from domain geometry.

Finally, we compare the time taken to evaluate the oPCF using both the path-finding algorithm described previously and the analytic expression (18). We consider domains of different sizes randomly occupied by agents such that 20% of the accessible sites contain an agent for different numbers of clusters of inaccessible sites. In Table I we present the time required to evaluate the oPCF if the domain is randomly generated with no restriction on the number of inaccessible sites, and in Table II we present the time required if the maximum number of inaccessible sites is restricted to 25% of the total number of sites. We note that the randomly generated domains considered here satisfy the restrictions required for the oPCF to be exact. In both cases, we observe that the numerical approach is always slower and requires between three and six times the computation time of the analytic approach. Interestingly, the numerical approach requires additional computation time for fewer inaccessible sites. Furthermore, as the numerical algorithm requires individual calculation of the distance from one site to all other sites for each site [approximately $(L_x L_y)^2$ algorithm realizations], memory issues become a considerable problem as the domain increases in size.

A. Approximation

The oPCF relies on a normalization term (18) that is exact only for certain configurations of inaccessible sites. As it is computationally intensive to determine the normalization term through many realizations of a path-finding algorithm [24], which is required for the exact counts of pair distances for

TABLE II. Average time taken to evaluate the oPCF for a randomly occupied domain at 20% density with the specified number of clusters of inaccessible sites for the analytic expression and the numerical path-finding algorithm. The maximum number of inaccessible sites is limited to 25% of the total number of sites. Times reported are the average time taken for 100 randomly generated domains.

Domain	50 × 50	50 × 50	100 × 100	100 × 100	150 × 150	150 × 150
Number of clusters	25	100	25	100	25	100
Analytic time (s)	3.81	3.16	62.24	56.23	345.20	352.57
Numerical time (s)	19.02	14.41	348.06	304.75	2053.61	2013.19

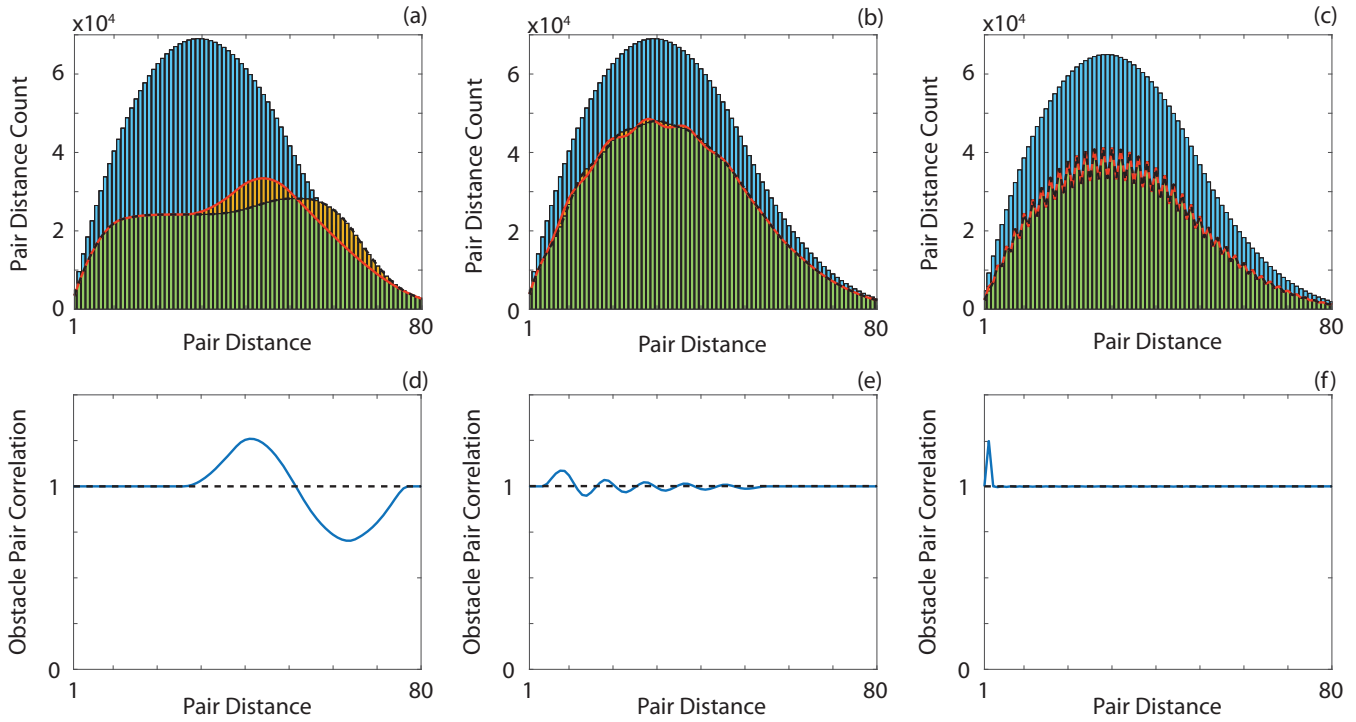

FIG. 12. (a)–(c) Pair distance count corresponding to the domains presented in Figs. 11(a), 11(d), and 11(g), respectively. Blue corresponds to pair distances obtained solely from $D_{\mathrm{NO}}(m)$ (i.e., uncorrected), green corresponds to pair distances from $D_{\mathrm{NO}}(m)$ corrected by $I(m)$ and $A(m)$, and orange represents the correction associated with $S(m)$. The corrected count of pair distances and the approximation are superimposed by the dashed black and red lines. (d)–(f) The approximate oPCF (blue) and exact oPCF (dashed black) for randomly located agents on the domains presented in Figs. 11(a), 11(d), and 11(g), respectively.

such configurations, it is of interest to examine whether an approximation of the analytic normalization term provides a sufficiently accurate alternative. As discussed previously, the corrected normalization term is composed of the standard pairs, accessible-inaccessible pairs, inaccessible-inaccessible pairs, and shifted pairs terms. The restriction to certain configurations of inaccessible sites is solely due to the shifted pairs, because the number of other pairs are calculated using standard distance metrics rather than path distance. Hence, if we do not consider the shifted pair distances, the restriction on inaccessible site configurations can be relaxed. We first examine the contributions of shifted pairs to the overall pair distances to determine the size of this contribution. As the shifted pairs consist of both negative and positive terms, corresponding to lost and gained pairs, respectively, the combined terms may provide only a small contribution to the total pairs. We note that for each lost pair there is a corresponding gained pair, and hence the total number of pairs is constant independently of whether the lost and gained pairs are considered. In Fig. 12 we present the contribution of the shifted pairs term to the overall count of pair distances for the three domains in Figs. 11(d), 11(g), and 11(j). The blue bars correspond to the standard pair distance counts, $D_{\mathrm{NO}}(m)$, the green bars correspond to standard pair distance counts corrected by accessible-inaccessible and inaccessible-inaccessible pairs, and the orange bars correspond to the correction associated with the shifted pairs. As such, the red line corresponds to the approximation of the pair distance count, and the black dashed line corresponds to the exact corrected pair distance count. We observe that the shifted pairs provide a small contribution, except in the case of a single large cluster of inaccessible sites [Fig. 12(a)]. As such, an approximation of the corrected normalization term may result in a valid approximation of the oPCF, provided that the domain is not dominated by a single large cluster of inaccessible sites. We note that for these domains we are able to compare the analytic pair distances with the approximation as the configuration of inaccessible sites means that the analytic function is exact. The approximation of the counts of the pair distances is

$$D_{\mathrm{approx}}(m) = D_{\mathrm{NO}}(m) - A(m) + I(m), \tag{19}$$

where $D_{\mathrm{NO}}(m)$, $A(m)$, and $I(m)$ are defined in (1), (7), and (8), respectively.

For the domains considered previously we present both the oPCF and the corresponding approximation in Figs. 12(d)–12(f). As expected, we see that for the domain with a large cluster of inaccessible sites, the approximation is poor for pair distances similar in size to the cluster. For the other two domains, the approximate oPCF performs well. Finally, we populate domains with inaccessible sites at random and calculate the approximate oPCF. All accessible sites on the domain are populated by agents such that the expected pair correlation is one for all pair distances. In Fig. 13 we present the average approximate oPCF for 50 random identically prepared domains for a range of numbers of inaccessible sites, as well as the mean error associated with each approximate oPCF. Intuitively, we observe that an increase in inaccessible sites corresponds to an increase in the distance between the

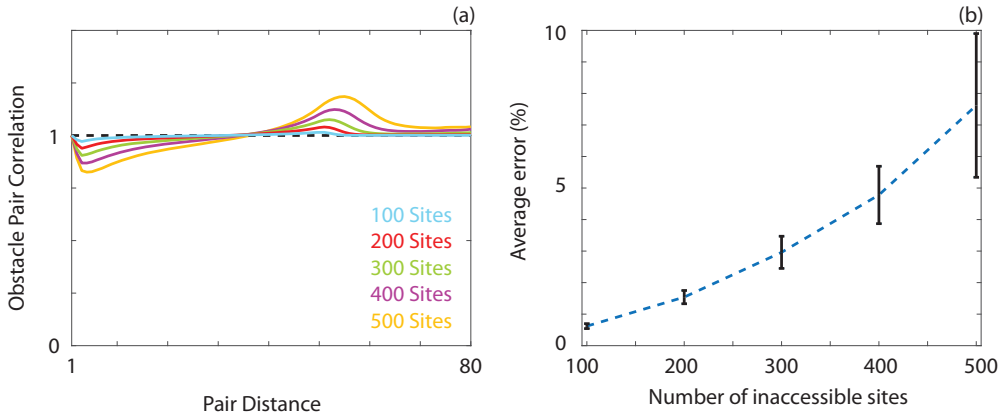

FIG. 13. (a) Approximate oPCF for 50 randomly generated domains containing 100 (cyan), 200 (red), 300 (green), 400 (purple), or 500 (orange) inaccessible sites for agents placed at random on accessible sites. The exact oPCF is shown via the dashed black line. (b) Error (mean $\pm$ one standard deviation) in the approximate oPCF, compared to the exact result, as a function of the number of inaccessible sites.

expected pair correlation and the approximation. Increasing the number of inaccessible sites while populating the domain with inaccessible sites at random introduces accessible sites that are not connected to the remainder of the domain, and hence we do not consider the oPCF approximation for higher numbers of inaccessible sites. As such, the approximation may prove useful for calculating pair correlations for domains where the inaccessible sites do not satisfy the conditions required for the exact pair distance calculation but have a modest proportion of inaccessible sites compared to accessible sites.

IV. DISCUSSION AND CONCLUSIONS

Analysis of the spatial structure present in experimental images provides valuable insight into the mechanisms governing behavior within the experiment [16,25,27]. Pair correlation functions have been employed in a variety of fields, including ecology [27], biology [22], and physics [18] and have proven useful for elucidating the presence and impact of spatial structure. Many experimental environments contain immobile obstacles that influence the transport and location of individuals within that environment [7–9]. Isolating the spatial structure associated with the mechanisms that govern transport, rather than the heterogeneous nature of the environment, is therefore of interest. Naively applying standard PCFs does not account for distances between pairs of individuals that must avoid obstacles and may result in the incorrect suggestion of spatial correlations.

Here we have presented an exact analytic expression for the normalization term of an obstacle PCF that incorporates a physical path distance between individuals and hence can be applied to environments with obstacles. We demonstrate that this oPCF is necessary for isolating the spatial correlation associated with the locations of individuals from the spatial correlation associated with the environment itself. Further, we highlight that the analytic expression allows for the oPCF to be calculated significantly faster than relying on a path-finding algorithm. We apply the oPCF to configurations arising from a lattice-based movement-birth random walk, which mimics cell motility and cell proliferation, where short-range correlation is known to exist, and demonstrate the oPCF recovers this correlation. Standard PCFs can introduce spurious correlations as well as oscillations in the correlation. Finally, we present an approximation to the oPCF that relaxes assumptions on the locations of the inaccessible sites within the domain and show that for modest numbers of inaccessible sites, the approximation is accurate.

The work and analysis presented here could be extended in a number of directions. One obvious application is to calculate the PCF for experimental data obtained from an environment that contains obstacles. Here we have focused on data resulting from simulations that mimic processes such as cell migration and proliferation [25] rather than explicitly using experimental data. As previous investigations involving the application of PCFs to experimental data have proved fruitful [17,22,25], the application of the oPCF to appropriate data may prove to be insightful. Another promising approach would be to examine how the pair correlation changes between two experiments on different domains. As the oPCF is able to isolate the correlation associated with the behavior of individuals, it would be instructive to consider whether the behavior is dependent on environment and, if so, quantify which mechanisms are are responsible for this change in behavior.

ACKNOWLEDGMENTS

This research was in part conducted and funded by the Australian Research Council Centre of Excellence in Convergent Bio-Nano Science and Technology (Project No. CE140100036). We thank Pradeep Rajasekhar and Daniel Poole for providing the image of the nervous system used in Fig. 1(a).

APPENDIX A: STANDARD COUNTS OF PAIR DISTANCES DERIVATION

Consider an arbitrary domain with no obstacles and L_x by L_y lattice sites. For pair distances that satisfy $1 \leqslant m \leqslant \min(L_x, L_y)$, the number of pairs of sites separated by a distance m was presented by Gavagnin *et al.* [24]. However, the maximum pair distance on such a domain with no-flux

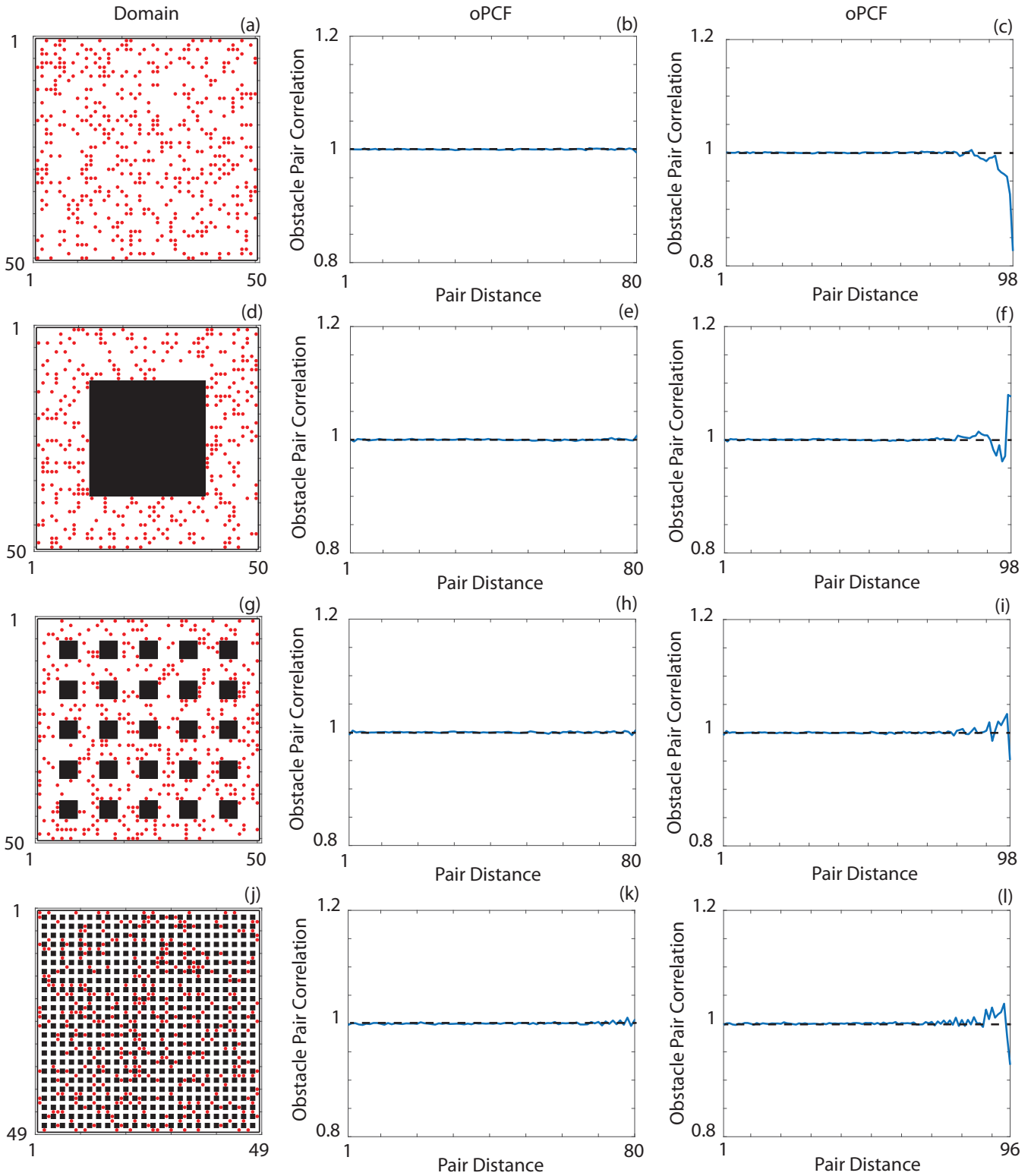

FIG. 14. (a), (d), (g), (j) Domains containing various configurations of inaccessible sites (black) with agents (red) randomly placed on accessible sites with the corresponding (b), (e), (h), (k) oPCF for $1 \leqslant m \leqslant 80$ and (c), (f), (i), (l) oPCF for $1 \leqslant m \leqslant L_x + L_y - 2$. The dashed black line corresponds to no correlation. All PCFs are the average of 1000 identically prepared domains.

boundary conditions is $L_x + L_y - 2$, and we therefore must derive counts of pair distances for $m > \min(L_x, L_y)$.

We first consider pair distances where $\min(L_x, L_y) < m < \max(L_x, L_y)$, that is, pairs of sites that cannot be connected via a path containing only "jumps" in the shorter of the horizontal or vertical directions. For example, if $L_y < L_x$, the path connecting pairs of sites separated by distance $\min(L_x, L_y) < m < \max(L_x, L_y)$ must contain either entirely

horizontal "jumps" or a combination of horizontal and vertical "jumps." For a particular m, the number of pairs of sites separated by m horizontal "jumps" is $L_y(L_x - m)$. This term is the product of the number of pairs of sites separated by m horizontal "jumps" within a single row, $L_x - m$, and the number of rows, L_y. The number of sites separated by $m - 1$ horizontal "jumps" and one vertical "jump" is $2(L_y - 1)(L_x - m + 1)$. Compared to the m horizontal "jumps" case, there is an additional number of pairs of sites separated by $m - 1$ horizontal "jumps," providing the $L_x - m + 1$ component. However, as each distance contains a vertical "jump," the rows containing the sites must be offset by one, and hence there is one less row where this can occur, which results in the $L_y - 1$ term. Finally, as the vertical "jump" can be in either the positive or negative vertical direction, this introduces the factor of two. Introducing additional vertical "jumps" increases the number of pairs of sites separated by the resulting smaller number of horizontal "jumps," while decreasing the effective number of rows. Noting that L_y and L_x are interchangeable, we therefore obtain

$$D_{\mathrm{NO}}(m) = \min(L_x, L_y)[\max(L_x, L_y) - m] + 2 \sum_{i=1}^{\min(L_x,L_y)} [\min(L_x, L_y) - i][\max(L_x, L_y) - m + i],$$

$$= D_{\mathrm{NO}}(\min(L_x, L_y)) - \min(L_x, L_y)^2(m - \min(L_x, L_y))$$

for $\min(L_x, L_y) < m < \max(L_x, L_y)$.

We next consider pair distances $m \geqslant \max(L_x, L_y)$, where distances between sites must include both horizontal and vertical "jumps." Again, without loss of generality, we assume that $L_y \leqslant L_x$. The minimum number of vertical "jumps" for a distance m is $m - L_x + 1$, and the maximum number is L_y. The corresponding number of pairs of rows that are separated by this vertical distance v is $L_y - v$, and the number of pairs of sites separated by this vertical distance and the requisite horizontal distance $h = m - v$ is $L_x - m + v$. Following the same process as above, taking a summation over the possible rows and columns, and noting that the vertical separation can be either in the positive or negative direction, we obtain

$$D_{\mathrm{NO}}(m) = 2 \sum_{i=m-\max(L_x,L_y)+1}^{\min(L_x,L_y)} [\min(L_x, L_y) - i][\max(L_x, L_y) - m + i]$$

$$= \frac{k(k+1)(k+2)}{3}, \quad \text{where } k = L_x + L_y - 1 - m$$

for $m \geqslant \max(L_x, L_y)$.

APPENDIX B: LARGE PAIR DISTANCES

As discussed previously, a standard choice is to present the pair correlation up to a threshold pair distance [19,24]. This is due to the small number of pairs of sites separated by large pair distances and the associated variability in the pair correlation at these distances when the domain is populated at random. For completeness, in Fig. 14 we present a comparison of the oPCF for $1 \leqslant m \leqslant 80$ and $1 \leqslant m \leqslant L_x + L_y - 2$. We note that the oPCF is calculated in an identical manner as in Fig. 10, with the exception that we use 1000 realizations of the domain population process instead of 100. We observe that the variability in the oPCF for $1 \leqslant m \leqslant 80$ is small [Figs. 14(b), 14(e), 14(h), and 14(k)] compared to the variability for $80 \leqslant m \leqslant L_x + L_y - 2$ [Figs. 14(c), 14(f), 14(i), and 14(l)]. As this variability is present for large m, even with an additional 900 realizations of the domain population process, and the small number of pairs of sites separated by large m, we make the standard choice to present the pair correlation up to a threshold pair distance.

[1] A. Lerner, Y. Chrysanthou, and D. Lischinski, Comput. Graph. Forum **26**, 655 (2007).
[2] F. Dietrich and G. Köster, Phys. Rev. E **89**, 062801 (2014).
[3] G. Gabella, J. Auton. Nerv. Syst. **30**, S59 (1990).
[4] R. Grima, Theor. Biol. Med. Model. **4**, 2 (2007).
[5] D. Helbing, L. Buzna, A. Johansson, and T. Werner, Transp. Sci. **39**, 1 (2005).
[6] F. Höfling and T. Franosch, Rep. Prog. Phys. **76**, 046602 (2013).
[7] M. Moussaïd, D. Helbing, and G. Theraulaz, Proc. Natl. Acad. Sci. USA **108**, 6884 (2011).
[8] E. Rehder and H. Kloeden, in *Proceedings of the IEEE International Conference on Computer Vision Workshops* (IEEE Computer Society, Washington, DC, 2015), pp. 50–58.
[9] F. Roosen-Runge, M. Hennig, F. Zhang, R. M. Jacobs, M. Sztucki, H. Schober, T. Seydel, and F. Schreiber, Proc. Natl. Acad. Sci. USA **108**, 11815 (2011).
[10] A. Rühl, Neurogastroenterol. Motil. **17**, 777 (2005).
[11] M. J. Simpson and M. J. Plank, Results Phys. **7**, 3346 (2017).
[12] A. Varas, M. D. Cornejo, D. Mainemer, B. Toledo, J. Rogan, V. Munoz, and J. A. Valdivia, Physica A **382**, 631 (2007).
[13] R. Alizadeh, Saf. Sci. **49**, 315 (2011).
[14] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, Piscataway, NJ, 2009), pp. 3931–3936.

[15] M. Chraibi, A. Seyfried, and A. Schadschneider, Phys. Rev. E **82**, 046111 (2010).
[16] J. N. Perry, A. M. Liebhold, M. S. Rosenberg, J. Dungan, M. Miriti, A. Jakomulska, and S. Citron-Pousty, Ecography **25**, 578 (2002).
[17] D. J. G. Agnew, J. E. F. Green, T. M. Brown, M. J. Simpson, and B. J. Binder, J. Theor. Biol. **352**, 16 (2014).
[18] N. A. Bahcall and R. M. Soneira, Astrophys. J. **270**, 20 (1983).
[19] B. J. Binder and M. J. Simpson, Phys. Rev. E **88**, 022705 (2013).
[20] B. J. Binder and M. J. Simpson, R. Soc. Open Sci. **2**, 140494 (2015).
[21] R. N. Binny, P. Haridas, A. James, R. Law, M. J. Simpson, and M. J. Plank, PeerJ **4**, e1689 (2016).
[22] S. Dini, B. J. Binder, and J. E. F. Green, J. Theor. Biol. **439**, 50 (2018).
[23] M. Holzmann and Y. Castin, Eur. Phys. J. D **7**, 425 (1999).
[24] E. Gavagnin, J. P. Owen, and C. A. Yates, Phys. Rev. E **97**, 062104 (2018).
[25] S. T. Johnston, M. J. Simpson, D. S. McElwain, B. J. Binder, and J. V. Ross, Open Biol. **4**, 140097 (2014).
[26] S. T. Johnston, J. V. Ross, B. J. Binder, D. S. McElwain, P. Haridas, and M. J. Simpson, J. Theor. Biol. **400**, 19 (2016).
[27] R. Law, J. Illian, D. F. Burslem, G. Gratzer, C. Gunatilleke, and I. Gunatilleke, J. Ecol. **97**, 616 (2009).
[28] A. J. Flügge, S. C. Olhede, and D. J. Murrell, Ecology **93**, 1540 (2012).
[29] T. Rajala, S. Olhede, and D. J. Murrell, J. Ecol. **107**, 711 (2019).
[30] https://github.com/DrStuartJohnston/obstacle-pair-correlation-function.
[31] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, Physica A **295**, 507 (2001).
[32] A. J. Ellery, R. E. Baker, and M. J. Simpson, Phys. Biol. **12**, 066010 (2015).
[33] A. J. Ellery, R. E. Baker, and M. J. Simpson, J. Chem. Phys. **144**, 171104 (2016).
[34] A. J. Ellery, R. E. Baker, and M. J. Simpson, Phys. Biol. **13**, 05LT02 (2016).
[35] D. V. Nicolau Jr., J. F. Hancock, and K. Burrage, Biophys. J. **92**, 1975 (2007).
[36] A. Wedemeier, H. Merlitz, C.-X. Wu, and J. Langowski, J. Chem. Phys. **131**, 064905 (2009).
[37] M. J. Simpson, K. A. Landman, and B. D. Hughes, Physica A **389**, 3779 (2010).