



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/245490/>

Version: Published Version

Article:

Kolesnik, B., Zakharov, G. and Zhukovskii, M. (2026) On the threshold for triangulations inside convex polygons. *Random Structures & Algorithms*, 69 (2). e70093. ISSN: 1042-9832

<https://doi.org/10.1002/rsa.70093>

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

RESEARCH ARTICLE OPEN ACCESS

On the Threshold for Triangulations Inside Convex Polygons

Brett Kolesnik¹  | Georgii Zakharov² | Maksim Zhukovskii³ 

¹Department of Statistics, University of Warwick, Warwick, UK | ²Department of Mathematics and Exeter College, University of Oxford, Oxford, UK | ³School of Computer Science, University of Sheffield, Sheffield, UK

Correspondence: Brett Kolesnik (brett.kolesnik@warwick.ac.uk)

Received: 10 December 2025 | **Revised:** 4 April 2026 | **Accepted:** 12 August 2026

Keywords: bootstrap percolation | Catalan numbers | critical threshold | graph bootstrap percolation | long-range dependency | oriented percolation | polluted bootstrap percolation | weak saturation

ABSTRACT

Start with a large convex polygon and add all other edges inside independently with probability p . At what critical threshold p_c do triangulations of the polygon begin to appear? The first author and Gravner asked this question and observed that $p_c = \Theta(1)$, using the relationship with the Catalan numbers and a coupling with oriented site percolation on $\mathbb{Z}^2$. More recently, Archer, Hartarsky, the first author, Olesker-Taylor, Schapira, and Valesin proved that $1/4 < p_c < p_c^o$, where $1/4$ is the Catalan exponential growth rate and p_c^o is the critical threshold for oriented percolation. The upper bound is strict, but nonquantitative and follows by a renormalization argument. In this work, we show that $p_c < 1/2$ using a simple ear clipping algorithm, which can be analyzed by comparing with random walks with negative drift. This bound is closer to the truth (perhaps near 0.4) and shows that most configurations of edges inside large convex polygons contain triangulations.

MSC2020 Classification: 60K35, 82B43

1 | Introduction

Let P_n be a convex polygon with vertices labeled by $\{1, \dots, n\}$. We further include a set $E_{n,p}$ of random edges obtained by adding all other edges inside P_n independently with probability p . We are interested in the critical point p_c at which triangulations of P_n appear. More formally, if $\mathcal{T}_{n,p}$ is the event that P_n can be triangulated using some subset of the edges of $E_{n,p}$, then

$$p_c = \inf\{p : \liminf_{n \rightarrow \infty} \mathbb{P}(\mathcal{T}_{n,p}) > 0\}$$

is the critical threshold at which triangulations begin to emerge.

In this work, we show the following:

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *Random Structures & Algorithms* published by Wiley Periodicals LLC.

Theorem 1.1. *There exists $\varepsilon > 0$ such that $p_c < 1/2 - \varepsilon$.*

We actually show that $p_c \leq p_*$, where $p_* \approx 0.4916$. Theorem 1.1 implies that most configurations of edges inside a large convex polygon can be used to triangulate the polygon.

1.1 | Previous Results

Recent work by Archer et al. (2026) shows that $p_c < p_c^o$, where p_c^o is the critical threshold for oriented site percolation on the integer lattice $\mathbb{Z}^2$; see Durrett (1984). Numerical simulations suggest that $p_c^o \approx .7055$; see Essam et al. (1988). On the other hand, the numerics in Archer et al. (2026) appear to indicate that p_c is much smaller, perhaps somewhere between 0.39 and 0.41; see Fig. 3 in Archer et al. (2026).

The connection with oriented percolation was already observed by Gravner and Kolesnik (2023) (see Theorem 1.3, Section 3 and Conjecture 6.1 therein), where the problem of finding p_c was referred to as *Catalan percolation*, as a special case of a certain transitive closure dynamics in polluted environments (see Section 1.2 below). The motivation in Gravner and Kolesnik (2023) was to bring together ideas from *weak saturation* (see Bollobás (1968) and Balogh et al. (2012)) and *polluted bootstrap percolation* (see Gravner and McDonald (1997)), in response to the final paragraph in Balogh et al. (2012) (see p. 439 therein).

Let us remark that although Theorem 1.1 improves the upper bound in Archer et al. (2026), it is the *proof* rather than the *result* that is the main contribution in Archer et al. (2026). Indeed, the coupling, first observed in Gravner and Kolesnik (2023), with oriented percolation is based on a significant restriction of the full Catalan dynamics (see Section 1.2 below). As such, $p_c < p_c^o$ is certainly not unexpected. However, due to long-range, non-decaying correlations in the model, it is not straightforward to deduce a *strict* inequality using standard techniques from percolation (e.g., the method of *essential enhancements* of Aizenman and Grimmett (1991) does not apply); see Section 1.2 in Archer et al. (2026) for a detailed discussion. As such, ideas in the proof in Archer et al. (2026) may be useful in analyzing other oriented percolation models, beyond the specific case of Catalan percolation. Indeed, the dynamics studied in Archer et al. (2026) can be thought of as a directed version of *Brochette percolation*, as studied by Duminil-Copin et al. (2018).

1.2 | Equivalence With Catalan Percolation

The perspectives taken in Gravner and Kolesnik (2023) and Archer et al. (2026) are conceptually different than ours, but are formally equivalent, as we will now explain.

In Gravner and Kolesnik (2023) the authors considered the following situation: start by initially *infecting* all directed nearest-neighbor edges along the integer line path from 1 to n . Then *open* all other leftward (resp. rightward) directed edges $i \leftarrow j$ (resp. $i \rightarrow j$), for $1 \leq i < j \leq n$ and $j - i > 1$, with some probability p_ℓ (resp. p_r). All other directed edges are *polluted* and can never become infected. Open edges, on the other hand, can become infected by the following *transitive closure* dynamics: if at some point there are directed edges $i \rightarrow j \rightarrow k$ which are both infected (initially or otherwise) then $i \rightarrow k$ becomes infected if it is open. This model is introduced in Gravner and Kolesnik (2023) as a simple model for the spread of information in the presence of censorship. (In fact, other initial graphs, besides paths, are considered in Gravner and Kolesnik (2023)).

What is called *Catalan percolation* in Gravner and Kolesnik (2023) is the special case in which $p_\ell = 0$ and $p_r > 0$. (When both $p_\ell, p_r > 0$ the behavior is very different, and still not fully understood.) In this case the dynamics have a simple graphical description, using undirected edges, in which open edges $\{i, k\}$ become infected if there are two infected edges $\{i, j\}$ and $\{j, k\}$ “underneath” the edge $\{i, k\}$.

In Archer et al. (2026), the authors let $\varphi_n(p)$ be the probability that the edge $\{1, n\}$ joining the endpoints of the path from 1 to n is eventually infected, conditional on it being open, and put $p_c = \inf\{p : \liminf_{n \rightarrow \infty} \varphi_n(p) > 0\}$. However, the eventual infection of $\{1, n\}$ by these dynamics, assuming that it is open, is equivalent to the existence of a triangulation of P_n using the edges in $E_{n,p}$; see Figure 1 below.

The Catalan numbers are, of course, related to many combinatorial objections. In Gravner and Kolesnik (2023) and Archer et al. (2026), the connection is, in some sense, viewed in terms of parenthesizations of a product. In the current work, we work with triangulations instead.

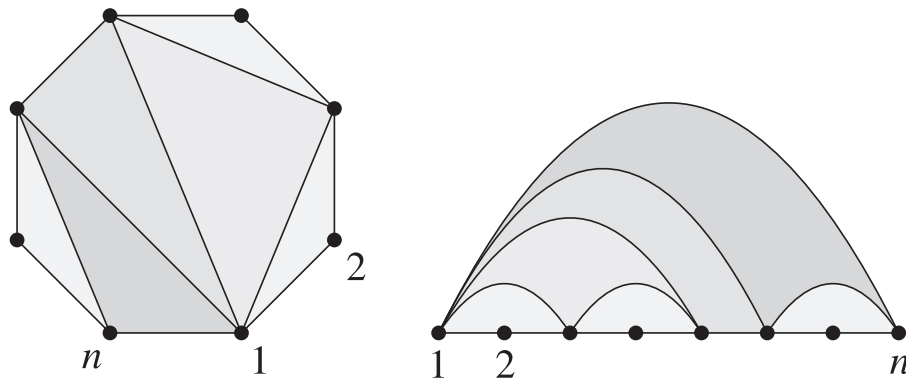


FIGURE 1 | Comparing Catalan percolation to the existence of a triangulation.

The coupling with oriented percolation, used in Gravner and Kolesnik (2023) and Archer et al. (2026), arises from a simple restriction of the Catalan percolation dynamics, where $\{i, k\}$ is only infected if at least one of the infected edges $\{i, j\}$ or $\{j, k\}$ underneath it is a nearest-neighbor edge.

1.3 | Our Strategy

In this work, we recast Catalan percolation in a new light, in terms of triangulations, which leads to better bounds and simpler proofs. The proof of Theorem 1.1 gives an upper bound of about 0.4916, and elaborations on our methods would yield further improvements. Our methods are rather elementary. We simply note that many natural ear clipping algorithms, using the random edges $E_{n,p}$, are related to random walks.

We recall that an *ear* in a triangulation of a polygon is a triangle with two adjacent edges along the boundary of the polygon. Removing such a triangle leaves us with a smaller polygon to triangulate. In discussing our recursive algorithms, when we refer to an ear, we mean an ear with respect to the convex polygon that currently remains, after all clippings up to this point have been performed. Such an ear may, of course, not be an ear with respect to the original polygon P_n .

In Sections 3 and 4, we use a simple greedy ear clipping algorithm to show that $p_c \leq 1/2$. The main idea is that, when $p > 1/2$, there is a natural comparison to simple random walks with negative drift. In this range of p , the greedy algorithm with high probability finds a triangulation of P_n in linear $O(n)$ time. In Section 5, we establish a strict inequality $p_c < 1/2$ by relaxing the greedy dynamics to allow for additional flexibility at the local decision level. We show that for p just below $1/2$ this continues to allow for a comparison to a (slightly more complicated) random walk with negative drift. Expanding this local neighborhood further would lead to better upper bounds on p_c but, as it would appear, at the expense of increasingly more elaborate arguments. See Section 6 below for more on this.

1.4 | Ear Clipping Comparison

Finally, to further compare with the previous works by Gravner and Kolesnik (2023) and Archer et al. (2026) discussed above, let us note that the proof of the oriented percolation bound can also be viewed in terms of a certain ear clipping strategy. This strategy, however, is much more restricted, requiring that each ear being clipped contains at least one edge in the original polygon P_n .

Basically, in the coupling with oriented percolation, the orientation of a step corresponds to either clipping directly to the left or right of the currently clipped region; see Fig. 3 in Gravner and Kolesnik (2023) and Fig. 6 in Archer et al. (2026). The strict inequality in Archer et al. (2026) essentially comes from studying a process that can clip slightly deeper into the polygon, by also clipping ears of P_n if they are adjacent to the currently clipped region; see Fig. 8 in Archer et al. (2026).

In this work, on the other hand, we consider exploration processes that can clip well into the polygon, and this opens up a lot more combinatorial freedom; see Figure 3 below.

1.5 | Related Works

In closing, let us mention a few other works in the literature with some connection to ours. First, we recall that Bollobás and Frieze (1991) found the thresholds for spanning maximal planar/outerplanar subgraphs of the Erdős–Rényi random graph $\mathcal{G}_{n,p}$. In particular, their result implies the threshold probability, up to a polylogarithmic factor, for the appearance of a triangulation of a triangle (i.e., a triangulation with the maximum possible number of “internal” vertices) and a triangulation of an n -gon (a somewhat opposite case when there are no internal vertices in a triangulation). A precise order of magnitude for the former threshold follows from a later result of Riordan (2000). The threshold for the latter case follows, up to a constant factor, from the recent results on the square of a Hamiltonian cycle by Kahn et al. (2021) and Zhukovskii (2025).

2 | Random Walks With Negative Drift

We recall that a *random walk* $(S_n, n \geq 0)$ on $\mathbb{Z}$ is determined by an independent and identically distributed sequence of random variables $(X_k, k \geq 1)$, called the *increments* of the walk. If the walk is started at $x \in \mathbb{Z}$ then $S_n = x + \sum_{k=1}^n X_k$. In this work, we use the following classical result, concerning the first time $\tau = \inf\{n > 0 : S_n < 0\}$ that the walk is negative.

Lemma 2.1. *Suppose that a random walk $(S_n, n \geq 0)$ started at $x \geq 0$ has a negative drift $\mu = \mathbb{E}(X_1) < 0$ and bounded increments; that is, for some $a, b > 0$, we have that $\mathbb{P}(-a \leq X_i \leq b) = 1$. Then $\mathbb{P}(\tau > n) = O(e^{-\gamma n})$, for some $\gamma > 0$.*

By the law of large numbers $S_n/n \rightarrow \mu < 0$, so even just $S_n \geq 0$ (which must hold if $\tau > n$) is an exponentially small large deviations event. In fact, the above result holds with the bounded increments condition replaced with the *Cramér condition* that $\mathbb{E}(e^{\lambda X_1}) < \infty$ for some $\lambda > 0$. The precise asymptotics of $\mathbb{P}(\tau > n)$ are known in many cases (see Doney (1989)) but the above bounds will serve our purposes.

3 | Greedy Ear Clipping Algorithm (GECA)

In this section, we describe a natural *greedy ear clipping algorithm* (GECA), which will play an important role in our proofs.

Input. As input, GECA takes in:

- a polygon P with vertices labelled by $v_1, \dots, v_n$ in counter-clockwise (CCW) order around the boundary of P ; and
- some set of edges E inside P .

Output. If successful, GECA returns:

- a polygon P' with vertices $v_1, v_\tau, \dots, v_n$, for some $3 \leq \tau \leq n$; and
- a triangulation of the region inside P to the right of $\{v_1, v_\tau\} \in E$ (i.e., the region delimited by the edge $\{v_1, v_\tau\}$ and the CCW path from v_1 to v_τ in P_n).

Algorithm. We start with $P^{(0)} = P$ and $\mathcal{L}^{(0)} = (v_1, v_2, v_3)$ and will construct recursively $P^{(k)}$ and $\mathcal{L}^{(k)}$. In the k th step of GECA, we have a polygon $P^{(k)}$ and a list

$$\mathcal{L}_k = (v_1 = v_1^{(k)}, \dots, v_{m_k}^{(k)})$$

of vertices in a CCW path along the boundary of $P^{(k)}$ starting from v_1 . Then, in the $(k+1)$ th step of GECA, we proceed as follows:

- **Clipping step:** If $\{v_{m_k-2}^{(k)}, v_{m_k}^{(k)}\} \in E$, then we obtain $P^{(k+1)}$ from $P^{(k)}$ by clipping the ear on $\{v_{m_k-2}^{(k)}, v_{m_k-1}^{(k)}, v_{m_k}^{(k)}\}$. In this case, we put

$$\mathcal{L}_{k+1} = (v_1^{(k)}, \dots, v_{m_k-2}^{(k)}, v_{m_k}^{(k)}).$$

That is, we obtain $\mathcal{L}_{k+1}$ from $\mathcal{L}_k$ by removing $v_{m_k-1}^{(k)}$.

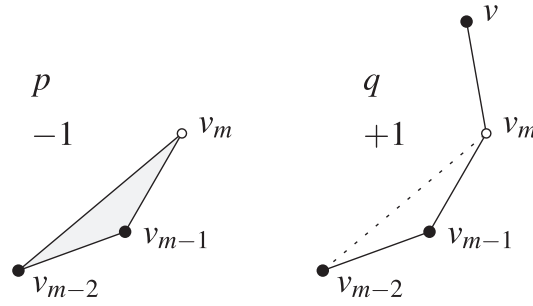


FIGURE 2 | In GECA, we clip an ear if possible, and otherwise move to the next vertex along the boundary of P . The key observation is that, when using random edges from $E_{n,p}$ inside the polygon, the length of the list decreases by 1 with probability p and increases by 1 with probability $q = 1 - p$. If $p > 1/2$ this gives rise to a random walk with negative drift.

- **Extending step:** Otherwise, if $\{v_{m_k-2}^{(k)}, v_{m_k}^{(k)}\} \notin E$, let $P^{(k+1)} = P^{(k)}$ and

$$\ell_{k+1} = (v_1^{(k)}, \dots, v_{m_k}^{(k)}, v),$$

where v is the next vertex after $v_{m_k}^{(k)}$ in CCW order along the boundary of P .

See Figure 2 for an illustration.

Termination. GECA continues in this way, until eventually in some step k either:

- **Success:** $\ell_k = (v_1, v_\tau)$, for some $3 \leq \tau \leq n$, in which case (by induction) GECA has triangulated the region inside P to the right of $\{v_1, v_\tau\} \in E$, and we terminate the algorithm and return $P' = P^{(k)}$; or else
- **Failure:** the last vertex in the list ℓ_k is $v_{m_k}^{(k)} = n$ and $\{v_{m_k-2}^{(k)}, v_{m_k}^{(k)}\} \notin E$, in which case we say that GECA has failed and terminate the algorithm.

The following observation will play a key role in our arguments, as it allow us to connect GECA with a random walk.

Lemma 3.1. *Starting with a polygon P , let us run GECA and reveal the edges E inside P on a need-to-know basis. Recall that the k th step of GECA is determined by whether the edge $\{v_{m_k-2}^{(k)}, v_{m_k}^{(k)}\}$ is in E (the step is clipping if so and extending otherwise). The status of this edge is revealed in step k , and is not involved in any earlier steps $j < k$.*

Proof. In each step of GECA, we check to see if the vertex v_m at the end of the current list (the open dot in Figure 2) is adjacent to some vertex v_{m-2} in the CCW path along P_n from 1 to v_m (i.e., to the “left” of v_m). In an extending step, we add a new vertex v to the end of the current list, and so, in the next step, this will be the first time that an edge incident to v has been checked. On the other hand, after a clipping step we clip an ear involving the edge most recently checked. Moreover, this edge will become a boundary edge of the polygon in the next step, and so will never be checked again. $\square$

4 | Greedy Triangulation Algorithm (GTA)

We show that if $p > 1/2$ then we can find a triangulation of P_n using a certain greedy algorithm based on GECA.

Proposition 4.1. *For every fixed $p > 1/2$, with high probability, we can find a triangulation of P_n in linear time. In particular, $p_c \leq 1/2$.*

The proof is almost as simple as iterating GECA, and using Lemmas 2.1 and 3.1 to bound the time between iterations. However, as we will see, some care is required as we near the end of our tour of the boundary of P_n .

To overcome this issue, we fix a set $B = \{n - b, \dots, n\}$ of vertices, which we will call the *buffer*. Here, $b = \beta \log n$ (ignoring insignificant rounding issues, here and throughout this work) where $\beta > 0$ is a small positive constant, to be determined below.

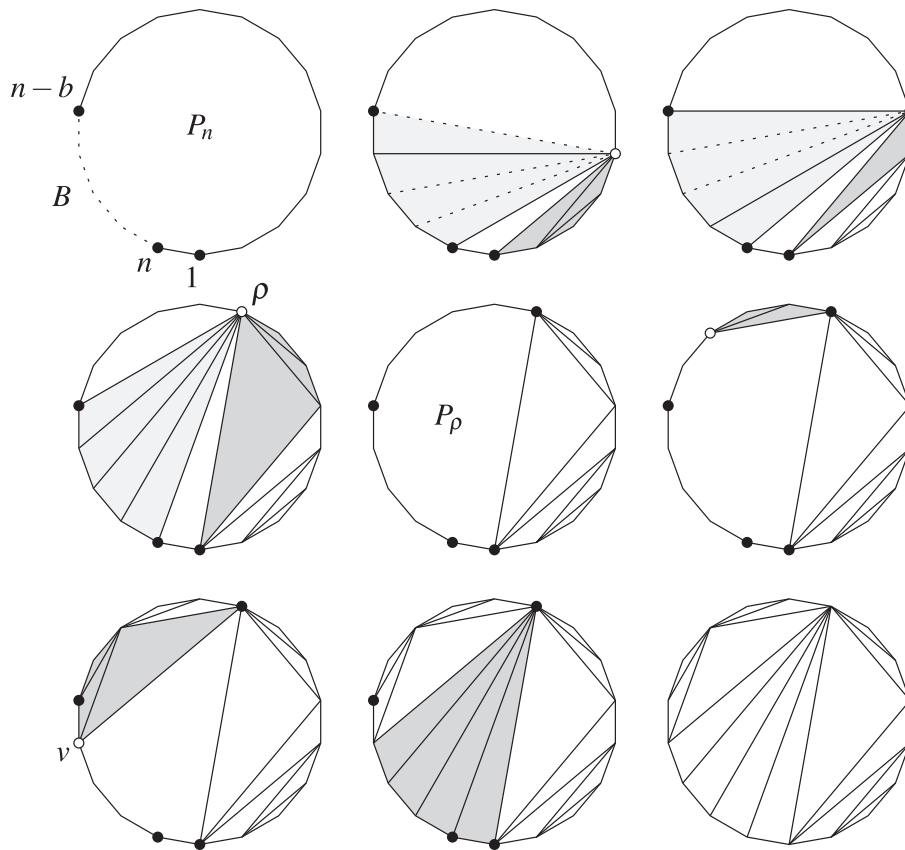


FIGURE 3 | Using GTA to triangulate P_n ; Iterations of GECA are shown in dark grey. The light grey regions depict attempts at finding a root (with dotted lines representing non-edges). After the third such attempt, we find ρ . We then continue to triangulate the remaining polygon P_ρ , until some iteration of GECA ends in the buffer B . Finally, we complete the triangulation using edges between ρ and B .

The *greedy triangulation algorithm* (GTA) proceeds as follows, in three main phases. (Of course, in the proof below, we will need to show that, with high probability, this procedure is well defined.) See Figure 3 for an illustrative example.

- **Root finding:** We iterate GECA using the edges in $E_{n,p}$. In the first application of GECA, we start with the polygon $P^{(0)} = P_n$ labelled by $1, 2, \dots, n$ in CCW order. Recall that (if successful) GECA outputs a polygon $P^{(1)}$ labelled by $1, v_{\tau_1}, \dots, n$ in CCW order, and a triangulation of the region delimited by the edge $\{1, v_{\tau_1}\}$ and the CCW path along P_n from 1 to v_{τ_1} . We continue recursively, in the k th step obtaining a polygon $P^{(k)}$ labelled by $1, v_{\tau_k}, \dots, n$ in CCW order, and a triangulation of the region delimited by the edge $\{1, v_{\tau_k}\}$ and the CCW path along P_n from 1 to v_{τ_k} . We continue until the first time, in step K say, that the vertex v_{τ_K} is neighbors with *all* vertices in the buffer B . At this point we halt the root finding phase. We denote $\rho = v_{\tau_K}$, and call it the *root*. We let P_ρ denote the polygon delimited by $\{1, \rho\}$ and the CCW path from ρ to 1 in P_n (i.e., the region that has yet to have been triangulated).
- **Reaching the buffer:** Once ρ is found, we continue to iterate GECA as before, but starting with P_ρ (labelled by $\rho, \dots, n, 1$ in CCW order) instead of P_n , and with ρ now playing the previous role of 1 . We continue to iterate until the first time that, after some application of GECA, for some $v \in B$ in the buffer, the region delimited by the edge $\{\rho, v\}$ and the CCW path along P_n from ρ to v has been triangulated.
- **Completion:** Finally, at this point, we complete the triangulation of P_n by using the triangulated portions of P_n found in the above two phases together with all edges between ρ and the vertices in B along the CCW path of P_n from v to n .

Proof of Proposition 4.1. Let $p > 1/2$. We will use GTA, as described above, together with Lemmas 2.1 and 3.1.

The key observation is to note that, by Lemma 3.1, each application of GECA in GTA corresponds to a simple random walk, with a negative drift $\mu = 1 - 2p < 0$. Indeed, recall that GECA starts with a list of length 3. In clipping steps the

list decreases by 1, and in extending steps it increases by 1; see Figure 2 above. In each step of GECA, a clipping step occurs with probability p and an extending step occurs with probability q . Moreover, by Lemma 3.1, this occurs in each step independently of all previous steps.

Recall that GECA terminates once the list reaches length 2. Therefore, by Lemma 2.1, the probability that any given application of GECA lasts for $A \log n$ steps is $O(n^{-\gamma A})$, for some $\gamma > 0$. In particular, taking a union bound over all the applications of GECA (which is less than n), we find that the probability that, in *any* such application of GECA during GTA, there are more than $A \log n$ extending steps is at most $O(n^{1-\gamma A}) \ll 1$, provided that $A > 1/\gamma$.

Therefore, for any such A , there will in particular be, with high probability, $\Omega(n/\log n)$ many opportunities to find a root ρ . Note that, at each such opportunity, we successfully find ρ with probability p^{b+1} , where recall that $b+1$ is the size of the buffer, with $b = \beta \log n$. Note that, if we take $\beta < -1/\log p$, then

$$\frac{n}{\log n} p^{b+1} = p \frac{n^{1+\beta \log p}}{\log n} \gg 1.$$

For the remainder of the proof, to be concrete, let us fix $A = 2/\gamma$ and $\beta = -1/(2 \log p)$. Then, with high probability, in the first stage of GTA we will find a root ρ somewhere in the CCW path from 1 to $\lfloor n/2 \rfloor$ in P_n .

Likewise, we observe that, with high probability, after some iteration of GECA in the second stage of GTA we will have triangulated a region delimited by the edge $\{\rho, u\}$ and the CCW path from ρ to u in P_n , for some u such that

$$(n - b) - 2A \log n \leq u \leq (n - b) - A \log n.$$

Finally, assuming that there is such a vertex u , it remains only to argue that we thereafter reach the buffer, meaning that after some subsequent step we have triangulated a region delimited by the edge $\{\rho, v\}$ and the CCW path from ρ to v in P_n , for some vertex $n - b \leq v \leq n$ in the buffer. To see this, we take another union bound. The vertex u is at distance at most $2A \log n$ from the buffer, and the buffer is of length $b = \beta \log n$. The probability that any application of GECA, started from any vertex between u and the buffer, has more than $b/2$ extending steps is at most $O(n^{-\gamma \beta/2 \log n}) \ll 1$. Therefore, with high probability, we will eventually reach the buffer, as required.

As discussed, once we reach the buffer, we can, by the choice of ρ , complete the triangulation of P_n using edges between ρ and the buffer.

Altogether, for any fixed $p > 1/2$, the above argument shows that GTA finds a triangulation of P_n with high probability, with success probability at least $1 - O(n^{-\epsilon} \log n)$, where $\epsilon = \min\{1, -\gamma/(4 \log p)\}$. $\square$

5 | Better Ear Clipping Algorithm (BECA)

Finally, we prove our main result Theorem 1.1. The proof is similar to that of Proposition 4.1. However, instead of proceeding greedily via GECA, we will be more judicious about ear clipping, using a certain *better ear clipping algorithm* (BECA). In contrast to GECA, this algorithm starts with a list ℓ_0 of length 4. In each iteration, some edges are revealed on a need-to-know basis, and then based on what is seen, BECA makes one of 7 moves, as shown in Figure 4 (cf. Figure 2 above). As we will see, these moves are chosen to maintain (when using the random edges in $E_{n,p}$ to triangulate P_n) a comparison to a random walk with a negative drift.

Let us give a formal definition of BECA. As in GECA, we start with a polygon P , with vertices labeled by $v_1, \dots, v_n$ in CCW order around the boundary of P , and some set of edges E inside of P . We start with the polygon $P^{(0)} = P$ and a list $\ell_0 = (v_1, v_2, v_3, v_4)$. Suppose that after the k th step we have a polygon $P^{(k)}$ and a list

$$\ell_k = (v_1 = v_1^{(k)}, \dots, v_{m_k}^{(k)})$$

of length at least 4. Then, by revealing edges inside of $P^{(k)}$ on a need-to-know basis, we make one of the following 7 mutually exclusive moves (cf. Figure 4). For simplicity in what follows, let us abbreviate $v_1^{(k)}, \dots, v_{m_k}^{(k)}$ by $u_1, \dots, u_m$ (although the values of u_i and m depend on k).

We let u, u', u'' be the next (in CCW order) three vertices after u_m along the boundary of $P^{(k)}$.

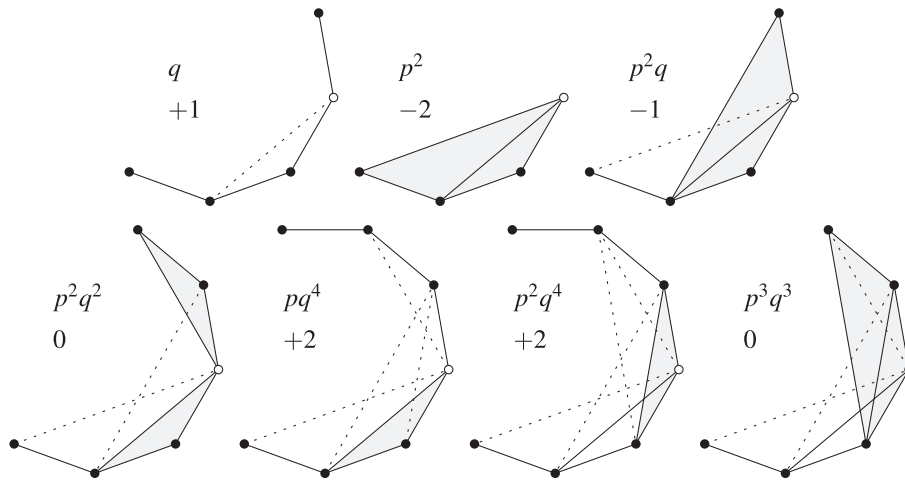


FIGURE 4 | In BECA, before clipping an ear, we consider the benefit going forward. In each step, we take one of the above mutually exclusive moves. To determine which move is made, we reveal some edges (and non-edges) on a need-to-know basis. This can be done by a simple search algorithm, as indicated by the order in which the moves are listed above. As in Figure 2 for GECA, the open dot represents the current end of the list ℓ . The vertex along the path furthest to its right (possibly itself) will be at the end of the list in the next step. The change in the length of the list is indicated, along with the probability with which such moves occur when using the random edges from $E_{n,p}$ inside the polygon.

1. If $\{u_{m-2}, u_m\} \notin E$ then
 - a. let $P^{(k+1)} = P^{(k)}$,
 - b. obtain ℓ_{k+1} from ℓ_k by adding u to the end of ℓ_k .
2. If $\{u_{m-2}, u_m\} \in E$ and $\{u_{m-3}, u_m\} \in E$ then:
 - a. obtain $P^{(k+1)}$ from $P^{(k)}$ by clipping the ear on $\{u_{m-2}, u_{m-1}, u_m\}$ and the ear on $\{u_{m-3}, u_{m-2}, u_m\}$,
 - b. and obtain ℓ_{k+1} from ℓ_k by removing u_{m-1}, u_{m-2} from ℓ_k .
3. If $\{u_{m-2}, u_m\} \in E$, $\{u_{m-3}, u_m\} \notin E$ and $\{u_{m-2}, u\} \in E$ then:
 - a. obtain $P^{(k+1)}$ from $P^{(k)}$ by clipping the ear on $\{u_{m-2}, u_{m-1}, u_m\}$ and the ear on $\{u_{m-2}, u_m, u\}$,
 - b. and obtain ℓ_{k+1} from ℓ_k by removing u_{m-1}, u_m from ℓ_k and then adding u to the end.

In all of the steps (4–7) below, we assume that $\{u_{m-2}, u_m\} \in E$, $\{u_{m-3}, u_m\} \notin E$ and $\{u_{m-2}, u\} \notin E$ (as otherwise we take one of the steps (1–3) above).

4. If $\{u_m, u'\} \in E$ then:
 - a. obtain $P^{(k+1)}$ from $P^{(k)}$ by clipping the ear on $\{u_{m-2}, u_{m-1}, u_m\}$ and the ear on $\{u_m, u, u'\}$,
 - b. and obtain ℓ_{k+1} from ℓ_k by removing u_{m-1} from ℓ_k (noting that u is not in ℓ_k to begin with, so does not need to be removed) and then adding u' to the end.
5. If $\{u_m, u'\} \notin E$ and $\{u_{m-1}, u\} \notin E$ then:
 - a. obtain $P^{(k+1)}$ from $P^{(k)}$ by clipping the ear on $\{u_{m-2}, u_{m-1}, u_m\}$,
 - b. and obtain ℓ_{k+1} from ℓ_k by removing u_{m-1} from ℓ_k and then adding u, u', u'' to the end.

Finally, in steps (6–7), we assume that $\{u_m, u'\} \notin E$ and $\{u_{m-1}, u\} \in E$ (and that $\{u_{m-2}, u_m\} \in E$, $\{u_{m-3}, u_m\} \notin E$ and $\{u_{m-2}, u\} \notin E$).

6. If $\{u_{m-1}, u'\} \notin E$ then:
 - a. obtain $P^{(k+1)}$ from $P^{(k)}$ by clipping the ear on $\{u_{m-1}, u_m, u\}$,
 - b. and obtain ℓ_{k+1} from ℓ_k by removing u_m from ℓ_k and then adding u, u', u'' to the end.
7. If $\{u_{m-1}, u'\} \in E$ then:
 - a. obtain $P^{(k+1)}$ from $P^{(k)}$ by clipping the ear on $\{u_{m-1}, u_m, u\}$ and the ear on $\{u_{m-1}, u, u'\}$,
 - b. and obtain ℓ_{k+1} from ℓ_k by removing u_m from ℓ_k (noting, once again, that u is not in ℓ_k , so does not need to be removed) and then adding u' to the end.

We continue to run BECA recursively in this way. By induction, after each step k , the regions delimited by the edges $\{v_1 = v_1^{(k)}, v_2^{(k)}\}, \dots, \{v_{m_k-1}^{(k)}, v_{m_k}^{(k)}\}$ between consecutive vertices in the list ℓ_k and the CCW path along P from $v_1 = v_1^{(k)}$ to $v_{m_k}^{(k)}$ have all been triangulated; only the polygon $P^{(k)}$ remains to be triangulated.

As before, when running GECA, if at some point we run out of vertices along the boundary of P , we simply halt the algorithm and say that it has failed. Recall that GECA is successful if eventually the list reaches length 2 (beginning with a list of length 3). On the other hand, for BECA, we start with a list of length 4 and say that the algorithm is successful if the length eventually reaches length 2 or 3.

Finally, as in GTA, we will want to iterate the BECA algorithm. In doing so, before applying BECA again, we will simply extend the list to length 4 by adding the next (1 or 2) unexplored vertices along the boundary of P (unless we run out of vertices along the boundary of P_n , in which case, once again, we simply halt the algorithm and say that it has failed).

Proof of Theorem 1.1. In a nutshell, to prove Theorem 1.1, we apply a natural analogue of the algorithm GTA used in the proof of Proposition 4.1, obtained simply by replacing the role of GECA with that of BECA. Since the proof is similar, we only discuss the parts that require adaptation.

We first need to establish an analogue of Lemma 3.1. That is, we need to argue that, when using the random edges from $E_{n,p}$ inside of P_n , each iteration of BECA corresponds to a random walk. To see this, it may be most helpful to refer to Figure 4. Note that, in each step of BECA, when checking to see if an edge $\{x, y\}$ is in $E_{n,p}$, at least one of the endpoints in $\{x, y\}$ belongs to the CCW path from the last vertex in the current list (the open dot in Figure 4) to n in P_n (i.e., in the formal definition of BECA, we have $\{x, y\} \cap \{u_m, u, u', u''\} \neq \emptyset$ for all such $\{x, y\}$). Furthermore, all such $\{x, y\}$ either belong to an ear that is clipped away (so that it is either removed or becomes a boundary edge after the clipping) before proceeding to the next step, or else both of its endpoints are *outside* the CCW path from the last vertex in the *next* list (the rightmost dot in Figure 4) to n in P_n (indeed, this is precisely why in some moves of BECA we extend the list). As a result, it follows that the sets of adjacencies revealed in any two steps of BECA are disjoint, yielding the desired Markov property.

Next, in order to apply Lemma 2.1, we need to check that the random walk corresponding to each iteration of BECA has a negative drift. Indeed, we observe (see Figure 4) that the expected change Δ in the length of a list ℓ after a BECA step (where $q = 1 - p$) is

$$\begin{aligned}\Delta &= q - 2p^2 - p^2q + 2pq^4 + 2p^2q^4 \\ &= 2p^6 - 6p^5 + 4p^4 + 5p^3 - 9p^2 + p + 1.\end{aligned}$$

We note that $\Delta < 0$ for all $p > p_*$, where $p_* \approx 0.4916$. Recall that we start each application of BECA with a list of length 4, and halt once the list reaches length 2 or 3. Therefore, for any such p , Lemma 2.1 applies as before in the proof of Proposition 4.1.

Recall that, in the root finding phase of GTA, after each iteration of GECA, when the list reaches length 2, and so is equal to some $\ell = (1, v)$, we check to see if v is neighbors will all vertices in the buffer (and, if so, we set $\rho = v$ and then proceed to the next phase of GTA). On the other hand, in the current proof, each iteration of BECA in the root finding phase will end with a list $\ell = (v_1 = 1, v_2, \dots, v_m)$ of length $m \in \{2, 3\}$. As such, in finding the root, we must now instead check to see if v_m is neighbors with all vertices in the buffer *and* the vertices $v_1, \dots, v_{m-1}$. Note that, when $m = 3$, the edge $\{v_1, v_m\}$ has not yet been revealed.

In the reaching the buffer phase, we continue to iterate BECA until the first time that, after some application of BECA, the last vertex v in the list is in the buffer. At this point, the list $\ell = (\rho, \dots, v)$ may be of length 2 or 3, but in either case we can complete to a triangulation using edges from ρ .

Apart from these changes, we can complete the proof along the same lines as Proposition 4.1. □

6 | Final Remarks

Let $p_c(n)$ be the value of p such that $\mathbb{P}(\mathcal{T}_{n,p}) = 1/2$. Clearly, $p_c \leq p_c(n)$. We conjecture that $\lim_{n \rightarrow \infty} p_c(n)$ exists and equals p_c .

Recall that the greedy GECA algorithm is used to show that $p_c \leq 1/2$. In each step, GECA simply checks if the edge between the current vertex v_m and v_{m-2} to its left is present; see Figure 2. The BECA algorithm expands the “decision

neighborhood,” checking edges between v_m and v_{m-2}, v_{m-3} to its left (and further vertices to its right as necessary); see Figure 4. This leads to an upper bound of about 0.4916. As already mentioned, the experiments in Archer et al. (2026) suggest that p_c might be between 0.39 and 0.41. We wonder if “local” search algorithms become “global” as the size of the “decision neighborhood” increases: For any fixed $\ell \geq 2$, one can define a generalization of BECA that checks the edges between v_m and $v_{m-2}, \dots, v_{m-\ell}$ (and vertices to the right as necessary). Each such algorithm would give some (increasingly more complicated) upper bound $p_c^{(\ell)} \geq p_c(n)$. We wonder if $p_c^{(\ell)} \rightarrow p_c$ as $\ell \rightarrow \infty$, and this would imply that $p_c = \lim_{n \rightarrow \infty} p_c(n) = \lim_{\ell \rightarrow \infty} p_c^{(\ell)}$. However, it could be that there is a second phase transition above p_c , such that $\lim_{\ell \rightarrow \infty} p_c^{(\ell)} > p_c$ and $p_c = (1 - o(1))p_c^{(\ell)}$ for some $\ell = \ell(n)$ growing with n . If so, it would be interesting to understand the growth rate of $\ell(n)$.

Acknowledgements

The authors thank the two anonymous referees, whose detailed comments improved this work. This work began while G.Z. and M.Z. were visiting B.K. at the University of Warwick. We thank the Centre for Research in Statistical Methodology (CRiSM) for funding this visit. G.Z. was supported by a Hill Foundation Scholarship.

Data Availability Statement

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

References

- Aizenman, M., and G. Grimmett. 1991. “Strict Monotonicity for Critical Points in Percolation and Ferromagnetic Models.” *Journal of Statistical Physics* 63, no. 5–6: 817–835.
- Archer, E., I. Hartarsky, B. Kolesnik, S. Olesker-Taylor, B. Schapira, and D. Valesin. 2026. “Catalan Percolation.” *Probability Theory and Related Fields* 195, no. 1–2: 413–449.
- Balogh, J., B. Bollobás, and R. Morris. 2012. “Graph Bootstrap Percolation.” *Random Structures & Algorithms* 41, no. 4: 413–440.
- Bollobás, B. 1968. *Weakly k-Saturated Graphs*, 25–31. B. G. Teubner Verlagsgesellschaft.
- Bollobás, B., and A. M. Frieze. 1991. “Spanning Maximal Planar Subgraphs of Random Graphs.” *Random Structures & Algorithms* 2, no. 2: 225–231.
- Doney, R. A. 1989. “On the Asymptotic Behaviour of First Passage Times for Transient Random Walk.” *Probability Theory and Related Fields* 81, no. 2: 239–246.
- Duminil-Copin, H., M. R. Hilário, G. Kozma, and V. Sidoravicius. 2018. “Brochette Percolation.” *Israel Journal of Mathematics* 225, no. 1: 479–501.
- Durrett, R. 1984. “Oriented Percolation in Two Dimensions.” *Annals of Probability* 12, no. 4: 999–1040.
- Essam, J. W., A. J. Guttmann, and K. De’Bell. 1988. “On Two-Dimensional Directed Percolation.” *Journal of Physics A* 21, no. 19: 3815–3832.
- Gravner, J., and B. Kolesnik. 2023. “Transitive Closure in a Polluted Environment.” *Annals of Applied Probability* 33, no. 1: 107–126.
- Gravner, J., and E. McDonald. 1997. “Bootstrap Percolation in a Polluted Environment.” *Journal of Statistical Physics* 87, no. 3–4: 915–927.
- Kahn, J., B. Narayanan, and J. Park. 2021. “The Threshold for the Square of a Hamilton Cycle.” *Proceedings of the American Mathematical Society* 149, no. 8: 3201–3208.
- Riordan, O. 2000. “Spanning Subgraphs of Random Graphs.” *Combinatorics, Probability and Computing* 9, no. 2: 125–148.
- Zhukovskii, M. 2025. “Sharp Thresholds for Spanning Regular Subgraphs.” preprint available at [arXiv:2502.14794](https://arxiv.org/abs/2502.14794).