Spectral Graph Theory Meets Coined Quantum Walks: Dispersion and Hitting Times on Simple Graph Families

Vishwesh Palani

December 21, 2025

Abstract

We study how spectral structure explains the qualitative gap between classical random walks and coined quantum walks on highly symmetric graphs. Classically, a walk is driven by a stochastic transition matrix whose eigenvalues in $[-1, 1]$ cause non-stationary modes to decay, producing diffusion and relaxation. Quantumly, a coined walk evolves under a unitary step operator whose eigenvalues lie on the unit circle, so amplitudes do not decay but instead rotate and interfere, yielding ballistic spreading and recurrence phenomena. We analyze and simulate these effects on cycles and paths using variance growth and return probability, and we compare target-finding on complete graphs by marking a vertex and tracking a time-to-p success metric under explicit measurement protocols. Across the tested instances, simulations show faster dispersion and pronounced revivals for the coined walk on C_n and P_n, and a search behavior on K_n consistent with a $\Theta(\sqrt{n})$ peak time versus a classical $\Theta(n)$ hitting-by-time baseline.

1 Motivation

Random walks are a powerful way to study graphs and quantify how quickly they spread and discover target nodes using nothing but local connectivity information. The classical formalism for a walk is described using a Markov chain with updates to the probability distribution being performed using a *stochastic* transition matrix whose spectrum (its eigenvalues) can be used to explain the behavior of the walk.

Coined quantum walks are the quantum analogue of classical random walks and use a *unitary* step operator to update probability *amplitudes*, which produce probabilities upon measurement.

This report will primarily use eigenstructure (eigenvalues and eigenvectors) to analyze both classical and quantum walks on graph families whose structure makes the relevant eigenstructure easy to compute and interpret. Finally, we will simulate both walks in a notebook by reducing updates to polynomial-time *linear* algebra to measure and compare observable metrics.

1.1 Why compare classical and quantum walks?

Comparing the two models on the same graph controls for the graph itself, so any difference in observed behavior comes from the update rule (stochastic versus unitary) rather than extra information or extra power given to one model. Concretely, the classical walk repeatedly *averages* probability over neighboring vertices, which tends to smooth distributions over time,

while the quantum walk updates *amplitudes* that can add together or cancel before measurement, producing patterns that are not simple averaging. Placing both walks on the same simple graphs therefore lets us isolate which effects are genuinely quantum (cancellation or reinforcement of amplitudes) and which are simply consequences of the graph's connectivity.

1.2 What is the "quantum advantage" metric in this report?

We compare the models using two operational notions of performance: how quickly the walk *spreads out* from its start (measured by a spread statistic such as the variance of the position distribution), and how quickly the walk can *find a designated target vertex*. For the second task, we will call the target vertex *marked* (i.e., labeled as the one we want to find) and measure performance as the smallest number of steps needed for an appropriately defined success probability (depending on when measurements occur) to exceed a fixed threshold (for example, 1/2). These choices give metrics that are directly measurable from simulated distributions and make classical and quantum comparisons meaningful on the same instance. When comparing target-finding on K_n, we use per-step checking for the classical baseline (to model hitting-by-time) and end-only measurement for the quantum walk (to preserve coherent amplitude buildup), and we report T_p under the corresponding S_t definitions.

1.3 Scope and graph families

We focus on *symmetric graph families*, meaning graphs where many vertices "look the same" from the perspective of connectivity (so relabeling vertices often does not change the graph). This symmetry makes the walk dynamics easier to analyze because it forces the transition rule to behave similarly at many locations, which often leads to a simple description of eigenvalues/eigenvectors and reduces the number of distinct cases to consider. Within this scope, cycles and paths serve as clean testbeds for spreading and return behavior, while complete graphs (and optionally hypercubes) serve as clean testbeds for target-finding because their high symmetry makes the relevant two-model comparison especially sharp.

1.4 Organization of the report

We first introduce the needed graph and walk definitions, then analyze cycles/paths (spreading and returns) and complete graphs (target-finding), and finally validate these behaviors with simulations and summarize limitations and extensions.

2 Background: Graphs, Spectra, and Random Walks

This section introduces the minimum *spectral graph theory* (SGT) and walk formalism needed to state and interpret our results. The central idea of SGT is that many properties of a graph can be read off from the *eigenvalues* and *eigenvectors* of matrices naturally associated with the graph, and that walk dynamics become especially transparent when expressed in an eigenvector basis.

2.1 Graphs and matrix representations

A (simple, undirected) graph is a pair $G = (V, E)$ where V is a set of vertices and E is a set of unordered pairs $\{u, v\}$ indicating edges. The *adjacency matrix* A of G is the $|V| \times |V|$

matrix with $A_{uv} = 1$ if $\{u, v\} \in E$ and $A_{uv} = 0$ otherwise. The *degree* of a vertex v is the number of its neighbors, written $\deg(v)$, and the *degree matrix* D is the diagonal matrix with $D_{vv} = \deg(v)$. These matrices encode the local connectivity of the graph in linear-algebraic form, which lets us describe walk updates as repeated matrix multiplication.

2.2 Spectral graph theory essentials

For any square matrix M, a scalar λ is an *eigenvalue* with *eigenvector* $x \neq 0$ if $Mx = \lambda x$; the collection of eigenvalues (counted with multiplicity) is called the *spectrum* of M. When a matrix has a full set of eigenvectors that form a basis (as happens for many symmetric matrices arising from undirected graphs), any vector can be written as a sum of eigenvector "modes," and multiplying by M acts on each mode by a simple scalar factor (or, for unitary matrices, a phase factor). This is the basic reason spectra are useful for walks: repeated updates become easy to understand once the update matrix is expressed in an eigenvector basis.

2.3 Classical random walks as Markov chains

A *Markov chain* on a finite state space is a process where the next state depends only on the current state, and it is specified by a *transition matrix* P whose entry P_{uv} is the probability of moving from u to v in one step. A matrix is *stochastic* if it has nonnegative entries and each row sums to 1, ensuring that $p_{t+1} = p_t P$ maps probability distributions to probability distributions. For a random walk on a graph, a common choice is to move uniformly to a random neighbor, in which case $P_{uv} = 1/\deg(u)$ when $\{u, v\} \in E$ and $P_{uv} = 0$ otherwise; in particular, if every vertex has the same degree d (the graph is *d-regular*), then $P = A/d$. A distribution π is called *stationary* if $\pi = \pi P$, meaning that if the chain starts in π then it stays in π after every step. On a finite connected graph, the simple random walk has a stationary distribution (uniform if the graph is regular), and if the chain is also *aperiodic* then p_t converges to π from any start. On periodic graphs (e.g., even cycles), p_t can oscillate rather than converge pointwise, but standard "mixing" statements still apply to time-averaged distributions.

2.4 Key classical quantities: mixing and hitting

To discuss convergence, we use a distance between probability distributions: the *total variation distance* between distributions p and q on V is $\frac{1}{2}\sum_{v \in V} |p(v) - q(v)|$, which measures how distinguishable they are by any event on V. The *mixing time* (informally) is the number of steps needed until the walk's distribution is close to a stationary distribution in total variation distance, starting from a worst-case initial vertex. For target-finding, fix a *marked* vertex m and define the *hitting time* as the (random) number of steps until the walk first visits m; we compare models using the expected value of this quantity or, equivalently, by the number of steps needed for the probability of having reached m to exceed a chosen threshold. Both mixing and hitting behavior are tightly connected to the spectrum of the transition matrix P because powers of P control how fast non-stationary modes shrink over time.

3 Coined Quantum Walks

A coined quantum walk is a discrete-time walk model in which the walker carries an extra finite register (the "coin") that determines which neighboring vertex it moves to at each step. Unlike a classical random walk, which updates a probability distribution using a stochastic matrix, a coined quantum walk updates a complex state vector using a unitary matrix, and probabilities are obtained only when we measure the state. The main benefit of this model for our purposes is that it enforces locality (moves occur along edges) while allowing interference between different paths through the graph.

3.1 Hilbert space and state representation

We work in a finite-dimensional complex vector space called a *Hilbert space*, which here is just C^N equipped with the usual inner product. For a graph $G = (V, E)$, the *position space* is $\mathcal{H}_{pos} = C^{|V|}$ with standard basis vectors $\{|v\rangle : v \in V\}$ representing the walker being at vertex v. The coined model introduces a *coin space* $\mathcal{H}_{coin} = C^d$, where d is the number of possible "move directions" from a vertex (for a d-regular graph, every vertex has d neighbors). The full state space is the tensor product $\mathcal{H} = \mathcal{H}_{pos} \otimes \mathcal{H}_{coin}$, with basis states $|v, c\rangle := |v\rangle \otimes |c\rangle$, and a walk state is a unit vector

$$|\psi\rangle = \sum_{v \in V} \sum_{c=1}^{d} \alpha_{v,c} \, |v, c\rangle,$$

where the complex numbers $\alpha_{v,c}$ are *amplitudes*.

3.2 Coin operator choices

The *coin operator* is a unitary matrix C acting only on $\mathcal{H}_{coin}$ that mixes the coin amplitudes at each vertex before the walker moves. Intuitively, C plays the role of "choosing a direction," except it does so coherently by redistributing amplitudes rather than sampling a random direction. A standard symmetric choice on d coin states is the *Grover coin*,

$$C \;=\; 2|s\rangle\langle s| - I \quad where \quad |s\rangle = \frac{1}{\sqrt{d}} \sum_{c=1}^{d} |c\rangle,$$

which reflects amplitudes about the uniform coin state. For degree 2 graphs (such as cycles and paths), one can also use the familiar two-dimensional *Hadamard* matrix as a simple balanced coin; in either case, the goal is to use a fixed, symmetric rule so observed differences come from the graph and the walk model rather than an asymmetric coin.

3.3 Shift operator and step unitary

After applying the coin, the walk uses a *shift operator* S, a unitary acting on $\mathcal{H}$ that moves amplitude along graph edges according to the coin label. Concretely, we interpret each coin basis state c at a vertex v as pointing to a specific neighbor of v, and S maps $|v, c\rangle$ to the corresponding neighbor state (possibly with an updated coin label to keep the mapping reversible). One full time step of the coined walk is the unitary

$$U \;=\; S\,(I \otimes C),$$

and the evolution from an initial state $|\psi_0\rangle$ is given by repeated application:

$$|\psi_t\rangle \;=\; U^t\,|\psi_0\rangle.$$

Because U is unitary, its eigenvalues lie on the unit circle; if $U|x\rangle = e^{i\theta}|x\rangle$, we call θ the *eigenphase* associated with eigenvector $|x\rangle$, and these eigenphases govern how different components of the state rotate and interfere over time.

3.4 Model conventions (coin, shift, and what counts as a step)

To make comparisons consistent, we fix a single coined-walk convention throughout the report: one *step* means "apply the coin operator everywhere, then apply the shift once." When the graph is d-regular, we use a d-dimensional coin with a fixed symmetric coin operator (Grover by default, with a two-dimensional balanced coin on degree-2 graphs). We also use a fixed reversible shift rule that maps each coin label at v to a unique neighboring vertex so that the overall update remains unitary. These conventions ensure that changes in behavior across graphs and between classical versus quantum walks are not artifacts of inconsistent update rules.

3.5 Measurement and induced position distributions

To compare with a classical probability distribution on vertices, we convert a quantum state into a vertex distribution by measuring only the position register. If the state at time t is $|\psi_t\rangle = \sum_{v,c} \alpha_{v,c}(t)\,|v,c\rangle$, then the probability of observing vertex v at time t is

$$p_t(v) \;=\; \sum_{c=1}^{d} |\alpha_{v,c}(t)|^2,$$

which is obtained by summing squared amplitudes over the coin states. This vertex distribution p_t is the main observable we plot in simulations, and it is the quantity we use to define spread statistics (such as variance) and target-finding success probabilities.

4 Spectral Perspective: Why Eigenstructure Matters

The unifying theme of this report is that both classical and quantum walks are linear update rules, so their long-term behavior can be understood by decomposing the initial state into eigenvector components of the update matrix. In the classical case, eigenvalues of a stochastic matrix lie in $[-1, 1]$ and repeated multiplication suppresses components with eigenvalue magnitude < 1, producing smoothing and convergence. In the quantum case, eigenvalues of a unitary matrix have magnitude 1 (they are complex numbers on the "unit circle"), so components do not decay; instead they rotate by accumulating phase, and observed probabilities change through interference between rotating components.

4.1 Classical mode decay and spectral gap intuition

Let P be the transition matrix of a classical random walk, and suppose P has a basis of eigenvectors $\{v_i\}$ with eigenvalues $\{\lambda_i\}$. Writing the initial distribution as a linear combination of

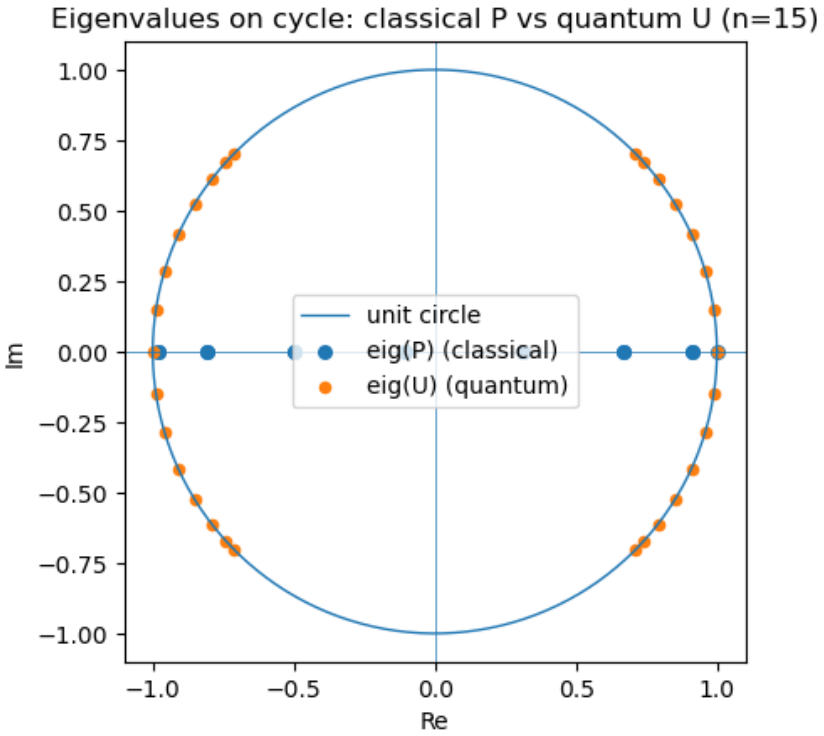


Figure 1: Spectral contrast (cycle example, $n = 15$): eigenvalues of the classical transition matrix P lie in $[-1, 1]$, while eigenvalues of the coined-walk unitary U lie on the unit circle.

eigenvectors, $p_0 = \sum_i a_i v_i$, we get

$$p_t \;=\; P^t p_0 \;=\; \sum_i a_i \lambda_i^t v_i,$$

so each eigenvector component is multiplied by λ_i^t after t steps. The eigenvalue 1 corresponds to the stationary component, while components with $|\lambda_i| < 1$ shrink toward 0 as t grows, explaining why the walk smooths out over time. A convenient one-number summary of the convergence rate is the *spectral gap*, defined here as $1 - \lambda_\star$ where $\lambda_\star$ is the largest eigenvalue of P strictly smaller than 1 (for many standard walks, $\lambda_\star$ is the second-largest eigenvalue). A larger gap means the non-stationary components shrink faster, so convergence to the stationary distribution and related relaxation effects occur in fewer steps.

4.2 Quantum eigenphases, interference, and recurrence

Let U be the unitary step operator of a coined quantum walk, and suppose U has eigenvectors $\{|x_i\rangle\}$ with eigenvalues $e^{i\theta_i}$, where each θ_i is an *eigenphase*. Decomposing the initial state as $|\psi_0\rangle = \sum_i b_i |x_i\rangle$ gives

$$|\psi_t\rangle \;=\; U^t|\psi_0\rangle \;=\; \sum_i b_i e^{it\theta_i}|x_i\rangle,$$

so no component decays; each component simply rotates by a phase that grows linearly with t. Observable probabilities change because the position probability at time t depends on sums of amplitudes, and those sums can either reinforce or cancel depending on the relative phases.

When many phases separate steadily over time, interference tends to spread amplitude across the graph (supporting fast dispersion), while when phases realign, the walk can return close to its starting behavior, producing large "return" or "revival" probabilities.

4.3 Mixing caveat for quantum walks

Because unitary evolution does not shrink eigenvector components, a coined quantum walk does not generally converge to a fixed stationary distribution in the same sense as a classical Markov chain. To still talk about a stable long-run behavior, we will sometimes use a *time-averaged* position distribution: for a horizon T, define

$$\bar{p}_T(v) \;=\; \frac{1}{T}\sum_{t=0}^{T-1} p_t(v),$$

where $p_t(v)$ is the measured position probability at time t. This averaging smooths out rapid oscillations and provides an operational proxy for "how spread out" the walk is over long times, even though the instantaneous distribution continues to fluctuate.

4.4 What we will test empirically

In the case studies, we will connect these spectral predictions to simulations by plotting (i) spread statistics derived from the position distribution over time and (ii) target-finding success probabilities over time, and comparing the resulting curves between classical and quantum walks under matched initial conditions.

5 Case Study I: Cycles and Paths (Dispersion and Revivals)

We begin with the simplest graph families that still exhibit nontrivial walk behavior: a *cycle* and a *path*. The cycle on n vertices, written C_n, consists of vertices $\{0, 1, \ldots, n-1\}$ with edges between i and $i \pm 1 \pmod{n}$, while the path on n vertices, written P_n, consists of vertices $\{0, 1, \ldots, n-1\}$ with edges between i and $i+1$. These graphs are useful because their one-dimensional geometry makes it natural to talk about how fast a walk spreads from its starting point and how often it returns close to where it began.

5.1 Classical walk on C_n and P_n

For the classical random walk on C_n, from vertex i the walker moves to $i+1$ or $i-1$ with equal probability $1/2$; on P_n the rule is the same except that at the endpoints there is only one available move. Starting from a single vertex, the probability distribution gradually "smears out" over nearby vertices, and the typical distance from the start grows slowly over time. On P_n, the endpoints create boundary effects (probability reflects and accumulates near the ends), while on C_n the walk eventually wraps around the cycle and begins to resemble a nearly uniform distribution.

5.2 Coined quantum walk on C_n and P_n

For the coined quantum walk on these graphs, the coin space has dimension 2, corresponding to the two directions "left" and "right." One step applies a fixed 2×2 unitary coin to mix

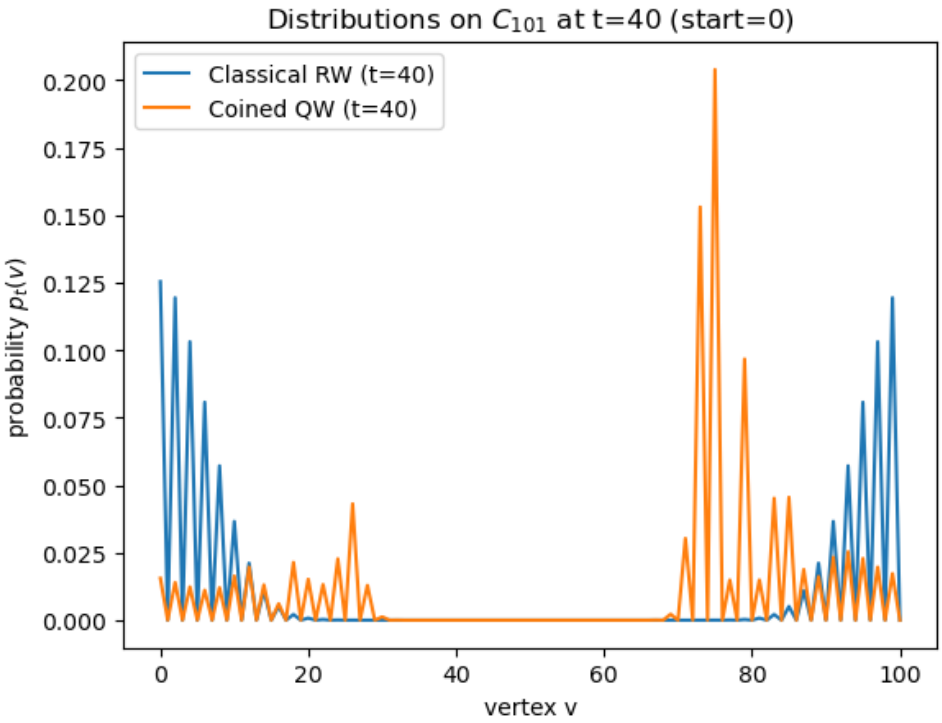


Figure 2: Position distribution on C_{101} at $t = 40$ (start $s = 0$). The coined QW shows pronounced outward-moving peaks and interference structure compared to the classical RW.

the directional amplitudes and then applies the shift to move the left-labeled amplitude one vertex left and the right-labeled amplitude one vertex right (on P_n we use a reversible endpoint "bounce" shift so amplitude pointing out of the path reflects back). In contrast to the classical walk, the measured position distribution typically develops two dominant peaks moving away from the start, together with oscillatory "ripples" caused by cancellation and reinforcement of amplitudes. Because evolution is coherent, the walk can also partially refocus, producing times when the probability of being near the start becomes unusually large compared to the classical walk.

5.3 Dispersion metric: variance growth rate

To quantify "how spread out" a walk is at time t, we use the *variance* of the position distribution. For a distribution p_t on vertices labeled by integers, define the mean $\mu_t = \sum_x x\, p_t(x)$ and the variance

$$\mathrm{Var}[X_t] \;=\; \sum_x (x-\mu_t)^2\, p_t(x),$$

where X_t denotes the measured position at time t. On C_n, to avoid wrap-around artifacts, we compute variance using the signed shortest displacement from the start vertex ("prewrap variance"): for start s and vertex x, let $\delta_s(x) \in \{-\lfloor n/2 \rfloor, \ldots, \lfloor n/2 \rfloor\}$ be the representative of $x - s \pmod n$ with minimum absolute value (ties broken consistently). We then compute the mean and variance using $\delta_s(x)$ in place of x. Empirically and theoretically, the classical walk exhibits approximately linear variance growth over time (typical distance grows like

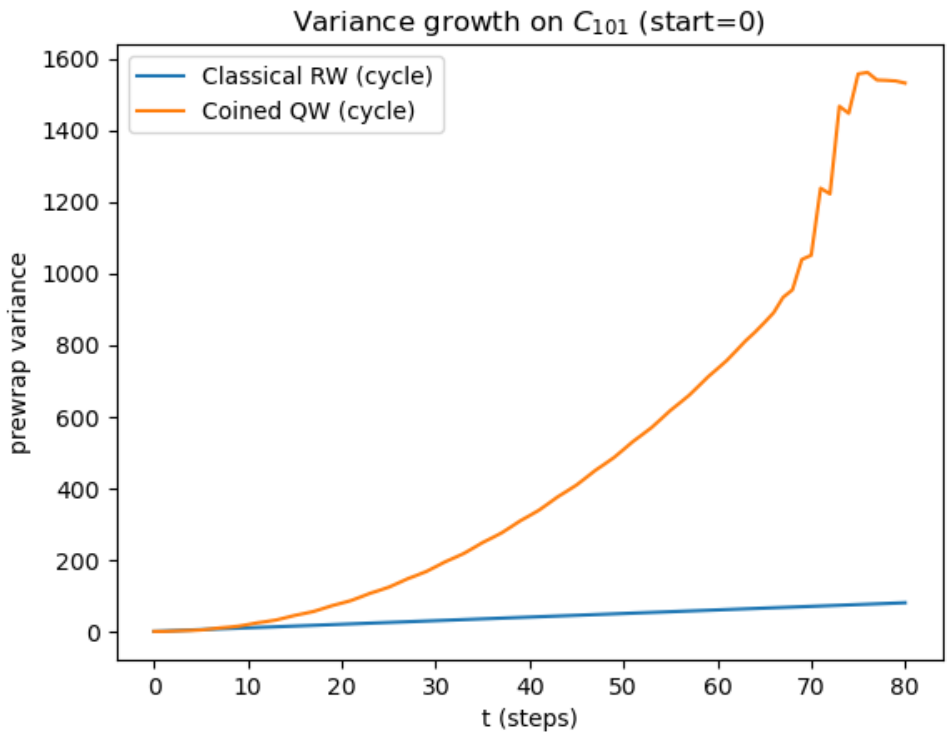


Figure 3: Variance growth on C_{101} from a localized start ($s = 0$). Classical RW shows diffusion-like growth, while the coined QW exhibits much faster growth prior to finite-size effects.

$\sqrt{t}$), while the coined quantum walk exhibits much faster variance growth over the same time window (typical distance grows roughly proportional to t). We will plot $\mathrm{Var}[X_t]$ versus t for matched initial conditions to make this contrast visible.

5.4 Revival metric: return probability

To quantify recurrence, we measure the *return probability*, defined as the probability that the walker is observed at the starting vertex at time t. If the start vertex is s, then the classical return probability is simply $p_t(s)$, while the quantum return probability is the position marginal $p_t(s) = \sum_c |\alpha_{s,c}(t)|^2$ obtained from the state amplitudes. In the classical walk, this quantity typically decreases and then stabilizes according to the long-run behavior of the chain, whereas in the coined quantum walk it can exhibit pronounced oscillations and peaks at specific times due to phase alignment. We will track $p_t(s)$ over time on both C_n and P_n to demonstrate recurrence behavior that is present in the quantum walk but absent (or much weaker) in the classical walk.

6 Case Study II: Complete Graphs (Hitting and Search)

We now study target-finding on the *complete graph* K_n, the graph on n vertices in which every pair of distinct vertices is connected by an edge. We designate one vertex m as *marked*

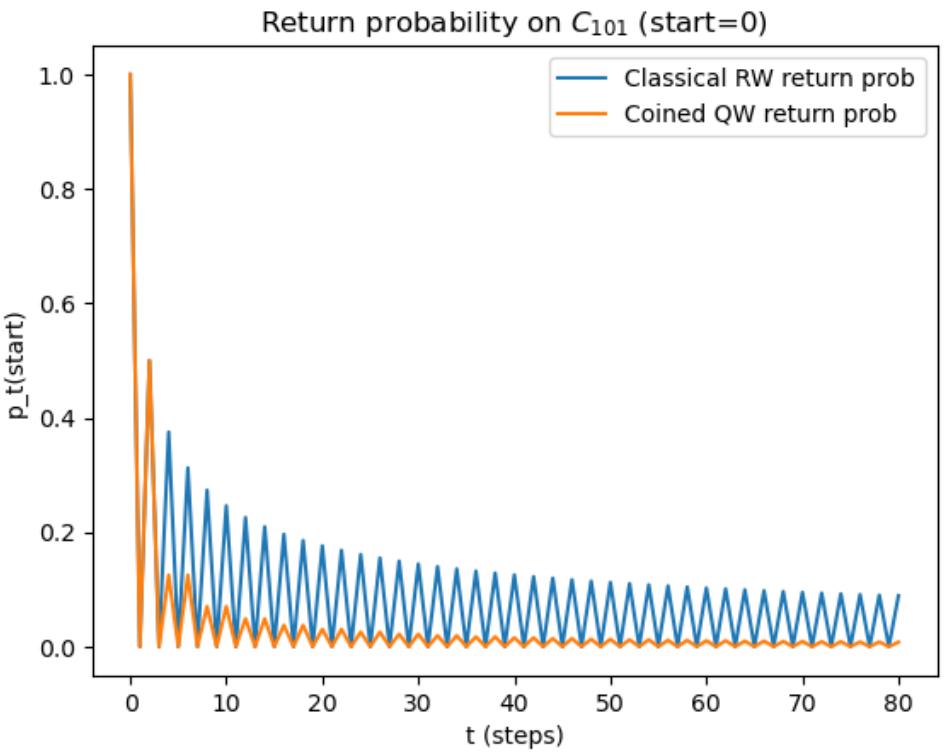


Figure 4: Return probability $p_t(s)$ on C_{101} (start $s = 0$). The coined QW exhibits oscillatory recurrence behavior compared to the classical RW.

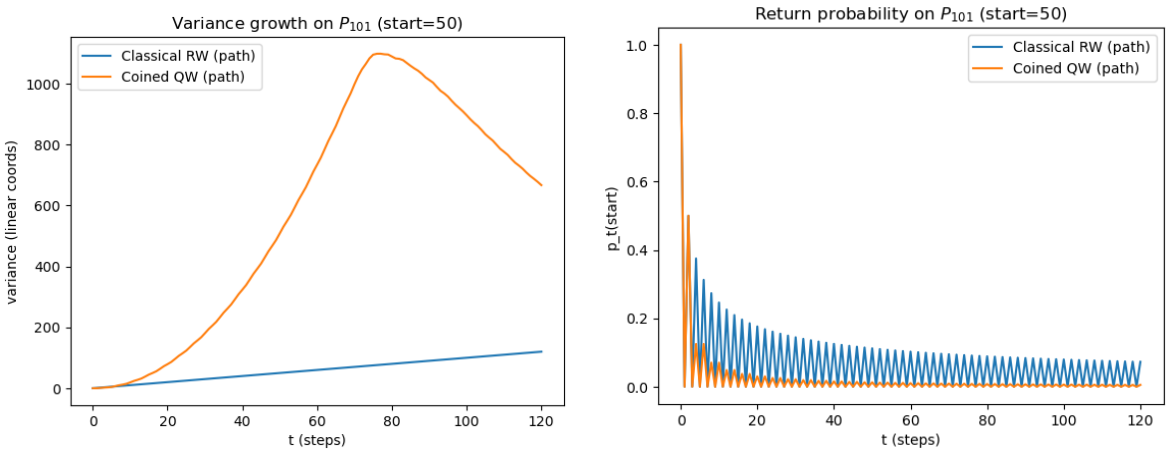
(the target we want to find) and compare how quickly a classical random walk versus a coined quantum walk can place large probability on m when measured. The high symmetry of K_n makes this comparison especially clean: away from the marked vertex, all unmarked vertices have identical connectivity, so the walk dynamics can often be described using only a small number of aggregate parameters.

6.1 Classical baseline on K_n

Consider the classical random walk that moves from its current vertex to a uniformly random neighbor at each step. In K_n, every vertex has degree $n-1$, and from any unmarked vertex the probability of moving to the marked vertex in the next step is exactly $1/(n-1)$. By symmetry, from any unmarked vertex the chance to move to m in the next step is $1/(n-1)$, so the time to hit m from an unmarked vertex is geometric with mean $n-1$ (and thus $\Theta(n)$). In simulations we will treat this $\Theta(n)$ scaling as the classical baseline for target-finding on K_n.

6.2 Quantum walk search viewpoint

To make the quantum walk "aware" of the target, we modify the walk so that amplitudes at the marked vertex acquire a sign flip each step (a phase of -1), while amplitudes at unmarked vertices evolve normally. On K_n, because all unmarked vertices are symmetric, starting from a symmetric initial state forces the evolution to remain in a low-dimensional invariant subspace.



(a) Variance on P_{101} (start $s = 50$).

(b) Return probability on P_{101} (start $s = 50$).

Figure 5: Path behavior: boundary effects appear in the coined QW variance and return curves while the classical walk remains diffusion-like.

In our simulations, we therefore implement the equivalent *symmetry-reduced* dynamics that tracks only the aggregate amplitude on the marked vertex versus the aggregate amplitude uniformly spread over unmarked vertices, rather than explicitly constructing the full coined walk operator on a $(n \cdot (n-1))$-dimensional position–coin space.

A key simplification on K_n is that symmetry forces the evolution to remain highly structured when we start from a symmetric initial state (for example, a uniform superposition over vertices with a symmetric coin state). In particular, throughout the evolution the amplitudes over all unmarked vertices remain equal in aggregate, so the state can be well-described using only two components: "amplitude on the marked vertex" versus "amplitude uniformly spread over the unmarked vertices." Under the marking rule above, the update acts like a repeated mixing between these two components, causing the marked-vertex probability to rise and fall periodically. The first large peak typically occurs after on the order of $\sqrt{n}$ steps, giving a quadratic improvement over the classical $\Theta(n)$ baseline in the number of steps needed to obtain a constant success probability.

6.3 Operational definition of quantum "hitting"

In a classical walk, "hitting" a vertex is an event that occurs during the trajectory without changing the underlying transition rule. In a quantum walk, by contrast, measuring whether the walker is at the marked vertex changes the state, so we must specify *when* measurements occur before defining a hitting-time-style metric. We therefore consider two standard, operational measurement protocols:

- **End-only measurement:** run the unitary update for t steps without measuring, then measure the position once at time t.
- **Per-step checking:** after each step, measure the position and stop the process immediately if the outcome is the marked vertex (otherwise continue with the post-measurement state restricted to unmarked outcomes).

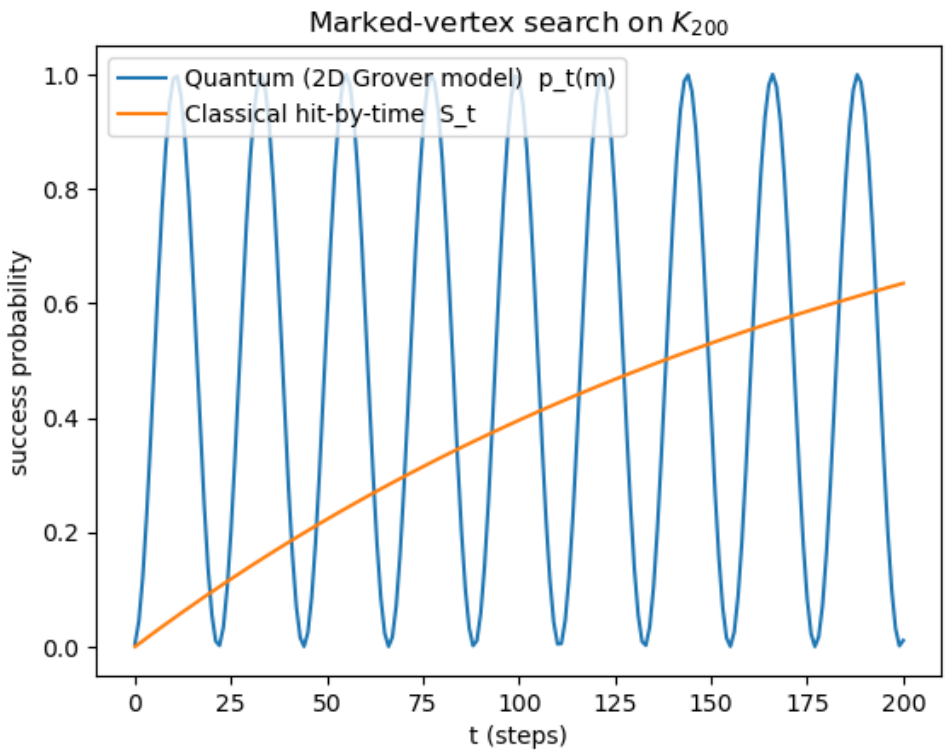


Figure 6: Marked-vertex success on K_{200}. Quantum: end-only measurement success $p_t(m)$ (oscillatory peaks). Classical: per-step checking hit-by-time S_t (monotone increase).

Both protocols are meaningful, but they answer slightly different questions; in this report we will state explicitly which protocol is used for each plotted curve.

6.4 Hitting metric: time-to-p success

To avoid ambiguity across measurement protocols, we define success using a *success-by-time* function $S_t \in [0, 1]$ and then extract a single step complexity from it. Under **end-only measurement**, we set $S_t := p_t(m)$ (the probability of observing the marked vertex when measuring only at time t). Under **per-step checking**, we set $S_t := \Pr(the process has stopped at the marked vertex by time t)$. For a fixed threshold $p \in (0, 1)$, we define the *time-to-p success* as

$$T_p \;=\; \min\{\, t \geq 0 \;:\; S_t \geq p \,\}.$$

In our experiments we typically use $p = 1/2$; for K_n, we report the classical baseline using per-step checking (so S_t corresponds to hitting-by-time), and we report the quantum curve using end-only measurement (so S_t is the marked-vertex observation probability at time t).

7 Simulation Methodology

All experiments are implemented in a single notebook so that the classical and quantum walks are compared under matched choices of graph, starting condition, and measurement

rule. We use the same vertex labeling across models and extract the same observable object—a probability distribution over vertices at each time step—so that plotted metrics are directly comparable.

7.1 Implementation overview

For a classical random walk, we represent the state at time t by a probability row vector p_t over vertices and update it by matrix multiplication:

$$p_{t+1} \;=\; p_t P,$$

where P is the stochastic transition matrix defined by the graph and the chosen classical walk rule. In the notebook implementation, these transition matrices are constructed by helper functions `P_cycle(n)`, `P_path(n)`, and `P_complete(n)`. For a coined quantum walk, we represent the state at time t by a complex unit vector $|\psi_t\rangle$ in the position–coin space and update it by applying the unitary step operator:

$$|\psi_{t+1}\rangle \;=\; U|\psi_t\rangle.$$

After each quantum update (or at selected times), we convert $|\psi_t\rangle$ into a vertex probability distribution by measuring only the position register, i.e., by summing squared amplitudes over coin states as in Section 3.

7.2 Initialization choices

To study transport on cycles and paths, we start the classical walk at a single vertex s by setting $p_0(s) = 1$ and $p_0(v) = 0$ for $v \neq s$. For the quantum walk, we use the corresponding localized position state $|s\rangle$ together with a symmetric coin state (uniform over coin labels) so that no direction is favored at time 0. For target-finding on K_n, we use an unbiased initialization in both models. For the quantum search curve, we start from the uniform superposition over all vertices (so $p_0(m) = 1/n$) and use end-only measurement to preserve coherent amplitude buildup. For the classical baseline, to model nontrivial *hitting-by-time* under per-step checking (and to match the notebook), we condition on starting at an *unmarked* vertex; from any unmarked vertex the one-step hit probability is $1/(n-1)$, hence

$$S_t \;=\; 1 - \Big(1 - \frac{1}{n-1}\Big)^t.$$

(If one instead starts classically from the uniform distribution over all vertices, this simply adds a trivial $1/n$ success probability at time 0 and does not change the $\Theta(n)$ scaling.)

7.3 Measurement protocol used in experiments

Whenever we report target-finding performance, we specify one of the two protocols introduced in Section 6: **end-only measurement** (measure the position only after t steps) or **per-step checking** (measure after each step and stop if the marked vertex is observed). For plots of spreading and return behavior on cycles and paths, we use end-only measurement at each time t to obtain the position distribution p_t without altering the unitary evolution between time steps. Stating the protocol explicitly is necessary because measuring during a quantum walk changes the state and therefore changes what "hitting" means operationally.

7.4 Graph sizes and computational constraints

All simulations use exact linear algebra on explicit vectors (and, when convenient, explicit matrices), so the main limitation is the dimension of the state space. For classical walks, the state has dimension $|V|$, while for coined quantum walks on a d-regular graph the state has dimension $|V| \cdot d$, which grows quickly for large graphs. We therefore choose values of n (and, for optional hypercubes, the dimension) that fit comfortably in memory while still allowing visible scaling trends in the measured metrics.

7.5 Metrics computed and plotted

For each time step t we compute the vertex distribution $p_t(v)$ for both models (directly for the classical walk, and via position measurement for the quantum walk). From p_t we compute: (i) a spread statistic: on C_n we use the prewrap variance (signed shortest displacement from the start), and on P_n we use the usual variance of the vertex index; (ii) the return probability $p_t(s)$ at the start vertex s; and (iii) for target-finding, the marked-vertex probability $p_t(m)$ and the derived time-to-p success T_p computed from the appropriate success-by-time function S_t for the chosen measurement protocol. These quantities are plotted as functions of t, and when relevant we repeat experiments over multiple graph sizes to compare empirical scaling between the classical and quantum models.

8 Results

The experiments reproduce the qualitative and scaling differences predicted by the spectral picture across the selected graph families. In each plot, the classical curves are produced by iterating the stochastic transition matrix on a probability distribution, and the quantum curves are produced by iterating the unitary step operator on amplitudes and then converting to a vertex distribution by measuring position (summing squared amplitudes over coin states). Unless stated otherwise, the starting vertex is fixed and the two models use matched initial conditions as described in the Simulation Methodology.

8.1 Dispersion plots on cycles/paths

On both C_n and P_n, the variance $\mathrm{Var}[X_t]$ grows much faster for the coined quantum walk than for the classical walk over the same time window. The classical curves show approximately linear growth in t before finite-size effects become visible (wrap-around on C_n and boundary effects on P_n), consistent with the distribution broadening by local averaging. In contrast, the quantum curves show substantially steeper growth consistent with the position distribution developing two outward-moving peaks, which increases typical distance from the start roughly proportionally to t and therefore increases variance roughly proportionally to t^2 in the pre-boundary regime.

8.2 Revivals on cycles

For cycles, the return probability at the start vertex, $p_t(s)$, behaves qualitatively differently between the two models. In the classical walk, $p_t(s)$ drops from 1 and then remains small; on aperiodic cases it approaches a near-uniform level, while on periodic cases (e.g., even cycles) it can exhibit mild parity-driven oscillations around that level. In the coined quantum walk,

$p_t(s)$ exhibits pronounced oscillations with intermittent peaks, indicating times when the walk partially refocuses near the start. These peaks are consistent with the phase-based picture from Section 4: because components of the state rotate rather than shrink, the measured distribution can repeatedly move away from and then back toward the starting location.

8.3 Search curves on complete graphs

On K_n with a marked vertex m, the classical baseline reaches constant success only on a $\Theta(n)$ step scale under per-step checking (equivalently, it has expected hitting time $\Theta(n)$). The quantum search dynamics behaves differently: under end-only measurement, the marked-vertex observation probability $p_t(m)$ rises sharply to a peak and then oscillates, with the first large peak occurring after far fewer steps than the classical baseline. When we repeat this experiment across increasing n, the location of the first peak shifts in a way consistent with scaling like $\sqrt{n}$ steps to reach a constant success probability.

8.4 Scaling summary

Across tested sizes, the cycle/path experiments support faster spreading for the quantum walk as measured by variance growth, and the complete-graph experiments support faster target-finding as measured by time-to-p success. In particular, the classical baseline on K_n reaches constant success only after $\Theta(n)$ steps (in the hitting-by-time sense), while the quantum search curve reaches constant observation probability after $\Theta(\sqrt{n})$ steps under end-only measurement. While constants depend on modeling choices (coin, shift, and measurement protocol), the qualitative separation is stable: classical updates smooth by averaging and suppress non-stationary components, whereas unitary updates preserve components and change observable probabilities through phase-driven cancellation and reinforcement.

Discussion

The main takeaway is that a spectral viewpoint provides a single, consistent explanation for the differences we observe between classical and coined quantum walks. In the classical case, repeated multiplication by a stochastic transition matrix shrinks all components except the stationary one, producing diffusion-like spreading and eventual stabilization. In the quantum case, repeated multiplication by a unitary step operator preserves all components but rotates them by different phases, so interference can both accelerate spreading and create recurrence peaks, and can also concentrate probability on a marked vertex under a simple marking rule.

Interpretation through spectra

The results on cycles and paths align with the idea that classical behavior is dominated by mode decay (components with eigenvalue magnitude less than 1 shrink over time), while quantum behavior is dominated by phase evolution (components rotate and can alternately cancel and reinforce). The return-probability peaks on cycles are naturally explained by partial phase realignment, and the sharp rise in marked-vertex probability on complete graphs is naturally explained by the fact that symmetry restricts evolution to a very small effective set of degrees of freedom when starting from a symmetric initial state. In both settings, the

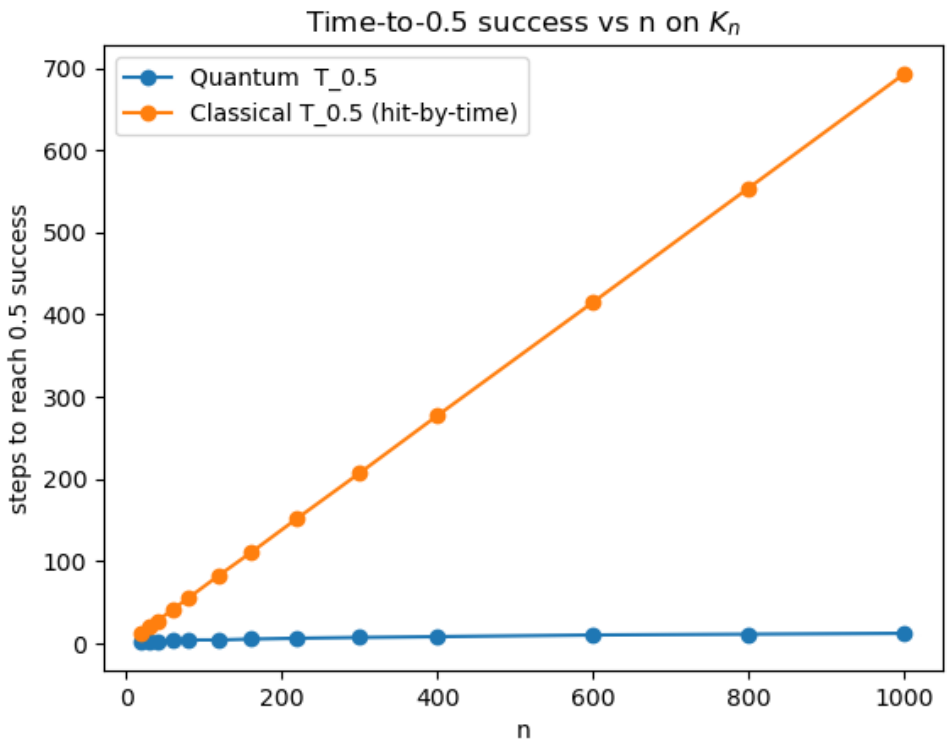


Figure 7: Empirical scaling of $T_{0.5}$ on K_n: classical hit-by-time grows linearly in n, while the quantum peak time grows much more slowly, consistent with $\Theta(\sqrt{n})$ behavior.

eigenstructure of the relevant update matrix is not just a descriptive tool but a predictive one: it indicates which behaviors are expected to persist as n grows.

Limitations

Our comparisons are strongest on highly structured graphs and under idealized unitary evolution. First, the coined quantum walk depends on explicit modeling choices (coin operator, shift rule, and measurement protocol), and different choices can change constants or even qualitative features on some graphs. Second, our experiments use end-only measurement for transport metrics and specified measurement rules for target-finding; measuring more frequently generally changes quantum dynamics. Third, we restrict attention to graphs where symmetry makes both analysis and simulation clean; on irregular graphs, interpreting behavior through eigenstructure can be less direct and may require additional tools.

Future directions

Several natural extensions fit within the same framework. One direction is to include hypercubes as an additional symmetric family and compare transport and target-finding under the same metrics. Another direction is to study time-averaged position distributions more systematically as a "mixing-like" proxy for quantum walks and compare them to classical stationary behavior on the same graphs. A final direction is to investigate robustness by adding

simple noise models (for example, occasional random measurements of the coin or position) to see how the behavior interpolates between coherent quantum evolution and classical diffusion.