Chapter 1

Universality of Quantum Walk

Quantum walk is an intuitive framework for developing quantum algorithms, inspired by the classical model of random walk. This framework has lead to examples of exponential speedups over classical computation [5], as well as optimal algorithms for element distinctness [1] and formula evaluation [6]. Additionally, the framework of Chapter **??** can be thought of as a special kind of continuos-time quantum walk.

With all of these algorithmic uses, we would then wonder at the computational power of this model. Using the ideas of graph scattering, Childs [3] was able to show that the model of continuous time quantum walk is universal for quantum computation (and later showed the same for discrete-time quantum walk [4]). This chapter will also show this result, but with a slightly different proof technique.

In particular, we will use results on the scattering behavior of finite-length wave-packets to implement single gates, and we will then show how to compose these scattering events for an entire computation. This chapter is really a primer for Chapter **??**, as many of the proof-techniques and ideas of this chapter will be used for the multi-particle case as well.

Most of this chapter will be devoted to showing how to simulate a circuit of a given form via graph scattering.

[**TO DO:** *Move wavepacket propagation to this chapter?*] [**TO DO:** *Make many figures (after writeup of other chapters)*] [**TO DO:** *Explict encodings for different momenta*]

1.1 Single qubit simulation

With our eventual goal of simulating an entire circuit via graph scattering, we will first need to understand how to perform single-qubit computations. We will use many of the results of Chapter **??**, and show that specific scattering behavior can be used as a computational tool. This section will first encode the qubit, then show how to have a simulate a single gate, and finally show how to simulate multiple single-qubit gates. These results will then be generalized for multiple-qubits in Section 1.2, and will be used nearly verbatim in Chapter **??**.

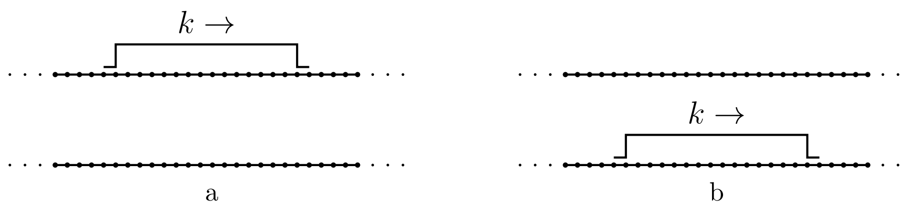

Figure 1.1: A qubit is encoded using single-particle wave packets at momentum k. (a) An encoded $|0\rangle$. (b) An encoded $|1\rangle$.

1.1.1 Single qubit encoding

In our search for simulations, we will first need to encode the logical state into some graph state. We can then take inspiration from the current literature, or else see motivation from classical asynchronous systems, to encode our logical system via dual-rails. In particular, a single qubit will correspond to two infinite paths, with a single wave-packet at specified momentum k traveling along one of the two paths. If the particle is located on the first (top) path, then the encoded qubit is in the logical state $|0\rangle$, while if the particle is on the second (bottom) path then the encoded qubit is in the logical state $|1\rangle$. Schematically, this can be seen in Figure 1.1.

[**TO DO:** *change figure to Gaussian as opposed to square?*]

If we didn't mind using an infinite Hilbert space to encode our qubits, we could actually use the eigenstates of the two paths to correspond to the two logical states, but we will eventually want to assume that the encodings have a well-defined position in space to ensure that we need only measure a (relatively) small number of qubits in order to determine the location of the particle with high probability. To ensure this localization in space (and to use some of our error bounds on the time evolution), we will assume that the logical states are encoded using a truncated Gaussian wave-packet, with four attributes that specify the state: the momentum k, the standard deviation σ, the center of mass μ, and the cutoff range L (which will be closely related to σ). With these four values, and assuming that the vertices of the infinite path are labeled as (x, z) for $x \in \mathbb{Z}$ and $z \in \mathbb{F}_2$, we then have that the logical qubit in our system will be encoded into the states

$$|z\rangle_{\log} = \gamma \sum_{x=\mu-L}^{\mu+L} e^{ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z\rangle. \tag{1.1}$$

It is important to realize that none of these four values depend on the value of the encoded qubit; this will allow us to interfere the wave-packets arising from different paths to the same computational basis path as there will be no extraneous information about the logical state.

This encoding is specifically chosen so that we can use Theorem **??**, and guarantee various attributes about the time evolution of such systems.

1.1.2 One single-qubit unitary

With an actual encoding of a logical qubit, the next step will be to apply an encoded unitary to the logical state. However, our current encoding is on two (disconnected) paths, and as such if we want to apply any unitary that mixes amplitudes among the two basis states we somehow need to connect the two paths. (Unitaries diagonal in the computational basis will use the same formalism, but they have additional constraints that might make them easier to apply.)

Note that Chapter **??** was all about connecting (semi-) infinite paths, where the amplitudes move from one path to another. As hinted in the chapter, we can implement encoded unitaries in this manner, if we restrict ourselves to specific momenta and specific scattering gadgets. Namely, we will examine graphs $\widehat{G}$ with four terminal vertices such that at the momentum k encoding the qubits, the scattering matrices take the form

$$S(k) = \begin{pmatrix} 0 & U^T \\ U & 0 \end{pmatrix}, \tag{1.2}$$

where U is a specific 2×2 unitary matrix. This will then allow us to apply the unitary U to the encoded qubit.

More explicitly, we will have four semi-infinite paths, and we will label the four paths by 0_{in}, 1_{in}, 0_{out}, and 1_{out} (where this labeling is the same as in equation (1.2)). We assume that the wave-packet encoding a qubit travels toward the graph $\widehat{G}$ along the two paths 0_{in} and 1_{in}. Far from the graph the evolution of this wavepacket is nearly identical to that of an infinite path, and thus our encoded qubit is well defined. As the wavepacket scatters through the graph $\widehat{G}$, the state of the qubit is not well defined, but after scattering, most of the amplitude is on the 0_{out} and 1_{out} paths, and is in the form of an encoded qubit.

For specific μ, L, σ, and t, we the have from Theorem **??** that the outgoing wave-packet for the two computational basis states is well approximated by the wave-packet corresponding to the state $U|z\rangle$. If we remember that the form of the wave-packet doesn't depend on the value of the initial encoded qubit, we can see that the evolution of the two basis states interfere, and thus for any encoded state $|\phi\rangle$, the outgoing wavepacket is well approximated by the encoded $U|\phi\rangle$. This is exactly what we were looking for.

[**TO DO:** *make graph?*]

1.1.3 Evolution on a finite graph

Unfortunately, a single unitary will not be sufficient for our purposes; while we could probably find a four-terminal graph that computes whether a given circuit accepts or rejects its input, most of the computation would be found in the construction of the underlying graph, as opposed to the evolution itself. To ensure that the computational power arises from the time evolution of the system, we will need to place multiple graphs as scattering obstacles for the computation. This causes problems, though, in that we extensively utilize the semi-infinite paths in our analysis; we somehow need to truncate the graph while maintaining our results about the time-evolution.

To do this, we will apply our truncation lemma (Lemma **??**), as it was designed specifically for this reason. Assuming that two Hamiltonians are identical on some set of basis states, and assuming that the support of the initial state is far (in some specified sense) from the difference, then the evolution of the state is the same for the two Hamiltonians, up to a small error term. By using this lemma on the scattering graph with semi-infinite paths,

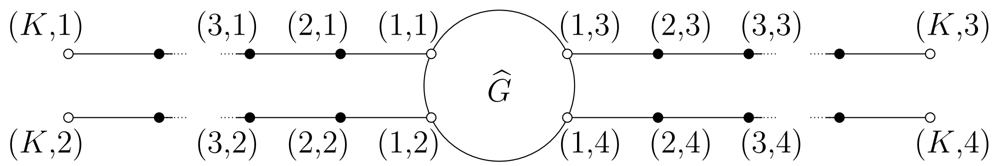

Figure 1.2: A graph $G(K)$ used to perform a single-qubit gate on an encoded qubit.

we can then see that if the paths are long enough (as compared to the location of the initial state), then the evolution of an initial wave-packet is relatively unchanged by the removal of the far vertices. Basically, Lemma **??** will allow us to prove an analog of Theorem **??** for finite graphs.

More concretely, let $H = A(G)$ be the Hamiltonian for a single particle scattering off of a finite graph $\widehat{G}$ with N paths. Let $G(K)$ be the finite graph obtained from G by truncating each of the paths to have a total length K (so that the endpoints of the paths are labeled (K, j) for $j \in [N]$), and choose $\widetilde{H} = A(G(K))$ (see Figure **??**). Let the subspace $\mathcal{K}$ be spanned by basis states corresponding to vertices in $G(K)$. Choose a momentum $k \in (-\pi, 0)$, a position μ, and a cutoff length L, and let $|\Phi\rangle = |\psi^j(0)\rangle$ be the same initial state as in Theorem **??**. We will choose the evolution time T so that for $0 \leq t \leq T$, the time-evolved state remains far from the vertices labeled (K, j) (for each $j \in \{1, \ldots, N\}$), and thus far from the effect of truncating the paths. Note that this requires $K > \mu + L$. More precisely, we will choose $T = \mathcal{O}(L)$ and $K = \mathcal{O}(L)$ so that, for times $0 \leq t \leq T$, the state $|\alpha^j(t)\rangle$ from Theorem **??** has no amplitude on vertices within a distance $N_0 = \Omega(L)$ from the endpoints of the paths. For such times t we have

$$(1 - P)\, H^r |\alpha^j\,(t)\rangle = 0 \text{ for all } 0 \leq r < N_0, \tag{1.3}$$

where P is the projector onto $\mathcal{K}$. With these values, we can apply Lemma **??** where $W = H = A(G)$, $|\gamma(t)\rangle = |\alpha^j(t)\rangle$, and the bound $\delta = \mathcal{O}(\sqrt{\log L/L})$ from Theorem **??**. The lemma then says that, for times t such that $0 \leq t \leq T$,

$$\left\| \left(e^{-iA(G)t} - e^{-iA(G(K))t}\right) |\psi^j(0)\rangle \right\| = \mathcal{O}\left(\sqrt{\frac{\log L}{L}}\right) \tag{1.4}$$

so, for $0 \leq t \leq T$, when combined with Theorem **??**, we can see

$$\left\| e^{-iA(G(K))t} |\psi^j(0)\rangle - |\alpha^j(t)\rangle \right\| = \mathcal{O}\left(\sqrt{\frac{\log L}{L}}\right). \tag{1.5}$$

In other words, for small enough evolution times, the conclusion of Theorem **??** still holds if we replace the full Hamiltonian $A(G)$ with the truncated Hamiltonian $A(G(K))$ (albeit with a larger constant). Note that this analysis is rather informal, and is more to give an intuition for the more exact analysis.

With the guaranteed bounds on the scattering behavior for finite graphs, we can give explicit bounds on the time-evolution of encoded qubits. In particular, let us assume that $\widehat{G}$ is a four-terminal gadget used to implement a unitary U at momentum k, and let us assume that our initial states are encoded as Gaussian wave-packets a distance μ from the graph, with a cutoff distance L. We will give explicit values of K, along with μ and L, so that the scattering event will cause the unitary U to be applied to the encoded qubits, along with bounds on the error term.

Explicitly, assuming that the four paths are labeled as in Figure 1.2, where 0_{in}, 1_{in}, 0_{out} and 1_{out} are labeled as 1, 2, 3, and 4, respectively, we have that our input logical basis states are

$$|z\rangle_{\text{log,in}} = \gamma \sum_{x=\mu-L}^{\mu+L} e^{ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+1\rangle, \tag{1.6}$$

where we assume that $\sigma = \frac{L}{2\sqrt{\log L}}$, as in Theorem **??**. Further, we can make us of the theorem, noting that $|z\rangle_{\text{log,in}}$ are of the form $|\alpha^{z+1}(0)\rangle$, to define output logical states as well:

$$|z\rangle_{\text{log,out}} = \gamma e^{-2iT\cos k} \sum_{x=\mu-L}^{\mu+L} e^{-ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+3\rangle, \tag{1.7}$$

where $T = \frac{\mu}{\sin|k|}$. Note that the momentum k for the output logical states implies that the particles are moving away from the graph $\widehat{G}$. In addition to these logical basis states, we can define logical superpositions for a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ as

$$|\psi\rangle_{\text{log,in}} = \alpha|0\rangle_{\text{log,in}} + \beta|1\rangle_{\text{log,in}} \tag{1.8}$$

and

$$|U\psi\rangle_{\text{log,out}} = \big(\alpha U_{00} + \beta U_{01}\big)|0\rangle_{\text{log,out}} + \big(\alpha U_{10} + \beta U_{11}\big)|1\rangle_{\text{log,out}}. \tag{1.9}$$

With these definitions, we will want to show that the input states evolve to the corresponding output states, in a manner similar to Theorem **??**. Working through the math, we then find:

Lemma 1. *Let $k \in (-\pi, 0)$, and let $\widehat{G}$ be a four-terminal gate gadget, such that its scattering matrix at momentum k is of the form (1.2). Letting the logical states $|z\rangle_{log,in}$ and $|z\rangle_{log,out}$ be defined as in (1.6) and (1.7), where $\mu \geq 2L$ and $K \geq \frac{5\mu}{3}$ and $T = \frac{\mu}{\sin|k|}$, we have that there exists some constant ξ such that for all $0 \leq t \leq T$*

$$\left\| e^{-iA(G(K))t}|\phi(0)\rangle - |\phi(t)\rangle \right\| \leq \xi \sqrt{\frac{\log L}{L}}, \tag{1.10}$$

where

$$|\phi(t)\rangle = \alpha|\alpha^1(t)\rangle + \beta|\alpha^2(t)\rangle, \tag{1.11}$$

*and the $|\alpha^j(t)\rangle$ are as defined in Theorem **??**. In particular, we have*

$$\left\| e^{-iA(G(K))T}|\psi\rangle_{log,in} - |U\psi\rangle_{log,out} \right\| \leq \xi \sqrt{\frac{\log L}{L}}. \tag{1.12}$$

Proof. Note that

$$\begin{aligned}
&\left\| e^{-iA(G(K))t} |\, \phi(0) \,\rangle - |\, \phi(t) \,\rangle \right\| \\
&\qquad \leq |\alpha| \left\| e^{-iA(G(K))t} |\, \alpha^1(0) \,\rangle - |\, \alpha^1(t) \,\rangle \right\| + |\beta| \left\| e^{-iA(G(K))t} |\, \alpha^2(0) \,\rangle - |\, \alpha^2(t) \,\rangle \right\|. \qquad (1.13)
\end{aligned}$$

We now have nearly have the form of the bound in Theorem **??**, but where we use truncated paths.

We will use Lemma **??**, with $H = A(G)$, $\tilde{H} = A(G(K))$, and $N_0 = K - \mu - L \geq \frac{\mu}{6}$, and where the error bound $\delta = \chi\sqrt{\frac{\log L}{L}}$ comes from Theorem **??**. Assuming that L is taken large enough so that $\delta < 1$, the lemma then gives us that for all $0 \leq t \leq T$,

$$\left\| \left(e^{-iA(G)t} - e^{-iA(G(K))t}\right) |\, \alpha^j(0) \,\rangle \right\| \leq \left(\frac{4e\|A(G)\|t}{N_0} + 2 \right) \left[\chi\sqrt{\frac{\log L}{L}} + 2^{-N_0}\left(1 - \chi\sqrt{\frac{\log L}{L}}\right) \right]. \qquad (1.14)$$

If we then note that $\|A(G)\|$ is bounded by the maximum degree of the graph G which is given by d (a constant), and our bounds on N_0, we then have

$$\left\| \left(e^{-iA(G)t} - e^{-iA(G(K))t}\right) |\, \alpha^j(0) \,\rangle \right\| \leq \left(\frac{12ed}{\mu} \frac{\mu}{\sin|k|} + 2 \right) (\chi + 1) \sqrt{\frac{\log L}{L}} \leq \zeta\sqrt{\frac{\log L}{L}}, \qquad (1.15)$$

where ζ is a constant (but does depend on k and the graph $\widehat{G}$).

We can then combine these results, as

$$\begin{aligned}
&\left\| e^{-iA(G(K))t} |\, \alpha^j(0) \,\rangle - |\, \alpha^j(t) \,\rangle \right\| \\
&\qquad \leq \left\| \left(e^{-iA(G(K))t} - e^{-iA(G)t}\right) |\, \alpha^j(0) \,\rangle \right\| + \left\| e^{-iA(G)t} |\, \alpha^j(0) \,\rangle - |\, \alpha^j(t) \,\rangle \right\| \leq (\chi + \zeta) \sqrt{\frac{\log L}{L}}. \qquad (1.16)
\end{aligned}$$

From this, we can then see that

$$\begin{aligned}
&\left\| e^{-iA(G(K))t} |\, \phi(0) \,\rangle - |\, \phi(t) \,\rangle \right\| \\
&\qquad \leq |\alpha| \left\| e^{-iA(G(K))t} |\, \alpha^1(0) \,\rangle - |\, \alpha^1(t) \,\rangle \right\| + |\beta| \left\| e^{-iA(G(K))t} |\, \alpha^2(0) \,\rangle - |\, \alpha^2(t) \,\rangle \right\| \qquad (1.17) \\
&\qquad \leq \big(|\alpha| + |\beta|\big) (\chi + \zeta) \sqrt{\frac{\log L}{L}}, \qquad (1.18)
\end{aligned}$$

and by setting $\xi = \sqrt{2}(\chi + \zeta)$ we have the requisite bound (for large enough L).

If we then note that $\phi(0) = |\psi\rangle_{\text{log,in}}$ and $\phi(T) = |U\psi\rangle_{\text{log,out}}$, we also have the particular bound we were looking for. □

Essentially, Lemma 1 tells us that even when truncated to finite length paths, the scattering events on our graphs apply an encoded unitary to the logical states. This is represented pictorially in Figure 1.3.

[**TO DO:** *correct Figure 1.3.*]

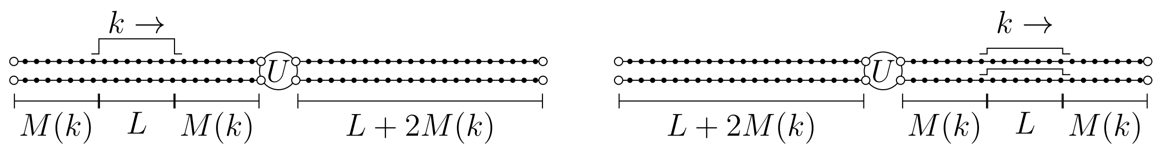

Figure 1.3: A single-qubit gate U acts on an encoded qubit. The wave packet starts on the paths on the left-hand side of the figure, a distance $M(k)$ from the ends of the paths. After time $t_{\mathrm{I}} = 3L/2$ the logical gate has been applied and the wave packet has traveled a distance $2M(k) + L$ (up to error terms that are bounded as $\emptyset(L^{-1/4})$).

1.1.4 Multi-gate computations

Now that we have a good approximation for the time evolution of a scattering event on a finite-sized graph, we can expand our results to multiple scattering events. In particular, if we can guarantee that the eventual graph corresponding to a given circuit locally looks like $G(K)$ for each unitary in the circuit, we will be able to use the results of Lemma 1 iteratively. Essentially, if the graph is two semi-infinite paths with regularly spaced scattering obstacles, our previous results will still apply.

[**TO DO:** *make a figure*]

Along these lines, let us assume that a single-qubit circuit $\mathcal{C}$ is composed of g unitaries, where the ith unitary applied is given by U_i. Moreover, let us assume that at a momentum k, the graphs $\widehat{G}_i$ have scattering matrices of the form (1.2) corresponding to the unitary U_i (i.e., at momentum k, the graph $\widehat{G}_i$ implements an encoded U_i). We can then construct a graph $G_{\mathcal{C}}$ which we will use to compute the circuit $\mathcal{C}$ using wave-packets at momentum k.

The graph $G_{\mathcal{C}}$ is constructed by combining the $G_i(K)$ into a single graph, where $G_i(K)$ is defined in Section **??**, by associating the output paths of $G_i(K)$ with the input paths of $G_{i+1}(K)$. Assuming that the vertices of $G_i(K)$ are labeled as (u, i), this essentially means that most of the vertices along the long paths have two labels, $(x, 3, i)$ and $(K+1-x, i+1)$ or $(x, 4, i)$ and $(K+1-x, 2, i+1)$. Equivalently, the graph $G_{\mathcal{C}}$ can be constructed by removing the input paths (paths 1 and 2) for all the $G_i(K)$ (except for $G_1(K)$), shorten each of the terminal paths by 1 (except for $G_g(K)$), and then connect the end of the paths for $G_i(K)$ to the input terminals of $G_{i+1}(K)$.

[**TO DO:** *make a simple figure*]

With this construction of $G_{\mathcal{C}}$, note that if we look only at the vertices supported within a distance of $K-2$ of $\widetilde{G}_i$, we actually have the graph $G_i(K-1)$. As such, we will be able to use Lemma **??** and Lemma 1 to determine the evolution while a Gaussian wave-packet is located near the graph $\widehat{G}_i$. If we assume that the initial wave-packet is located in the correct position near $\widehat{G}_i$, we can iteratively apply this idea, where the "input" logical state for the $(i+1)$th scattering event is simply the "output" from the ith scattering event. As such, the logical state after the gth scattering event will correspond to the logical state after the circuit $\mathcal{C}$ has been applied.

Concretely, let us choose some cutoff length L, set $\sigma = \frac{L}{2\sqrt{\log L}}$, chose $\mu = 2L$, $K = 2\mu - 1$ and $T = \frac{\mu}{\sin|k|}$. With these choices, our initial logical state will be nearly identical to (1.6), but where the basis states also have a label corresponding to fact that there are multiple long paths:

$$|z\rangle_{\text{log,in}} = \gamma \sum_{x=\mu-L}^{\mu+L} e^{ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+1, 1\rangle. \tag{1.19}$$

In a similar manner, the final state of the qubit will be defined as

$$|z\rangle_{\text{log,out}} = \gamma e^{-2iTg\cos k - 2ik(g-1)\mu} \sum_{x=\mu-L}^{\mu+L} e^{-ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+3, g\rangle, \tag{1.20}$$

where the additional global phase arises from the change in bases between the various scattering events. We will also need to define logical states at several times throughout the computation, corresponding to the states after each applied unitary. We will thus define the logical state after the jth scattering event (and before the $(j+1)$th scattering event) as

$$|z\rangle_{\text{log},j} = \gamma e^{-2iTj\cos k - 2ik\mu(j-1)} \sum_{x=\mu-L}^{\mu+L} e^{-ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+3, j\rangle \tag{1.21}$$

$$= \gamma e^{-2iTj\cos k - 2ijk\mu} \sum_{x=\mu-L}^{\mu+L} e^{ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+1, j+1\rangle. \tag{1.22}$$

Except for the additional (global) phase arising from the change of basis, these are exactly the logical states necessary for Lemma 1.

We can now bound the error in approximating the time evolution of an initial state with the output state corresponding to the logical application of the circuit:

$$\begin{aligned}&\left\| e^{-iA(G_{\mathcal{C}})gT} |\psi\rangle_{\text{log,in}} - |U_{\mathcal{C}}\psi\rangle_{\text{log,out}} \right\| \\ &\qquad \leq \sum_{j=0}^{g-1} \left\| e^{-iA(G_{\mathcal{C}})T} |U_j U_{j-1} \cdots U_1 \psi\rangle_{\text{log},j} - |U_{j+1} U_j \cdots U_1 \psi\rangle_{\text{log},j+1} \right\|\end{aligned} \tag{1.23}$$

Note that each individual term is close to that in Lemma 1, but where the Hamiltonian is given by $A(G_{\mathcal{C}})$ as opposed to $A(G(K))$. However, we can use Lemma **??**, with $H = A(G_{\mathcal{C}})$, $\tilde{H} = G(K-1)$, $N_0 = \frac{\mu}{4}$, and the error δ from Lemma 1, we have that for all logical states $|\phi\rangle$,

$$\begin{aligned}&\left\| e^{-iA(G_{\mathcal{C}})T} |\phi\rangle_{\text{log},j} - |U_{j+1}\phi\rangle_{\text{log},j+1} \right\| \\ &\leq \left(\frac{16e\|A(G_{\mathcal{C}})\|T}{\mu} + 2 \right) \left[\xi \sqrt{\frac{\log L}{L}} + 2^{-\mu+L+2} \left(1 - \chi \sqrt{\frac{\log L}{L}} \right) \right] \qquad (1.24) \\ &\leq \kappa \sqrt{\frac{\log L}{L}}, \qquad (1.25)\end{aligned}$$

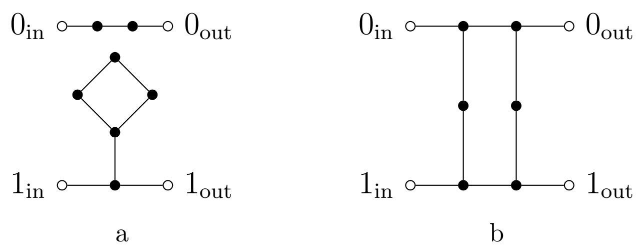

Figure 1.4: Encoded one-qubit gates at $k = -\frac{\pi}{4}$. (a) A phase gate. (b) Basis-changing gate.

where κ depends on k and the maximum degree of $G_{\mathcal{C}}$ (which we assume to be constant). We can then see that

$$\begin{aligned}
&\left\| e^{-iA(G_{\mathcal{C}})gT} |\,\psi\,\rangle_{\text{log,in}} - |\,U_{\mathcal{C}}\psi\,\rangle_{\text{log,out}} \right\| \\
&\quad \leq \sum_{j=0}^{g-1} \left\| e^{-iA(G_{\mathcal{C}})T} |\,U_j U_{j-1} \cdots U_1 \psi\,\rangle_{\text{log},j} - |\,U_{j+1} U_j \cdots U_1 \psi\,\rangle_{\text{log},j+1} \right\| \qquad (1.26)\\
&\quad \leq g\kappa \sqrt{\frac{\log L}{L}}. \qquad (1.27)
\end{aligned}$$

As such, if we take L larger than $g^2\kappa^2 \log^2(g\kappa)$ we find that the error can be made arbitrarily small, and thus we were able to simulate the circuit $\mathcal{C}$ via scattering on a constant-degree graph with $\mathcal{O}(g^3 \log^2 g)$ vertices.

1.1.5 Explicit encodings

[**TO DO:** *Find universal gate set for several momenta ($\pi/2$, $\pi/4$, $\pi/3$, etc.)*]

Thus far, we have given a general recipe for constructing graphs that implement particular encoded unitaries for wave-packets at a particular momentum k, assuming that we have scattering gadgets at momentum k that implement a universal set of one-qubit unitaries. While the framework is useful, in that it works for any such momentum, we need to show that there exists at least one momentum for which a universal set of gadgets exist.

While we could search ourselves for these gadgets, Blumer, Underwood, and Feder ([2]) have already searched over all graphs with up to 9 vertices for graphs that implement encoded unitaries at particular momentum. Combined with the universal gadgets given by Childs [3], these will be enough for our purposes.

In particular, at momentum $k = -\frac{\pi}{4}$ or at momentum $p = -\frac{3\pi}{4}$, the two gadgets from [3] are depicted in (Create Figure). Moreover, they implement the unitaries:

For momentum $\pi/2$, we have that the graph (create Figure) implements a Hadamard gate. Additionally, we have from [2] that the graphs in Figure ???? (find) implement unitaries that form the V-basis.

(I'd really like to also have universal gates at $\pi/3$ and $2\pi/3$, since they move at the same speed, and we have a momentum switch between them. However, I don't have one yet.

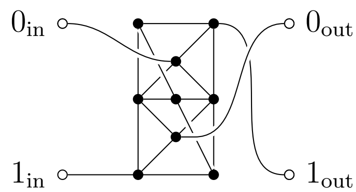

Figure 1.5: Graph implementing a Hadamard gate at $k = -\frac{\pi}{2}$.

1.2 Multi-qubit computations

With our understanding of single-qubit circuit simulation via scattering, we can now try to apply our results to multi-qubit circuts. The intuitive construction will remain the same, but the requisite number of vertices will become rather large. In particular, our construction will require an exponential number of long paths, corresponding to the exponential size of the corresponding Hilbert space.

To begin, let us give the encoding of n qubits in our scattering framework. As in Section 1.1.1, we will encode the state as a wave-packet traveling along an infinite path, where the value of the qubit is encoded in the path on which the particle is located. For a single qubit, this meant that we had two infinite paths, corresponding to logical 0 and 1. For n qubits, however, this means that we need 2^n infinite paths, one path corresponding to each logical basis state.

We will still have four important quantities that are independent of the state of the qubit, namely the momentum of the wave-packet k, the position of the center of the wave-packet μ (which depends on t), the cutoff distance L, and the standard deviation of the Guassian σ. As such, if we label the 2^n infinite paths by the strings $\mathbf{z} \in \mathbb{F}_2^n$, and the vertices as $(x, \mathbf{s})$ for $x \in \mathbb{Z}$ and $\mathbf{z} \in \mathbb{F}_2^n$, we have that the logical states are encoded in the wave-packets

$$|\mathbf{z}\rangle_{\log} = \gamma \sum_{x=\mu-L}^{\mu+L} e^{ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, \mathbf{z}\rangle. \tag{1.28}$$

Note that we again use this encoded form so that we will be able to analyze the dynamics via Theorem **??**.

1.2.1 One multi-qubit unitary

Now that we have an encoding of our qubits, we will need to somehow apply encoded unitary gates. We have already done most of the work in Section 1.1, and we just need to show how to use the single-qubit results in our larger encoding, and how to perform multi-qubit entangling gates.

Our implementation of single-qubit unitaries for multi-qubit computations is to use many copies of the single-qubit implementation. In particular, since a single qubit unitary U acting on qubit $w \in [n]$ can be written as

$$\mathbb{I}_{2^{w-1}} \otimes U \otimes \mathbb{I}_{2^{n-w}} = \sum_{x \in \mathbb{F}_2^{w-1}, y \in \mathbb{F}_2^{n-w}} |x\rangle\langle x| \otimes U \otimes |y\rangle\langle y|, \tag{1.29}$$

we can apply the unitary U on the encoded w qubit by ensuring that the scattering occurs for each computational basis state of the other qubits. This means that by placing 2^{n-1} copies of the graph $\widehat{G}$ as obstacles in the paths, one for each computational basis state, the scattering behavior is exactly as expended. We will call the infinite graph corresponding to the unitary U acting on n qubits G_U^n.

[**TO DO:** *make multi qubit gate figure*]

For multi-qubit entangling unitaries, the solution is even more simple; we simply relabel the output paths. If we note that many multi-qubit gates such as a controlled-NOT gate or a Toffoli gate simply permutes the computational basis states, along with the fact that the particular path a particle travels along corresponds to its logical state, by relabeling the paths, or equivalently by permuting the paths, we apply an encoded entangling gate. Schematically, this can be seen in Figure **??**. Note that this method of applying a multi-qubit gate is independent of the encoding momenta, and thus can be used for all momenta (so that we don't need to find additional graphs with scattering matrices at the encoding momentum).

Assuming that a given two-qubit unitary V occurs after some single-qubit unitary U, we construct the graph G_{VU}^n implementing VU by taking a copy of G_U^n, and then permuting its output paths. Note that this means that for a given copy of $\widehat{G}_j$, the logical states corresponding to the input paths might be different from the logical states corresponding to the output paths. However, since V is only a two-qubit permutation, the output paths can be efficiently determined for a given copy of $\widehat{G}_j$.

As in Section 1.1.3, we can then define $G_{VU}^n(K)$ as the graph that that remains after truncating each of the infinite paths of G_{VU}^n to length K.

[**TO DO:** *make Figure* **??** *entangling gate*] .

With all of this, and assuming that the 2^{n+1} paths are labeled as in Figure **??** where the path for $\mathbf{z}_{\text{in}}$ for $\mathbf{z} \in \mathbb{F}_2^n$ is labeled as $z+1$ while the $\mathbf{z}_{\text{out}}$ are labeled as $z + 2^n + 1$, we have that our input logical basis states are

$$|\mathbf{z}\rangle_{\text{log,in}} = \gamma \sum_{x=\mu-L}^{\mu+L} e^{ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+1\rangle, \tag{1.30}$$

where we assume that $\sigma = \frac{L}{2\sqrt{\log L}}$, as in Theorem **??**. Note that this generalizes the single particle case (1.6) to more input and output paths. We can then take inspiration from the single qubit case, and define the output logical states as well:

$$|\mathbf{z}\rangle_{\text{log,out}} = \gamma e^{-2iT\cos k} \sum_{x=\mu-L}^{\mu+L} e^{-ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+2^n+1\rangle, \tag{1.31}$$

where $T = \frac{\mu}{\sin|k|}$. In addition to these logical basis states, for an arbitrary state $|\psi\rangle = \sum_{\mathbf{z}\in\mathbb{F}_2^N} \alpha_{\mathbf{z}} |\mathbf{z}\rangle$ in $\mathbb{C}^{\mathbb{F}_2^n}$

$$|\psi\rangle_{\text{log,in}} = \sum_{\mathbf{z}\in\mathbb{F}_2^n} \alpha_{\mathbf{z}} |\mathbf{z}\rangle_{\text{log,in}} \tag{1.32}$$

and

$$|U\psi\rangle_{\text{log,out}} = \sum_{\mathbf{y},\mathbf{z}\in\mathbb{F}_2^N} U_{\mathbf{z},\mathbf{y}}\alpha_{\mathbf{y}}|\mathbf{z}\rangle_{\text{log,out}}, \tag{1.33}$$

where U is thought of as a unitary on n qubits. With these definitions, we will want to show that the input states evolve to the corresponding output states. This nearly follows from Lemma 1, but we need to do some small work showing that the errors don't grow like the number of paths.

Corollary 1. *Let $k \in (-\pi, 0)$, let $\widehat{G}$ be a four-terminal gate gate, such that its scattering matrix at momentum k is of the form (1.2) for a given unitary U, and let V be a permutation of the underlying basis states. Letting the logical states $|\mathbf{z}\rangle_{log,in}$ and $|\mathbf{z}\rangle_{log,out}$ be as in (1.30) and (1.31), where $\mu \geq 2L$ and $K \geq \frac{5\mu}{3}$ and $T = \frac{\mu}{\sin|k|}$, we have that there exists some constant ξ such that for all $0 \leq t \leq T$,*

$$\left\| e^{-iA(G^n_{VU}(K))t}|\phi(0)\rangle - |\phi(t)\rangle \right\| \leq \xi\sqrt{\frac{\log L}{L}}, \tag{1.34}$$

where

$$|\phi(t)\rangle = \sum_{\mathbf{z}\in\mathbb{F}_2^n} \beta_{\mathbf{z}}|\alpha^{z+1}(t)\rangle, \tag{1.35}$$

*and the $|\alpha^j(t)\rangle$ are as defined in Theorem **??**. In particular, we have for any $|\psi\rangle \in \mathbb{C}^{\mathbb{F}_2^n}$,*

$$\left\| e^{-iA(G^n_V(K))T}|\psi\rangle_{log,in} - |U\psi\rangle_{log,out} \right\| \leq \xi\sqrt{\frac{\log L}{L}}. \tag{1.36}$$

Proof. Note that $G^n_{VU}(K)$ is a disconnected graph, with 2^{n-1} components. As such, we have that $e^{-iA(G^n_{VU}(K)t}$ decomposes into the product of 2^{n-1} commuting operators, all acting on disjoint Hilbert spaces. Because of this, error cannot interfere between the various disconnected paths, and thus the total error is bounded by the maximum error on any individual component.

In each component, however, we can use Lemma 1 to see that the first part of the corollary holds, with the appropriate error (and with a constant equal to that of Lemma 1). Hence, the total error is bounded by $\xi\sqrt{\frac{\log L}{L}}$.

For the second part of the corollary, we can use Lemma 1 to see that the result holds on each component of $G^n_{VU}(K)$, and thus holds in general. □

1.2.2 Multi-gate computations

At this point, we have most of the requirements to simulate a multi-qubit circuit. We know from Section 1.2.1 how to apply a single encoded unitary on multiple qubits via scattering on a finite graph, and we know from Section 1.1.4 how to apply multiple single-qubit gates. We need only to combine these two results.

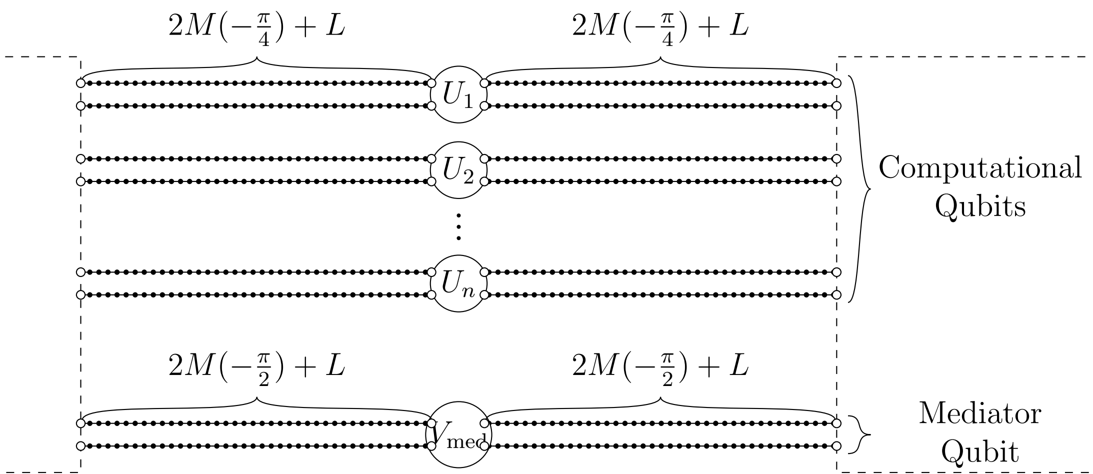

Figure 1.6: The intuitive idea for a single-particle block.

We will make use of the same block structure as in Section 1.1.4, where the graph corresponding to a single unitary is shown in Figure 1.6. Additionally, we will assume that the circuit we want to simulate only consists of a single-qubit gates followed by a two-qubit gate. This assumption isn't difficult to enforce, as these gates can simply consist of identity operations. The circuit that we want to simulate is then given by

$$U_{\mathcal{C}} = V_g U_g V_{g-1} U_{g-1} \cdots V_1 U_1, \tag{1.37}$$

where each V_j is a two-qubit gate, and each U_1 is a one-qubit gate.

As in Section 1.1.4, we will construct a graph for this circuit, $G_{\mathcal{C}}$, by examining the graphs $G^n_{V_j U_j}$ for each $j \in [g]$, and then combining them by associating the output paths of $G^n_{V_j U_j j}(K)$ with the input paths of $G^n_{V_{j+1} U_{j+1}}(K)$. Explicitly, the vertices along the output paths of $G^n_{V_j U_j}(K)$ labeled as $(x, 2^n + 1 + z, j)$ for $\mathbf{z} \in \mathbb{F}_2^n$ are defined to be the same vertices on the input paths of $G^n_{V_{j+1} U_{j+1}}(K)$ labeled as $(K + 1 - x, z + 1, j + 1)$. This can be seen pictorially in Figure 1.6 for one of these blocks.

[**TO DO:** *fix figure Figure 1.6*]

With this construction, we will use exactly the same idea as in the Section 1.1.4 to analyze the time-evolution of a particular initial logical state, with the analysis proceeding accordingly. Using Lemma **??** to focus on the vertices close to a the wavepacket (and in particular the nearest copy of $G^n_{V_j U_j}(K)$) we can the use Corollary 1 to approximate the evolution.

Concretely, let us choose some cutoff length L, set $\sigma = \frac{L}{2\sqrt{\log L}}$, chose $\mu = 2L$ and $K = 2\mu - 1$ and $T = \frac{\mu}{\sin |k|}$. With these choices, our initial logical state will be nearly identical to (1.6), but where the labels for the basis states also have a label corresponding to fact that there are multiple long paths:

$$|\mathbf{z}\rangle_{\text{log,in}} = \gamma \sum_{x=\mu-L}^{\mu+L} e^{ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+1, 1\rangle. \tag{1.38}$$

In a similar manner, the final state of the qubit will be defined as

$$|\mathbf{z}\rangle_{\text{log,out}} = \gamma e^{-2iTg\cos k - 2i(g-1)\mu} \sum_{x=\mu-L}^{\mu+L} e^{-ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+2^n+1, g\rangle, \tag{1.39}$$

Additionally, we will need to define logical states at several times throughout the computation, corresponding to the states after each applied unitary. As such, we will define the logical state after the jth scattering event (and before the $(j+1)$th scattering event) as

$$|\mathbf{z}\rangle_{\text{log},j} = \gamma e^{-2iTj\cos k - 2ik\mu(j-1)} \sum_{x=\mu-L}^{\mu+L} e^{-ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+2^n+1, j\rangle \tag{1.40}$$

$$= \gamma e^{-2iTj\cos k - 2ijk\mu} \sum_{x=\mu-L}^{\mu+L} e^{ikx} e^{-\frac{(x-\mu)^2}{2\sigma^2}} |x, z+1, j+1\rangle, \tag{1.41}$$

where as in the single-qubit case, the additional phase arises from the change of labeling of the vertices.

From here, we can now bound the error in approximating the time evolution of $|\mathbf{z}\rangle_{\text{log,in}}$ for a time Tg by the output logical state corresponding to an application of the circuit, by breaking the evolution into blocks of length T. Explicitly:

$$\left\| e^{iA(G_\mathcal{C})gT} |\psi\rangle_{\text{log,in}} - |U_\mathcal{C}\psi\rangle_{\text{log,out}} \right\|$$

$$\leq \sum_{j=0}^{g-1} \left\| e^{iA(G_\mathcal{C})T} |U_j U_{j-1} \cdots U_1 \psi\rangle_{\text{log},j} - |U_{j+1} U_j \cdots U_1 \psi\rangle_{\text{log},j+1} \right\| \tag{1.42}$$

Note that each individual term is close to that in Corollary 1, but where the Hamiltonian is given by $A(G_\mathcal{C})$ as opposed to $A(G_{V_jU_j}(K))$. However, using Lemma **??** with $H = A(G_\mathcal{C})$, $\tilde{H} = G_{V_jU_j}(K-1)$, $N_0 = \frac{\mu}{4}$, and the error δ from Corollary 1, we have that for all logical states $|\phi\rangle$,

$$\left\| e^{iA(G_\mathcal{C})T} |\phi\rangle_{\text{log},j} - |U_{j+1}\phi\rangle_{\text{log},j+1} \right\|$$

$$\leq \left(\frac{16e\|A(G_\mathcal{C})\|T}{\mu} + 2 \right) \left[\xi \sqrt{\frac{\log L}{L}} + 2^{-\mu+L+2} \left(1 - \chi \sqrt{\frac{\log L}{L}} \right) \right] \tag{1.43}$$

$$\leq \kappa \sqrt{\frac{\log L}{L}}, \tag{1.44}$$

where κ is depends on k and the maximum degree of $G_\mathcal{C}$ (which we assume to be constant). We can then see that

$$\left\| e^{iA(G_\mathcal{C})gT} |\psi\rangle_{\text{log,in}} - |U_\mathcal{C}\psi\rangle_{\text{log,out}} \right\|$$

$$\leq \sum_{j=0}^{g-1} \left\| e^{iA(G_\mathcal{C})T} |U_j U_{j-1} \cdots U_1 \psi\rangle_{\text{log},j} - |U_{j+1} U_j \cdots U_1 \psi\rangle_{\text{log},j+1} \right\| \tag{1.45}$$

$$\leq g\kappa \sqrt{\frac{\log L}{L}}, \tag{1.46}$$

exactly as in Section 1.1.4 and our error bound does not explicitly depend on n. As such, if we take L larger than $g^2\kappa^2\log^2(g\kappa)$ we find that the error can be made arbitrarily small, and thus we can simulate a particular n-qubit g-gate circuit with constant error via graph scattering on a graph of $\mathcal{O}(2^n g^3 \log^2 g)$ vertices for a time $\mathcal{O}(g^3\log^2 g)$.

1.3 Universality via single-particle scattering

[**TO DO:** *I'm not happy with the wording of much of this section*]

Thus far, we have shown how to simulate the application of a quantum circuit on a computational basis state, when a circuit is of a particular form. This nearly shows that quantum walk is universal for quantum computation, but to make things precise we need to explicitly define a decision problem.

[**TO DO:** *Change definition of problem?*] As such, let us define the problem QUANTUM WALK EVOLUTION:

Problem 1 (QUANTUM WALK EVOLUTION)**.** Given a d-sparse, row-computable simple graph G on $\Theta(2^n)$ vertices, an efficiently preparable initial state $|\phi^G_{\text{init}}\rangle$, a subset of the vertices $\mathcal{V}_{\text{accept}}$ (for which membership is easily verified), and a time $T \in \text{poly}(n)$, determine whether

- $\langle\phi^G_{\text{init}}|e^{iA(G)T}\Pi_{\mathcal{V}}e^{-iA(G)T}|\phi^G_{\text{init}}\rangle \geq \frac{2}{3}$ (a yes instance), or
- $\langle\phi^G_{\text{init}}|e^{iA(G)T}\Pi_{\mathcal{V}}e^{-iA(G)T}|\phi^G_{\text{init}}\rangle \leq \frac{1}{3}$ (a no instance),

where we are guaranteed that one case holds.

Note that the efficient preparation of $|\phi^G_{\text{init}}\rangle$, and the fact that membership $\mathcal{V}$ is easily verified, are used to insure that the bulk of the computational power arises from the time evolution on G, and that we are not hiding the computation in the preparation of the initial state or in our measurements.

1.3.1 Simulation of a quantum circuit

Let us first show that an arbitrary quantum circuit can be simulated by the time evolution of an efficiently specifiable state on a succinctly represented graph. This is the point at which we will use our graph scattering techniques.

In particular, choose some $k \in (-\pi, 0)$ for which a universal set of quantum scattering gates are known (such as those in Section 1.1.5, and let $\mathcal{C}_x$ be a circuit corresponding to the QUANTUM CIRCUIT VERIFICATION instance x, where $\mathcal{C}_x$ can be decomposed into a sequence of one- and two-qubit gates from the universal gate set $\mathcal{S}_k$ and controlled-not gates. We assume that the circuit $\mathcal{C}_x$ acts on m qubits (where $m \in \text{poly}(n)$), and that the circuit satisfies:

- if $x \in$ QUANTUM CIRCUIT VERIFICATION, then $\Pr(\mathcal{C}_x, 0^m) > 1 - 2^{-|x|}$,
- if $x \notin$ QUANTUM CIRCUIT VERIFICATION, then $\Pr(\mathcal{C}_x, 0^m) < 2^{-|x|}$.

In other words, with high probability the circuit $\mathcal{C}_x$ computes whether the instance x is in Quantum Circuit Verification.

Note that we almost have from Section 1.2 that such a simulation is possible, we just need to ensure that the unitary $\mathcal{C}_x$ has the decomposition required, that the associated graph is d-sparse and row-computable, that the initial state is efficiently preparable, and that the subset of vertices corresponding to an accepting computation is easily verifiable.

For the first problem, note that we can easily change the circuit $\mathcal{C}_x$ into a circuit of the form described in Section 1.2 by doubling the number of unitaries in the circuit. In particular, if $\mathcal{C}_x = U_g U_{g-1} \cdots U_1$, the quantum walk will simulate the circuit $\mathcal{D}_x = V_g V_{g-1} \cdots V_1$, where

$$V_j = \begin{cases} U_j \mathbb{I}_2^{(1)} & U_j \text{ is a 2-qubit gate} \\ \mathbb{I}_4^{(1,2)} U_j & U_j \text{ is a 1-qubit gate.} \end{cases} \tag{1.47}$$

In this manner, $\mathcal{D}_x$ is an alternating sequence of 1- and 2-qubit gates. For a given L, we can thus use the construction of Section 1.2 for the graph $\mathcal{D}_x$.

To see that the Hamiltonian corresponding to $G_{\mathcal{D}_x}(K)$ is d-sparse, note that the Hamiltonian simply corresponds to the adjacency matrix of the graph. As such, the sparsity is exactly equal to the maximum degree of the graph, which is bounded by some constant $d_{\max}$, which depends on the scattering gadgets for the momentum k.

To see that the graph is row-computable, note that our construction already gives us the ability to efficiently determine the neighbors of a vertex. Given a vertex's label, we can easily determine its location in the graph, and in particular determine whether it is located along a path or within a copy of $\widehat{G}_j$ for some $j \in [g]$. If the vertex is on a path, then its only neighbors are the two vertices immediately before and immediately after it on the path (i.e, $(x, \mathbf{z}, j)$ is connected to $(x \pm 1, \mathbf{z}, j)$). If a vertex is an internal vertex within a copy of $\widehat{G}_j$ we ca easily determine its neighbors in $\widehat{G}_j$, and thus determine its neighbors in the total graph. The only small difficulty arises if a vertex is a terminal vertex of some copy of $\widehat{G}_j$, but in this case we can efficiently determine the path to which it is connected, along with the vertices within the copy of $\widehat{G}_j$. Hence, $G_{\mathcal{D}_x}(K)$ is row-computable.

As for the initial state, note that the construction of Section 1.2 only has non-zero amplitude on $2L+1$ vertices, which only depends on $\mathcal{C}$ in terms of the length. This is both efficiently specifiable and constructable.

Finally, we have that the states corresponding to an accepting computation are easy to check; they are those vertices on the final $(g+1)$ long paths with the first qubit in logical state 1.

Putting this together, we can then use the results of Section 1.2 to show that any Quantum Circuit Verification instance can be reduced to an instance of Quantum Walk Evolution. Hence, Quantum Walk Evolution is **BQP**-hard.

1.3.2 Simulation of a quantum walk

Let us now show that Quantum Walk Evolution is in **BQP**. It relatively easily follows from Theorem **??**, since we simply need to evolve a given Hamiltonian for the time T, and then measure in the computational basis.

In particular, the fact that the initial state is efficiently preparable means that there exists a quantum circuit $\mathcal{C}_{\text{prep}}$ that prepares the given initial state from the state $|0^n\rangle$, possibly including some ancilla states. Moreover, the quantum circuit only consists of poly($|x|$) one- and two-qubit gates, since the state is efficiently preparable. (Note that this bound depends on the efficiency of the given instance.)

From the initial state, we can then use Theorem **??** to evaluate the quantum walk Hamiltonian for the time T. Since the Hamiltonian is d-sparse, and since the Hamiltonian is an adjacency matrix, we have that its maximal eigenvalue is at most d. Assuming that we want to perform the simulation with error $1/10$, we can then simulate the evolution of the quantum walk in an efficient manner. (Again, the efficiency depends on the given instance.)

Finally, the quantum circuit can measure the state of the system in the vertex basis. If the measured vertex belongs to $\mathcal{K}$ (a post-processing procedure that can be done efficiently), then the circuit accepts. Otherwise, the circuit rejects.

As the quantum walker fails with probability at most $\frac{1}{3}$ (i.e., accepts the state for a yes instance with probability at least $\frac{2}{3}$, and accepts a no-instance with probability at most $\frac{1}{3}$), with the error arising from the simulation we still have that the yes and no instances of our simulations have a constant acceptance gap. Hence, we have that Quantum Walk Evolution is in **BQP**.

Put together with our results using scattering, we then have that Quantum Walk Evolution is **BQP**-complete, or that quantum walks are universal for quantum computation.

1.4 Discussion and extensions

Note that these results are essentially already known. In particular, Childs gave a similar result using scattering theory, but where his time evolution arose from the theory of stationary phases.

This discussion has essentially been plug-and-play for the momentum k, assuming that a set of universal scattering gadgets are known for the momenta. However, we only have these gadgets for a small set of momenta. It might be interesting to find these universal gate sets for a larger group, or show that they do not exist for some particular momentum.

As a possible extension of this work, the permutation of the underlying paths corresponding to a two-qubit unitary plays a key part in our encoded gates. However, these permutations are intrinsically non-planar (assuming that there are more than three non-identity 2-qubit gates). One might ask whether any planar graph could be used to encode a computation, or whether non-planarity is required to capture the entire power of quantum walk. In a similar manner, one might ask what other properties of the underlying graph are needed to maintain the same computational power.

Additionally, our current construction assumes no errors, and error-correction cannot trivially be implemented in our scheme. We are unsure how error correction could be implemented using this construction, since both non-local measurements and measurements in the middle of the computation seem rather difficult. Including error correction in our graph scattering (or quantum walk in general) would be of great help in the applications toward universality. This might also help us in the next chapter, as we use similar methods to show the universality of multi-qubit quantum walk.

References

[1] Andris Ambainis, *Quantum walk algorithm for element distinctness*, SIAM Journal on Computing **37** (2007), no. 1, 210–239, quant-ph/0311001, Preliminary version in FOCS 2004.

[2] Benjamin A. Blumer, Michael S. Underwood, and David L. Feder, *Single-qubit unitary gates by graph scattering*, Physical Review A **84** (2011), no. 6, 062302, arXiv:1111.5032.

[3] Andrew M. Childs, *Universal computation by quantum walk*, Physical Review Letters **102** (2009), no. 18, 180501, arXiv:0806.1972.

[4] Andrew M. Childs, *On the relationship between continuous- and discrete-time quantum walk*, Communications in Mathematical Physics **294** (2010), 581–603, arXiv:0810.0312.

[5] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman, *Exponential algorithmic speedup by quantum walk*, Proceedings of the 35th ACM Symposium on Theory of Computing, pp. 59–68, 2003, quant-ph/0209131.

[6] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann, *A quantum algorithm for the Hamiltonian NAND tree*, Theory of Computing **4** (2008), no. 1, 169–190, quant-ph/0702144.