A Note on Scalable Pseudorandom Unitaries

Giulio Malavolta*

Abstract

Pseudorandom unitaries (PRUs) are efficiently computable keyed families of unitaries that are computationally indistinguishable from Haar-random unitaries; scalable PRUs allow the security parameter to be chosen independently of the number of qubits. Assuming quantum-secure pseudorandom functions and Vaikuntanathan and Zamir's conjecture on sequential Kac walks, we propose a construction of scalable PRUs. At a technical level, we show how to implement $n \cdot \mathrm{polylog}(n)$ steps of a sequential Kac's (pseudo)random walk by a quantum circuit of size $\mathrm{polylog}(n)$.

1 Introduction

Pseudorandom unitaries (PRUs), introduced by Ji, Liu, and Song [1], are efficiently-computable keyed families of unitaries that are computationally indistinguishable from Haar-random unitaries under quantum oracle access. Random unitaries have applications throughout quantum information, including quantum algorithms for state estimation [2], quantum advantage experiments [3], quantum learning [4], and cryptographic protocols [5]. While Haar-random unitaries generally require circuits of exponential size, PRUs provide efficient substitutes that preserve the behavior observable by polynomial-time quantum algorithms. They also offer efficient models of chaotic quantum systems, including black-hole dynamics [9, 11], and are connected to computational hardness of learning physical systems [10].

Early constructions of PRUs achieved security for restricted classes of inputs [6] and against arbitrary nonadaptive queries [7, 8]. Ma and Huang [9] subsequently constructed PRUs secure against adaptive queries from quantum-secure one-way functions, analyzing a scheme proposed in [7]. Their construction also achieves *strong* pseudorandomness, which allows the adversary to query both the unitary and its inverse. Further work developed shallow PRU constructions [10], including constructions secure under queries to the inverse, transpose, and complex conjugate [11], and constant-depth implementations [12]. Another line of work [13, 14] constructed PRUs in the Haar random oracle model and gave generic transformations reducing their key length, or proving security under inverse queries.

Despite these developments, an open problem was to construct *scalable* PRUs: Families whose security parameter can be chosen independently of the number of qubits, with key generation and evaluation polynomial in both parameters [18, 19]. Scalability had been achieved for pseudorandom states [15], for pseudorandom state scramblers [16], and for pseudorandom function-like state generators [18]. Recently, Brakerski and Yuen [19] studied the scalability question directly. They showed that scalable PRUs in the random oracle model are connected to the unitary synthesis problem [21]

*Bocconi University. giulio.malavolta@unibocconi.it

and proved that existing candidates cannot achieve scalability in that model. Huang, Ramkumar, and Wright [20] subsequently ruled out statistically secure scalable PRUs in the random oracle model, motivating the need for additional computational assumptions.

In this note, we put forward a candidate construction for scalable PRUs. We show that quantum-secure pseudorandom functions and (the post-quantum version of) a conjecture on sequential Kac's walks introduced by Vaikuntanathan and Zamir [22], see also [23], imply the existence of scalable PRUs. Kac's random walks were already shown to be useful in the context of PRUs [16, 17]. The difference is that [16, 17] consider $O(\log n)$ rounds of *parallel* walks, whereas we efficiently implement $n \cdot \mathrm{polylog}(n)$ rounds of a *sequential* walk, allowing us to appeal to the conjecture in [22].

1.1 AI Statement

ChatGPT Astra was used at all stages of this work for generating ideas, suggesting proofs of technical claims, sanity-checking the math, and writing assistance. The final version of this manuscript is written by the author, who takes full responsibility for its content.

2 Preliminaries

An n-dimensional quantum register has state space $\mathbb{C}^n$, with computational basis $\{|x\rangle : 0 \le x < n\}$; composite registers are described by tensor products. A quantum state is a positive semidefinite matrix of trace one, with a unit vector $|\psi\rangle$ representing the pure state $|\psi\rangle\langle\psi|$. A quantum channel is a completely positive, trace-preserving linear map. A unitary $\mathbf{U}$ induces the channel $\rho \mapsto \mathbf{U}\rho\mathbf{U}^\dagger$, and we identify unitaries with their channels when writing diamond norms. For channels Φ, Ψ on $\mathbb{C}^n$,

$$\|\Phi - \Psi\|_\diamond = \max_\rho \big\|((\Phi - \Psi) \otimes \mathrm{id}_n)(\rho)\big\|_1, \qquad \|\mathbf{A}\|_1 = \mathrm{Tr}\sqrt{\mathbf{A}^\dagger\mathbf{A}},$$

where ρ ranges over states on $\mathbb{C}^n \otimes \mathbb{C}^n$. For unitaries,

$$\frac{1}{2}\|\mathbf{U} - \mathbf{V}\|_\diamond \le \|\mathbf{U} - \mathbf{V}\|_{\mathrm{op}}, \qquad \|\mathbf{A}\|_{\mathrm{op}} = \max_{\|v\|_2=1} \|\mathbf{A}v\|_2.$$

Oracle access to a Boolean function f means access to the unitary $|x, b\rangle \mapsto |x, b \oplus f(x)\rangle$, extended linearly to superpositions. The single-qubit Pauli operators satisfy $\mathbf{X}|b\rangle = |b \oplus 1\rangle$ and $\mathbf{Z}|b\rangle = (-1)^b|b\rangle$. Finally, $\mathrm{Haar}(G)$ denotes the probability measure on a compact group G invariant under left and right multiplication.

We recall the definition of a (quantum secure) pseudorandom function.

Definition 2.1 (Pseudorandom Function)**.** *A keyed family $\{f_k : \{0,1\}^{\ell(\lambda)} \to \{0,1\}\}_k$ is a quantum-secure pseudorandom function family if for every polynomial-size quantum distinguisher $\mathcal{A}$,*

$$\left| \Pr_{k \leftarrow \mathrm{Gen}(1^\lambda)} \left(\mathcal{A}^{f_k}(1^\lambda) = 1\right) - \Pr_f \left(\mathcal{A}^{f}(1^\lambda) = 1\right) \right| \le \mathrm{negl}(\lambda).$$

3 Kac's Random Walks

3.1 Background on Kac's Random Walks

We start by recalling Kac's random walks. For $0 \le i < j < n$ and $\theta \in [0, 2\pi)$ let $\mathbf{R}_{i,j,\theta} \in \mathbb{R}^{n\times n}$ be the unique rotation matrix such that

$$\mathbf{R}_{i,j,\theta}\,|i\rangle = \cos\theta\,|i\rangle + \sin\theta\,|j\rangle\,, \qquad \mathbf{R}_{i,j,\theta}\,|j\rangle = -\sin\theta\,|i\rangle + \cos\theta\,|j\rangle\,, \qquad \mathbf{R}_{i,j,\theta}\,|k\rangle = |k\rangle\,, \qquad (k \notin \{i,j\}).$$

Let $\mathcal{M}$ be the uniform measure over $\{\mathbf{R}_{i,j,\theta}\}_{i,j,\theta}$, where $i < j$ and θ are sampled uniformly at random. Let $\mathbf{Q}_0 = \mathbf{I}_{n\times n} \in \mathbb{R}^{n\times n}$, define $\mathbf{Q}_t = \mathbf{Q}_{t-1} \cdot \mathbf{R}_{i,j,\theta}$ where $\mathbf{R}_{i,j,\theta} \sim \mathcal{M}$. A conjecture from Vaikuntanathan and Zamir [22] postulates that a sufficiently large t results in a matrix that is computationally indistinguishable from Haar-random in the special orthogonal group. Here we state the analogous conjecture for quantum distinguishers.

Conjecture 3.1 ([22])**.** *There exists some $t(n) = n \cdot \mathrm{polylog}(n)$ such that for every polynomial-size quantum distinguisher $\mathcal{A}$,*

$$\left| \Pr_{\mathbf{Q}_t \sim \mathrm{Kac}(t(n))} (\mathcal{A}(\mathbf{Q}_t) = 1) - \Pr_{\mathbf{O} \sim \mathrm{Haar}(\mathrm{SO}(n))} (\mathcal{A}(\mathbf{O}) = 1) \right| \le \mathrm{negl}(n).$$

We can easily convert a sampler from the special orthogonal group to a sampler for the full orthogonal group by the following simple fact.

Lemma 3.2. $\mathrm{diag}(b, 1, \ldots, 1) \cdot \mathbf{O} \overset{d}{=} \mathbf{P} : \mathbf{O} \sim \mathrm{Haar}(\mathrm{SO}(n)), \mathbf{P} \sim \mathrm{Haar}(\mathrm{O}(n)), b \sim \mathrm{Unif}\{\pm 1\}$.

Proof. Immediate by the decomposition $\mathrm{O}(n) = \mathrm{SO}(n) \sqcup \mathrm{diag}(-1, 1, \ldots, 1)\mathrm{SO}(n)$. □

3.2 Polynomial-Time Implementation

We show how to implement $t = n \cdot \mathrm{polylog}(n)$ sequential steps of the Kac walk with a quantum circuit of size polylogarithmic in n and access to a random function $f : \{0,1\}^* \to \{0,1\}$. First, we show how to implement a random permutation with access to a random function.

Lemma 3.3. *For every even $m \ge 2$ and every $0 < \varepsilon < 1$ there are uniformly generated quantum circuits of size $\mathrm{poly}(\log m, \log(1/\varepsilon))$ that, with oracle access to a random function f, implement a permutation π_f and its inverse:*

$$|x\rangle|0\rangle \longmapsto |\pi_f(x)\rangle|0\rangle, \qquad |x\rangle|0\rangle \longmapsto |\pi_f^{-1}(x)\rangle|0\rangle.$$

Moreover, the induced distribution over π_f is within ε total-variation distance from $\mathrm{Unif}(S_m)$.

Proof. Use the shuffle of Morris and Rogaway [24, Theorem 4], see also [25], with error $\varepsilon/2$. The resulting permutation is within $\varepsilon/2$ of uniform in total variation, and evaluation in either direction takes $O(\log m \log(m/\varepsilon))$ rounds in the worst case (we use disjoint oracle bits for each round). For each round key, use rejection sampling with $O(\log(m/\varepsilon))$ trials, returning zero if all trials fail. With a sufficiently large implicit constant, the probability of any failure is at most $\varepsilon/2$. Reversible evaluation of the shuffle and its inverse gives the claimed circuits. □

Next, we show how to sample from the binomial distribution, again assuming oracle access to a random function.

Lemma 3.4. *For every $\ell \geq 1$ and every $0 < \varepsilon < 1$, there is a uniformly generated reversible quantum circuit of size* $\mathrm{poly}(\ell, \log(1/\varepsilon))$ *that, with oracle access to a random function f, implements the map*

$$|m\rangle|z\rangle|0\rangle \longmapsto |m\rangle|z \oplus B^f_{\ell,\varepsilon}(m)\rangle|0\rangle \qquad (0 \leq m < 2^\ell).$$

For every fixed m, the induced distribution over $B^f_{\ell,\varepsilon}(m)$ is supported on $\{0, \ldots, m\}$ and is within ε total-variation distance from $\mathrm{Bin}(m, 1/2)$.

Proof. Use exact samplers, e.g., [26, 27, 28], truncated after $\mathrm{polylog}(1/\varepsilon)$ operations, so that the probability of failure is bounded by ε. Return 0 if this time bound is exceeded. With the oracle bits fixed, reversibly simulate the truncated sampler's operations, XOR its output into z, and reverse the computation to clear the workspace. The total variation distance from $\mathrm{Bin}(m, 1/2)$ is bounded by ε, and the truncated computation can be implemented with a quantum circuit of size $\mathrm{poly}(\ell, \log(1/\varepsilon))$. □

Next, we show how to implement efficiently a sequential Kac's random walk. First, we define a data structure that is going to be useful for our implementation. Intuitively, for each basis state x, we define a list of all rotations (in a given batch) that are in the "light cone" of x.

Let (i_h, j_h) be the independent uniform pairs sampled by a sequential Kac's random walk. Partition the rotations into batches of size $n/256$, then for each batch $0 \leq b < 256t/n$ define the following lists. The index h is local to its batch. For any $x \in \{0, \ldots, n-1\}$, add x to $\mathcal{C}_{f,b}(x)$, then define a list $\mathcal{L}_{f,b}(x)$ iteratively as follows:

(i) For each $y \in \mathcal{C}_{f,b}(x)$, look up all the rotations in batch b involving y.

(ii) For each such rotation let h be its index, then:

- Add (h, i_h, j_h) to $\mathcal{L}_{f,b}(x)$, if h is not already present.
- Add i_h and j_h to $\mathcal{C}_{f,b}(x)$, unless they are already present.

(iii) Go back to step (i), unless all coordinates in $\mathcal{C}_{f,b}(x)$ are processed.

(iv) Return the list $\mathcal{L}_{f,b}(x)$ with entries sorted by h.

Lemma 3.5. *For any b, x, y, if $x \in \mathcal{C}_{f,b}(y)$, then $\mathcal{L}_{f,b}(x) = \mathcal{L}_{f,b}(y)$.*

Proof. Steps (i)–(iii) compute the connected component of x in the undirected multigraph with vertex set $\{0, \ldots, n-1\}$ and edges $\{i_h, j_h\}$ for rotations in batch b: Every added coordinate is connected to x, and once the algorithm terminates then every incident edge has been examined. Hence

$$x \in \mathcal{C}_{f,b}(y) \quad \Longrightarrow \quad \mathcal{C}_{f,b}(x) = \mathcal{C}_{f,b}(y).$$

By steps (ii) and (iv),

$$\mathcal{L}_{f,b}(x) = \mathrm{sort}_h\big\{(h, i_h, j_h) : h \text{ lies in batch } b,\ \{i_h, j_h\} \subseteq \mathcal{C}_{f,b}(x)\big\}.$$

Substituting $\mathcal{C}_{f,b}(y)$ gives $\mathcal{L}_{f,b}(x) = \mathcal{L}_{f,b}(y)$. □

Lemma 3.6. *For every $0 < \varepsilon < 1$,*

$$\Pr_f\left(\max_{b,x} |\mathcal{L}_{f,b}(x)| > \log^2\left(\frac{256nt}{\varepsilon}\right)\right) \leq \varepsilon.$$

Proof. Assume $n \geq 256$ as otherwise every list has size at most 1. For each batch b, consider the graph whose vertices are the rotation indices in that batch. Two distinct vertices h, h' are adjacent exactly when $\{i_h, j_h\} \cap \{i_{h'}, j_{h'}\} \neq \varnothing$. By the definition of $\mathcal{L}_{f,b}(x)$, its rotation indices, when nonempty, form a connected component of this graph. Thus, if $|\mathcal{L}_{f,b}(x)| \geq k$, the graph contains a tree on k distinct rotation indices.

Fix an integer $k \geq 2$, a batch b, and a tree T on k distinct indices of that batch. Root T at its smallest index. For each non-root vertex h, let h' be its parent. For every fixed pair at h',

$$\left|\left\{(i_h, j_h) : 0 \leq i_h < j_h < n,\ \{i_h, j_h\} \cap \{i_{h'}, j_{h'}\} \neq \varnothing\right\}\right| = \binom{n}{2} - \binom{n-2}{2} = 2n - 3.$$

The root pair has $\binom{n}{2}$ choices, therefore exactly $\binom{n}{2}(2n-3)^{k-1}$ assignments cover all edges of T, denoted by $E(T)$. Thus,

$$\Pr_f\left(\{i_h, j_h\} \cap \{i_{h'}, j_{h'}\} \neq \varnothing \text{ for every } \{h, h'\} \in E(T)\right) = \frac{\binom{n}{2}(2n-3)^{k-1}}{\binom{n}{2}^k} \leq \left(\frac{4}{n}\right)^{k-1}.$$

There are at most t nonempty batches, at most $\binom{n/256}{k}$ choices of indices per batch, and k^{k-2} trees on each choice. Hence

$$\Pr_f\left(\max_{b,x} |\mathcal{L}_{f,b}(x)| \geq k\right) \leq t\binom{n/256}{k} k^{k-2} \left(\frac{4}{n}\right)^{k-1} \leq \frac{nt}{4k^2}\left(\frac{e}{64}\right)^k \leq nte^{-k}.$$

Taking $k = \log^2\left(\frac{256nt}{\varepsilon}\right)$, we get $nte^{-k} \leq \frac{\varepsilon}{256}$. □

Lemma 3.7. *For every power-of-two $n \geq 2$, every $0 < \varepsilon < 1$, and every b there is a uniformly generated quantum circuit of size* $\mathrm{poly}(\log n, \log t, \log(1/\varepsilon))$ *that, with oracle access to a random function f, implements*

$$|x\rangle|z\rangle|0\rangle \longmapsto |x\rangle|z \oplus \mathcal{L}_{f,b}(x)\rangle|0\rangle.$$

With probability at least $1 - \varepsilon$ over f, the map holds simultaneously for every b, x.

Proof. We first describe a sampler for each batch of rotation indices, using coins from a random function f. Let $C(\varnothing) = n/128$, for $|v| < \log_2 n$, define

$$C(v0) = B^f_{\log_2(n+1),\, \varepsilon^2/(256n^2t^2)}(C(v)), \qquad C(v1) = C(v) - C(v0)$$

using the sampler from Lemma 3.4, whose output lies in $\{0, \ldots, C(v)\}$. In batch b and repetition r, the binomial sampler at node v uses the oracle $z \mapsto f(\mathrm{bin}, b, r, v, z)$, and the permutation sampler uses $z \mapsto f(\mathrm{perm}, b, r, z)$ (we henceforth suppress this dependence). This procedure defines a data structure that defines the count of sampled indices with prefix v. We define the sorted sequence by

$$a_j = x \quad \Longleftrightarrow \quad \sum_{y<x} C(y) \leq j < \sum_{y \leq x} C(y).$$

Sample a random permutation π_f of $\{0, \ldots, n/128 - 1\}$ via Lemma 3.3, then for each index h define the h-rotation to act on the pair $(a_{\pi_f(2h)}, a_{\pi_f(2h+1)})$. Repeat this entire procedure $8\log(4t/\varepsilon)$ times with independent coins derived from f. For each rotation index h, use its pair from the first repetition in which $a_{\pi_f(2h)} \neq a_{\pi_f(2h+1)}$. If no repetition satisfies these conditions, assign the identity to that step. Sort each selected pair so that $i_h < j_h$.

Note that this sampling procedure allows us to efficiently compute, on input a given x, the indices of the rotations involving x. In each repetition, enumerate the candidate indices

$$\sum_{y<x} C(y) \le j < \sum_{y\le x} C(y), \qquad h = \left\lfloor \frac{\pi_f^{-1}(j)}{2} \right\rfloor,$$

where the two sums can be computed with $O(\log n)$ recursive evaluations of C. For each candidate h, recompute its first successful repetition and retain h only if the pair selected there contains x, discarding duplicate indices and identity steps. Compute $\mathcal{L}_{f,b}(x)$ by the iterative procedure above. Before enumerating candidates at a visited coordinate y, return the empty list if $C(y) > \log^2(1024nt/\varepsilon)$ in any repetition. Also return the empty list as soon as the number of distinct rotation indices collected exceeds $\log^2(1024nt/\varepsilon)$. Thus, for every fixed f, all starting coordinates in the same component return the same list: The full sorted list if both bounds hold throughout the component, and the empty list otherwise.

Overall, this allows us to compute, with a circuit of size polynomial in $\log n, \log t, \log(1/\varepsilon)$, the map

$$\begin{aligned} |x\rangle|z\rangle|0\rangle &\longmapsto |x\rangle|z\rangle|\mathcal{L}_{f,b}(x)\rangle \\ &\longmapsto |x\rangle|z \oplus \mathcal{L}_{f,b}(x)\rangle|\mathcal{L}_{f,b}(x)\rangle \\ &\longmapsto |x\rangle|z \oplus \mathcal{L}_{f,b}(x)\rangle|0\rangle. \end{aligned}$$

We next bound the running time of the stated procedure. For each fixed batch, repetition, and coordinate x, exact sampling gives

$$C(x) \sim \mathrm{Bin}(C(\varnothing), 1/n), \qquad \mathbb{E}[e^{C(x)}] = \left(1 + \frac{e-1}{n}\right)^{C(\varnothing)} < 6.$$

Therefore the probability that any such count exceeds $\log^2(1024nt/\varepsilon)$ is at most

$$6nt8\log\frac{4t}{\varepsilon}\exp\left(-\log^2\frac{1024nt}{\varepsilon}\right) \le \frac{\varepsilon}{4}.$$

Outside this event, every enumeration has polynomially many terms in $\log n, \log t, \log(1/\varepsilon)$. Together with the stated bound on the collected list, this bounds the total running time by

$$O\big(\log^2(4t/\varepsilon)\ \log^4(1024nt/\varepsilon)\big)\ \mathrm{poly}(\log n, \log t, \log(1/\varepsilon)).$$

It remains to bound the probability of an incorrect list. Assuming exact sampling, the two coordinates at each step are independent and uniform in $\{0, \dots, n-1\}$. They are distinct with probability $n(n-1)/n^2 \ge 1/8$. Hence the probability that some rotation index has no repetition with distinct coordinates is at most

$$t\exp\left(-\frac{1}{8}8\log\frac{4t}{\varepsilon}\right) \le \frac{\varepsilon}{4}.$$

Moreover, Lemma 3.6 with error $\varepsilon/4$ bounds the probability that any list exceeds $\log^2(1024nt/\varepsilon)$ by $\varepsilon/4$.

Finally, apply Lemma 3.3 with total-variation error $\varepsilon^2/(256n^2t^2)$, the same error used for each invocation of Lemma 3.4. Couple the implemented samplers to their exact counterparts. A union bound over at most $2nt \cdot 8\log(4t/\varepsilon)$ invocations bounds the probability of any disagreement by $\varepsilon/4$. Overall, the probability of an incorrect list is at most ε by a union bound. □

Lemma 3.8. *For every power-of-two $n \geq 2$, every integer $t \geq 1$, and every $0 < \varepsilon < 1$, there is a uniformly generated quantum oracle circuit $K_{n,t,\varepsilon}$ of size $(1 + t/n)\,\mathrm{poly}(\log n, \log(t+1), \log(1/\varepsilon))$ such that there exists a coupling of f with an exact t-step Kac walk $\mathbf{Q}_t$ such that*

$$\mathbb{E}\frac{1}{2}\left\|K^f_{n,t,\varepsilon} - \mathbf{Q}_t\right\|_\diamond \leq \varepsilon.$$

Proof. Apply Lemma 3.7 with error $\varepsilon/3$ and set $r = \log_2 \frac{6\pi t}{\varepsilon}$. For each batch b, define $\theta_h|_r$ as 2π times the binary fraction formed by $f(\text{angle}, b, h, 1), \ldots, f(\text{angle}, b, h, r)$. The angle registers below store the corresponding r-bit fractions. On the success event of Lemma 3.7, the circuit performs

$$\begin{aligned}
|x\rangle\,|0\rangle\,|0\rangle &\longmapsto |x\rangle\,|\mathcal{L}_{f,b}(x)\rangle\,|0\rangle \\
&\longmapsto |x\rangle\,|\mathcal{L}_{f,b}(x)\rangle\,\Big|(\theta_h|_r)_{(h,i_h,j_h)\in\mathcal{L}_{f,b}(x)}\Big\rangle \\
&\longmapsto \left(\prod_{(h,i_h,j_h)\in\mathcal{L}_{f,b}(x)} \mathbf{R}_{i_h,j_h,\theta_h|_r}\,|x\rangle\right)|\mathcal{L}_{f,b}(x)\rangle\,\Big|(\theta_h|_r)_{(h,i_h,j_h)\in\mathcal{L}_{f,b}(x)}\Big\rangle \\
&\longmapsto \left(\prod_{(h,i_h,j_h)\in\mathcal{L}_{f,b}(x)} \mathbf{R}_{i_h,j_h,\theta_h|_r}\,|x\rangle\right)|0\rangle\,|0\rangle = \left(\prod_{h \text{ in batch } b} \mathbf{R}_{i_h,j_h,\theta_h|_r}\right)|x\rangle\,|0\rangle\,|0\rangle
\end{aligned}$$

The factors in each product are ordered by increasing h from left to right, so the rotations are applied in decreasing order. The last step first uncomputes the angle registers using the unchanged list register, and then uncomputes the list. Since the above equation holds for all basis states $|x\rangle$, it extends linearly to all quantum states.

Each rotation in $\mathcal{L}_{f,b}(x)$ acts on two coordinates in $\mathcal{C}_{f,b}(x)$ and therefore preserves $\mathrm{span}\{|y\rangle : y \in \mathcal{C}_{f,b}(x)\}$. Applying these rotations to $|x\rangle$ gives a state in this subspace. For every $y \in \mathcal{C}_{f,b}(x)$, Lemma 3.5 gives $\mathcal{L}_{f,b}(y) = \mathcal{L}_{f,b}(x)$, which justifies uncomputing the list. Implement each rotation with operator-norm error at most $\varepsilon/(3t)$. Since half-diamond distance between channels is at most one,

$$\mathbb{E}\frac{1}{2}\left\|K^f_{n,t,\varepsilon} - \mathbf{Q}_t\right\|_\diamond \leq \frac{\varepsilon}{3} + 2\pi t\,2^{-r} + t\frac{\varepsilon}{3t} \leq \varepsilon.$$

There are $O(1 + t/n)$ batches, each requiring $\mathrm{poly}(\log n, \log(t+1), \log(1/\varepsilon))$ gates. □

4 Simulating the Spectrum of Haar-Random Unitaries

4.1 Spectral Simulation

Let $n \geq 2$ be even, with coordinates indexed by $0, \ldots, n-1$. Sample mutually independently $\theta_0, \ldots, \theta_{n-1}$ uniformly from $[0, 2\pi)$ and $\gamma_0, \ldots, \gamma_{n-2}$ according to

$$\Pr(\gamma_j \leq t) = (\sin t)^{n-j-1}, \qquad 0 \leq t \leq \frac{\pi}{2}.$$

For each $j = 0, \ldots, n-2$, consider the block

$$\mathbf{T}_j = \begin{bmatrix} \cos\gamma_j\, e^{-i\theta_j} & \sin\gamma_j \\ \sin\gamma_j & -\cos\gamma_j\, e^{i\theta_j} \end{bmatrix}$$

on coordinates $j, j+1$. Define

$$\mathbf{T} = \begin{bmatrix} 1 & & & & & & \\ & \mathbf{T}_1 & & & & & \\ & & \mathbf{T}_3 & & & & \\ & & & \ddots & & & \\ & & & & \mathbf{T}_{n-3} & & \\ & & & & & e^{-i\theta_{n-1}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{T}_0 & & & & & & \\ & \mathbf{T}_2 & & & & & \\ & & \mathbf{T}_4 & & & & \\ & & & \ddots & & & \\ & & & & \mathbf{T}_{n-4} & & \\ & & & & & \mathbf{T}_{n-2} \end{bmatrix}$$

Let Spec denote the spectrum of an operator, counted with multiplicities. The following is a consequence of Theorem 1 of [29].

Theorem 4.1 ([29])**.** $\mathrm{Spec}(\mathbf{T}) \stackrel{d}{=} \mathrm{Spec}(\mathbf{U}^\intercal\mathbf{U}) : \mathbf{U} \sim \mathrm{Haar}(\mathrm{U}(n))$.

We want to define a unitary operator $\mathbf{V}$ such that $\mathbf{V}^\intercal\mathbf{V}$ has the same spectrum of $\mathbf{T}$. We do this by defining block operators

$$\mathbf{V}_j = \frac{1+i}{2} \begin{bmatrix} (1 - i\cos\gamma_j)e^{-i\theta_j/2} & -i\sin\gamma_j\, e^{i\theta_j/2} \\ -i\sin\gamma_j\, e^{-i\theta_j/2} & (1 + i\cos\gamma_j)e^{i\theta_j/2} \end{bmatrix}.$$

Note that each column of $\mathbf{V}_j$ has square norm 1 and the two columns are orthogonal, so $\mathbf{V}_j$ (and consequently $\mathbf{V}_j^\intercal$) is unitary. We define the operator

$$\mathbf{V} = \begin{bmatrix} 1 & & & & & & \\ & \mathbf{V}_1 & & & & & \\ & & \mathbf{V}_3 & & & & \\ & & & \ddots & & & \\ & & & & \mathbf{V}_{n-3} & & \\ & & & & & e^{-i\theta_{n-1}/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_0^\intercal & & & & & & \\ & \mathbf{V}_2^\intercal & & & & & \\ & & \mathbf{V}_4^\intercal & & & & \\ & & & \ddots & & & \\ & & & & \mathbf{V}_{n-4}^\intercal & & \\ & & & & & \mathbf{V}_{n-2}^\intercal \end{bmatrix}$$

Since each block is unitary, it holds that $\mathbf{V} \in \mathrm{U}(n)$. Next, we show that the distribution of $\mathrm{Spec}(\mathbf{V}^\intercal\mathbf{V})$ is identical to the one induced by a Haar-random unitary.

Lemma 4.2. $\mathrm{Spec}(\mathbf{V}^\intercal\mathbf{V}) \stackrel{d}{=} \mathrm{Spec}(\mathbf{U}^\intercal\mathbf{U}) : \mathbf{U} \sim \mathrm{Haar}(\mathrm{U}(n))$.

Proof. By definition of $\mathbf{V}_j$,

$$\mathbf{V}_j^\intercal\mathbf{V}_j = \begin{bmatrix} \cos\gamma_j\, e^{-i\theta_j} & \sin\gamma_j \\ \sin\gamma_j & -\cos\gamma_j\, e^{i\theta_j} \end{bmatrix} = \mathbf{T}_j.$$

From the definition of $\mathbf{V}$, we obtain the explicit change-of-basis identity

$$\mathrm{diag}(\mathbf{V}_0, \mathbf{V}_2, \ldots, \mathbf{V}_{n-2})^{-1}\, \mathbf{V}^\intercal\mathbf{V}\, \mathrm{diag}(\mathbf{V}_0, \mathbf{V}_2, \ldots, \mathbf{V}_{n-2}) = \mathbf{T}.$$

The change-of-basis matrix is unitary, hence invertible. Thus $\mathbf{V}^\intercal\mathbf{V}$ and $\mathbf{T}$ are isomorphic as linear operators and therefore have the same spectrum. The desired identity follows from Theorem 4.1. □

Finally, we show that multiplying this matrix on the left and right by two Haar-random orthogonal matrices results in a Haar-random unitary.

Lemma 4.3. $\mathbf{O}_0\mathbf{V}\mathbf{O}_1 \stackrel{d}{=} \mathbf{U} : \mathbf{O}_0, \mathbf{O}_1 \sim \mathrm{Haar}(\mathrm{O}(n)), \mathbf{U} \sim \mathrm{Haar}(\mathrm{U}(n))$.

Proof. Every $\mathbf{W} \in \mathrm{U}(n)$ admits a factorization

$$\mathbf{W} = \mathbf{P}\mathbf{D}\mathbf{Q}^{\intercal}, \qquad \mathbf{P}, \mathbf{Q} \in \mathrm{O}(n),$$

where $\mathbf{D}$ is diagonal and unitary. Indeed, the real and imaginary parts of $\mathbf{W}^{\intercal}\mathbf{W}$ are commuting real symmetric matrices, so some $\mathbf{Q} \in \mathrm{O}(n)$ satisfies $\mathbf{Q}^{\intercal}\mathbf{W}^{\intercal}\mathbf{W}\mathbf{Q} = \mathbf{D}^2$. Then $\mathbf{P} = \mathbf{W}\mathbf{Q}\mathbf{D}^{-1}$ is both unitary and complex orthogonal, hence real orthogonal.

For independent Haar orthogonals $\mathbf{O}_0, \mathbf{O}_1$, Haar invariance gives, for each fixed $\mathbf{W}$,

$$\mathbf{O}_0\mathbf{W}\mathbf{O}_1 \overset{d}{=} \mathbf{O}_0\mathbf{D}\mathbf{O}_1.$$

The spectrum of $\mathbf{W}^{\intercal}\mathbf{W}$ determines $\mathbf{D}$ up to permutations and signs of its diagonal entries. These changes are absorbed by the Haar orthogonal factors. Thus the distribution above depends only on $\mathrm{Spec}(\mathbf{W}^{\intercal}\mathbf{W})$. By Lemma 4.2 and Haar invariance,

$$\mathbf{O}_0\mathbf{V}\mathbf{O}_1 \overset{d}{=} \mathbf{O}_0\mathbf{U}\mathbf{O}_1 \overset{d}{=} \mathbf{U}. \qquad \square$$

4.2 Polynomial-Time Implementation

We show how to implement $\mathbf{V}$ in polynomial time, given oracle access to a random function f. More precisely, let $f : \{0,1\} \times \{0,\dots,n-1\} \times \mathbb{N}_{>0} \longrightarrow \{0,1\}$ have independent uniform output bits, and set

$$\theta_j = 2\pi \sum_{r \geq 1} 2^{-r} f(0, j, r) \qquad (0 \leq j \leq n-1),$$

$$\gamma_j = \arcsin\left(\left(\sum_{r \geq 1} 2^{-r} f(1, j, r) \right)^{1/(n-j-1)} \right) \qquad (0 \leq j \leq n-2).$$

Then the resulting $\mathbf{V}(f)$ has precisely the same distribution as $\mathbf{V}$.

Lemma 4.4. *For every even $n \geq 2$ and every $0 < \varepsilon < 1$, there is a uniformly generated quantum circuit $C_{n,\varepsilon}$ of size $\mathrm{poly}(\log n, \log(1/\varepsilon))$, with coherent access to f and querying at bit positions bounded by $r = O(\log n + \log(1/\varepsilon))$, such that*

$$\mathbb{E}_f \frac{1}{2} \left\| C^f_{n,\varepsilon} - \mathbf{V}(f) \right\|_{\diamond} \leq \varepsilon.$$

Proof. Set $r = \log_2(100n/\varepsilon)$. Write $\theta_j|_r$, $\gamma_j|_r$, and $\mathbf{V}_j|_r$ for the expressions obtained by truncating their defining oracle-bit sums after r terms. For even j and $i \in \{0,1\}$, perform

$$\begin{aligned}
|j+i\rangle|0\rangle|0\rangle &\longmapsto |j\rangle|i\rangle|0\rangle \\
&\longmapsto |j\rangle|i\rangle\,|(f(0,j,k), f(1,j,k))_{k=1}^{r}\rangle \\
&\longmapsto |j\rangle\big((\mathbf{V}_j|_r)^{\intercal}|i\rangle\big)\,|(f(0,j,k), f(1,j,k))_{k=1}^{r}\rangle \\
&\longmapsto |j\rangle\big((\mathbf{V}_j|_r)^{\intercal}|i\rangle\big)|0\rangle \\
&\longmapsto \sum_{k=0}^{1} (\mathbf{V}_j|_r)_{ik}|j+k\rangle|0\rangle|0\rangle.
\end{aligned}$$

The second and fourth maps each use $2r$ oracle queries. For the remaining factor, proceed as above with odd j, applying $\mathbf{V}_j|_r$ instead of its transpose. The operations on singletons are

$$|0\rangle \longmapsto |0\rangle, \qquad |n-1\rangle \longmapsto e^{-i(\theta_{n-1}|_r)/2}|n-1\rangle$$

and can be implemented with controlled rotations. These operations take $\mathrm{poly}(\log n, r, \log(1/\varepsilon))$ gates and can achieve total half-diamond error at most $\varepsilon/2$ relative to the displayed maps. It therefore suffices to bound the truncation errors. Indeed, $0 \le \theta_j - \theta_j|_r \le 2\pi\, 2^{-r}$ and

$$\mathbb{E}_f\left(\gamma_j - \gamma_j|_r\right) \le 2^{-r} \sum_{k=0}^{2^r-1} \arcsin\Big(((k+1)2^{-r})^{1/(n-j-1)}\Big) - \arcsin\Big((k2^{-r})^{1/(n-j-1)}\Big) = \frac{\pi}{2}\, 2^{-r}.$$

Since the operator-norm error of each block is bounded by the truncation errors of its angles

$$\left\|\mathbf{V}_j - \mathbf{V}_j|_r\right\|_{\mathrm{op}} \le \sqrt{2}\left|\sin\frac{\gamma_j - \gamma_j|_r}{2}\right| + 2\left|\sin\frac{\theta_j - \theta_j|_r}{4}\right| \le \gamma_j - \gamma_j|_r + \frac{1}{2}\left(\theta_j - \theta_j|_r\right)$$

we can conclude that

$$\mathbb{E}_f \frac{1}{2}\left\|C^f_{n,\varepsilon} - \mathbf{V}(f)\right\|_\diamond \le \frac{\varepsilon}{2} + \sum_{j=0}^{n-2} \mathbb{E}_f\left\|\mathbf{V}_j - \mathbf{V}_j|_r\right\|_{\mathrm{op}} + \frac{1}{2}\mathbb{E}_f\big(\theta_{n-1} - \theta_{n-1}|_r\big) \le \frac{\varepsilon}{2} + \frac{3\pi n}{2}\, 2^{-r} \le \varepsilon. \quad \square$$

5 Scalable Pseudorandom Unitaries

We recall the definition of a scalable pseudorandom unitary (PRU).

Definition 5.1 (Scalable pseudorandom unitaries)**.** *A keyed family $\{\mathbf{U}_{n,\lambda,k}\}_k \subseteq \mathrm{U}(n)$ is a scalable pseudorandom unitary family if for every distinguisher $\mathcal{A}$ of size $\mathrm{poly}(\lambda + \log n)$ satisfies*

$$\left|\Pr_{k\sim\mathrm{Gen}(n,1^\lambda)}\left(\mathcal{A}^{\mathbf{U}_{n,\lambda,k}}(n,1^\lambda)=1\right) - \Pr_{\mathbf{U}\sim\mathrm{Haar}(\mathrm{U}(n))}\left(\mathcal{A}^{\mathbf{U}}(n,1^\lambda)=1\right)\right| \le \mathrm{negl}(\lambda+\log n).$$

The family is a *strong PRU* if the guarantee also holds when the distinguisher can query the inverse channel. The following is our main theorem.

Theorem 5.2. *Assume Conjecture 3.1 and quantum-secure pseudorandom functions. There exists a scalable strong PRU family in every power-of-two dimension n, with key generation and evaluation time $\mathrm{poly}(\lambda, \log n)$.*

Proof. Set $\varepsilon = 2^{-(\lambda+\log n)}$. If $n \le \lambda$, sample a circuit within expected half-diamond distance ε of a Haar-random unitary in time $\mathrm{poly}(n, \log(1/\varepsilon))$, and let the key contain its description. Its evaluation and inverse are efficient, and the distinguishing advantage after polynomially many queries is at most $\mathrm{poly}(\lambda+\log n)\varepsilon$.

For $n > \lambda$, set $t = t(n)$ as in Conjecture 3.1. The algorithm Gen samples a quantum-secure PRF seed at security parameter $\lambda + \log n$, with input length sufficient to encode all oracle queries, and independent uniform signs $b_0, b_1 \in \{\pm 1\}$. Define

$$\mathbf{U}_{n,\lambda,k} = \mathbf{U}(f_k, b_0, b_1) = \mathrm{diag}(b_0, 1, \ldots, 1) K^{f_k,(0)}_{n,t,\varepsilon} C^{f_k}_{n,\varepsilon}\, \mathrm{diag}(b_1, 1, \ldots, 1) K^{f_k,(1)}_{n,t,\varepsilon}.$$

By Lemmas 4.4 and 3.8, the resulting circuit has size $\mathrm{poly}(\lambda, \log n)$.

Quantum PRF security first allows us to replace f_k by a random function f in these finite circuits. This changes the acceptance probability of any quantum distinguisher of size $\mathrm{poly}(\lambda+\log n)$ by at most $\mathrm{negl}(\lambda+\log n)$. Under the independent couplings from Lemmas 4.4 and 3.8,

$$\mathbb{E}\frac{1}{2}\left\|\mathbf{U}(f,b_0,b_1)-\mathrm{diag}(b_0,1,\ldots,1)\mathbf{Q}_t^{(0)}\mathbf{V}\,\mathrm{diag}(b_1,1,\ldots,1)\mathbf{Q}_t^{(1)}\right\|_\diamond \le 3\varepsilon.$$

Here the two exact Kac walks and $\mathbf{V}$ are mutually independent. A hybrid over the oracle queries therefore changes the acceptance probability by at most $3\,\mathrm{poly}(\lambda+\log n)\varepsilon$.

Next apply Conjecture 3.1 to the two Kac factors one at a time. Given the challenge matrix, each reduction can simulate the remaining factors and both forward and inverse queries to error negligible in n in time $\mathrm{poly}(n,\lambda)$. Since $n>\lambda$, these reductions have size $\mathrm{poly}(n)$, and $\lambda+\log n\le 2n$ makes their $\mathrm{negl}(n)$ advantage negligible in $\lambda+\log n$. The independent signs convert the resulting special orthogonals into independent Haar orthogonals, so the final ideal distribution is

$$\mathbf{O}^{(0)}\mathbf{V}\mathbf{O}^{(1)} \overset{d}{=} \mathbf{U}, \qquad \mathbf{U}\sim\mathrm{Haar}(\mathrm{U}(n)),$$

by Lemma 4.3. Finally, inverse queries are implemented by reversing the finite circuits. The half-diamond distance between two unitary channels equals that between their inverses, so the same approximation and hybrid bounds apply. The total advantage is therefore $\mathrm{negl}(\lambda+\log n)$, also with inverse queries. □

Acknowledgements

Work supported by the European Research Council through an ERC Starting Grant (Grant agreement No. 101077455, ObfusQation) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA – 390781972.

References

[1] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom Quantum States. In *CRYPTO 2018*, pages 126–152. https://doi.org/10.1007/978-3-319-96878-0_5. 1

[2] Hsin-Yuan Huang and Richard Kueng. Predicting Features of Quantum Systems using Classical Shadows. Presented at *QIP 2020*. https://arxiv.org/abs/1908.08909. 1

[3] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. "Quantum Supremacy" and the Complexity of Random Circuit Sampling. In *ITCS 2019*, pages 15:1–15:2. https://doi.org/10.4230/LIPIcs.ITCS.2019.15. 1

[4] Sitan Chen, Jordan Cotler, Hsin-Yuan Huang, and Jerry Li. Exponential Separations Between Learning With and Without Quantum Memory. In *FOCS 2021*, pages 574–585. https://doi.org/10.1109/FOCS52979.2021.00063. 1

[5] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from Pseudorandom Quantum States. In *CRYPTO 2022*, pages 208–236. https://doi.org/10.1007/978-3-031-15802-5_8. 1

[6] Zvika Brakerski and Nir Magrafta. Real-Valued Somewhat-Pseudorandom Unitaries. In *TCC 2024*, pages 36–59. https://doi.org/10.1007/978-3-031-78017-2_2. 1

[7] Tony Metger, Alexander Poremba, Makrand Sinha, and Henry Yuen. Simple Constructions of Linear-Depth t-Designs and Pseudorandom Unitaries. In *FOCS 2024*, pages 485–492. https://doi.org/10.1109/FOCS61266.2024.00038. 1

[8] Chi-Fang Chen, Jordan Docter, Michelle Xu, Adam Bouland, Fernando G. S. L. Brandão, and Patrick Hayden. Efficient Unitary Designs from Random Sums and Permutations. In *FOCS 2024*, pages 476–484. https://doi.org/10.1109/FOCS61266.2024.00037. 1

[9] Fermi Ma and Hsin-Yuan Huang. How to Construct Random Unitaries. In *STOC 2025*, pages 806–809. https://doi.org/10.1145/3717823.3718254. 1

[10] Thomas Schuster, Jonas Haferkamp, and Hsin-Yuan Huang. Random Unitaries in Extremely Low Depth. Presented at *QIP 2025*. https://arxiv.org/abs/2407.07754. 1

[11] Thomas Schuster, Fermi Ma, Alex Lombardi, Fernando Brandão, and Hsin-Yuan Huang. Strong Random Unitaries and Fast Scrambling. Presented at *QIP 2026*. https://arxiv.org/abs/2509.26310. 1

[12] Ben Foxman, Natalie Parham, Francisca Vasconcelos, and Henry Yuen. Random Unitaries in Constant (Quantum) Time. In *ITCS 2026*, pages 61:1–61:25. https://doi.org/10.4230/LIPIcs.ITCS.2026.61. 1

[13] Prabhanjan Ananth, John Bostanci, Aditya Gulati, and Yao-Ting Lin. Pseudorandomness in the (Inverseless) Haar Random Oracle Model. In *EUROCRYPT 2025*, pages 138–166. https://doi.org/10.1007/978-3-031-91098-2_6. 1

[14] Prabhanjan Ananth, John Bostanci, Aditya Gulati, and Yao-Ting Lin. Pseudorandom Unitaries in the Haar Random Oracle Model. In *CRYPTO 2025*, pages 301–333. https://doi.org/10.1007/978-3-032-01878-6_10. 1

[15] Zvika Brakerski and Omri Shmueli. Scalable Pseudorandom Quantum States. In *CRYPTO 2020*, pages 417–440. https://doi.org/10.1007/978-3-030-56880-1_15. 1

[16] Chuhan Lu, Minglong Qin, Fang Song, Penghui Yao, and Mingnan Zhao. Quantum Pseudorandom Scramblers. In *TCC 2024*, pages 3–35. https://doi.org/10.1007/978-3-031-78017-2_1. 1, 2

[17] Chuhan Lu, Minglong Qin, Fang Song, Penghui Yao, and Mingnan Zhao. Parallel Kac's Walk Generates PRU. To appear in *ASIACRYPT 2026*. Full version: https://arxiv.org/abs/2504.14957. 2

[18] Rishabh Batra, Zhili Chen, Rahul Jain, and YaoNan Zhang. Scalable, Quantum-Accessible, and Adaptive Pseudorandom Quantum State and Pseudorandom Function-Like Quantum State Generators. In *CRYPTO 2026*, pages 215–246. https://doi.org/10.1007/978-3-032-35409-9_8. 1

[19] Zvika Brakerski and Henry Yuen. On Scalable Pseudorandom Unitaries and the Unitary Synthesis Problem. In *CRYPTO 2026*, pages 372–401. https://doi.org/10.1007/978-3-032-35409-9_13. 1

[20] Andrew Huang, Akshar Ramkumar, and John Wright. Random-Oracle Unitary Synthesis is Impossible. arXiv:2610.06700, 2026. https://arxiv.org/abs/2610.06700. 2

[21] Scott Aaronson and Greg Kuperberg. Quantum versus Classical Proofs and Advice. In *CCC 2007*, pages 115–128. https://doi.org/10.1109/CCC.2007.27. 1

[22] Vinod Vaikuntanathan and Or Zamir. Improving Algorithmic Efficiency using Cryptography: Trapdoored Matrices and Applications. In *Proceedings of SODA*, pages 2554–2574, 2026. https://doi.org/10.1137/1.9781611978971.92. Full version: https://arxiv.org/abs/2502.13065. 2, 3

[23] Natesh S. Pillai, Aaron Smith, and Vinod Vaikuntanathan. On the Pseudo-Mixing of Kac's Walk. arXiv:2608.17374, 2026. https://arxiv.org/abs/2608.17374. 2

[24] Ben Morris and Phillip Rogaway. Sometimes-Recurse Shuffle: Almost-Random Permutations in Logarithmic Expected Time. In *Advances in Cryptology – EUROCRYPT 2014*, pages 311–326, 2014. https://eprint.iacr.org/2013/560. 3

[25] Mark Zhandry. A Note on Quantum-Secure PRPs. *Quantum*, 9:1696, 2025. https://doi.org/10.22331/q-2025-04-08-1696. 3

[26] Luc Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, 1986. https://luc.devroye.org/rnbookindex.html. 4

[27] J. H. Ahrens and U. Dieter. Sampling from binomial and Poisson distributions: A method with bounded computation times. *Computing*, 25:193–208, 1980. https://doi.org/10.1007/BF02241999. 4

[28] Karl Bringmann, Fabian Kuhn, Konstantinos Panagiotou, Ueli Peter, and Henning Thomas. Internal DLA: Efficient Simulation of a Physical Growth Model. In *Proceedings of ICALP*, Part I, volume 8572 of *Lecture Notes in Computer Science*, pages 247–258. Springer, 2014. https://doi.org/10.1007/978-3-662-43948-7_21. Full version: https://people.mpi-inf.mpg.de/~kbringma/paper/2014ICALP.pdf. 4

[29] Rowan Killip and Irina Nenciu. Matrix models for circular ensembles. *International Mathematics Research Notices*, 2004(50):2665–2701, 2004. https://doi.org/10.1155/S1073792804141597. 8

A Proof of Theorem 4.1

For completeness, we present a short and self-contained proof of Theorem 4.1.

Proof. Let $\mathbf{U}$ be Haar in $\mathrm{U}(m)$ for $m \geq 2$, let $\mathbf{v}$ be its first column, and set $j = n - m$. Write $\mathbf{v}^\intercal \mathbf{v} = \cos\gamma_j \, e^{-i\theta_j}$ and set

$$\mathbf{w} = \frac{\overline{\mathbf{v}} - \cos\gamma_j \, e^{i\theta_j}\mathbf{v}}{\sin\gamma_j}.$$

Almost surely, $\mathbf{v}, \mathbf{w}$ are orthonormal. Complete them to a unitary matrix $\mathbf{Q}$ by choosing a real orthonormal basis perpendicular to $\operatorname{Re}\mathbf{v}$ and $\operatorname{Im}\mathbf{v}$. Since $\mathbf{v}^\intercal\overline{\mathbf{v}} = 1$ and $\mathbf{w}^\intercal\overline{\mathbf{v}} = 0$, we obtain

$$\mathbf{v}^\intercal\mathbf{w} = \frac{1 - \cos^2\gamma_j}{\sin\gamma_j} = \sin\gamma_j, \qquad \mathbf{w}^\intercal\mathbf{w} = -\frac{\cos\gamma_j \, e^{i\theta_j}\mathbf{w}^\intercal\mathbf{v}}{\sin\gamma_j} = -\cos\gamma_j \, e^{i\theta_j}.$$

Thus $\mathbf{Q}^\intercal\mathbf{Q} = \mathbf{T}_j \oplus \mathrm{id}_{m-2}$.

Conditional on $\mathbf{v}$, the remaining columns of $\mathbf{U}$ form a Haar-random orthonormal basis of $\mathbf{v}^{\perp}$. Hence $\mathbf{U} = \mathbf{Q}\mathbf{D}$, where $\mathbf{D} = 1 \oplus \mathbf{H}$ and $\mathbf{H}$ is Haar in $\mathrm{U}(m-1)$ and independent of $\mathbf{v}$. It follows that $\mathbf{U}^{\intercal}\mathbf{U} = \mathbf{D}^{\intercal}(\mathbf{T}_j \oplus \mathrm{id}_{m-2})\mathbf{D}$. Consequently,

$$\mathbf{D}(\mathbf{U}^{\intercal}\mathbf{U})\mathbf{D}^{-1} = (1 \oplus \mathbf{H}\mathbf{H}^{\intercal})(\mathbf{T}_j \oplus \mathrm{id}_{m-2}),$$
$$(\mathbf{D}^{\intercal})^{-1}(\mathbf{U}^{\intercal}\mathbf{U})\mathbf{D}^{\intercal} = (\mathbf{T}_j \oplus \mathrm{id}_{m-2})(1 \oplus \mathbf{H}\mathbf{H}^{\intercal}).$$

Both changes of basis preserve the spectrum and fix the first coordinate. Since $\mathbf{H}^{\intercal}$ is Haar, we may repeat in dimension $m-1$. Starting at $m = n$, use the first identity for even j and the second for odd j. At step j the change of basis fixes coordinates $0, \dots, j$, so it commutes with every earlier block. This gives the product defining $\mathbf{T}$, with independent blocks and a final scalar equal to the square of a uniform phase, hence itself uniform.

It remains to verify the angles. Since $\mathbf{v}$ is uniform on the real sphere S^{2m-1}, the density of $\|\operatorname{Re}\mathbf{v}\|^2$ is proportional to $s^{m/2-1}(1-s)^{m/2-1}$ for each $s \in (0,1)$. Thus $\operatorname{Re}(\mathbf{v}^{\intercal}\mathbf{v}) = 2\|\operatorname{Re}\mathbf{v}\|^2 - 1$ has density proportional to $(1-s^2)^{(m-2)/2}$ on $s \in (-1,1)$, the density of the first coordinate of a uniform point on S^m. By rotational invariance, $\mathbf{v}^{\intercal}\mathbf{v}$ is distributed as the first two coordinates of that point viewed as a complex number, since all their one-dimensional projections agree. Consequently, θ_j is uniform on $[0, 2\pi)$ and independent of γ_j. Write the norms of the first two and the remaining $m-1$ coordinates as $\cos u$ and $\sin u$, respectively. The corresponding spherical volume element is proportional to $\cos u\,(\sin u)^{m-2}\,du$, so the squared norm $s = \sin^2 u$ has normalized density $\frac{m-1}{2}s^{(m-3)/2}$ on $(0,1)$. Thus,

$$\Pr(\gamma_j \le t) = \Pr\bigl(1 - |\mathbf{v}^{\intercal}\mathbf{v}|^2 \le \sin^2 t\bigr) = \int_0^{\sin^2 t} \frac{m-1}{2} s^{(m-3)/2}\,ds = (\sin t)^{n-j-1}.$$

This gives the prescribed distribution of each block. □