# On the best generators of PU(2) II


Terrence Blackman[1] **Zachary Stier**[2]

[1]Medgar Evers College

[2]UC Berkeley


Joint Mathematics Meetings
AMS Special Session on Quaternions
6 April 2022
Seattle, WA

# Outline

- **Setup**: factorization with icosahedral super golden gates
- **Inspiration**: short paths in LPS Ramanujan graphs
- **Diagonal factorization**
- **General factorization**
- **Analysis**
- **Sums of squares**: hurdles in algorithmic algebraic number theory
- **Examples**

# Icosahedral super golden gates

$\rho$ is the matrix corresponding to $i + (\varphi - 1)j + \varphi k$
$\sigma$ $1 + i + j + k$
$\tau$ $(2 + \varphi)i + j + k$

# Icosahedral super golden gates

$\rho$ is the matrix corresponding to $i + (\varphi - 1)j + \varphi k$
$\sigma$ $\quad 1 + i + j + k$
$\tau$ $\quad (2 + \varphi)i + j + k$
Let $S := \{\rho, \sigma, \tau\}$ be the **icosahedral super golden gates**.

# Icosahedral super golden gates

$\rho$ is the matrix corresponding to $i + (\varphi - 1)j + \varphi k$
$\sigma$ $1 + i + j + k$
$\tau$ $(2 + \varphi)i + j + k$
Let $S := \{\rho, \sigma, \tau\}$ be the **icosahedral super golden gates**.
$\langle \rho, \sigma \rangle \cong A_5$ (hence *icosahedral*), and $\Gamma := \langle S \rangle$ is dense in $\mathrm{PU}(2)$.

# Icosahedral super golden gates, cont.

Fix $\mathfrak{p} := 7 + 5\varphi$. Notice that $\mathrm{N}(\rho) = \mathrm{N}(\sigma) = 4$ while $\mathrm{N}(\tau) = \mathfrak{p}$.

# Icosahedral super golden gates, cont.

Fix $\mathfrak{p} := 7 + 5\varphi$. Notice that $\mathrm{N}(\rho) = \mathrm{N}(\sigma) = 4$ while $\mathrm{N}(\tau) = \mathfrak{p}$.
This makes factoring in $\Gamma$ easy:

# Icosahedral super golden gates, cont.

Fix $\mathfrak{p} := 7 + 5\varphi$. Notice that $\mathrm{N}(\rho) = \mathrm{N}(\sigma) = 4$ while $\mathrm{N}(\tau) = \mathfrak{p}$. This makes factoring in $\Gamma$ easy: we access elements

$$\Gamma \ni \gamma = a_0 \tau a_1 \cdots \tau a_m$$

as lifts $\widehat{\gamma} \in \mathbb{H}(\mathbb{Z}[\varphi])$ where $\mathfrak{p} \nmid \widehat{\gamma}$ (as a scalar); $a_m$ is detectable as corresponding to the unique $\widehat{a} \in \widehat{A_5}$ for which $\mathfrak{p} \mid \mathrm{N}(\widehat{\gamma}\widehat{a}\tau)$.

# Icosahedral super golden gates, cont.

Fix $\mathfrak{p} := 7 + 5\varphi$. Notice that $\mathrm{N}(\rho) = \mathrm{N}(\sigma) = 4$ while $\mathrm{N}(\tau) = \mathfrak{p}$. This makes factoring in $\Gamma$ easy: we access elements

$$\Gamma \ni \gamma = a_0 \tau a_1 \cdots \tau a_m$$

as lifts $\widehat{\gamma} \in \mathbb{H}(\mathbb{Z}[\varphi])$ where $\mathfrak{p} \nmid \widehat{\gamma}$ (as a scalar); $a_m$ is detectable as corresponding to the unique $\widehat{a} \in \widehat{A_5}$ for which $\mathfrak{p} \mid \mathrm{N}(\widehat{\gamma}\widehat{a}\tau)$. Thus, factoring in $\Gamma$ is $O(m)$.

# Short paths in $X^{p,q}$

Recall the LPS Ramanujan graphs $X^{p,q}$.

**Theorem (Carvalho Pinto–Petit '18)**

There exists a factorization of any element of $X^{p,q}$ into $(7/3 + o(1)) \log_p(q^3)$ generators.

## Short paths in $X^{p,q}$

Recall the LPS Ramanujan graphs $X^{p,q}$.

**Theorem (Carvalho Pinto–Petit '18)**

There exists a factorization of any element of $X^{p,q}$ into $({}^7\!/_3 + o(1)) \log_p(q^3)$ generators.

The idea is to compute, given $g \in X^{p,q}$, elements $\gamma_1, \gamma, \gamma_2 \in X^{p,q}$ where: $\gamma_1$ and $\gamma_2$ are diagonal; $\gamma$ has particularly short factorization; and $g = \gamma_1 \gamma \gamma_2$.

# Our main result

**Theorem**

There exists a factorization of any element sufficiently far from the identity of PU(2) using $\tau$-count at most $(7/3 + o(1)) \log_{59}(1/\varepsilon^3)$.

# Our main result

**Theorem**

There exists a factorization of any element sufficiently far from the identity of PU(2) using $\tau$-count at most $(7/3 + o(1)) \log_{59}(1/\varepsilon^3)$.

We are concerned with $\tau$-count because for certain engineering purposes, the $A_5$ gates are simpler to construct while the $\tau$ involution is extremely costly. (Similar gate cost models are used with other generators, in particular Clifford+$T$.)

# Some quick shorthand

Because every element of SU(2) takes the form

$$g = \begin{pmatrix} \alpha & \beta \\ -\overline{\beta} & \overline{\alpha} \end{pmatrix}$$

for some $\alpha, \beta \in \mathbb{C}$, we say that $g = u(\alpha, \beta)$; and diagonals take the form

$$\delta = \begin{pmatrix} e^{i\theta} & \\ & e^{-i\theta} \end{pmatrix}$$

for some $\theta \in \mathbb{R}$, whence we write $\delta = u(\theta)$.

We also identify elements of PU(2) with their lifts to SU(2), as appropriate.

# Technical substantiation

This result enables the finite–continuous analogy to go through.

**Lemma ("tuning;" S. '21)**

Select absolute constants $\delta, \varepsilon_0 > 0$ and put $C = \sqrt{\frac{1}{2} + \frac{1}{2}\left(\frac{2+\delta}{\varepsilon_0}\right)^2}$. Take $\gamma_1, \gamma_2 \in \mathrm{PU}(2)$ and write them as $\gamma_\ell = u(\alpha_\ell, \beta_\ell)$. If $||\alpha_1| - |\alpha_2|| < \varepsilon$ for some $\varepsilon < \delta$ and $\min\{|\alpha_1|, |\alpha_2|\} < \sqrt{1-\varepsilon_0^2}$ then for

$$\begin{aligned} \theta_1 &= \frac{1}{2}(\arg\alpha_1 - \arg\alpha_2 + \arg\beta_1 - \arg\beta_2), & \delta_1 &= u(\theta_1) \\ \theta_2 &= \frac{1}{2}(\arg\alpha_1 - \arg\alpha_2 - \arg\beta_1 + \arg\beta_2), & \delta_2 &= u(\theta_2) \end{aligned}$$

we have the approximation $\delta_1\gamma_2\delta_2$ to $\gamma_1$, satisfying

$$d(\gamma_1, \delta_1\gamma_2\delta_2) < C\varepsilon.$$

# Diagonal elements

For given diagonal $\delta = u(\theta)$ and $\varepsilon$, we seek $\gamma \in \Gamma$ with $d(\delta, \gamma) < \varepsilon$ where

$$\gamma = \begin{pmatrix} x_0 + x_1 i & x_2 + x_3 i \\ -x_2 + x_3 i & x_0 - x_1 i \end{pmatrix}$$

for $x_0, x_1, x_2, x_3 \in \mathbb{Z}[\varphi]$ satisfying

$$x_0^2 + x_1^2 + x_2^2 + x_3^2 = \mathfrak{p}^m \tag{$\dagger$}$$

for some $m \in \mathbb{N}$ (factorization of length $m$). Ross–Selinger deduce that

$$x_0 \cos\theta + x_1 \sin\theta \geqslant \mathfrak{p}^{m/2}(1 - 2\varepsilon^2) \tag{$\ddagger$}$$

is sufficient.

# Diagonal elements, cont.

Algebraic manipulation and Galois conjugates reduce (†) and (‡) to consideration of $x_1 =: c + d\varphi$ and the following sufficient conditions:

$$\begin{aligned} (c+d\varphi)\sin\theta &\leqslant \mathfrak{p}^{m/2}(1-\varepsilon^2) \\ |c+d\sigma_\pm\varphi| &\leqslant (\sigma_\pm\mathfrak{p})^{m/2} \\ \left|c+d\varphi-\mathfrak{p}^{m/2}(1-\varepsilon^2)\sin\theta\right| &\leqslant \mathfrak{p}^{m/2}\,|\cos\theta|\,\sqrt{2-\varepsilon^2}\varepsilon. \end{aligned}$$

Lenstra's algorithm finds all such points efficiently.

# Diagonal elements, cont.

Generically, the solution set is grid points contained in a long, thin, tilted rectangle:

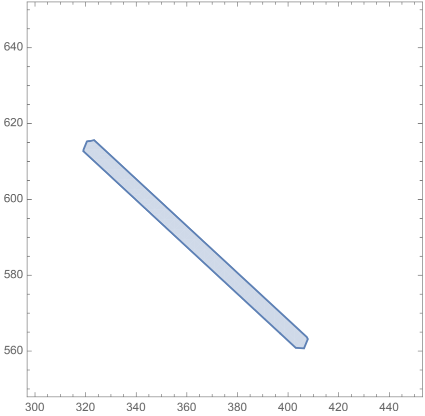


Figure: Feasible set for $\theta = \pi/8$, $m = 6$, and $\varepsilon = 1/10^3$.

# Diagonal elements, cont.

Then, we seek seek $x_0 =: a + b\varphi$ from

$$\begin{aligned} |a + b\sigma_\pm\varphi| &\leqslant (\sigma_\pm\mathfrak{p})^{m/2}\sqrt{1-(\sigma_\pm x_1)^2} \\ (a+b\varphi)\cos\theta &\leqslant \mathfrak{p}^{m/2}(1 - x_1\sin\theta) \\ (a+b\varphi)\cos\theta &\geqslant \mathfrak{p}^{m/2}(1 - \varepsilon^2 - x_1\sin\theta) \end{aligned}$$

again using Lenstra's algorithm.

# Diagonal elements, cont.

Then, we seek seek $x_0 =: a + b\varphi$ from

$$
\begin{aligned}
|a + b\sigma_\pm\varphi| &\leqslant (\sigma_\pm\mathfrak{P})^{m/2}\sqrt{1-(\sigma_\pm x_1)^2} \\
(a+b\varphi)\cos\theta &\leqslant \mathfrak{P}^{m/2}(1 - x_1\sin\theta) \\
(a+b\varphi)\cos\theta &\geqslant \mathfrak{P}^{m/2}(1 - \varepsilon^2 - x_1\sin\theta)
\end{aligned}
$$

again using Lenstra's algorithm.

We complete the search, having found candidates $x_0$ and $x_1$, by writing $\mathfrak{P}^m - x_0^2 - x_1^2$ as a sum of two squares $x_2^2 + x_3^2$ in $\mathbb{Z}[\varphi]$.

## Diagonal elements, cont.

Then, we seek seek $x_0 =: a + b\varphi$ from

$$|a + b\sigma_\pm\varphi| \leqslant (\sigma_\pm\mathfrak{p})^{m/2}\sqrt{1-(\sigma_\pm x_1)^2}$$
$$(a + b\varphi)\cos\theta \leqslant \mathfrak{p}^{m/2}(1 - x_1\sin\theta)$$
$$(a + b\varphi)\cos\theta \geqslant \mathfrak{p}^{m/2}(1 - \varepsilon^2 - x_1\sin\theta)$$

again using Lenstra's algorithm.

We complete the search, having found candidates $x_0$ and $x_1$, by writing $\mathfrak{p}^m - x_0^2 - x_1^2$ as a sum of two squares $x_2^2 + x_3^2$ in $\mathbb{Z}[\varphi]$.

The factorization length, if we start from $m = 1$, will be exactly the $m$ on which we halt.

# General elements

For given element $g = u(\alpha, \beta)$ and $\varepsilon$, we seek $\gamma \in \Gamma$ with $d(g, \gamma) < \varepsilon$ where

$$\gamma = \begin{pmatrix} x_0 + x_1 i & x_2 + x_3 i \\ -x_2 + x_3 i & x_0 - x_1 i \end{pmatrix}$$

for $x_0, x_1, x_2, x_3 \in \mathbb{Z}[\varphi]$ satisfying

$$x_0^2 + x_1^2 + x_2^2 + x_3^2 = \mathfrak{p}^m \tag{$\star$}$$

for some $m \in \mathbb{N}$ (factorization of length $m$). All we need to apply tuning is have $\sqrt{\frac{x_0^2 + x_1^2}{\mathfrak{p}^m}} \approx |\alpha|$.

# General elements, cont.

This transforms into

$$\left|x_0^2 + x_1^2 - |\alpha|^2\,\mathfrak{p}^m\right| < \varepsilon\,|\alpha|\,\mathfrak{p}^m.$$

Studying Galois conjugates of $(\star)$ give the added condition

$$\sigma_\pm(x_0^2 + x_1^2) \leqslant (\sigma_\pm\mathfrak{p})^m.$$

# General elements, cont.

This transforms into

$$\left| x_0^2 + x_1^2 - |\alpha|^2 \mathfrak{p}^m \right| < \varepsilon |\alpha| \mathfrak{p}^m.$$

Studying Galois conjugates of $(\star)$ give the added condition

$$\sigma_\pm(x_0^2 + x_1^2) \leqslant (\sigma_\pm \mathfrak{p})^m.$$

Viewing $x_0^2 + x_1^2$ as the element $a + b\varphi \in \mathbb{Z}[\varphi]$ (a rank-two lattice) we apply Lenstra's algorithm to

$$\begin{aligned} \left| a + b\varphi - |\alpha|^2 \mathfrak{p}^m \right| &< \varepsilon |\alpha| \mathfrak{p}^m \\ a + b\sigma_\pm\varphi &\leqslant (\sigma_\pm \mathfrak{p})^m \\ a + b\sigma_\pm\varphi &\geqslant 0. \end{aligned}$$

# General elements, cont.

Arriving at a candidate pair $(a, b)$, we find the squares $x_0$ and $x_1$, and then use the same sum-of-squares algorithm to find $x_2$ and $x_3$ satisfying $(\star)$.

# General elements, cont.

Arriving at a candidate pair $(a, b)$, we find the squares $x_0$ and $x_1$, and then use the same sum-of-squares algorithm to find $x_2$ and $x_3$ satisfying $(\star)$.

We perform this task for each $m$, starting with $m = 1$, until a valid quadruple is found.

# General elements, cont.

Arriving at a candidate pair $(a, b)$, we find the squares $x_0$ and $x_1$, and then use the same sum-of-squares algorithm to find $x_2$ and $x_3$ satisfying $(\star)$.

We perform this task for each $m$, starting with $m = 1$, until a valid quadruple is found.

Then, we compute the phases for the tuning lemma and compute the corresponding diagonal approximations.

# Analysis

The algorithm for diagonals is fundamentally the same as Ross–Selinger's, so we defer that analysis to their paper. The runtime is $O(\operatorname{poly}\log(1/\varepsilon))$ and the factorization length is $(1+o(1))\log(1/\varepsilon^3)$.

# Analysis

The algorithm for diagonals is fundamentally the same as Ross–Selinger's, so we defer that analysis to their paper. The runtime is $O(\operatorname{poly}\log(1/\varepsilon))$ and the factorization length is $(1+o(1))\log(1/\varepsilon^3)$.

For the general algorithm, we expect to halt when the planar region has area $\Theta(\operatorname{poly}\log(1/\varepsilon))$. As the area is exactly $\frac{2|\alpha|}{\sqrt{5}}59^m\varepsilon$, we expect to halt when $m \approx \log_{59}(1/\varepsilon)$.

# Analysis

The algorithm for diagonals is fundamentally the same as Ross–Selinger's, so we defer that analysis to their paper. The runtime is $O(\operatorname{poly}\log({}^1\!/\!_\varepsilon))$ and the factorization length is $(1+o(1))\log({}^1\!/\!_{\varepsilon^3})$.

For the general algorithm, we expect to halt when the planar region has area $\Theta(\operatorname{poly}\log({}^1\!/\!_\varepsilon))$. As the area is exactly $\frac{2|\alpha|}{\sqrt{5}}59^m\varepsilon$, we expect to halt when $m \approx \log_{59}({}^1\!/\!_\varepsilon)$. Thus the runtime remains $O(\operatorname{poly}\log({}^1\!/\!_\varepsilon))$ and the total length is $({}^7\!/\!_3 + o(1))\log({}^1\!/\!_{\varepsilon^3})$, as claimed.

# Analysis

The algorithm for diagonals is fundamentally the same as Ross–Selinger's, so we defer that analysis to their paper. The runtime is $O(\operatorname{poly}\log(1/\varepsilon))$ and the factorization length is $(1 + o(1))\log(1/\varepsilon^3)$.

For the general algorithm, we expect to halt when the planar region has area $\Theta(\operatorname{poly}\log(1/\varepsilon))$. As the area is exactly $\frac{2|\alpha|}{\sqrt{5}}59^m\varepsilon$, we expect to halt when $m \approx \log_{59}(1/\varepsilon)$. Thus the runtime remains $O(\operatorname{poly}\log(1/\varepsilon))$ and the total length is $(7/3 + o(1))\log(1/\varepsilon^3)$, as claimed.

The tuning lemma gives that precision is lossy only up to a constant prefactor.

# Sums of squares

The main obstacle in pure algebraic number theory to overcome in this work is the task of computing $x, y \in \mathbb{Z}[\varphi]$, given (WLOG irreducible) $z \in \mathbb{Z}[\varphi]$, for which

$$z = x^2 + y^2.$$

(Assume efficient integer factorization.)

# Sums of squares

The main obstacle in pure algebraic number theory to overcome in this work is the task of computing $x, y \in \mathbb{Z}[\varphi]$, given (WLOG irreducible) $z \in \mathbb{Z}[\varphi]$, for which

$$z = x^2 + y^2.$$

(Assume efficient integer factorization.)

Recall that for $p \in \mathbb{Z}$, this can be done by computing $w$ for which $w^2 + 1 \equiv 0 \pmod{p}$ and then finding $\gcd(p, w + i) \in \mathbb{Z}[i]$.

# Sums of squares

The main obstacle in pure algebraic number theory to overcome in this work is the task of computing $x, y \in \mathbb{Z}[\varphi]$, given (WLOG irreducible) $z \in \mathbb{Z}[\varphi]$, for which

$$z = x^2 + y^2.$$

(Assume efficient integer factorization.)

Recall that for $p \in \mathbb{Z}$, this can be done by computing $w$ for which $w^2 + 1 \equiv 0 \pmod{p}$ and then finding $\gcd(p, w + i) \in \mathbb{Z}[i]$. Crucially, **$\boldsymbol{p}$ is 1 mod 4 and $\mathbb{Z}[\boldsymbol{i}]$ is a Euclidean domain.**

# Sums of squares, cont.

We prove the following:

**Theorem**

$\mathbb{Q}(i, \varphi)$ is norm-Euclidean.

# Sums of squares, cont.

We prove the following:

**Theorem**

$\mathbb{Q}(i, \varphi)$ is norm-Euclidean.

**Corollary**

Let $u \in \mathbb{Z}[\varphi]$ be irreducible and $\mathrm{N}(u)$ be either $p$ or $p^2$. By passing up to $\mathbb{Z}[i, \varphi]$, if $p \equiv 1, 3, 7, 9, 13, 17 \pmod{20}$ then either $u$ or $u\varphi$ is a sum of two squares.

# Code

Our algorithm has been implemented in Python. Visit `https://math.berkeley.edu/~zstier/icosahedral` to download the code and for some documentation.

# Examples

Recall the generators of the Clifford+$T$ gate set:

$$H = \frac{i}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad\qquad T = \begin{pmatrix} e^{i\pi/8} & \\ & e^{-i\pi/8} \end{pmatrix}.$$

We demonstrate factorizations of both, to precision $\varepsilon = {}^1\!/_{10^{10}}$.

# Example: $T$

$$
\begin{aligned}
T \approx\ & (\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\sigma\rho\sigma\sigma)\tau(\sigma\rho\sigma\rho\sigma)\tau(\rho\sigma\rho\sigma\rho\sigma)\tau(\rho\sigma\sigma\rho \\
& \sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho)\tau(\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho) \\
& \tau(\rho)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\rho\sigma\rho\sigma\rho\sigma\sigma)\tau(\sigma\sigma\rho\sigma\rho\sigma\sigma\rho) \\
& \tau(\sigma\rho\sigma\rho\sigma\sigma)\tau(\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\sigma\rho \\
& \sigma\rho\sigma)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\sigma\rho\sigma)
\end{aligned}
$$

This has $\tau$-count 19, against predicted 16.9, and is accurate up to ${}^{1.28}/_{10^{10}}$ in $d$.

# Example: $H$

$$
\begin{aligned}
\gamma &= (\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\sigma)\tau(\sigma\rho\sigma\rho\sigma\sigma)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\rho\sigma\sigma\rho)\tau \\
&\quad (\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\sigma)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma)\tau(\rho\sigma\sigma\rho\sigma\rho) \\
\gamma_1 &= (\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\rho\sigma\sigma\rho\sigma\sigma)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\rho\sigma)\tau(\sigma \\
&\quad \rho\sigma\rho)\tau(\sigma\sigma\rho\sigma)\tau(\sigma\sigma\rho\sigma\rho\sigma)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma)\tau(\rho \\
&\quad \sigma\rho\sigma\sigma\rho)\tau(\sigma\sigma\rho\sigma)\tau(\sigma\sigma\rho\sigma\rho\sigma\sigma)\tau(\sigma\sigma)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\sigma)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\sigma\sigma\rho \\
&\quad \sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\rho\sigma) \\
\gamma_2 &= (\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\sigma\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\sigma)\tau(\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\sigma\sigma\rho\sigma\rho\sigma)\tau(\rho\sigma\rho \\
&\quad \sigma\sigma)\tau(\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\sigma)\tau(\rho\sigma\sigma\rho\sigma)\tau(\rho\sigma\rho\sigma\rho\sigma\sigma\rho \\
&\quad \sigma\rho)\tau(\rho\sigma\sigma\rho)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\rho\sigma\sigma\rho\sigma)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma) \\
&\quad \tau(\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\sigma\rho)\tau(\rho)
\end{aligned}
$$

The overall $\tau$-count is 45, against predicted 39.4, and $\gamma_1\gamma\gamma_2$ is accurate up to ${}^{1.28}/{}_{10^{10}}$ in $d$.

# Future directions

In the past decade the almost-guarantees have been reduced from $\operatorname{poly}\log$ to $c\log$, while the $c = 1$ case is provably NP-complete. Recent work (including this) has reduced $c$ to as low as $7/3$. How close can one get to $c = 1$?

# Future directions

In the past decade the almost-guarantees have been reduced from $\operatorname{poly}\log$ to $c\log$, while the $c = 1$ case is provably NP-complete. Recent work (including this) has reduced $c$ to as low as $7/3$. How close can one get to $c = 1$?

Maybe CNOTs plus universal single-qubit sets aren't optimal for (say) the two-qubit gates. What is? (This study is already underway, see e.g. Evra–Parzanchevski's work on PU(3).)

# Acknowledgements

Thank you to my coauthor Terrence Blackman; Peter Sarnak; and the organizers of this Special Session.

This work was supported by an NSF Graduate Research Fellowship, grants DGE-1752814/2146752.

# References

The following references were cited directly in this presentation. Please see our paper for a full list of references.


- E. Carvalho Pinto and C. Petit, "Better path-finding algorithms in LPS Ramanujan graphs," *Journal of Mathematical Cryptology* 2018.
- S. Evra and O. Parzanchevski, "Ramanujan complexes and Golden Gates in PU(3)," arXiv:1810.04710.
- H. Lenstra, "Integer Programming with a Fixed Number of Variables," *Mathematics of Operations Research* 1983.
- O. Parzanchevski and P. Sarnak, "Super-Golden-Gates for PU(2)," *Advances in Mathematics* 2018.
- N. Ross and P. Selinger, "Optimal ancilla-free Clifford+$T$ approximation of $z$-rotations," *Quantum Information & Computation* 2016.
- Z. Stier, "Short paths in PU(2)," *Quantum Information & Computation* 2021.