Tutorial

Breaking and Making Quantum Speedups Workshop

Siddhartha Jain, Seyoon Ragavan

What is quantum computing?

A primer on BQP

What is quantum computing?

A primer on BQP

Start with a randomized algorithm (BPP) and replace the probabilities by complex “amplitudes” with ℓ_2 norm 1.

What is quantum computing?

A primer on BQP

Start with a randomized algorithm (BPP) and replace the probabilities by complex "amplitudes" with ℓ_2 norm 1.

A qubit is $\alpha|0\rangle + \beta|1\rangle$ where $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $|\alpha|^2 + |\beta|^2 = 1$.

What is quantum computing?

A primer on BQP

Start with a randomized algorithm (BPP) and replace the probabilities by complex "amplitudes" with ℓ_2 norm 1.

A qubit is $\alpha|0\rangle + \beta|1\rangle$ where $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $|\alpha|^2 + |\beta|^2 = 1$.

Entanglement:

What is quantum computing?

A primer on BQP

Start with a randomized algorithm (BPP) and replace the probabilities by complex "amplitudes" with ℓ_2 norm 1.

A qubit is $\alpha|0\rangle + \beta|1\rangle$ where $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$ and $|\alpha|^2 + |\beta|^2 = 1$.

Entanglement:

$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ cannot be written as a tensor product ($\approx$ concatenation) of two qubits.

What is quantum computing?

A primer on BQP

What is quantum computing?

A primer on BQP

There are two main operations in quantum computing.

What is quantum computing?

A primer on BQP

There are two main operations in quantum computing.

Measurement:

What is quantum computing?

A primer on BQP

There are two main operations in quantum computing.

Measurement:

On measuring state $|\psi\rangle = \sum_{x \in 0,1^n} a_x |x\rangle$ we see x with probability $|a_x|^2$

What is quantum computing?

A primer on BQP

There are two main operations in quantum computing.

Measurement:

On measuring state $|\psi\rangle = \sum_{x \in 0,1^n} a_x |x\rangle$ we see x with probability $|a_x|^2$

Unitary evolution:

What is quantum computing?

A primer on BQP

There are two main operations in quantum computing.

Measurement:

On measuring state $|\psi\rangle = \sum_{x \in 0,1^n} a_x |x\rangle$ we see x with probability $|a_x|^2$

Unitary evolution:

Map $|\psi\rangle \rightarrow U|\psi\rangle$ where $UU^\dagger = U^\dagger U = I$, norm-preserving & invertible linear transform

What is quantum computing?

A primer on BQP

What is quantum computing?

A primer on BQP

Our measure of complexity is (uniform) **circuit size** after picking your favorite gate set (does not matter due to Solovay-Kitaev theorem), mine is Toffoli (T) + Hadamard (H).

What is quantum computing?

A primer on BQP

Our measure of complexity is (uniform) **circuit size** after picking your favorite gate set (does not matter due to Solovay-Kitaev theorem), mine is Toffoli (T) + Hadamard (H).

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$ thus

$$H|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), H|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

$$H\frac{|0\rangle + |1\rangle}{\sqrt{2}} = |0\rangle$$

What is quantum computing?

A primer on BQP

Our measure of complexity is (uniform) **circuit size** after picking your favorite gate set (does not matter due to Solovay-Kitaev theorem), mine is Toffoli (T) + Hadamard (H).

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$ thus

$$H|0\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big), H|1\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big)$$

$$T\,|\,x, y, z\rangle = |\,x, y, z \oplus (x \wedge y)\rangle$$

$$H\frac{|0\rangle + |1\rangle}{\sqrt{2}} = |0\rangle$$

What is quantum computing?

A primer on BQP

What is quantum computing?

A primer on BQP

What can we do with this? A lot. Most famously,

What is quantum computing?

A primer on BQP

What can we do with this? A lot. Most famously,

[Shor94] Factoring is in BQP.

Besides that, we have expected for a long time that it is useful for

What is quantum computing?

A primer on BQP

What can we do with this? A lot. Most famously,

[Shor94] Factoring is in BQP.

Besides that, we have expected for a long time that it is useful for

- Simulating quantum physics and chemistry (exponential speedups)

What is quantum computing?

A primer on BQP

What can we do with this? A lot. Most famously,

[Shor94] Factoring is in BQP.

Besides that, we have expected for a long time that it is useful for

- Simulating quantum physics and chemistry (exponential speedups)
- Many search problems (quadratic speedups)

What is quantum computing?

A primer on BQP

What can we do with this? A lot. Most famously,

[Shor94] Factoring is in BQP.

Besides that, we have expected for a long time that it is useful for

- Simulating quantum physics and chemistry (exponential speedups)
- Many search problems (quadratic speedups)

Where is quantum computing?

Answer: it's an exciting time

Where is quantum computing?

Answer: it's an exciting time

In the lab

Where is quantum computing?

Answer: it's an exciting time

In the lab

- Google's Willow chip demonstrated an error rate below the surface code threshold.

Where is quantum computing?

Answer: it's an exciting time

In the lab

- Google's Willow chip demonstrated an

Article | Open access | Published: 09 December 2024

Quantum error correction below the surface code threshold

Google Quantum AI and Collaborators

Nature **638**, 920–926 (2025) | Cite this article

Abstract

Quantum error correction[1,2,3,4] provides a path to reach practical quantum computing by combining multiple physical qubits into a logical qubit, in which the logical error rate is suppressed exponentially as more qubits are added. However, this exponential suppression only occurs if the physical error rate is below a critical threshold. Here we present two below-threshold surface code memories on our newest generation of superconducting processors, Willow: a distance-7 code and a distance-5 code integrated with a real-time decoder. The logical error rate of our larger quantum memory is suppressed by a factor of $\Lambda = 2.14 \pm 0.02$ when increasing the code distance by 2, culminating in a 101-qubit distance-7 code with 0.143% ± 0.003 per cent error per cycle of error correction. This logical memory is also

Where is quantum computing?

Answer: it’s an exciting time

In the lab

- Google’s Willow chip demonstrated an error rate below the surface code threshold.

Where is quantum computing?

Answer: it's an exciting time

In the lab

- Google's Willow chip demonstrated an error rate below the surface code threshold.
- Quantum computers from IBM, Quantinuum, QuEra, PsiQuantum & more already performing experiments and poised to scale up.

Where is quantum computing?

Answer: it's an exciting time

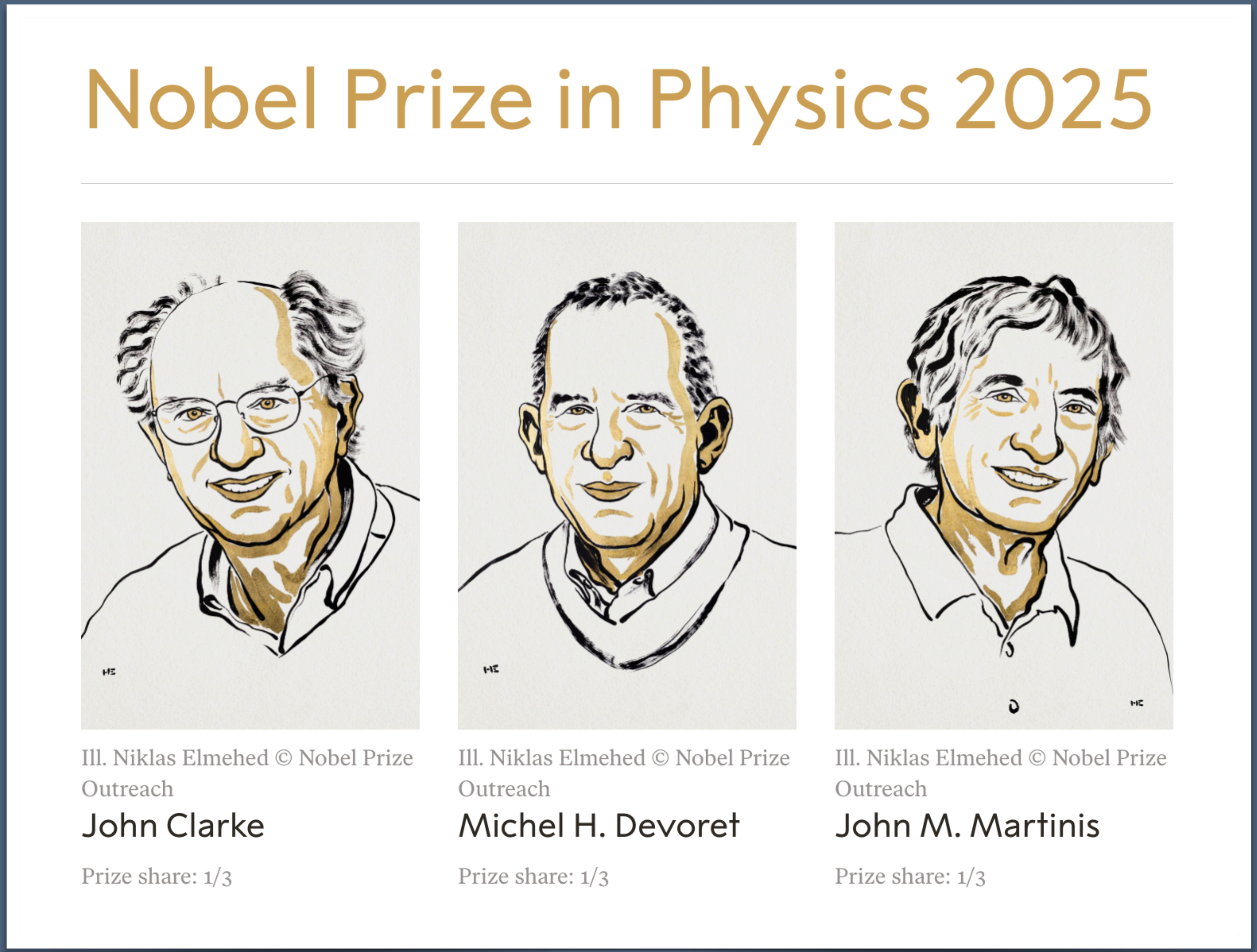
In the lab

- Google's Willow chip demonstrated an error rate below the surface code threshold.
- Quantum computers from IBM, Quantinuum, QuEra, PsiQuantum & more already performing experiments and poised to scale up.

Nobel Prize in Physics this year awarded to John Clarke, Michel H. Devoret and John M. Martinis for "for the discovery of macroscopic quantum mechanical tunnelling and energy quantisation in an electric circuit."

Where is quantum computing?

Answer: it's an exciting time



Where is quantum computing?

Answer: it’s an exciting time

Where is quantum computing?

Answer: it's an exciting time

In theory

Where is quantum computing?

Answer: it’s an exciting time

In theory

New speedups!

Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advantage without Structure (on the inputs)!

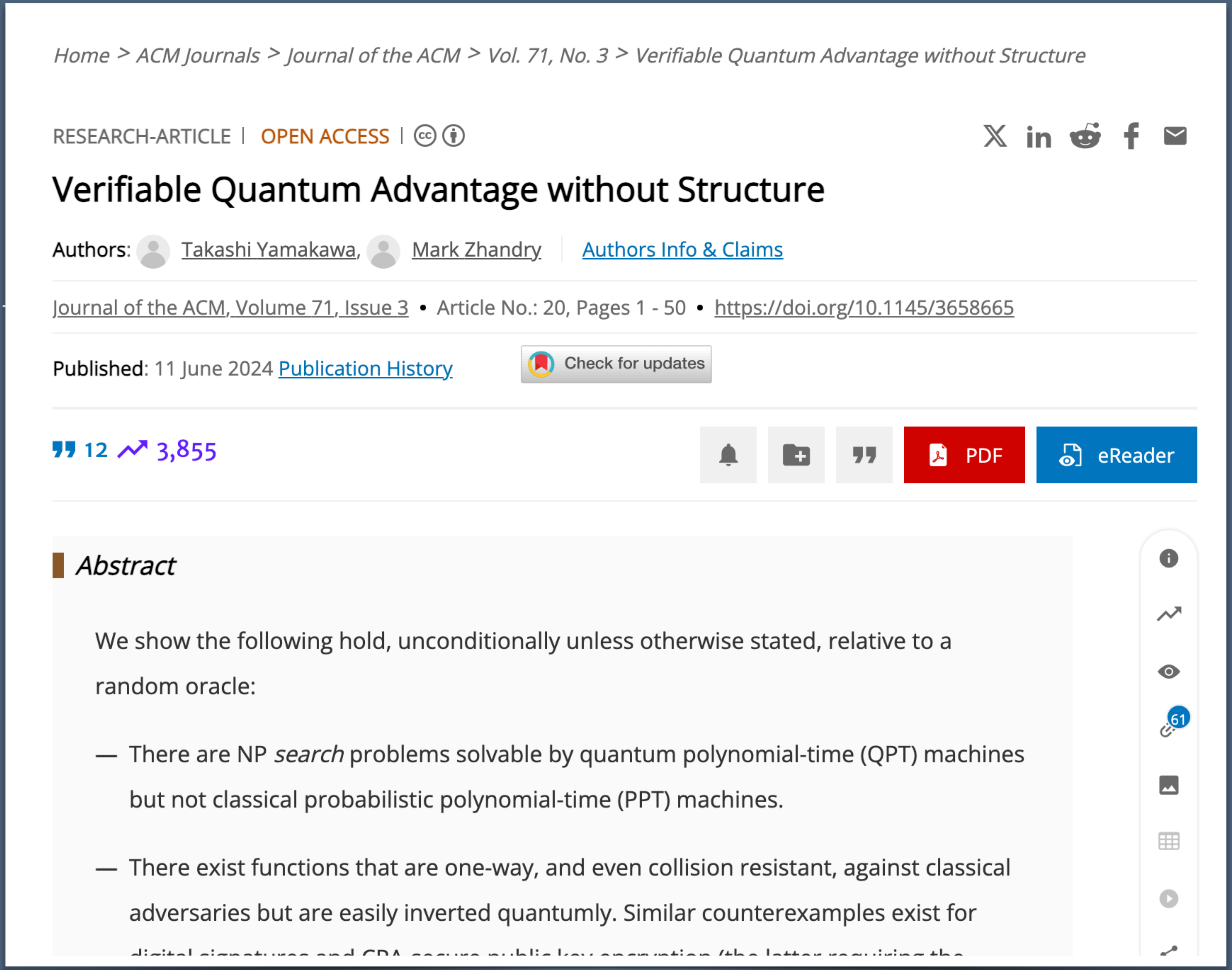
Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advantage without Struc



Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advantage without Structure (on the inputs)!

Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advantage without Structure (on the inputs)!
- Using a similar framework (Regev's reduction), Decoded Quantum Interferometry for approximate optimization

Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advantage without Structur
- Using a similar framework (Regev's reduction), D approximate optimization

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › articles › article

Article | Open access | Published: 22 October 2025

Optimization by decoded quantum interferometry

Stephen P. Jordan, Noah Shutty, Mary Wootters, Adam Zalcman, Alexander Schmidhuber, Robbie King, Sergei V. Isakov, Tanuj Khattar & Ryan Babbush

Nature **646**, 831–836 (2025) | Cite this article

Abstract

Achieving superpolynomial speed-ups for optimization has long been a central goal for quantum algorithms[1]. Here we introduce decoded quantum interferometry (DQI), a quantum algorithm that uses the quantum Fourier transform to reduce optimization problems to decoding problems. When approximating optimal polynomial fits over finite fields, DQI achieves a superpolynomial speed-up over known classical algorithms. The speed-up arises because the algebraic structure of the problem is reflected in the decoding problem, which can be solved efficiently. We then investigate whether this approach can achieve a speed-up for optimization problems that lack an algebraic structure but have sparse clauses. These problems reduce to decoding low-density parity-check codes, for which powerful decoders are known[2,3]. To test this, we construct a max-XORSAT instance for which DQI finds an

Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advantage without Structure (on the inputs)!
- Using a similar framework (Regev's reduction), Decoded Quantum Interferometry for approximate optimization

Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advantage without Structure (on the inputs)!
- Using a similar framework (Regev's reduction), Decoded Quantum Interferometry for approximate optimization
- Quartic quantum speedups for planted inference

Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advant)!
- Using a similar framework (um Interferometry for approximate optimization

Classical and Quantum Algorithms for Tensor Principal Component Analysis

Matthew B. Hastings

Station Q, Microsoft Research, Santa Barbara, CA 93106-6105, USA
Microsoft Quantum and Microsoft Research, Redmond, WA 98052, USA

Published:	2020-02-27, volume 4, page 237
Eprint:	arXiv:1907.12724v2
Doi:	https://doi.org/10.22331/q-2020-02-27-237
Citation:	Quantum 4, 237 (2020).

GET FULL TEXT PDF | READ ON ARXIV VANITY

Find this paper interesting or want to discuss? Scite or leave a comment on SciRate.

Abstract

We present classical and quantum algorithms based on spectral methods for a problem in tensor principal component analysis. The quantum algorithm achieves a *quartic* speedup while using exponentially smaller space than the fastest classical spectral algorithm, and a super-polynomial speedup over classical algorithms that use only polynomial space. The classical algorithms that we present are related to, but slightly different from those presented recently in Ref. [1]. In particular, we have an improved threshold for recovery and the algorithms we present work for both even and odd order tensors. These results suggest that large-scale inference problems are a promising future application for quantum computers.

OPEN ACCESS

Quartic Quantum Speedups for Planted Inference

Alexander Schmidhuber [1,2], Ryan O'Donnell [3], Robin Kothari [1], and Ryan Babbush [1]

Show more

Phys. Rev. X **15**, 021077 – **Published 2 June, 2025**

er

Quartic quantum speedups for community detection

Alexander Schmidhuber* — MIT

Alexander Zlokapa† — MIT

October 9, 2025

Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advantage without Structure (on the inputs)!
- Using a similar framework (Regev's reduction), Decoded Quantum Interferometry for approximate optimization

OPEN ACCESS

Quartic Quantum Speedups for Planted Inference

Alexander Schmidhuber[1,2], Ryan O'Donnell[3], Robin Kothari[1], and Ryan Babbush[1]

Show more

Phys. Rev. X **15**, 021077 – **Published 2 June, 2025**

Quartic quantum speedups for community detection

Alexander Schmidhuber* MIT
Alexander Zlokapa† MIT

October 9, 2025

Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advantage without Structure (on the inputs)!
- Using a similar framework (Regev's reduction), Decoded Quantum Interferometry for approximate optimization
- Quartic quantum speedups for planted infere

Quartic quantum speedups for community detection

Alexander Schmidhuber* MIT

Alexander Zlokapa† MIT

October 9, 2025

Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

- Verifiable Quantum Advantage without Structure (on the inputs)!
- Using a similar framework (Regev's reduction), Decoded Quantum Interferometry for approximate optimization
- Quartic quantum speedups for planted inference

Where is quantum computing?

Answer: it's an exciting time

In theory

New speedups!

With new caveats?

- Verifiable Quantum Advantage without Structure (on the inputs)!
- Using a similar framework (Regev's reduction), Decoded Quantum Interferometry for approximate optimization
- Quartic quantum speedups for planted inference

Plan for today

Welcome Tea

9:00-10:00 Tutorial

Coffee break ☕

10:30-12:00 Breaking Quantum Speedups

Lunch 🍽️

1:30-3:00 Making Quantum Speedups I

Coffee break ☕

3:30-4:30 Making Quantum Speedups II

Plan for today

Welcome Tea

9:00-10:00 Tutorial

Coffee break ☕

10:30-12:00 Breaking Quantum Speedups

Lunch 🍽️

1:30-3:00 Making Quantum Speedups I

Coffee break ☕

3:30-4:30 Making Quantum Speedups II

Plan for today

Welcome Tea

9:00-10:00 **Tutorial** You are here ★

Coffee break ☕

10:30-12:00 **Breaking Quantum Speedups**

Lunch 🍽️

1:30-3:00 **Making Quantum Speedups I**

Coffee break ☕

3:30-4:30 **Making Quantum Speedups II**

Plan for today

Welcome Tea

9:00-10:00 Tutorial — You are here ★

Coffee break ☕

10:30-12:00 Breaking Quantum Speedups — DQI (Robin), planted inference (William)

Lunch 🍽️

1:30-3:00 Making Quantum Speedups I

Coffee break ☕

3:30-4:30 Making Quantum Speedups II

Plan for today

Welcome Tea

9:00-10:00 Tutorial

You are here ★

Coffee break ☕

10:30-12:00 Breaking Quantum Speedups

DQI (Robin), planted inference (William)

Lunch 🍽️

Verifiable quantum advantage (Soumik),

1:30-3:00 Making Quantum Speedups I

Coffee break ☕

3:30-4:30 Making Quantum Speedups II

Plan for today

Welcome Tea

9:00-10:00 Tutorial — You are here ★

Coffee break ☕

10:30-12:00 Breaking Quantum Speedups — DQI (Robin), planted inference (William)

Lunch 🍽️

1:30-3:00 Making Quantum Speedups I — Verifiable quantum advantage (Soumik), Quantum MCMC (Anthony)

Coffee break ☕

3:30-4:30 Making Quantum Speedups II

Plan for today

Welcome Tea

9:00-10:00 **Tutorial** — You are here ★

Coffee break ☕

10:30-12:00 **Breaking Quantum Speedups** — DQI (Robin), planted inference (William)

Lunch 🍽️

1:30-3:00 **Making Quantum Speedups I** — Verifiable quantum advantage (Soumik), Quantum MCMC (Anthony)

Coffee break ☕

3:30-4:30 **Making Quantum Speedups II** — Characters of S_n (Vojtěch)

Exponential Speedups from the Quantum Fourier Transform

Act I: period finding

Exponential Speedups from the Quantum Fourier Transform

Act I: period finding

Act II: building cryptography on the hardness of lattice problems

Exponential Speedups from the Quantum Fourier Transform

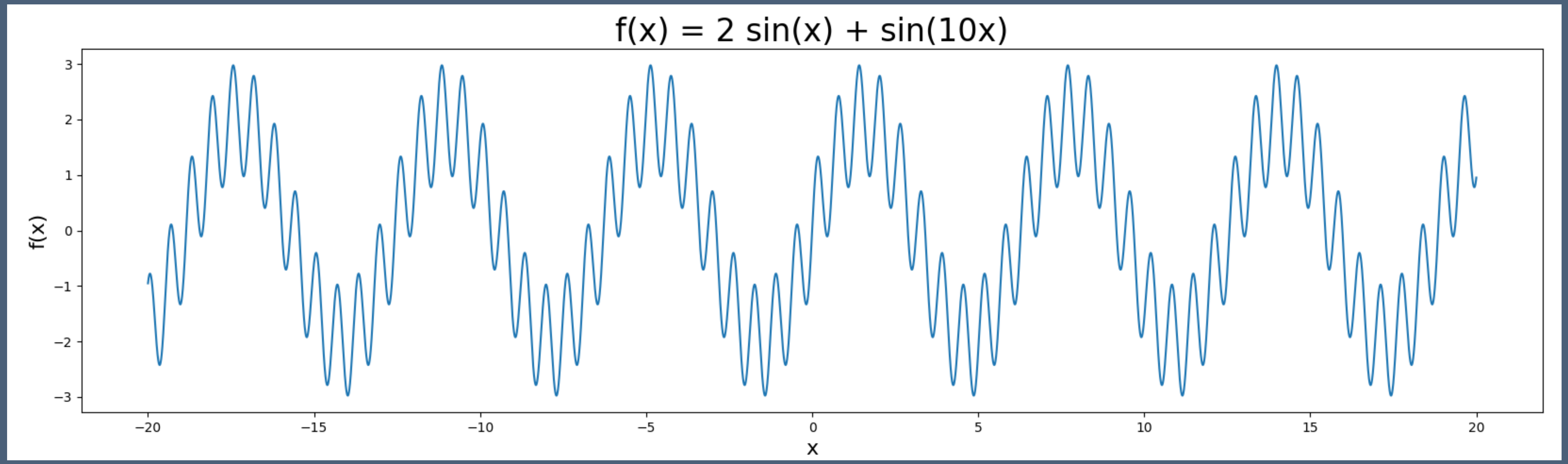
Act I: period finding

Act II: building cryptography on the hardness of lattice problems

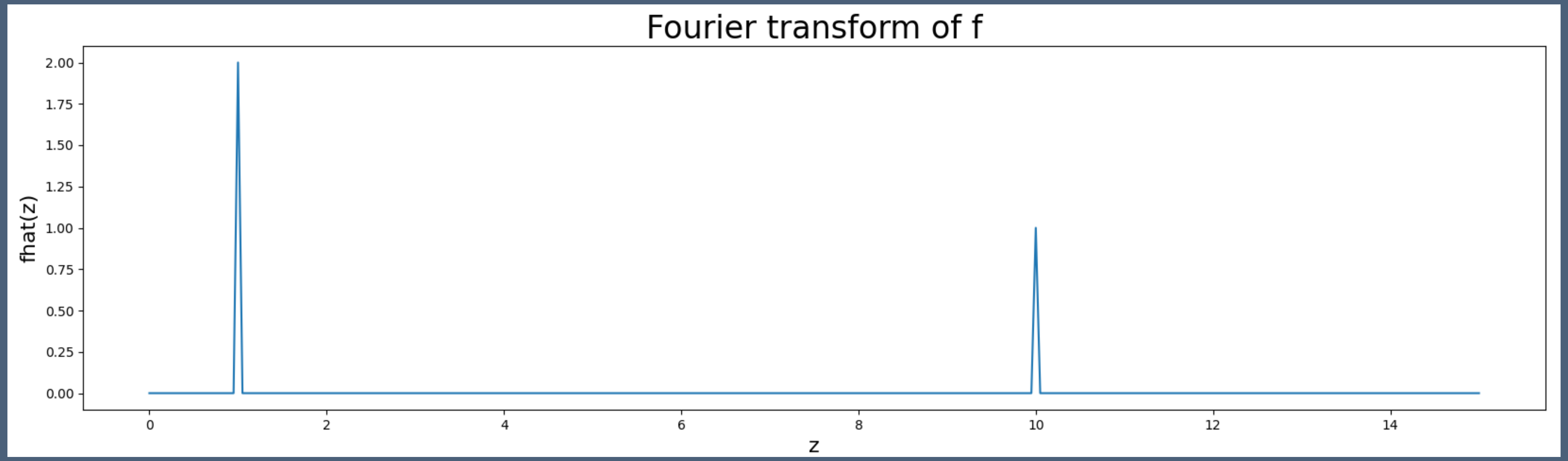
Act III: new quantum algorithms from Regev's reduction

Quantum Fourier Transform

Classical Fourier transform: extracts information about a <u>signal's</u> periodicity



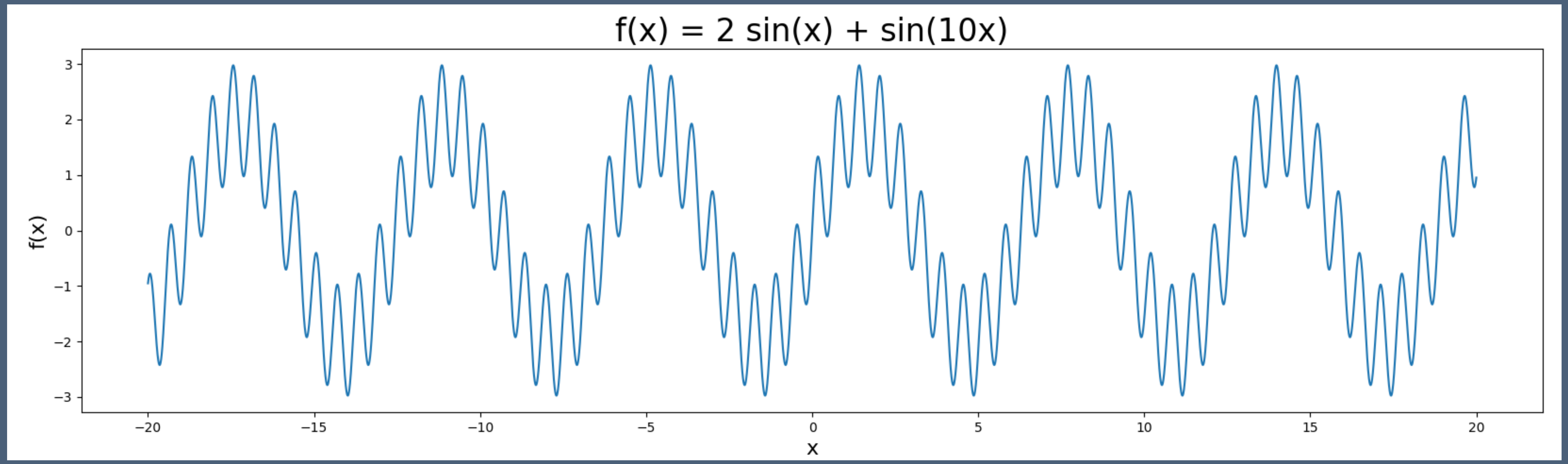
$f(x)$

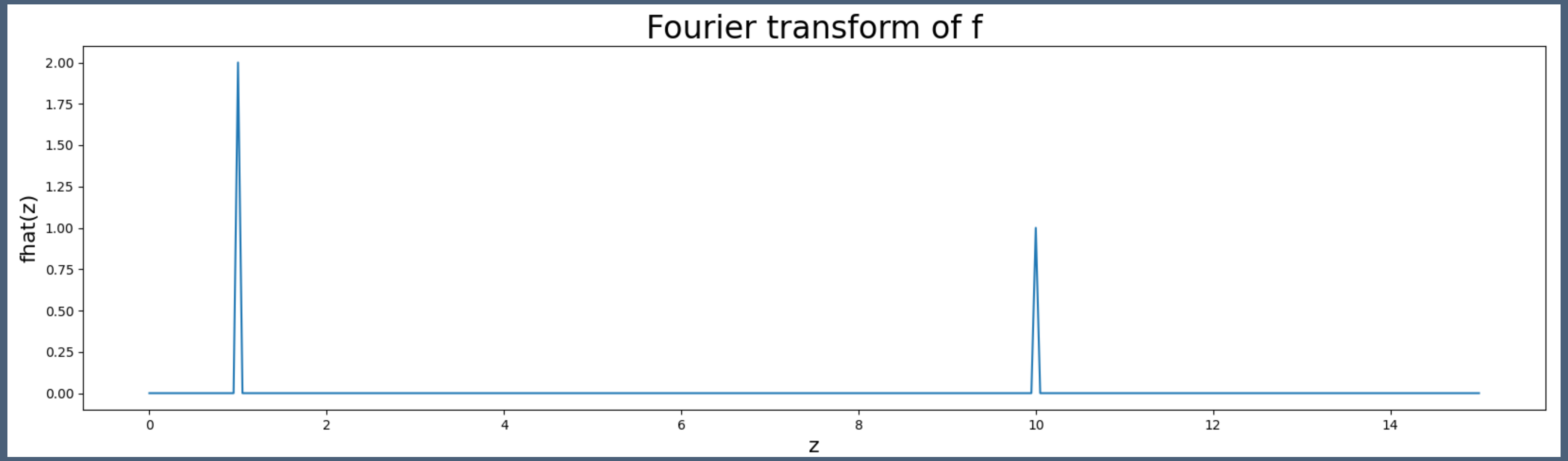


$\hat{f}(z)$

Quantum Fourier Transform

Classical Fourier transform: extracts information about a <u>signal's</u> periodicity



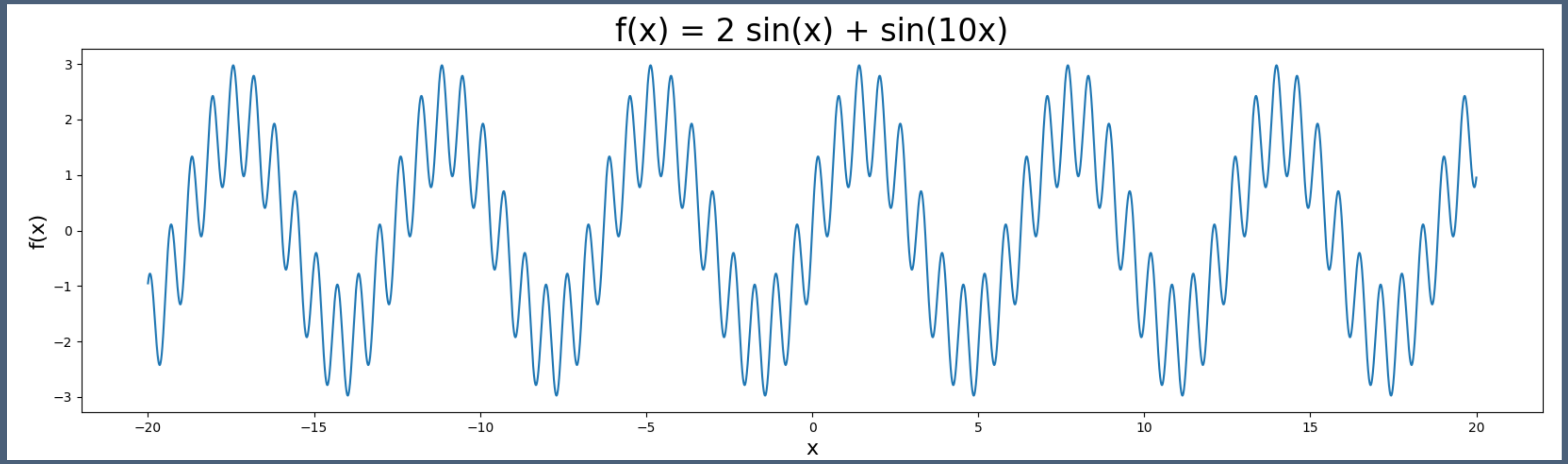


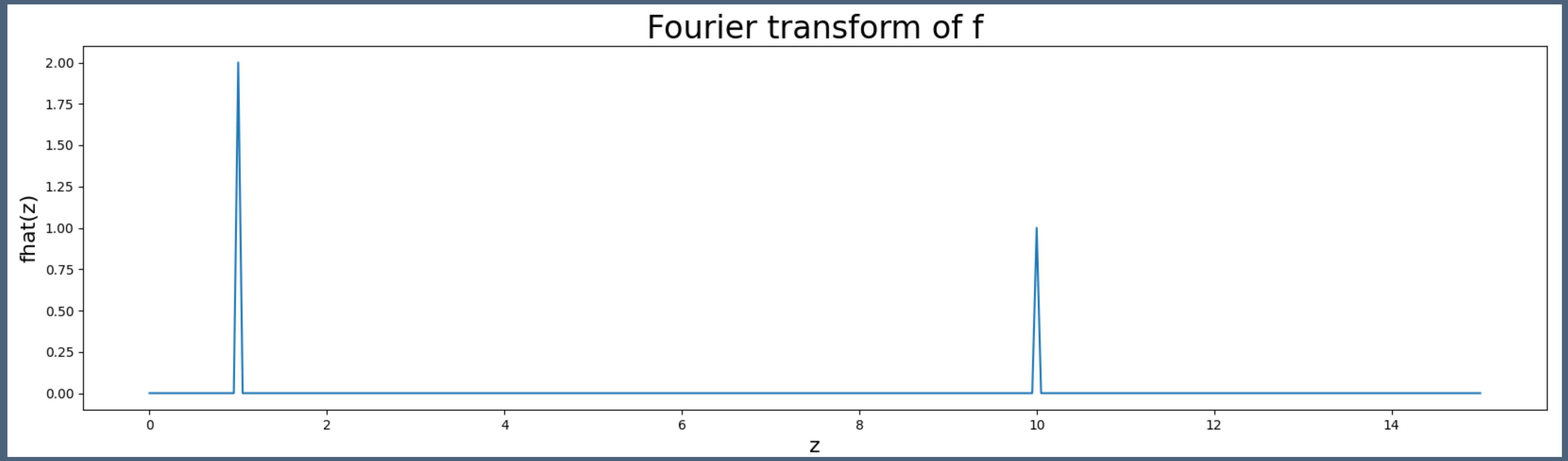
$$f(x) \longrightarrow \sum_x f(x)\,|x\rangle$$

$$\hat{f}(z) \longrightarrow \sum_z \hat{f}(z)\,|z\rangle$$

Quantum Fourier Transform

Classical Fourier transform: extracts information about a <u>signal's</u> periodicity





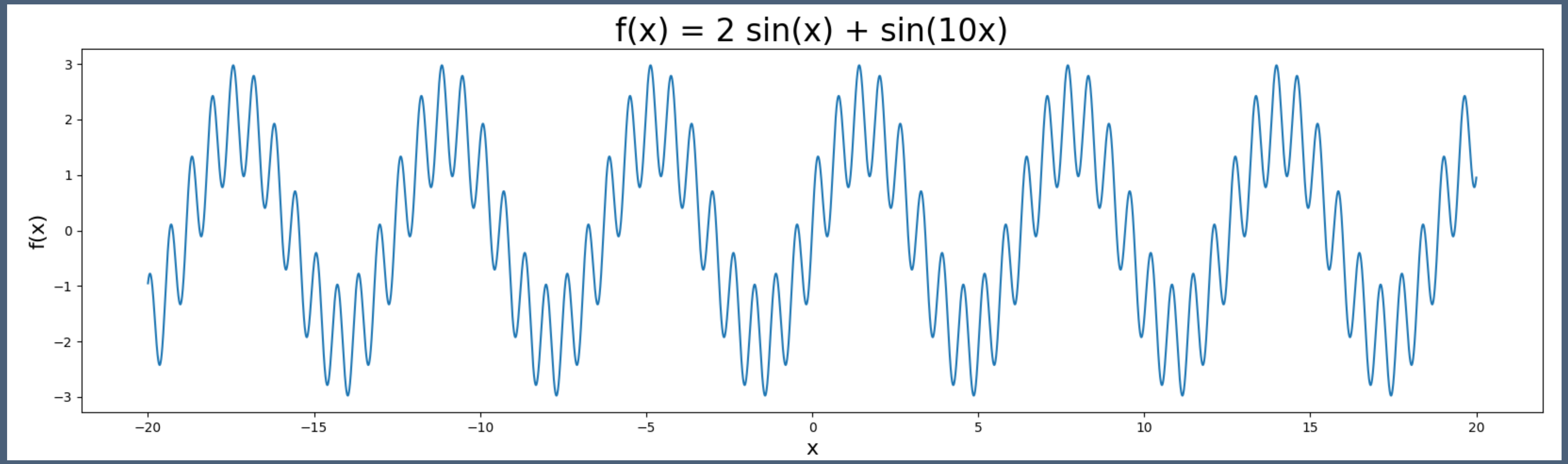
$$f(x) \qquad\qquad \hat{f}(z)$$

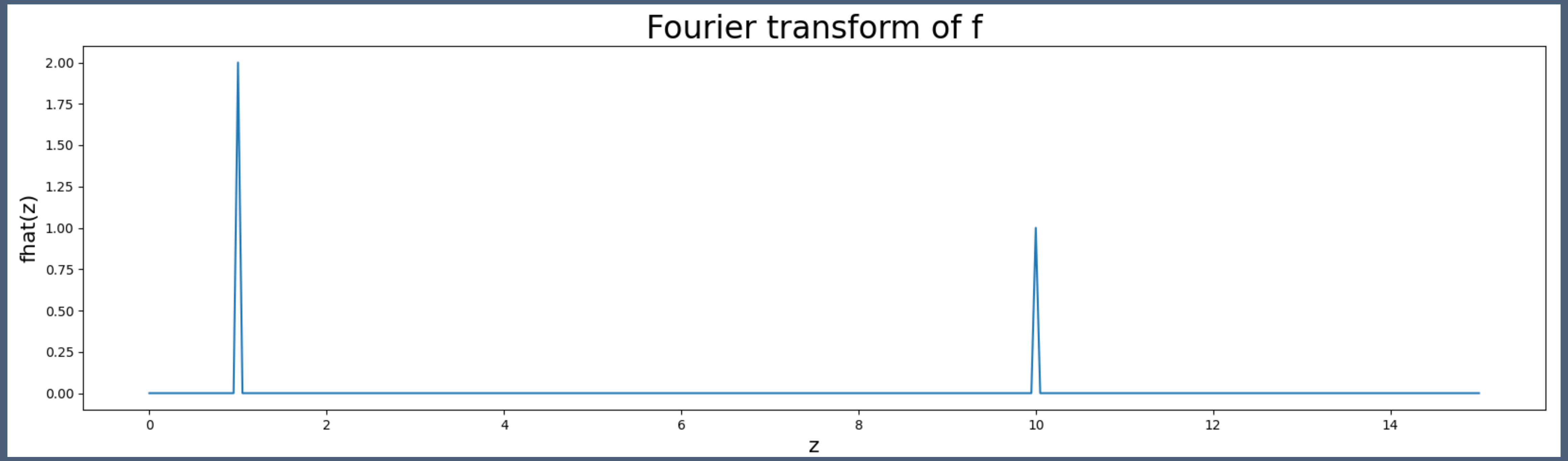
$$\sum_x f(x)\,|x\rangle \longrightarrow \sum_z \hat{f}(z)\,|z\rangle$$

Quantum Fourier transform: extracts information about a <u>quantum state's</u> periodicity

Quantum Fourier Transform

Classical Fourier transform: extracts information about a <u>signal's</u> periodicity





$$f(x) \longrightarrow \hat{f}(z)$$

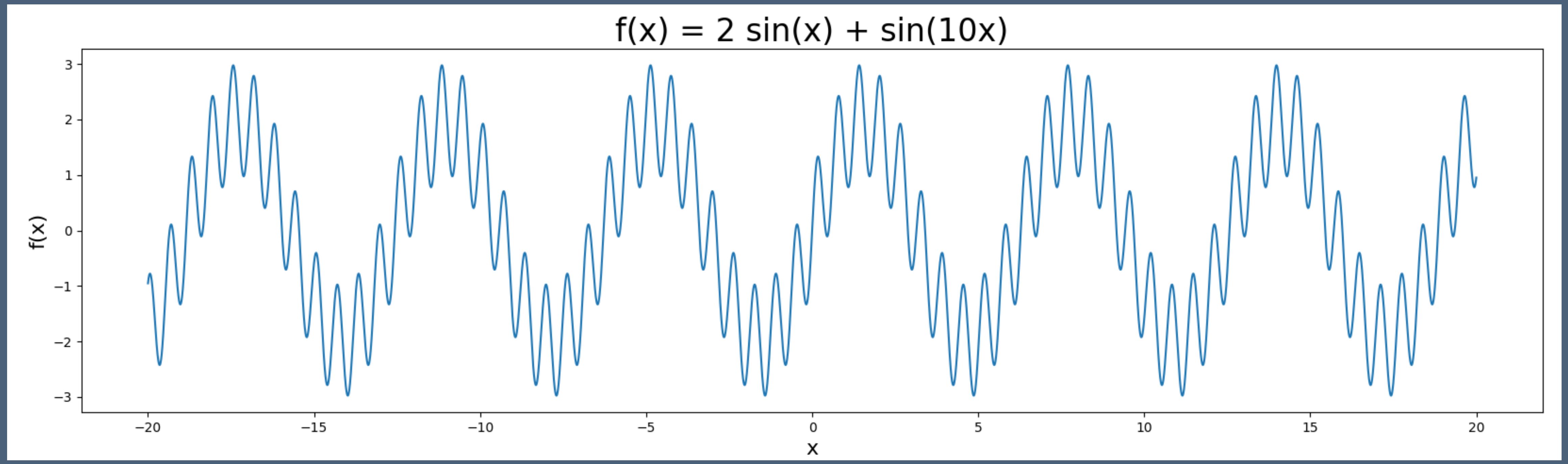
$$\sum_x f(x)\,|x\rangle \longrightarrow \sum_z \hat{f}(z)\,|z\rangle$$

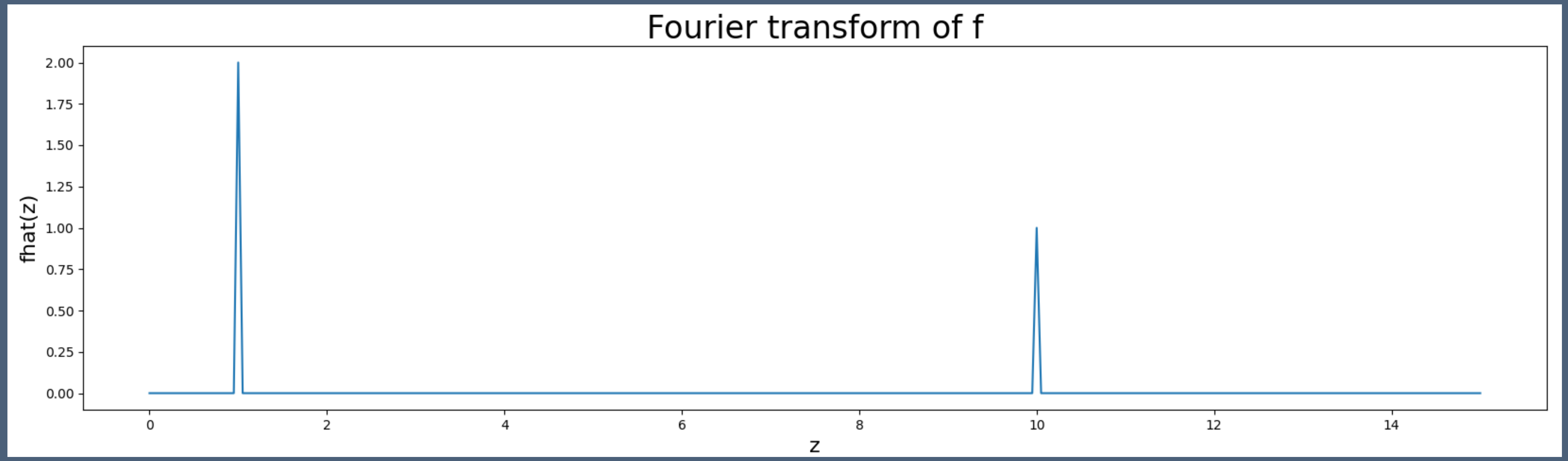
Quantum Fourier transform: extracts information about a <u>quantum state's</u> periodicity

- Classical FT: explicitly stores N values of f, computes $\hat{f}$ in $O(N \log N)$ time

Quantum Fourier Transform

Classical Fourier transform: extracts information about a <u>signal's</u> periodicity





$f(x)$ $\qquad$ $\hat{f}(z)$

$$\sum_x f(x)\,|x\rangle \longrightarrow \sum_z \hat{f}(z)\,|z\rangle$$

Quantum Fourier transform: extracts information about a <u>quantum state's</u> periodicity

- Classical FT: explicitly stores N values of f, computes $\hat{f}$ in $O(N \log N)$ time
- Quantum FT: implicitly stores f in a state on $\log N$ qubits, implicitly computes $\hat{f}$ in $O(\log^2 N)$ time

Exponential Speedups from the Quantum Fourier Transform

Act I: period finding

Act II: building cryptography on the hardness of lattice problems

Act III: new quantum algorithms from Regev's reduction

Period Finding

Period Finding

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown but exponentially large period T
  - $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{T}$

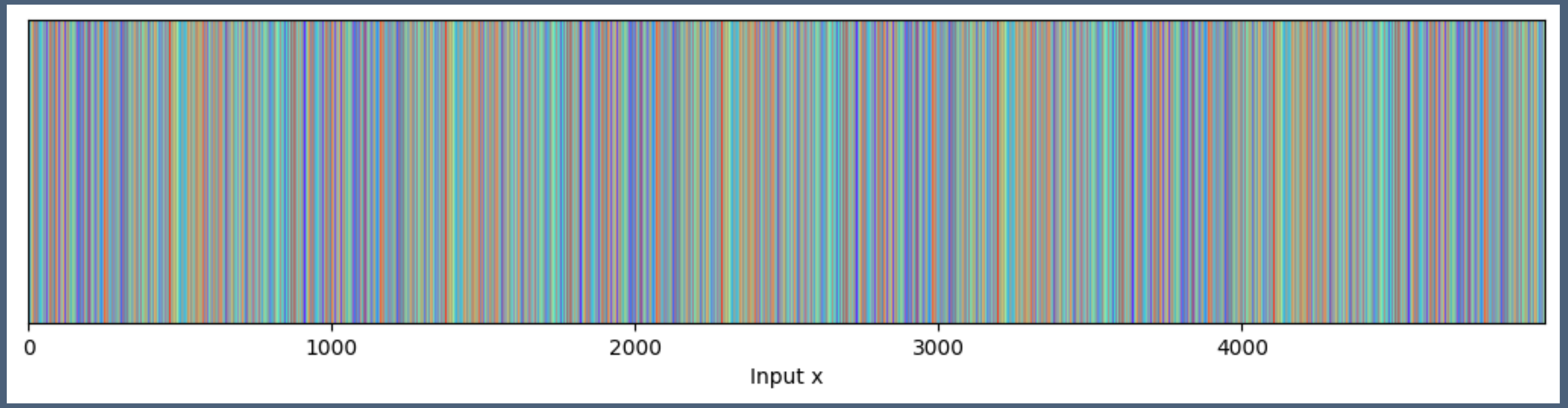
Period Finding

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown but exponentially large period T
  - $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{T}$
- Theorem: can quantumly recover T in $\mathrm{poly}(\log T)$ time

Period Finding

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown but exponentially large period T
  - $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{T}$
- Theorem: can quantumly recover T in $\mathsf{poly}(\log T)$ time

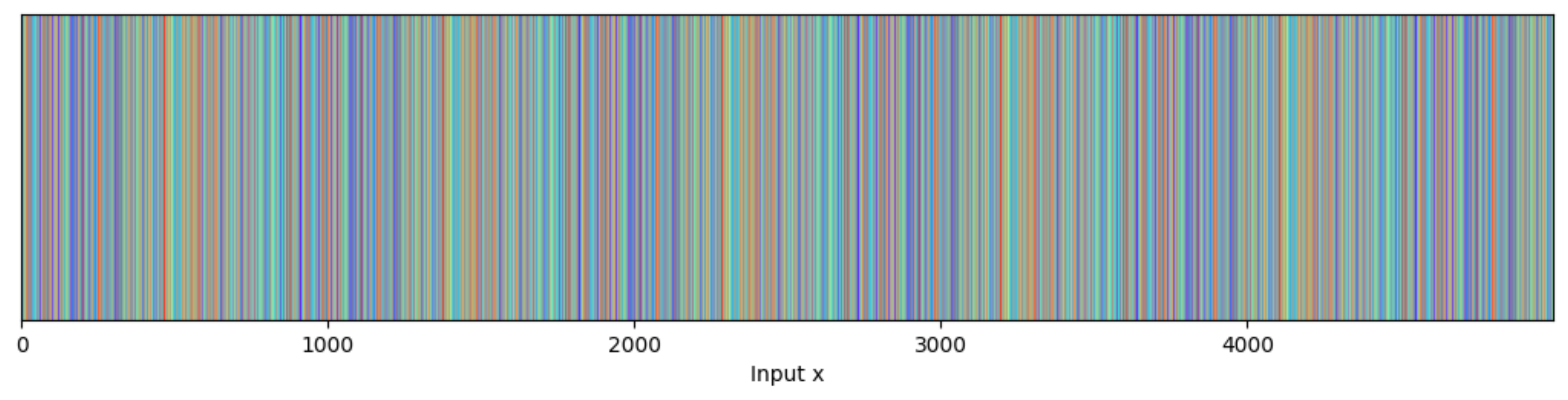
1. Prepare the superposition $\displaystyle\sum_{x=1}^{\mathsf{poly}(T)} |x\rangle |f(x)\rangle$



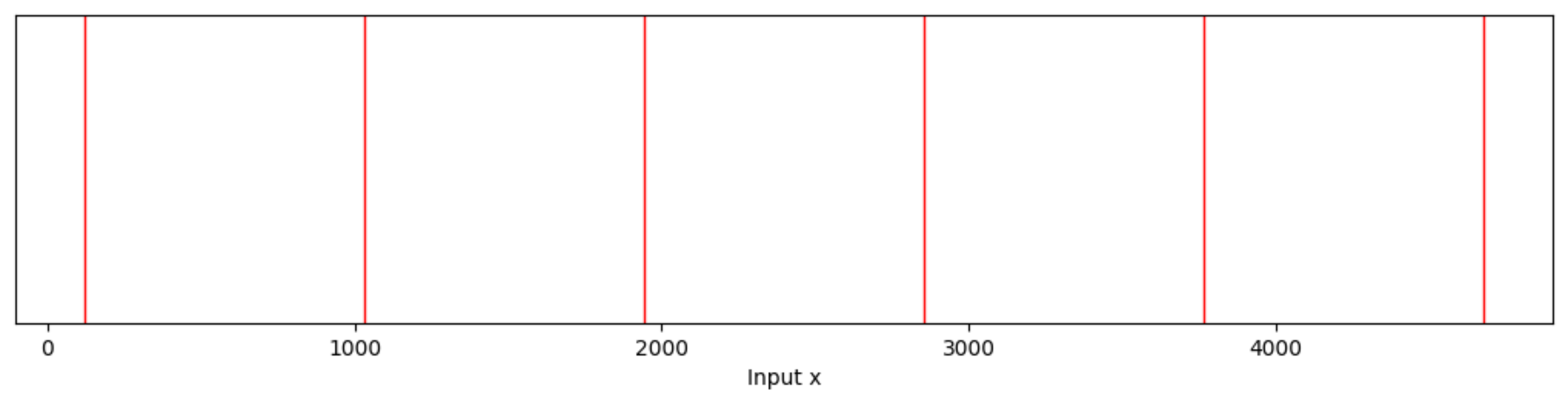
Period Finding

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown but exponentially large period T
  - $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{T}$
- Theorem: can quantumly recover T in **poly**$(\log T)$ time

1. Prepare the superposition $\sum_{x=1}^{\text{poly}(T)} |x\rangle |f(x)\rangle$



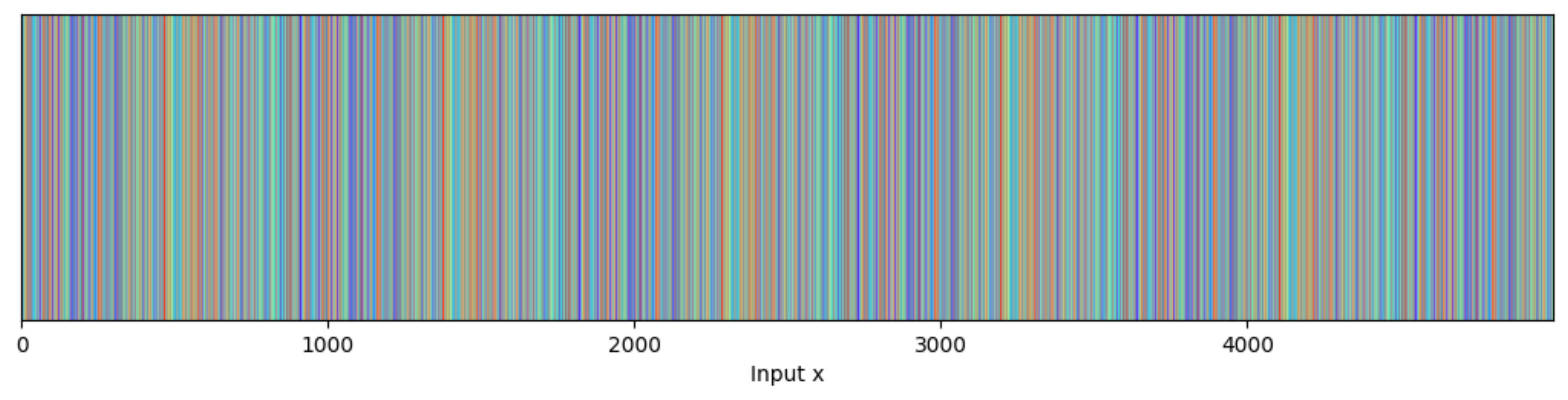
2. Measure ("condition on") the second register → signal $|x_0\rangle + |x_0 + T\rangle + |x_0 + 2T\rangle + \ldots$ has period T



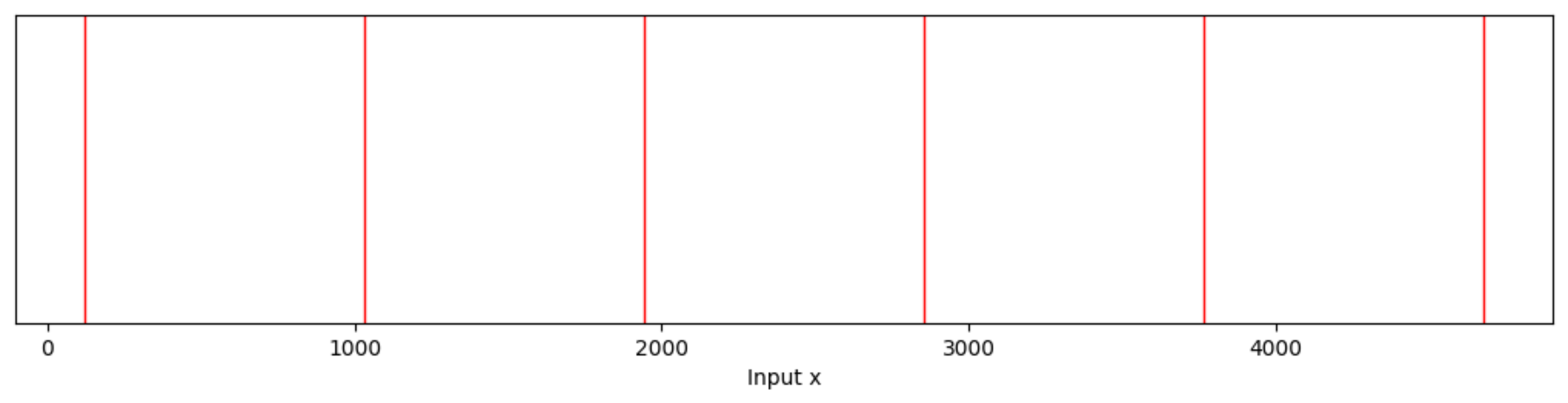
Period Finding

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown but exponentially large period T
  - $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{T}$
- Theorem: can quantumly recover T in **poly**$(\log T)$ time

1. Prepare the superposition $\sum_{x=1}^{\text{poly}(T)} |x\rangle |f(x)\rangle$



2. Measure ("condition on") the second register → signal $|x_0\rangle + |x_0 + T\rangle + |x_0 + 2T\rangle + \ldots$ has period T



3. QFT and measure the state → recover T

From Period Finding to Factoring

From Period Finding to Factoring

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown but exponentially large period T
  - $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{T}$
- Theorem: can quantumly recover T in **poly**$(\log T)$ time
- **Corollary (Shor): factoring (and discrete logarithm) is in BQP**

From Period Finding to Factoring

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown but exponentially large period T
  - $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{T}$
- Theorem: can quantumly recover T in **poly**$(\log T)$ time
- **Corollary (Shor): factoring (and discrete logarithm) is in BQP**

1. To factor N: consider $f(x) = a^{2x} \bmod N$ for randomly chosen a

From Period Finding to Factoring

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown but exponentially large period T
  - $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{T}$
- Theorem: can quantumly recover T in **poly**$(\log T)$ time
- **Corollary (Shor): factoring (and discrete logarithm) is in BQP**

1. To factor N: consider $f(x) = a^{2x} \bmod N$ for randomly chosen a
2. Period T is such that $a^{2T} \equiv (a^T)^2 \equiv 1 \pmod{N} \Rightarrow N \mid (a^T - 1)(a^T + 1)$

From Period Finding to Factoring

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown but exponentially large period T
  - $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{T}$
- Theorem: can quantumly recover T in **poly**$(\log T)$ time
- **Corollary (Shor): factoring (and discrete logarithm) is in BQP**

1. To factor N: consider $f(x) = a^{2x} \bmod N$ for randomly chosen a
2. Period T is such that $a^{2T} \equiv (a^T)^2 \equiv 1 \pmod{N} \Rightarrow N \mid (a^T - 1)(a^T + 1)$
3. With good probability: $\gcd(a^T - 1, N)$ is a nontrivial factor of N

From Period Finding to Factoring

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown but exponentially large period T
  - $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{T}$
- Theorem: can quantumly recover T in poly(log T) time
- **Corollary (Shor): factoring (and discrete logarithm) is in BQP**

Implication: large-scale quantum computers would break essentially all 20th century public-key encryption (RSA, Diffie-Hellman)!

Cryptographic goal: find public-key encryption that is secure against quantum attackers...

1. To factor N: consider $f(x) = a^{2x} \mod N$ for randomly chosen a
2. Period T is such that $a^{2T} \equiv (a^T)^2 \equiv 1 \pmod{N} \Rightarrow N \mid (a^T - 1)(a^T + 1)$
3. With good probability: $\gcd(a^T - 1, N)$ is a nontrivial factor of N

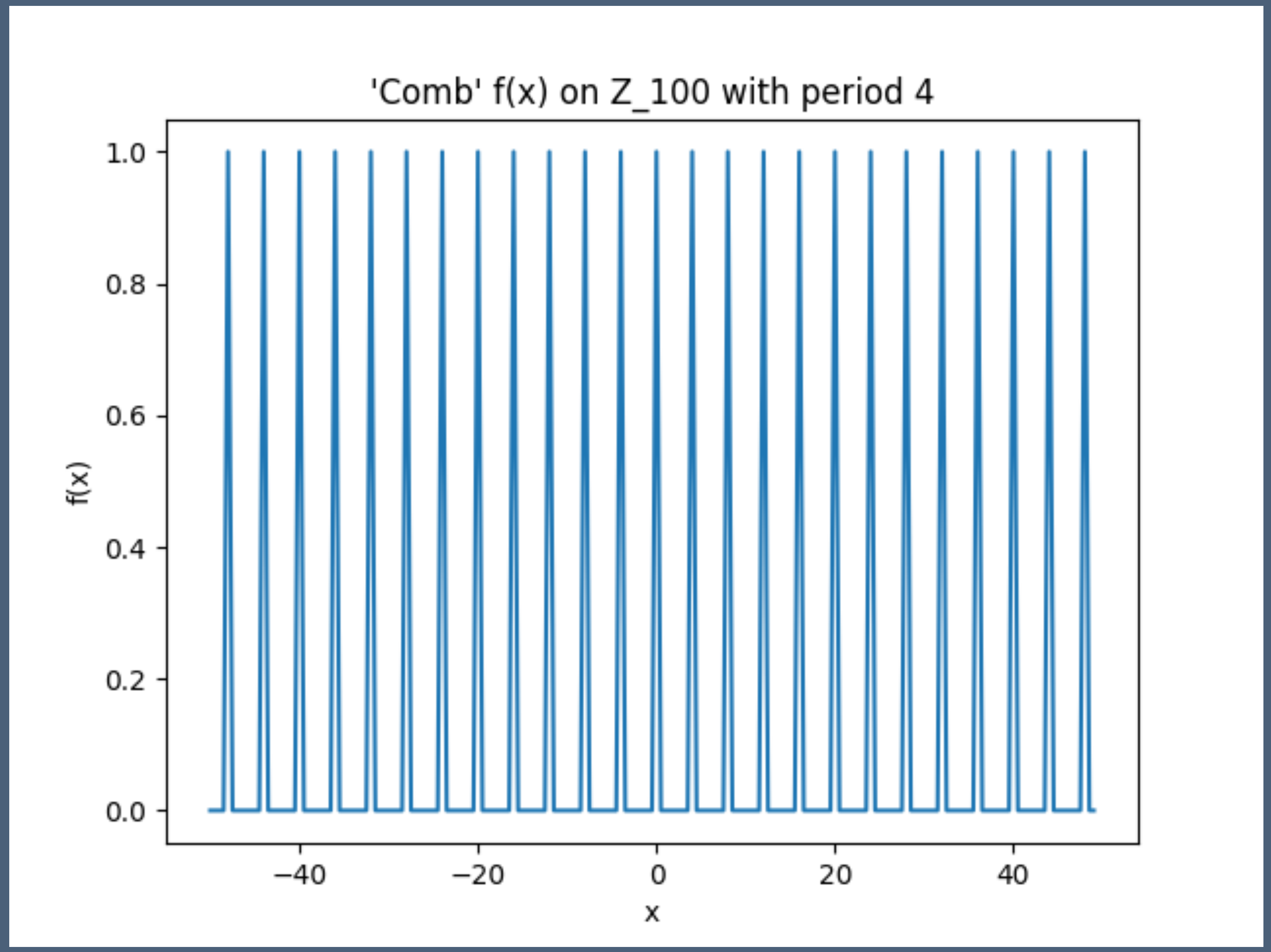
<u>Exponential</u> Speedups from the Quantum Fourier Transform

Act I: period finding

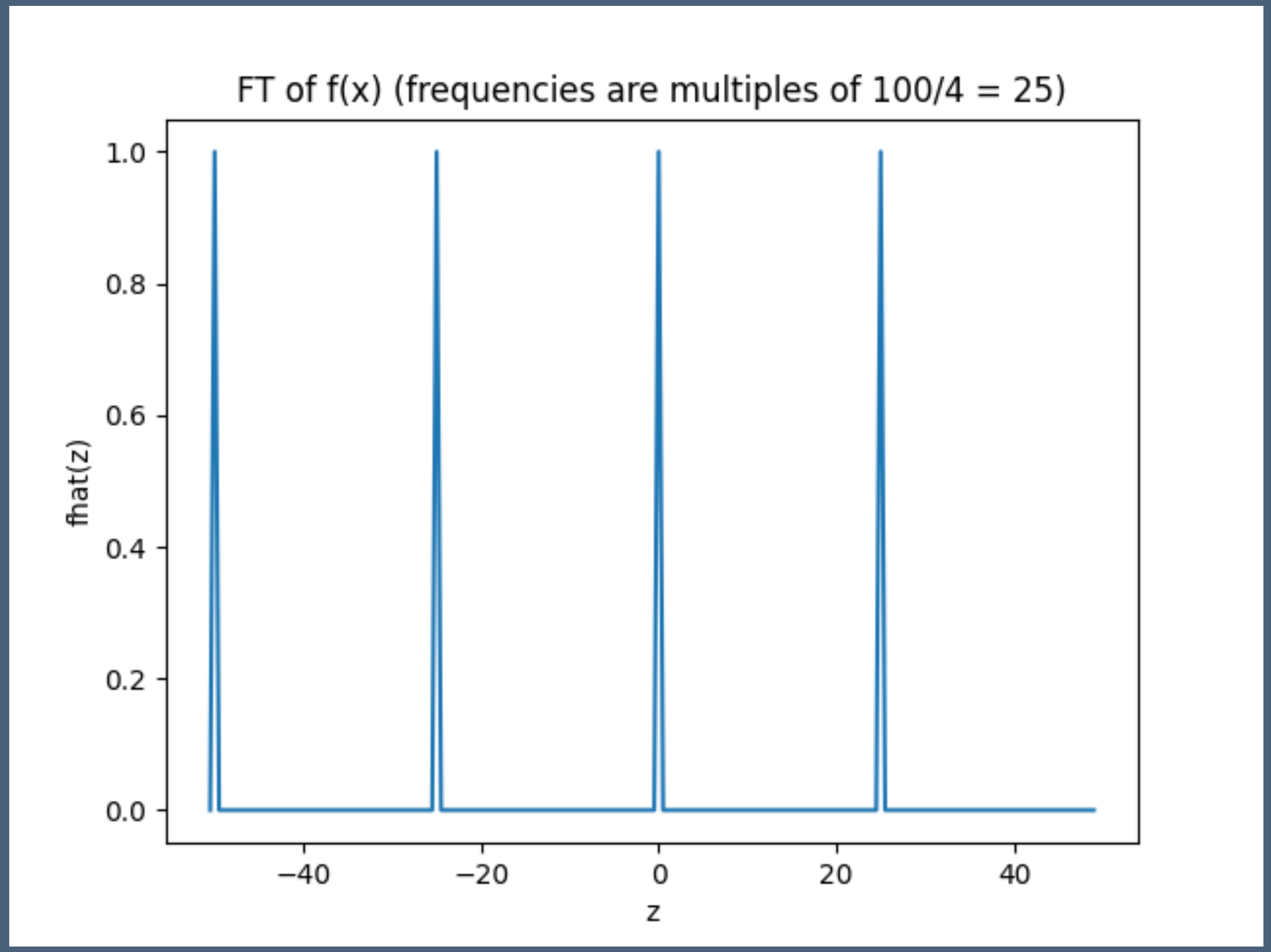
Act II: building cryptography on the hardness of lattice problems

Act III: new quantum algorithms from Regev's reduction

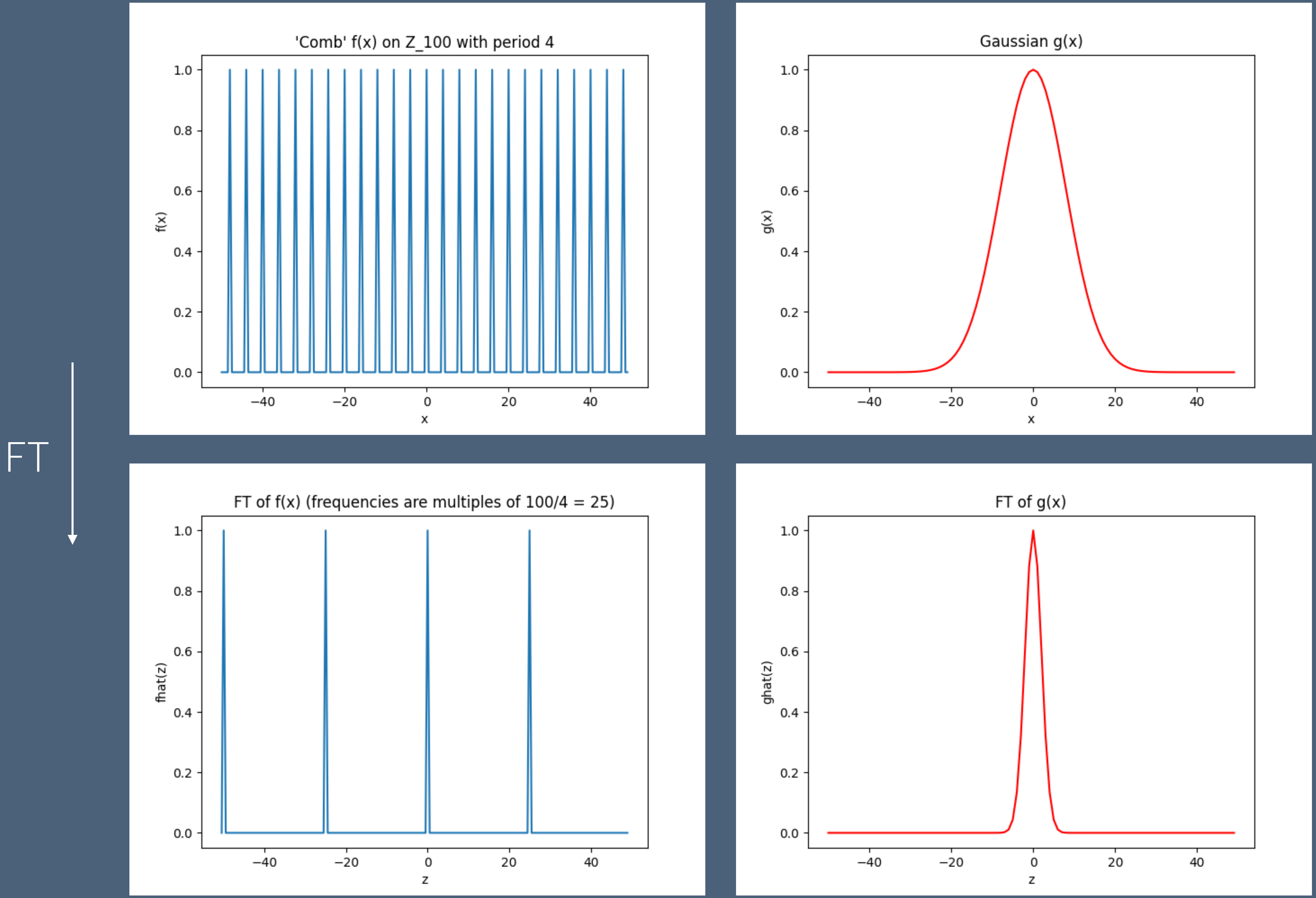
Fourier Convolution Theorem



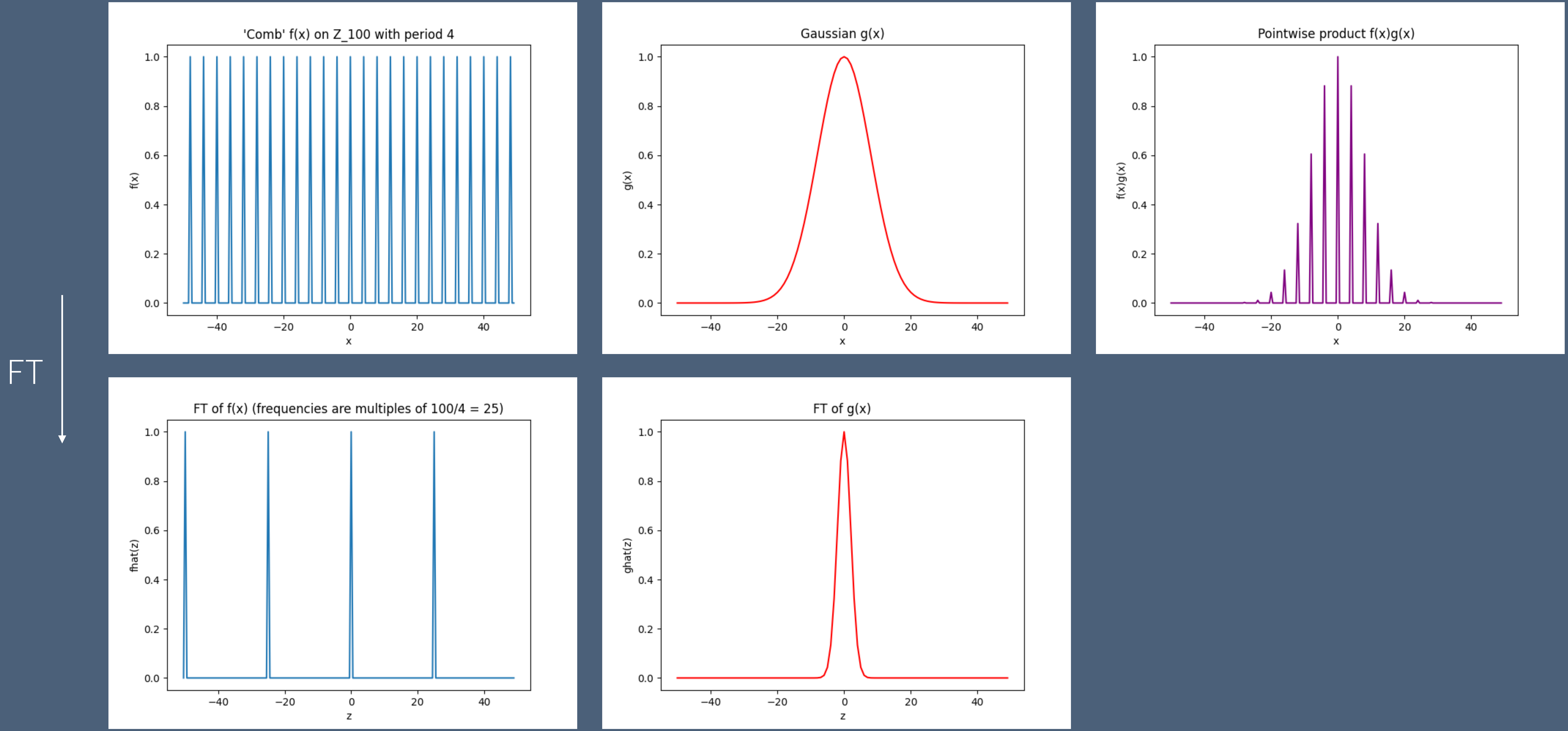
FT



Fourier Convolution Theorem

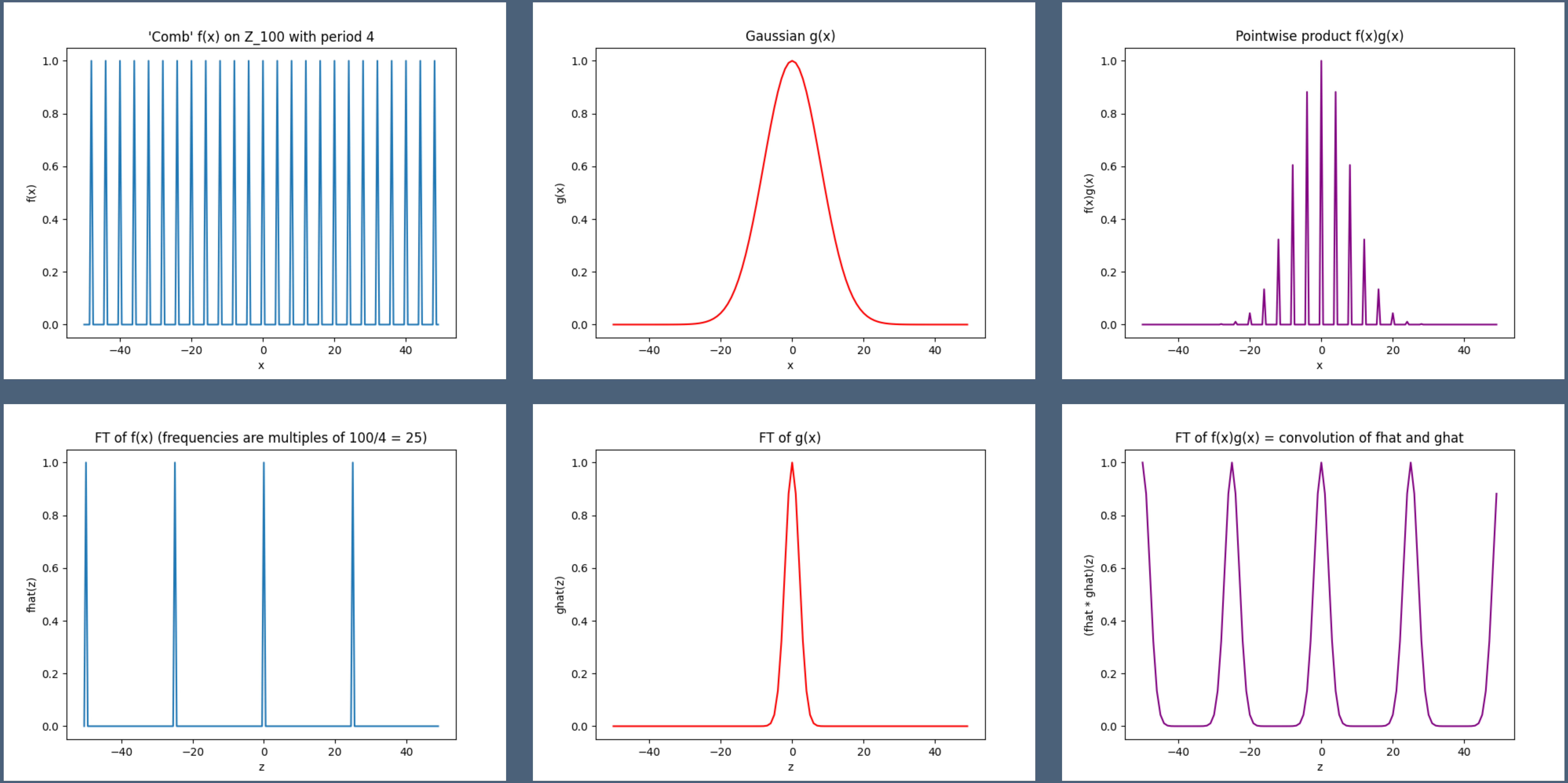


Fourier Convolution Theorem



Fourier Convolution Theorem

FT



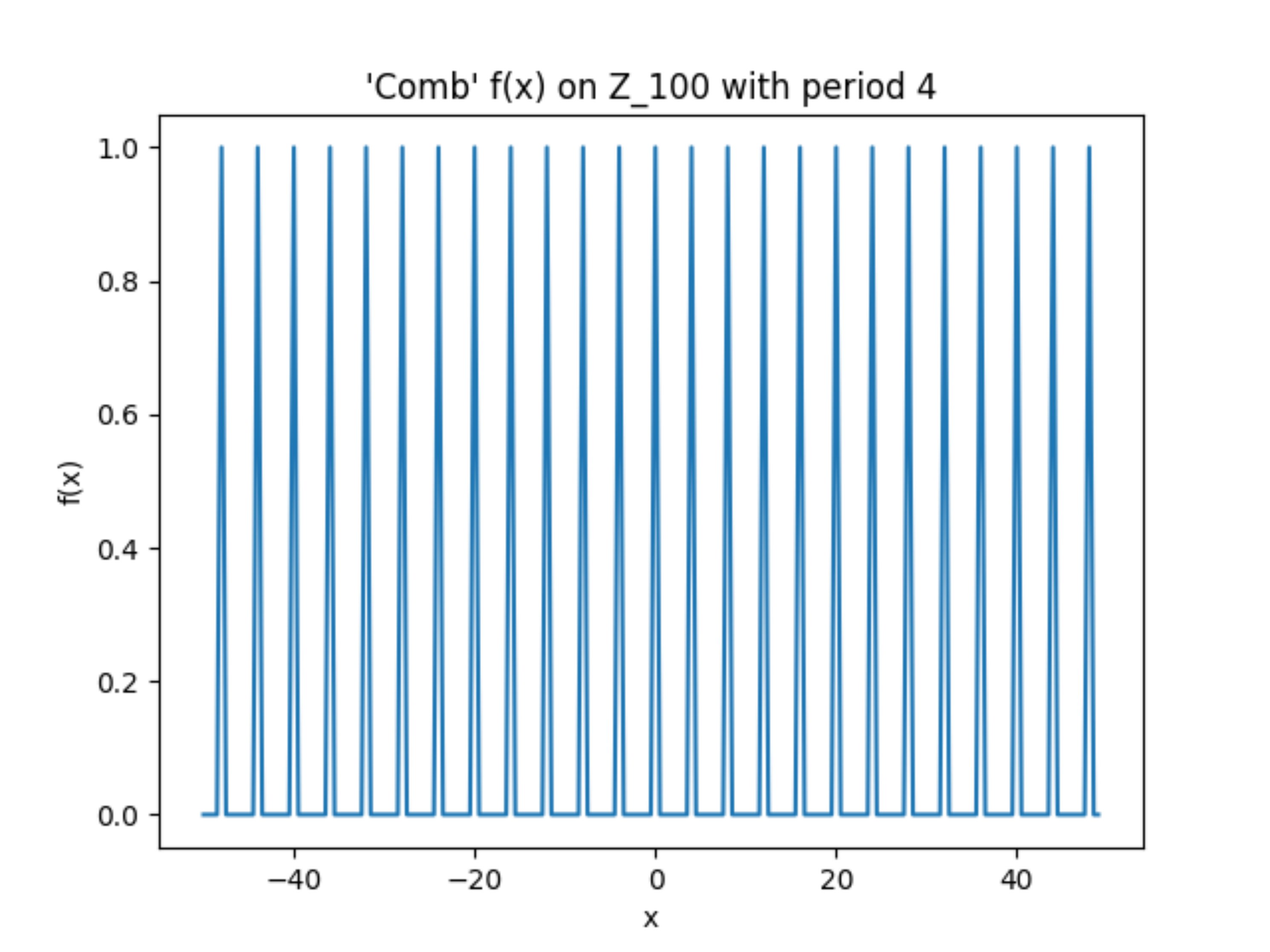
Regev's Reduction

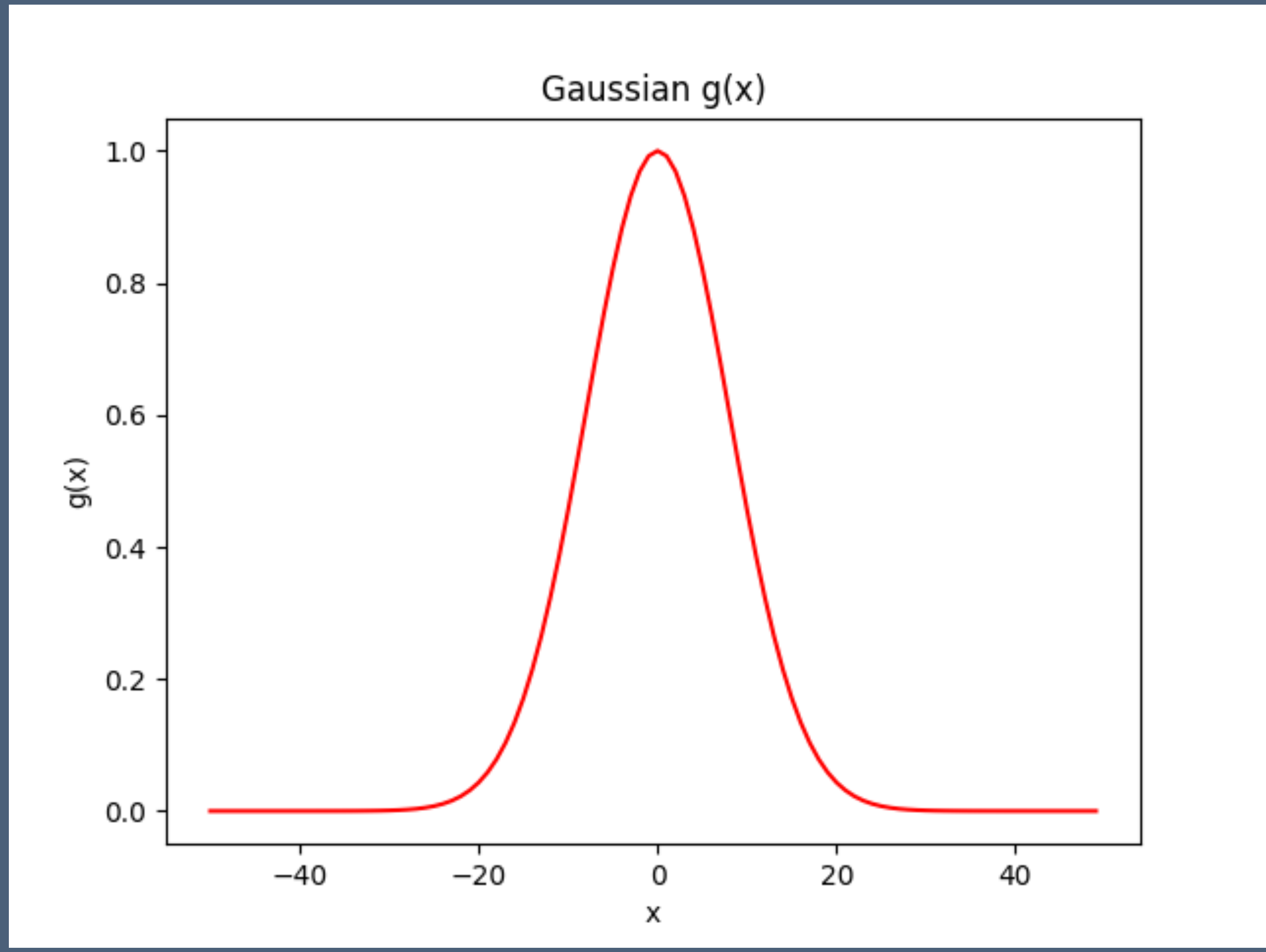
Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

Regev's Reduction

Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

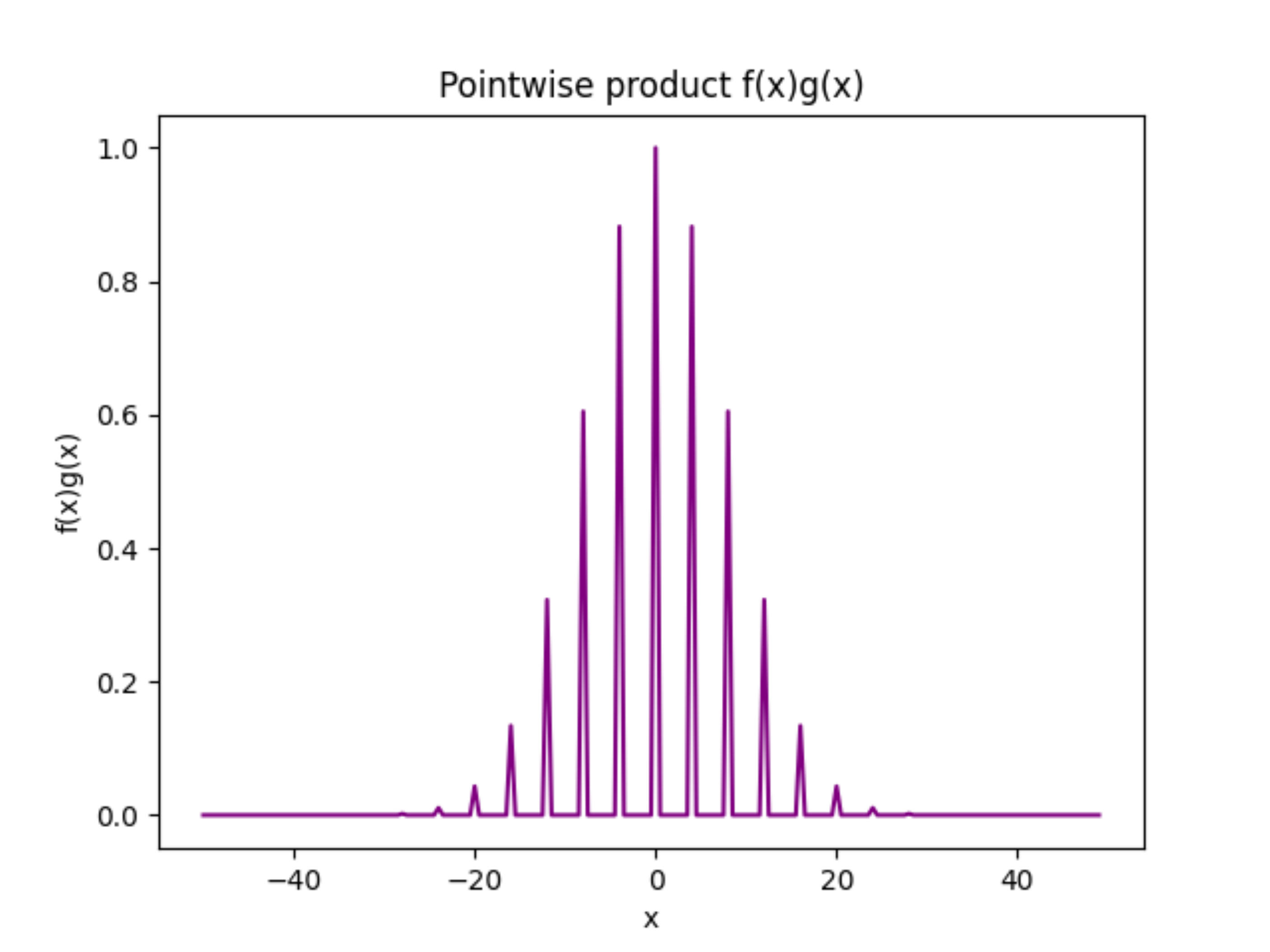
- Let $f(\mathbf{x}) = \mathbb{I}[\mathbf{x} \in \mathscr{C}]$ and $g(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/R^2)$ for suitable R





Regev's Reduction

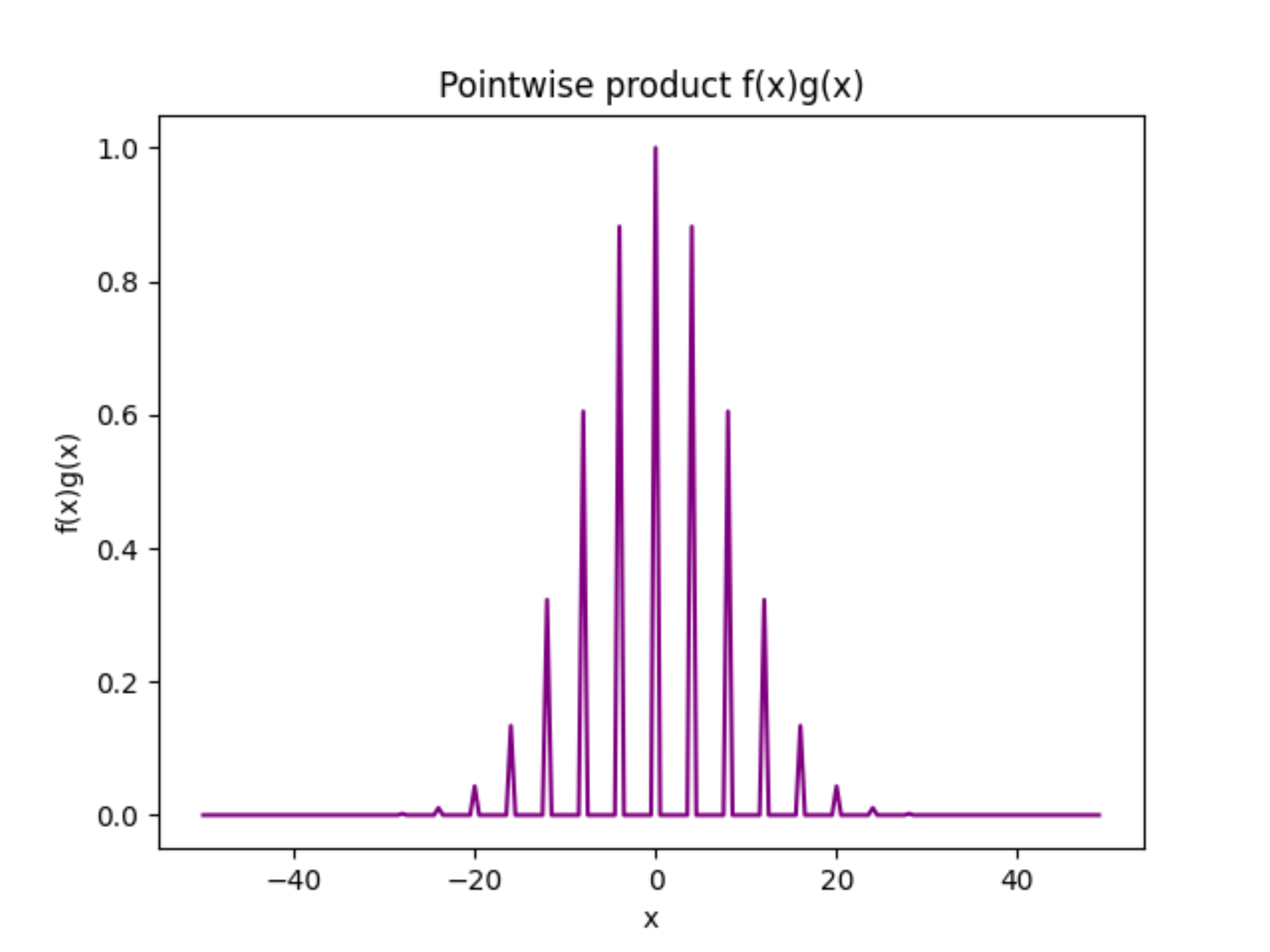
Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0



Regev's Reduction

Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

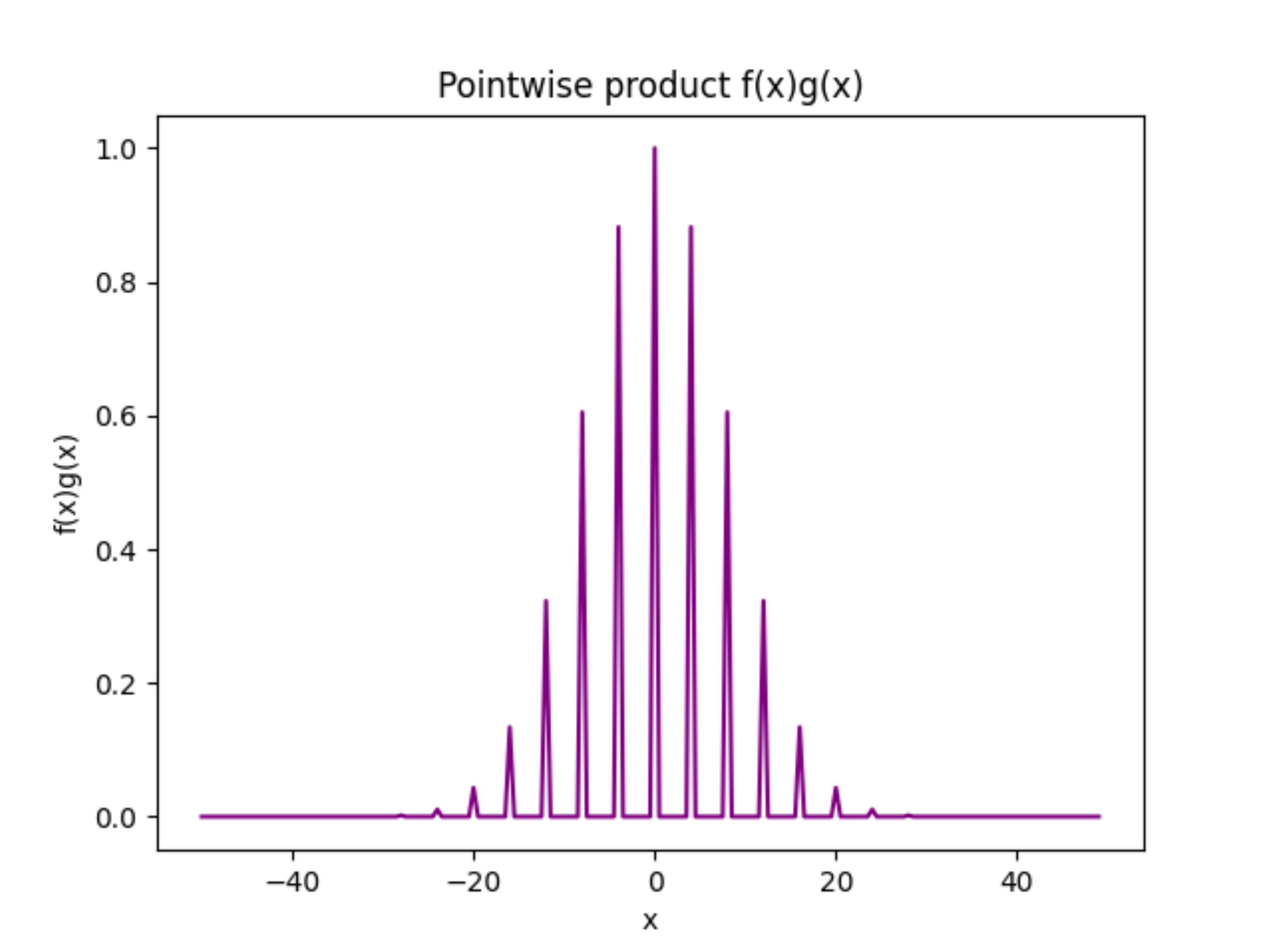
- Let $f(\mathbf{x}) = \mathbb{I}[\mathbf{x} \in \mathscr{C}]$ and $g(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/R^2)$ for suitable R
- Suffices to construct the state $\sum_{\mathbf{x}\in\mathbb{Z}_q^m} f(\mathbf{x})g(\mathbf{x})\,|\,\mathbf{x}\rangle$

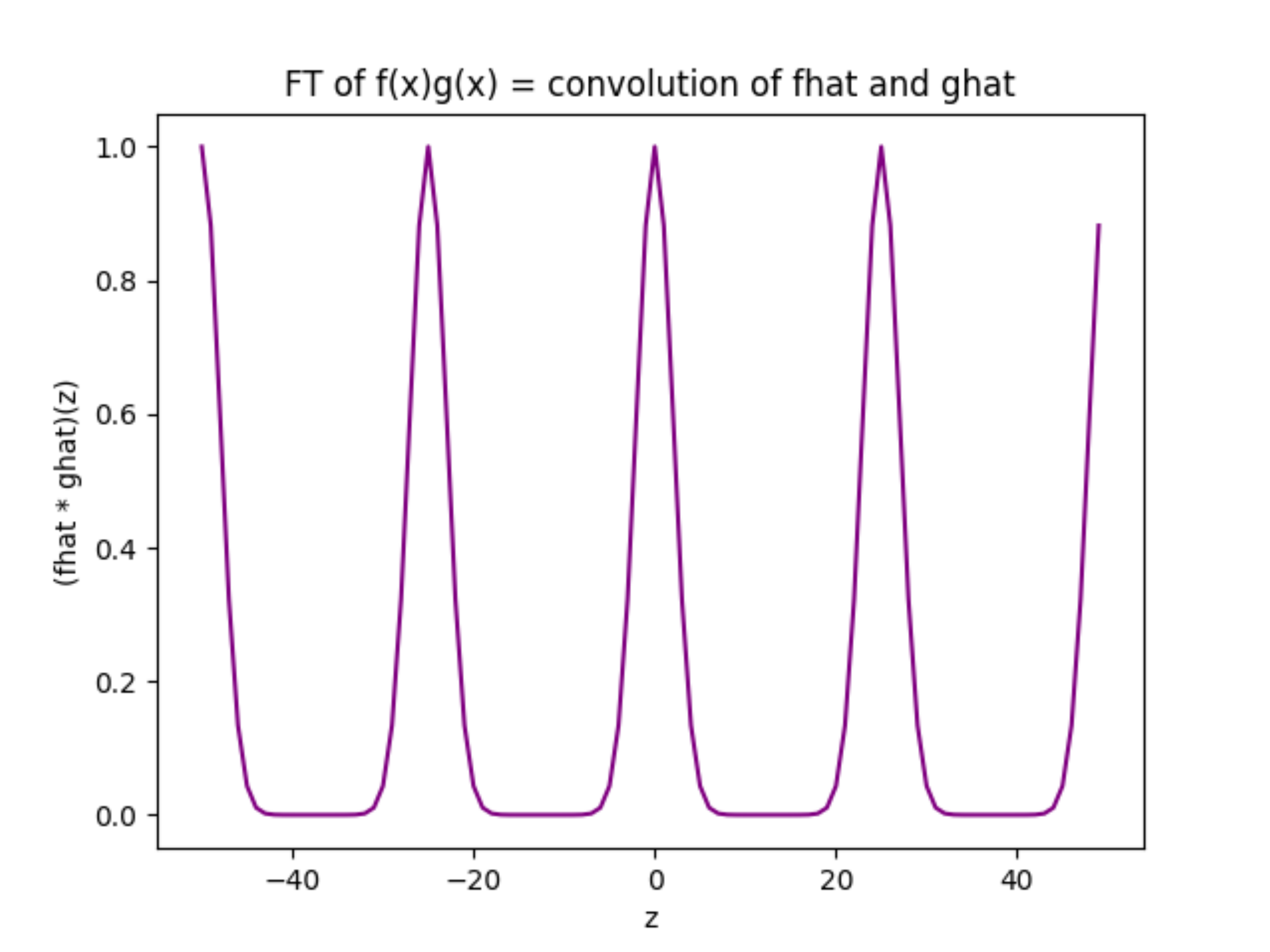


Regev's Reduction

Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

- Let $f(\mathbf{x}) = \mathbb{1}[\mathbf{x} \in \mathscr{C}]$ and $g(\mathbf{x}) = \exp(-\pi \|\mathbf{x}\|^2 / R^2)$ for suitable R
- Suffices to construct the state $\sum_{\mathbf{x} \in \mathbb{Z}_q^m} f(\mathbf{x}) g(\mathbf{x}) \,|\, \mathbf{x} \rangle$
- QFT → suffices to construct the convolution of the Fourier transforms





Regev's Reduction

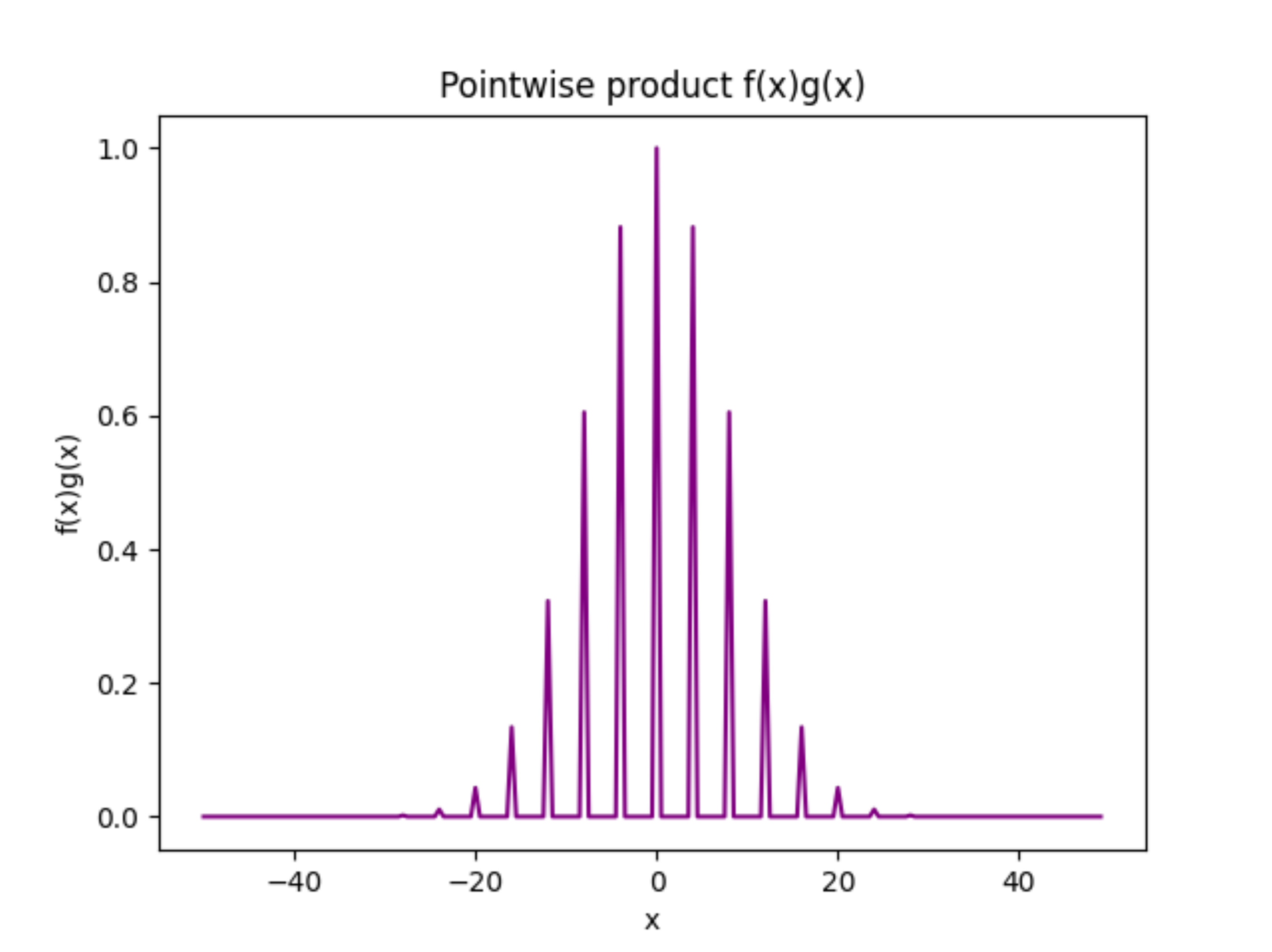
Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

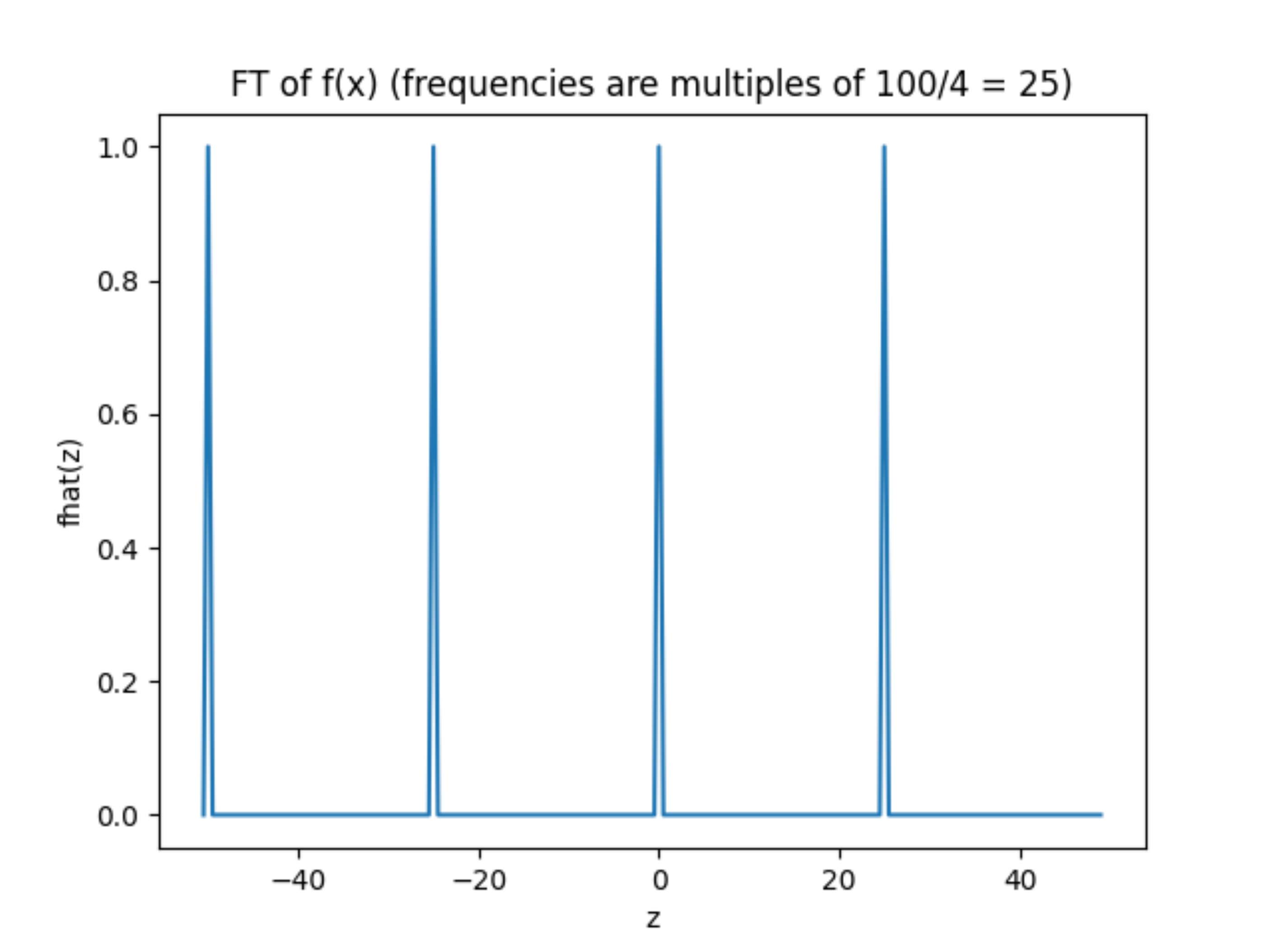
- Let $f(\mathbf{x}) = \mathbb{I}[\mathbf{x} \in \mathscr{C}]$ and $g(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/R^2)$ for suitable R

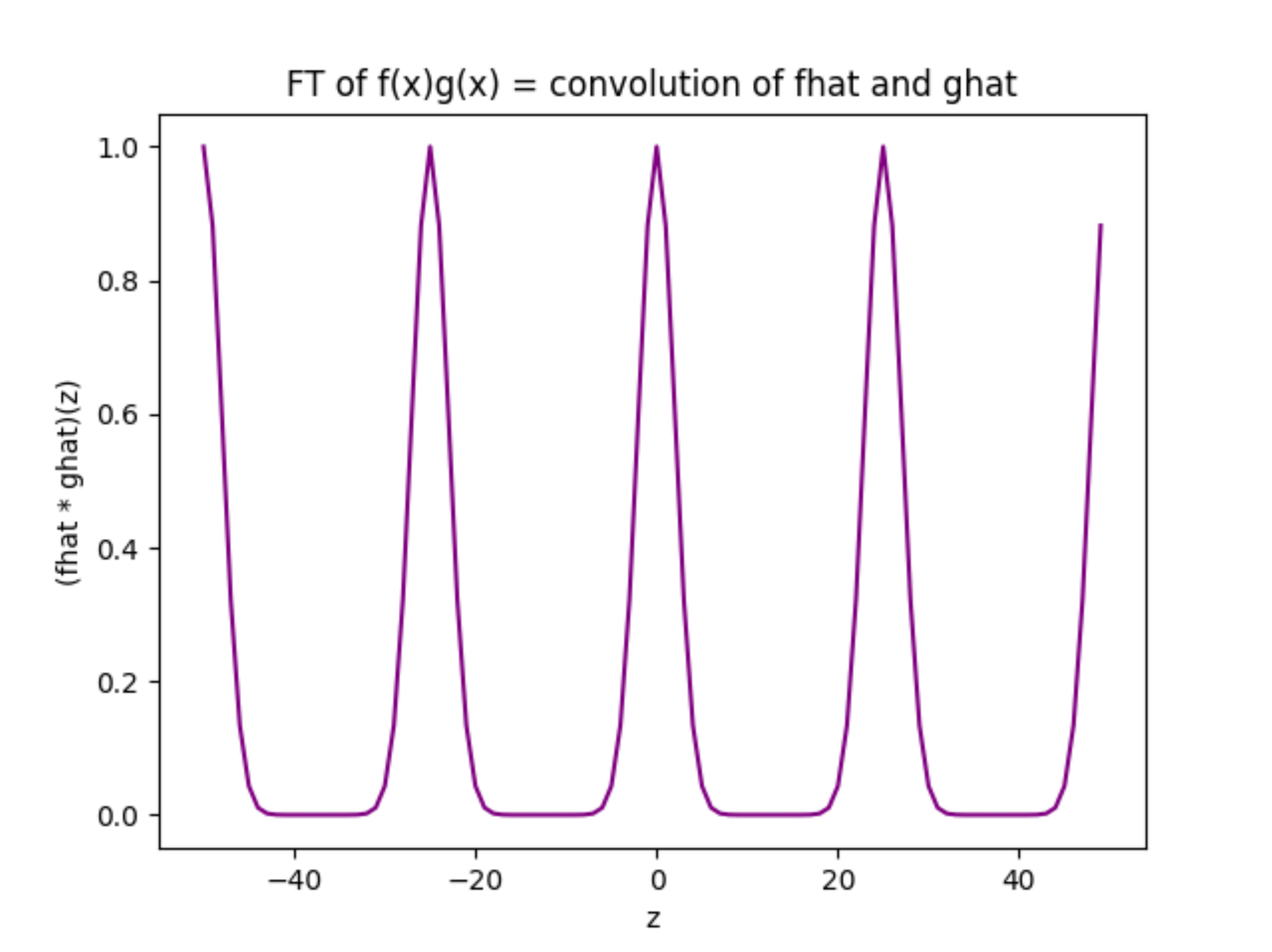
- Suffices to construct the state $\sum_{\mathbf{x}\in\mathbb{Z}_q^m} f(\mathbf{x})g(\mathbf{x})\,|\,\mathbf{x}\rangle$

- QFT → suffices to construct the convolution of the Fourier transforms

  - $\hat{f}(\mathbf{z}) \propto \mathbb{I}[\mathbf{z} \in \mathscr{C}^\perp]$



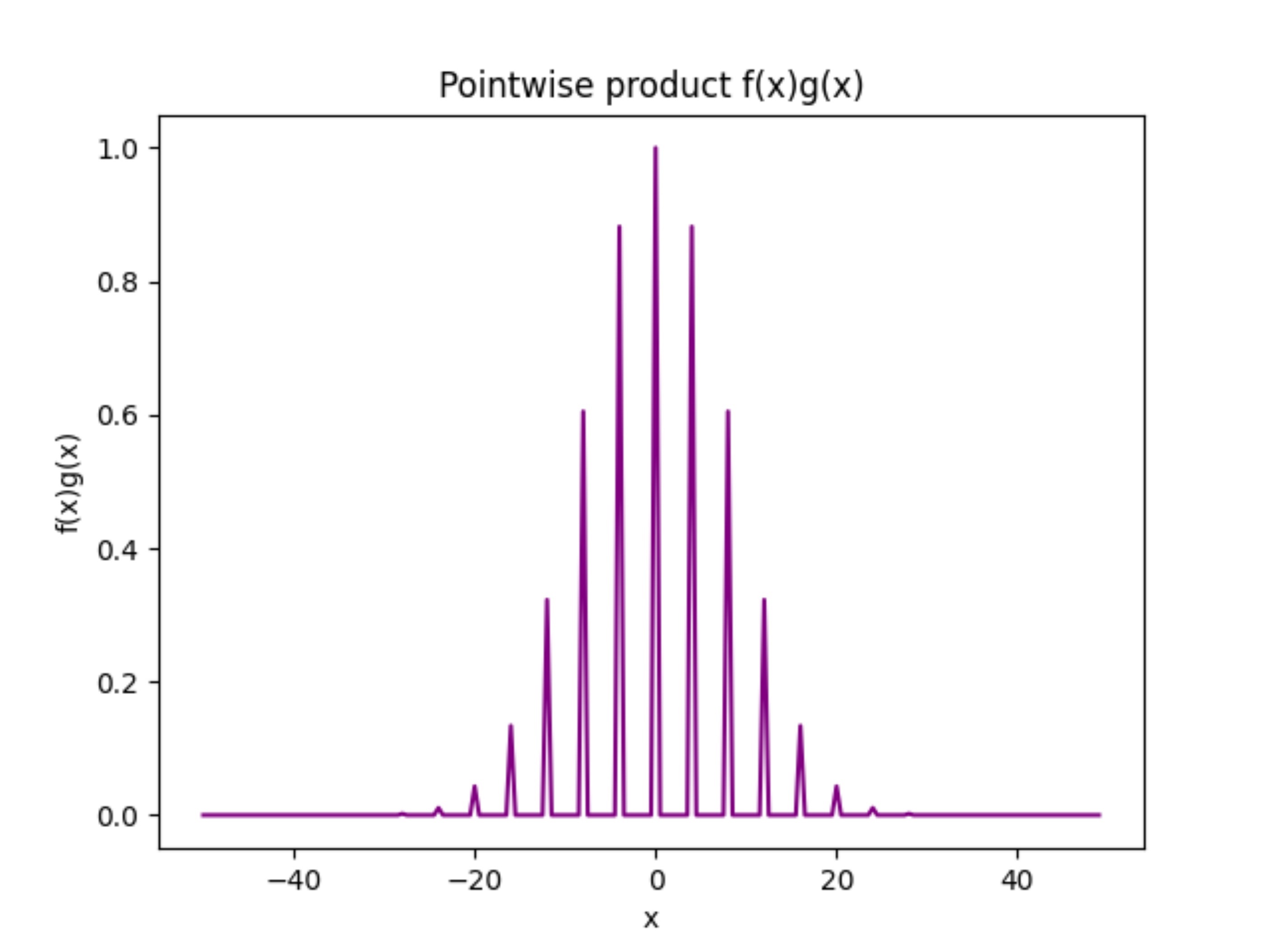


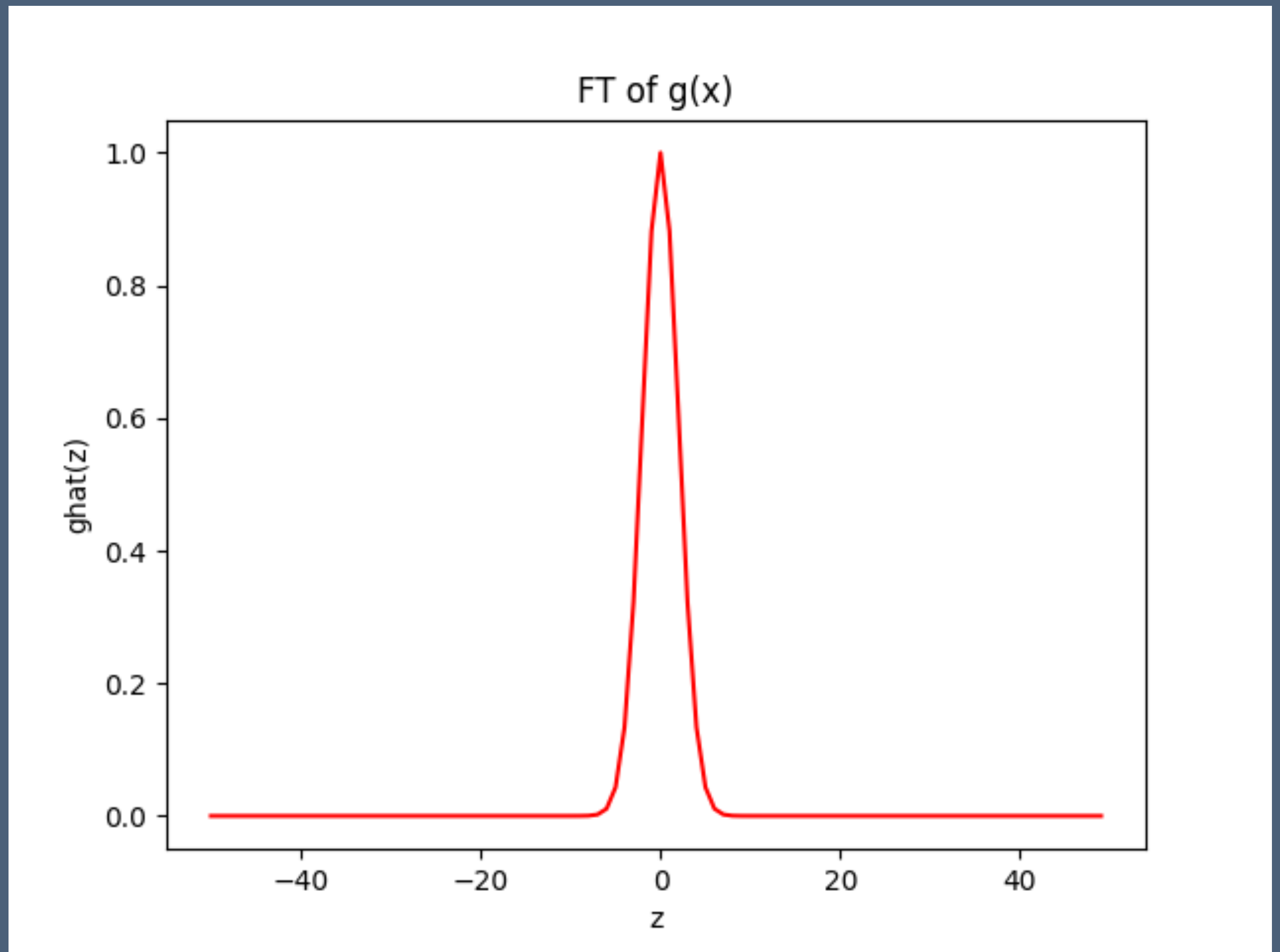


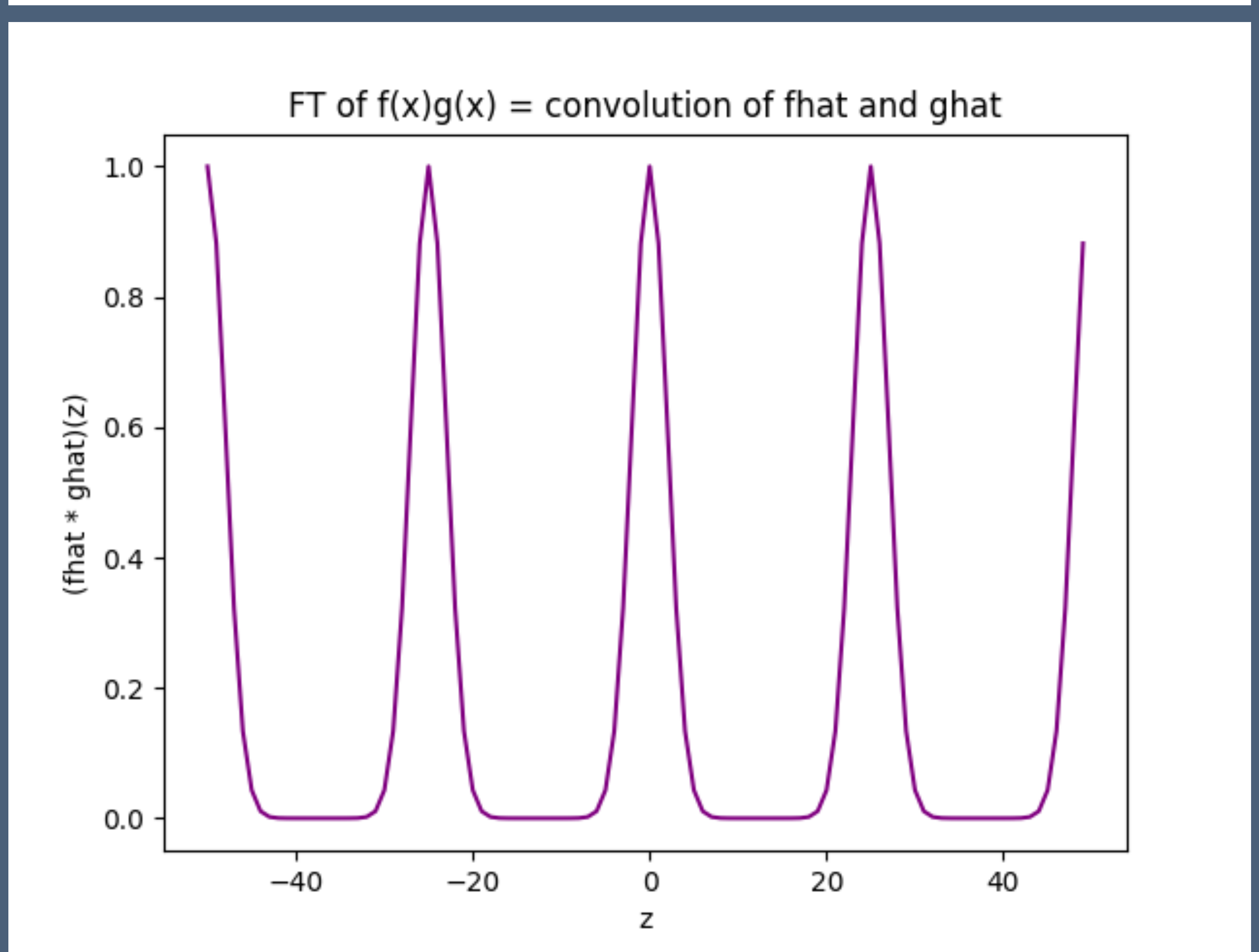
Regev's Reduction

Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

- Let $f(\mathbf{x}) = \mathbb{I}[\mathbf{x} \in \mathscr{C}]$ and $g(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/R^2)$ for suitable R
- Suffices to construct the state $\sum_{\mathbf{x} \in \mathbb{Z}_q^m} f(\mathbf{x})g(\mathbf{x})\,|\,\mathbf{x}\rangle$
- QFT → suffices to construct the convolution of the Fourier transforms
  - $\hat{f}(\mathbf{z}) \propto \mathbb{I}[\mathbf{z} \in \mathscr{C}^\perp]$
  - $\hat{g}(\mathbf{z}) \propto \exp(-\pi R^2\|\mathbf{z}\|^2/q^2)$







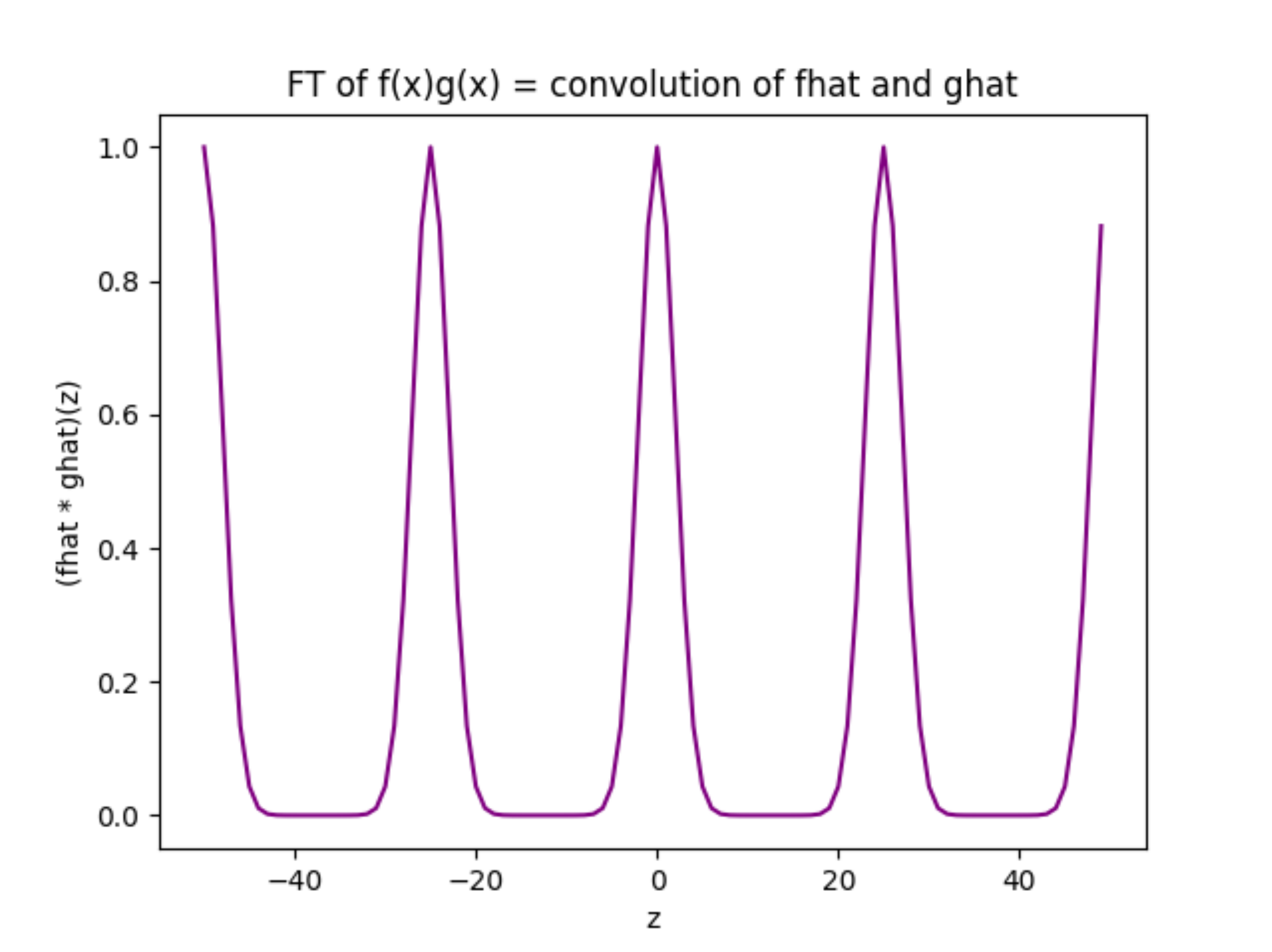
Regev's Reduction

Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

Regev's Reduction

Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

- Suffices to prepare $\displaystyle\sum_{\mathbf{c}\in\mathscr{C}^\perp}\sum_{\mathbf{e}\in\mathbb{Z}_q^m} \exp(-\pi R^2\|\mathbf{e}\|^2/q^2)\,|\,\mathbf{c}+\mathbf{e}\rangle$



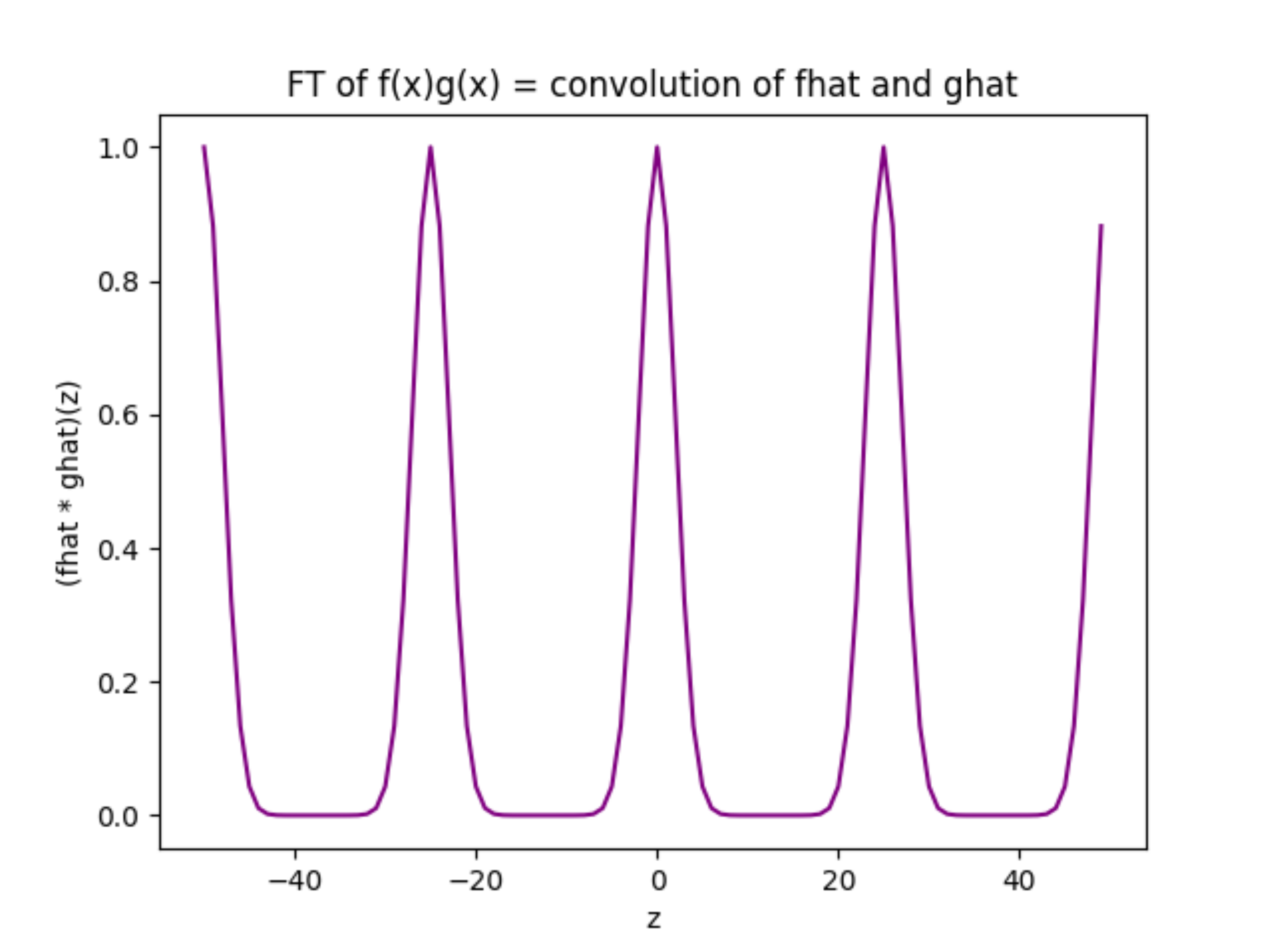
Regev's Reduction

Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

- Suffices to prepare $\displaystyle\sum_{\mathbf{c}\in\mathscr{C}^\perp}\sum_{\mathbf{e}\in\mathbb{Z}_q^m} \exp(-\pi R^2\|\mathbf{e}\|^2/q^2)\,|\,\mathbf{c}+\mathbf{e}\rangle$

- Algorithm:

  1. Separately prepare $\displaystyle\sum_{\mathbf{c}\in\mathscr{C}^\perp} |\,\mathbf{c}\rangle$ and $\displaystyle\sum_{\mathbf{e}\in\mathbb{Z}_q^m} \exp(-\pi R^2\|\mathbf{e}\|^2/q^2)\,|\,\mathbf{e}\rangle$



Regev's Reduction

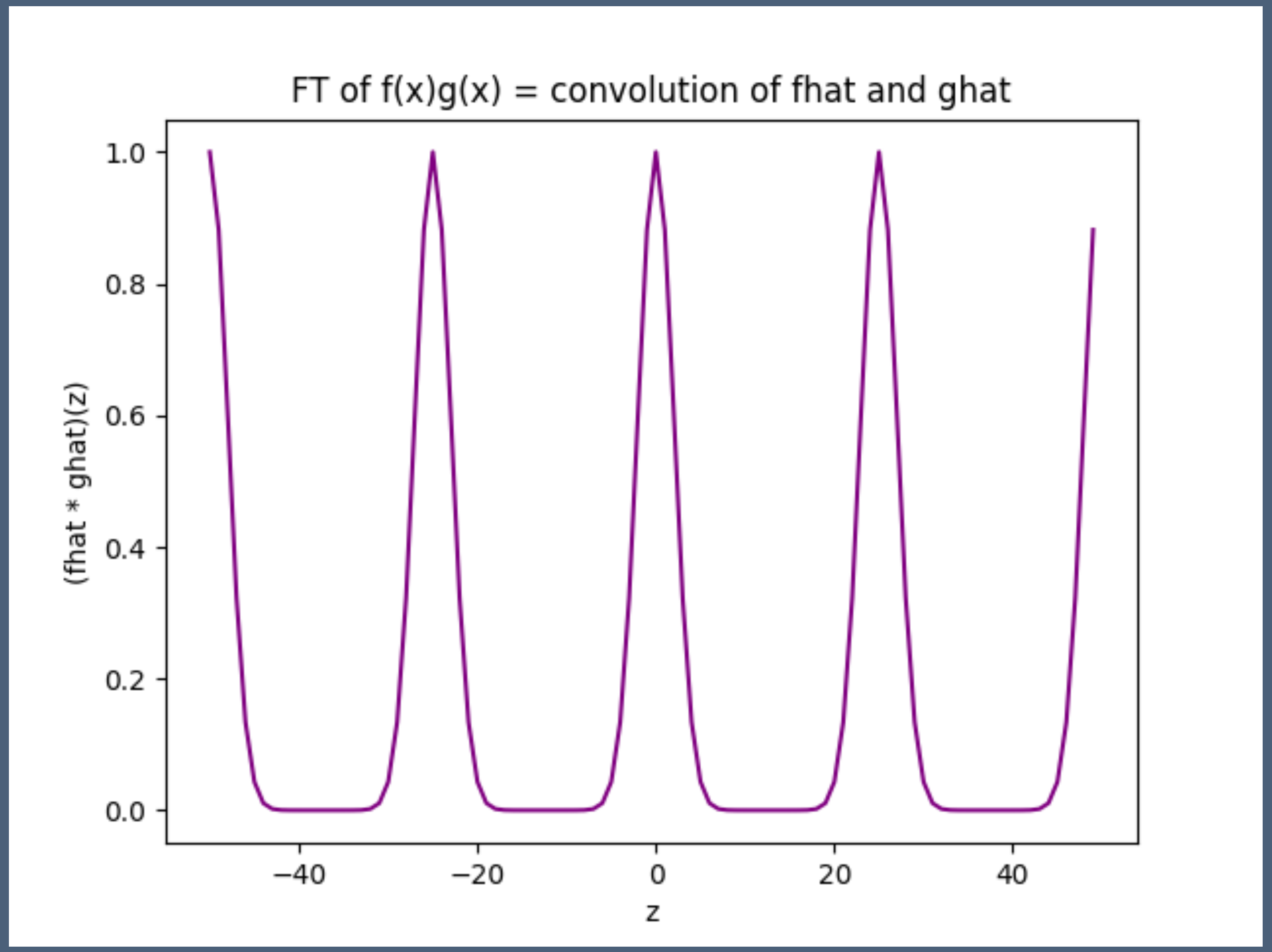
Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

- Suffices to prepare $\sum_{\mathbf{c}\in\mathscr{C}^\perp} \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \exp(-\pi R^2\|\mathbf{e}\|^2/q^2)\,|\mathbf{c}+\mathbf{e}\rangle$

- Algorithm:

  1. Separately prepare $\sum_{\mathbf{c}\in\mathscr{C}^\perp} |\mathbf{c}\rangle$ and $\sum_{\mathbf{e}\in\mathbb{Z}_q^m} \exp(-\pi R^2\|\mathbf{e}\|^2/q^2)\,|\mathbf{e}\rangle$

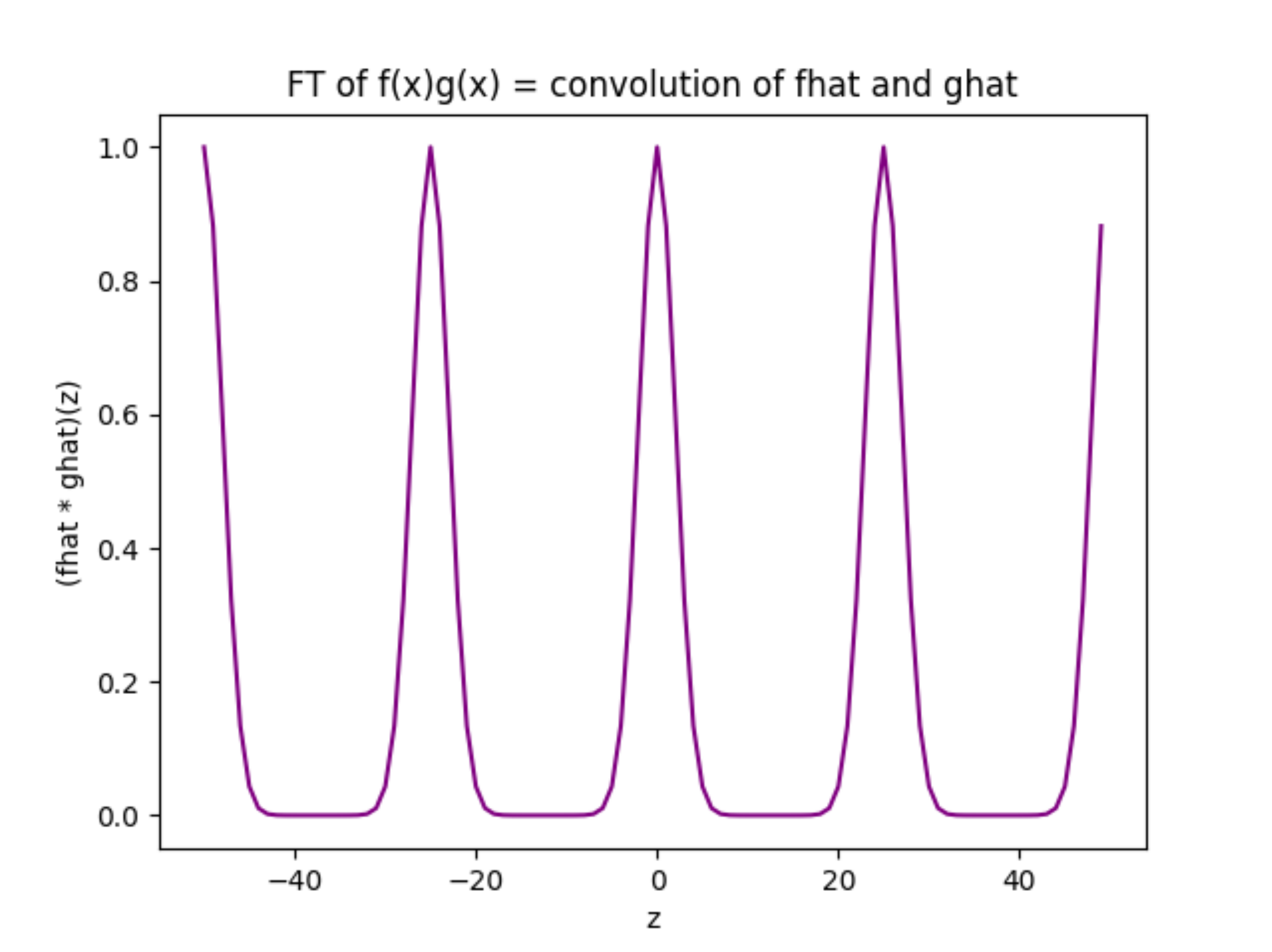
  2. Entangle them by computing $\mathbf{c}+\mathbf{e}$: $\sum_{\mathbf{c}\in\mathscr{C}^\perp} \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \exp(-\pi R^2\|\mathbf{e}\|^2/q^2)\,|\mathbf{c}\rangle\,|\mathbf{e}\rangle\,|\mathbf{c}+\mathbf{e}\rangle$



Regev's Reduction

Goal: output a codeword from linear $\mathscr{C} \subset \mathbb{Z}_q^m$ with all entries close to 0

- Suffices to prepare $\displaystyle\sum_{\mathbf{c} \in \mathscr{C}^\perp} \sum_{\mathbf{e} \in \mathbb{Z}_q^m} \exp(-\pi R^2 \|\mathbf{e}\|^2 / q^2) \,|\, \mathbf{c} + \mathbf{e}\rangle$



- Algorithm:

1. Separately prepare $\displaystyle\sum_{\mathbf{c} \in \mathscr{C}^\perp} |\mathbf{c}\rangle$ and $\displaystyle\sum_{\mathbf{e} \in \mathbb{Z}_q^m} \exp(-\pi R^2 \|\mathbf{e}\|^2 / q^2) \,|\, \mathbf{e}\rangle$

2. Entangle them by computing $\mathbf{c} + \mathbf{e}$: $\displaystyle\sum_{\mathbf{c} \in \mathscr{C}^\perp} \sum_{\mathbf{e} \in \mathbb{Z}_q^m} \exp(-\pi R^2 \|\mathbf{e}\|^2 / q^2) \,|\mathbf{c}\rangle\,|\mathbf{e}\rangle\,|\, \mathbf{c} + \mathbf{e}\rangle$

3. We now need a <u>decoder</u> to recover $\mathbf{c}, \mathbf{e}$ from $\mathbf{c} + \mathbf{e}$ to "erase" these registers

Regev's Reduction: Summary

Cryptography assuming the hardness of lattice problems

Regev's Reduction: Summary

Cryptography assuming the hardness of lattice problems

- Regev's encryption scheme is secure, unless there exists an algorithm for decoding $\mathbf{c} + \mathbf{e} \mapsto \mathbf{c}, \mathbf{e}$ for $\mathbf{c} \in \mathscr{C}^{\perp}$ and $\mathbf{e}$ of low ℓ_2 norm

Regev's Reduction: Summary

Cryptography assuming the hardness of lattice problems

- Regev's encryption scheme is secure, unless there exists an algorithm for decoding $\mathbf{c} + \mathbf{e} \mapsto \mathbf{c}, \mathbf{e}$ for $\mathbf{c} \in \mathscr{C}^\perp$ and $\mathbf{e}$ of low ℓ_2 norm
- Given such an algorithm, we could combine it with the QFT to output a codeword in $\mathscr{C}$ of low ℓ_2 norm

Regev's Reduction: Summary

Cryptography assuming the hardness of lattice problems

- Regev's encryption scheme is secure, unless there exists an algorithm for decoding $\mathbf{c} + \mathbf{e} \mapsto \mathbf{c}, \mathbf{e}$ for $\mathbf{c} \in \mathscr{C}^\perp$ and $\mathbf{e}$ of low ℓ_2 norm

- Given such an algorithm, we could combine it with the QFT to output a codeword in $\mathscr{C}$ of low ℓ_2 norm

- Classical reductions → this would allow us to solve lattice problems (e.g. approximate shortest vector)

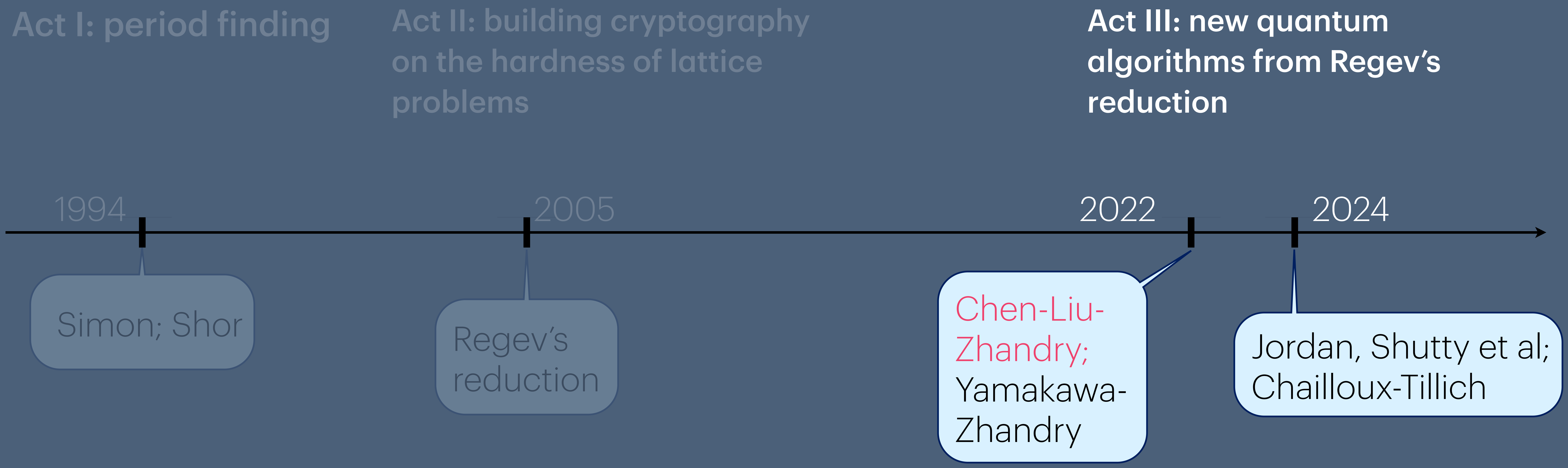
Regev's Reduction: Summary

Cryptography assuming the hardness of lattice problems

- Regev's encryption scheme is secure, unless there exists an algorithm for decoding $\mathbf{c} + \mathbf{e} \mapsto \mathbf{c}, \mathbf{e}$ for $\mathbf{c} \in \mathscr{C}^{\perp}$ and $\mathbf{e}$ of low ℓ_2 norm
- Given such an algorithm, we could combine it with the QFT to output a codeword in $\mathscr{C}$ of low ℓ_2 norm
- Classical reductions → this would allow us to solve lattice problems (e.g. approximate shortest vector)

Running this backwards: if lattice problems are hard then Regev's encryption scheme is secure!
Reduction inherently quantum; relies on the QFT

Exponential Speedups from the Quantum Fourier Transform



Algorithms from Regev’s Reduction!

Algorithms from Regev's Reduction!

- Regev's reduction: a framework for quantumly solving search/optimisation problems (governed by some "score function" $g : \mathbb{Z}_q^m \to \mathbb{C}$) over linear codes $\mathscr{C} \subset \mathbb{Z}_q^m$

Algorithms from Regev's Reduction!

- Regev's reduction: a framework for quantumly solving search/optimisation problems (governed by some "score function" $g : \mathbb{Z}_q^m \to \mathbb{C}$) over linear codes $\mathscr{C} \subset \mathbb{Z}_q^m$

  - Key ingredient: an algorithm for decoding noisy codewords $\mathbf{c} + \mathbf{e}$, where $\mathbf{c} \leftarrow \mathscr{C}^\perp$ and $\mathbf{Pr}[\mathbf{e}] \propto |\hat{g}(\mathbf{e})|^2$

Algorithms from Regev's Reduction!

- Regev's reduction: a framework for quantumly solving <u>search/optimisation problems</u> (governed by some "score function" $g : \mathbb{Z}_q^m \to \mathbb{C}$) over linear codes $\mathscr{C} \subset \mathbb{Z}_q^m$

    - Key ingredient: an <u>algorithm for decoding</u> noisy codewords $\mathbf{c} + \mathbf{e}$, where $\mathbf{c} \leftarrow \mathscr{C}^\perp$ and $\mathbf{Pr}[\mathbf{e}] \propto |\hat{g}(\mathbf{e})|^2$

- Regev: assume search is hard → show that decoding is hard

Algorithms from Regev's Reduction!

- Regev's reduction: a framework for quantumly solving search/optimisation problems (governed by some "score function" $g : \mathbb{Z}_q^m \to \mathbb{C}$) over linear codes $\mathscr{C} \subset \mathbb{Z}_q^m$
  - Key ingredient: an algorithm for decoding noisy codewords $\mathbf{c} + \mathbf{e}$, where $\mathbf{c} \leftarrow \mathscr{C}^\perp$ and $\mathbf{Pr}[\mathbf{e}] \propto |\hat{g}(\mathbf{e})|^2$
- Regev: assume search is hard → show that decoding is hard
- Recent works flip the script! Set up $\mathscr{C}^\perp, \hat{g}$ so that decoding is easy → search is also easy!

Algorithms from Regev's Reduction!

- Regev's reduction: a framework for quantumly solving search/optimisation problems (governed by some "score function" $g : \mathbb{Z}_q^m \to \mathbb{C}$) over linear codes $\mathscr{C} \subset \mathbb{Z}_q^m$
  - Key ingredient: an algorithm for decoding noisy codewords $\mathbf{c} + \mathbf{e}$, where $\mathbf{c} \leftarrow \mathscr{C}^\perp$ and $\mathbf{Pr}[\mathbf{e}] \propto |\hat{g}(\mathbf{e})|^2$
- Regev: assume search is hard → show that decoding is hard
- Recent works flip the script! Set up $\mathscr{C}^\perp, \hat{g}$ so that decoding is easy → search is also easy!
  - $\mathscr{C}^\perp$ as a code from low-degree polynomials (e.g. Reed-Solomon): Yamakawa-Zhandry '22, Jordan-Shutty et al '24, Chailloux-Tillich '24

Algorithms from Regev's Reduction!

- Regev's reduction: a framework for quantumly solving search/optimisation problems (governed by some "score function" $g : \mathbb{Z}_q^m \to \mathbb{C}$) over linear codes $\mathscr{C} \subset \mathbb{Z}_q^m$
    - Key ingredient: an algorithm for decoding noisy codewords $\mathbf{c} + \mathbf{e}$, where $\mathbf{c} \leftarrow \mathscr{C}^\perp$ and $\mathbf{Pr[e]} \propto |\hat{g}(\mathbf{e})|^2$
- Regev: assume search is hard → show that decoding is hard
- Recent works flip the script! Set up $\mathscr{C}^\perp, \hat{g}$ so that decoding is easy → search is also easy!
    - $\mathscr{C}^\perp$ as a code from low-degree polynomials (e.g. Reed-Solomon): Yamakawa-Zhandry '22, Jordan-Shutty et al '24, Chailloux-Tillich '24
    - $\mathscr{C}^\perp$ as a very low rate linear code: Chen-Liu-Zhandry '22

Talk 1: Dequantising Chen-Liu-Zhandry

Robin Kothari (Google Quantum AI)

No exponential quantum speedup for SIS^∞ anymore

Robin Kothari* Ryan O'Donnell† Kewen Wu‡

Abstract

In 2021, Chen, Liu, and Zhandry presented an efficient quantum algorithm for the average-case ℓ_∞-Short Integer Solution (SIS^∞) problem, in a parameter range outside the normal range of cryptographic interest, but still with no known efficient classical algorithm. This was particularly exciting since SIS^∞ is a simple problem without structure, and their algorithmic techniques were different from those used in prior exponential quantum speedups.

We present efficient classical algorithms for all of the SIS^∞ and (more general) Constrained Integer Solution problems studied in their paper, showing there is no exponential quantum speedup anymore.

- Regev's reduction: a framework for quantumly solving search/optimisation problems (governed by some "scor
  - Key ingredie … $\mathscr{C}^\perp$ and $\Pr[\mathbf{e}] \propto |\hat{g}($
- Regev: assum
- Recent works … lso easy!
  - Codes from … y '22, Jordan-Shutty et al
  - Very low rank codes: Chen-Liu-Zhandry '22

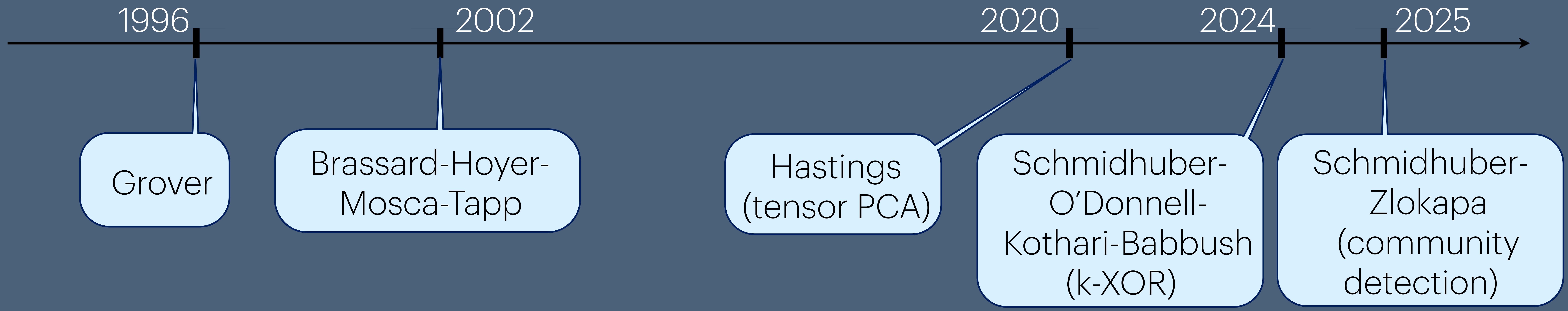
Polynomial Speedups for Search Problems

Act I: generic quadratic speedups

Polynomial Speedups for Search Problems

Act I: generic quadratic speedups

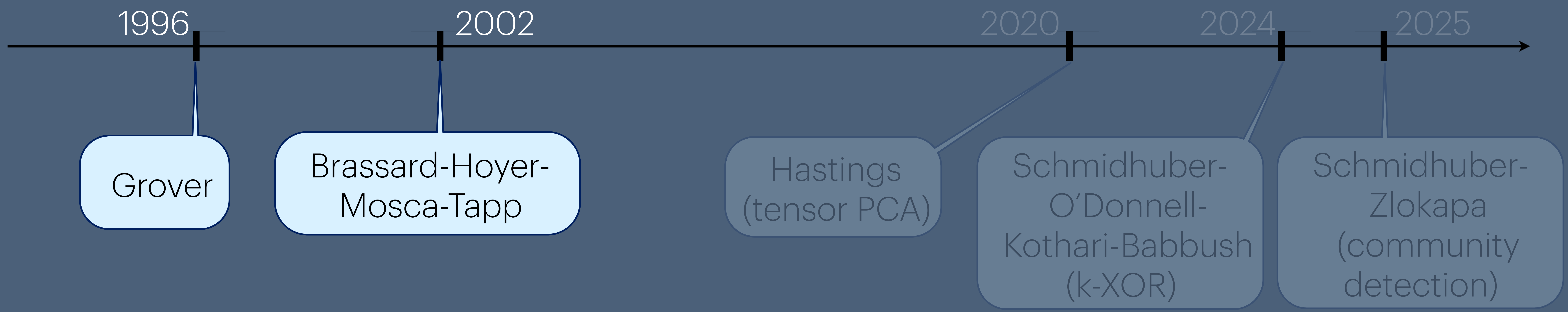
Act II: quartic speedups for planted inference problems



Polynomial Speedups for Search Problems

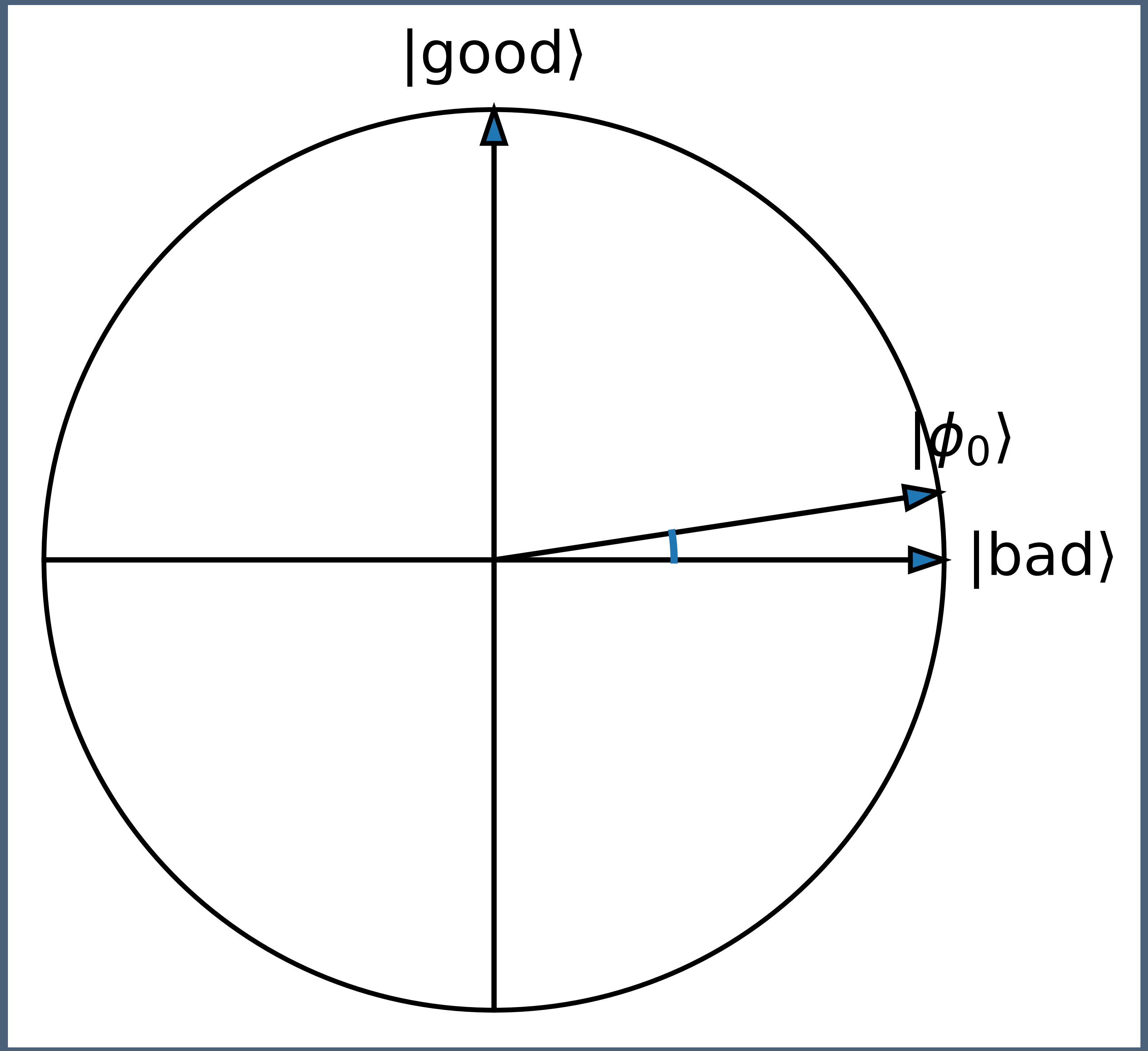
Act I: generic quadratic speedups

Act II: quartic speedups for planted inference problems



Amplitude Amplification

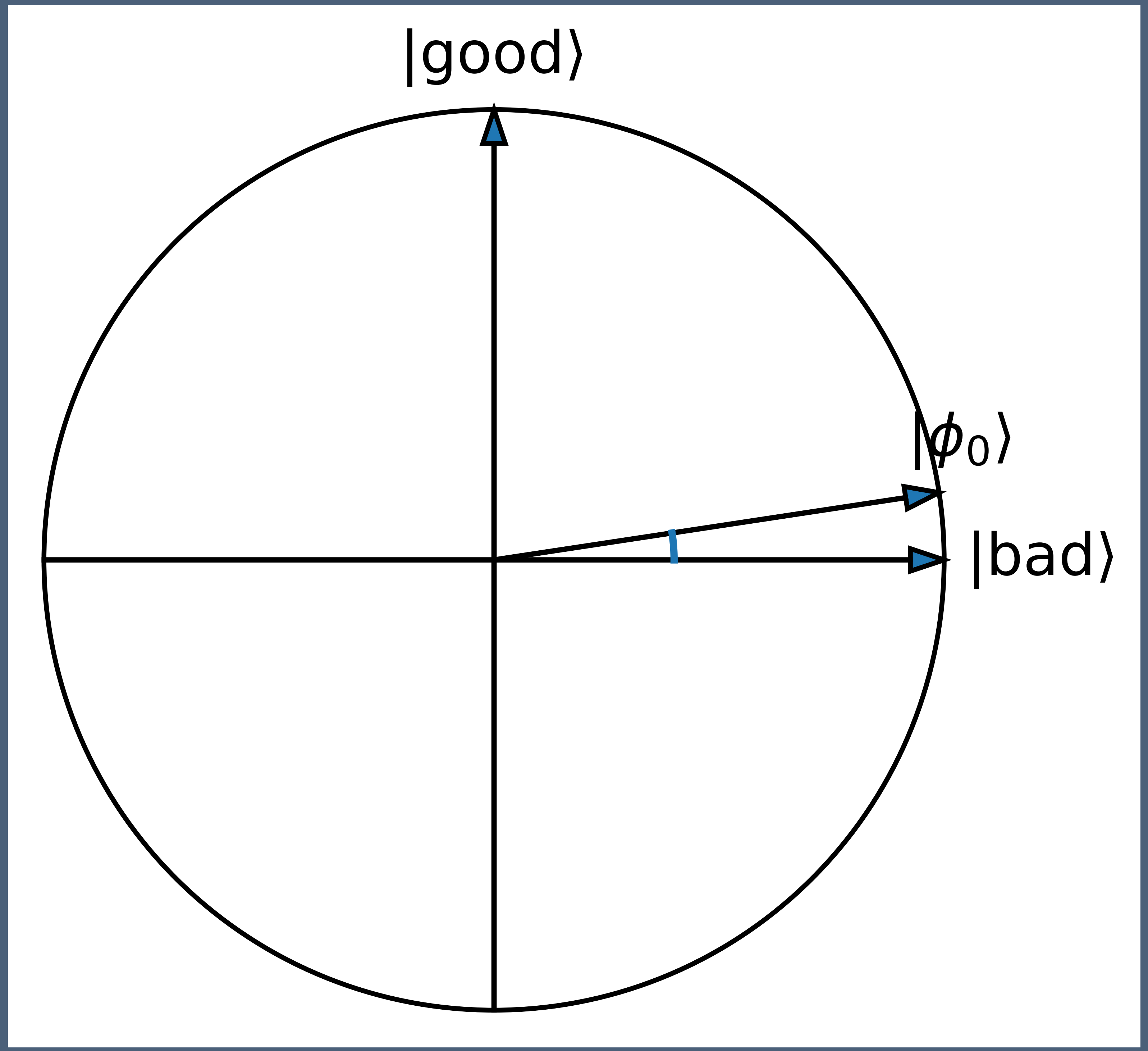
Often referred to as "Grover search"



Amplitude Amplification

Often referred to as "Grover search"

- Setting: we have (copies of) a starting state $|\phi_0\rangle = \cos\theta|\text{bad}\rangle + \sin\theta|\text{good}\rangle$, that we want to sanitise into $|\text{good}\rangle$

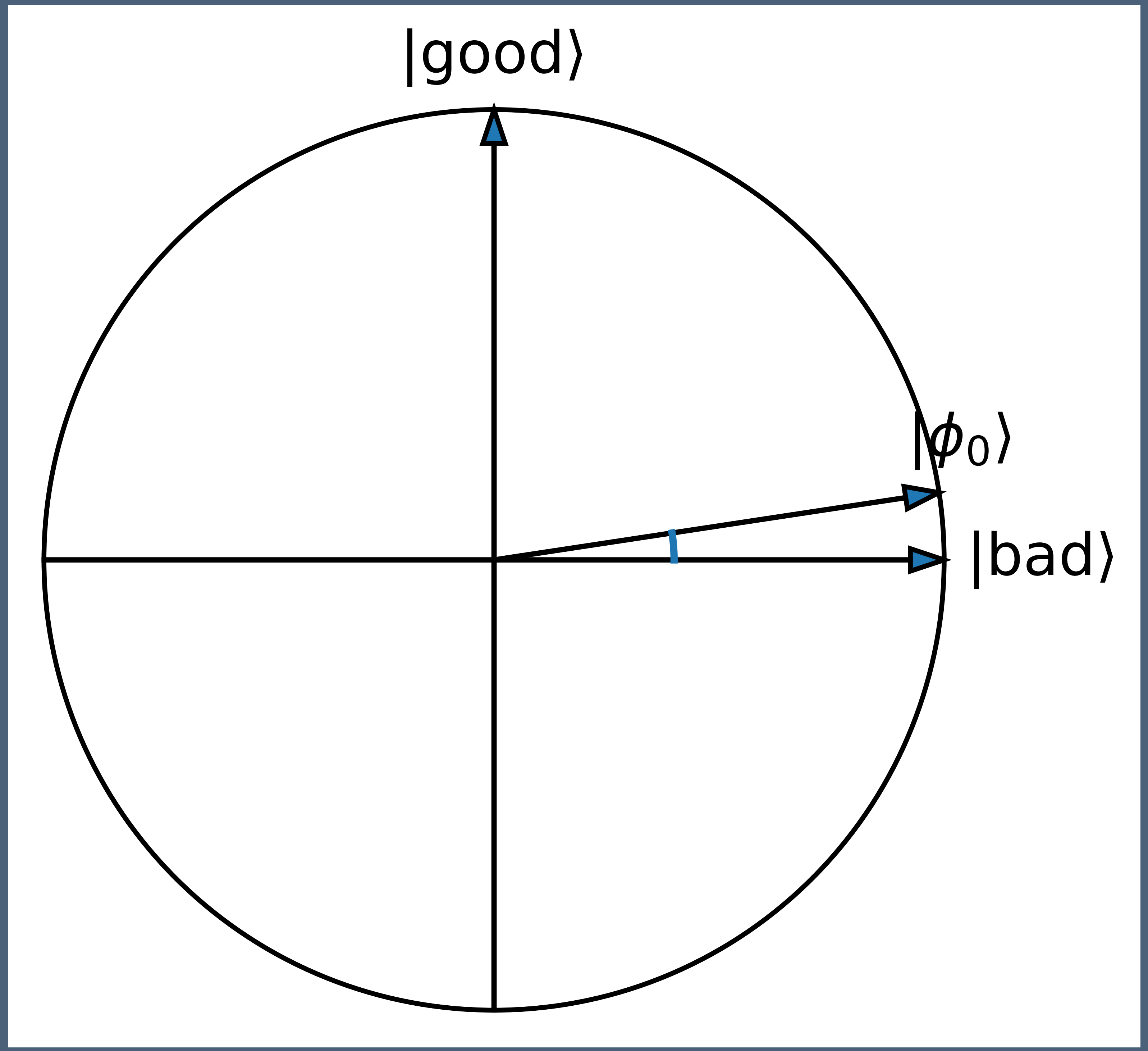


Amplitude Amplification

Often referred to as "Grover search"

- Setting: we have (copies of) a starting state $|\phi_0\rangle = \cos\theta|\mathsf{bad}\rangle + \sin\theta|\mathsf{good}\rangle$, that we want to sanitise into $|\mathsf{good}\rangle$

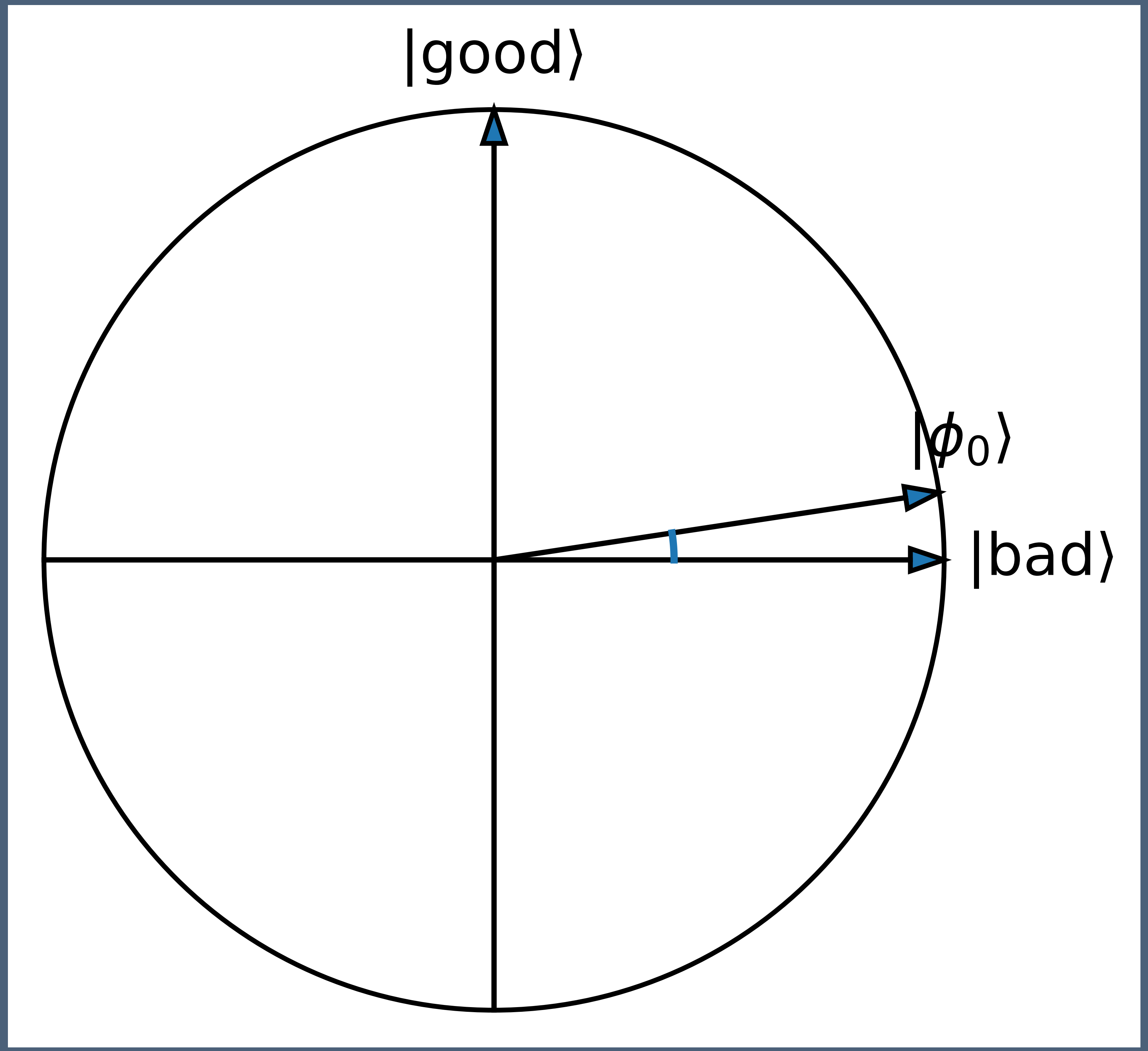
- Naive idea: measure $|\phi_0\rangle$ → success probability $\sin^2\theta \approx \theta^2$, so runtime is $O(1/\theta^2)$



Amplitude Amplification

Often referred to as "Grover search"

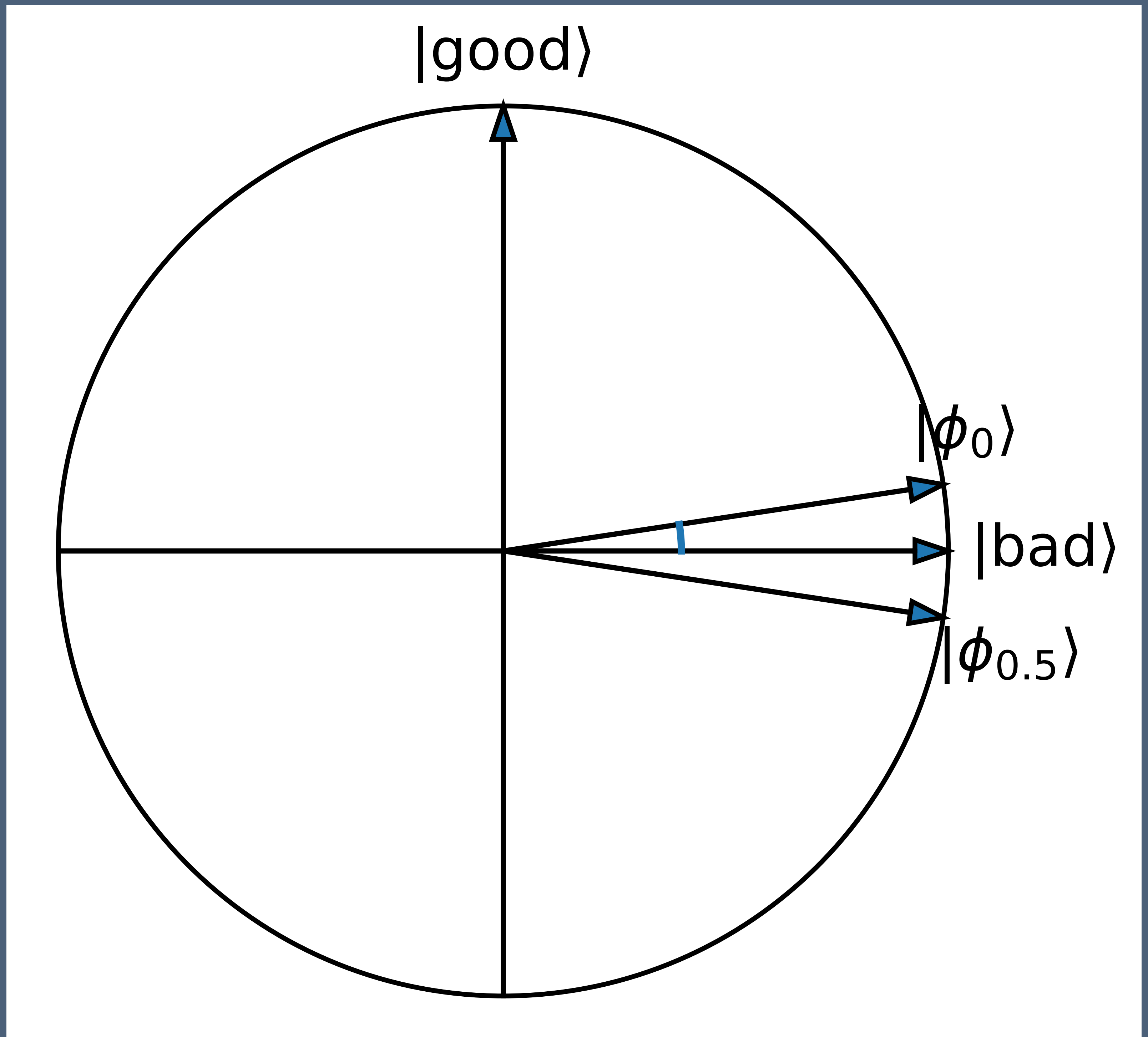
- Setting: we have (copies of) a starting state $|\phi_0\rangle = \cos\theta\,|\mathsf{bad}\rangle + \sin\theta\,|\mathsf{good}\rangle$, that we want to sanitise into $|\mathsf{good}\rangle$
- Naive idea: measure $|\phi_0\rangle$ → success probability $\sin^2\theta \approx \theta^2$, so runtime is $O(1/\theta^2)$
- Better idea: gently rotate towards $|\mathsf{good}\rangle$ by reflecting over $|\mathsf{bad}\rangle$ then $|\phi_0\rangle$



Amplitude Amplification

Often referred to as "Grover search"

- Setting: we have (copies of) a starting state $|\phi_0\rangle = \cos\theta|\text{bad}\rangle + \sin\theta|\text{good}\rangle$, that we want to sanitise into $|\text{good}\rangle$
- Naive idea: measure $|\phi_0\rangle$ → success probability $\sin^2\theta \approx \theta^2$, so runtime is $O(1/\theta^2)$
- Better idea: gently rotate towards $|\text{good}\rangle$ by reflecting over $|\text{bad}\rangle$ then $|\phi_0\rangle$



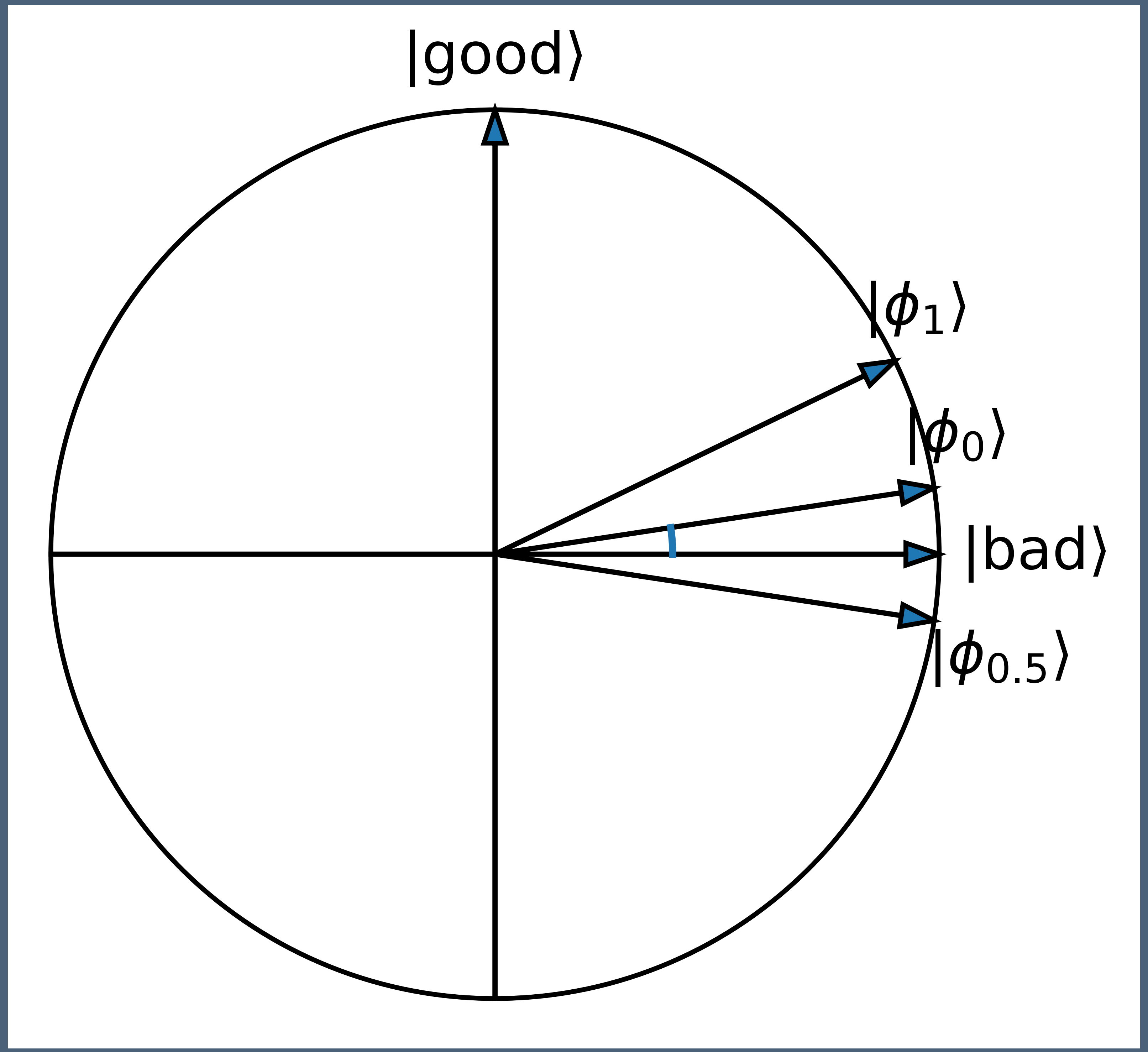
Amplitude Amplification

Often referred to as "Grover search"

- Setting: we have (copies of) a starting state $|\phi_0\rangle = \cos\theta|\text{bad}\rangle + \sin\theta|\text{good}\rangle$, that we want to sanitise into $|\text{good}\rangle$

- Naive idea: measure $|\phi_0\rangle$ → success probability $\sin^2\theta \approx \theta^2$, so runtime is $O(1/\theta^2)$

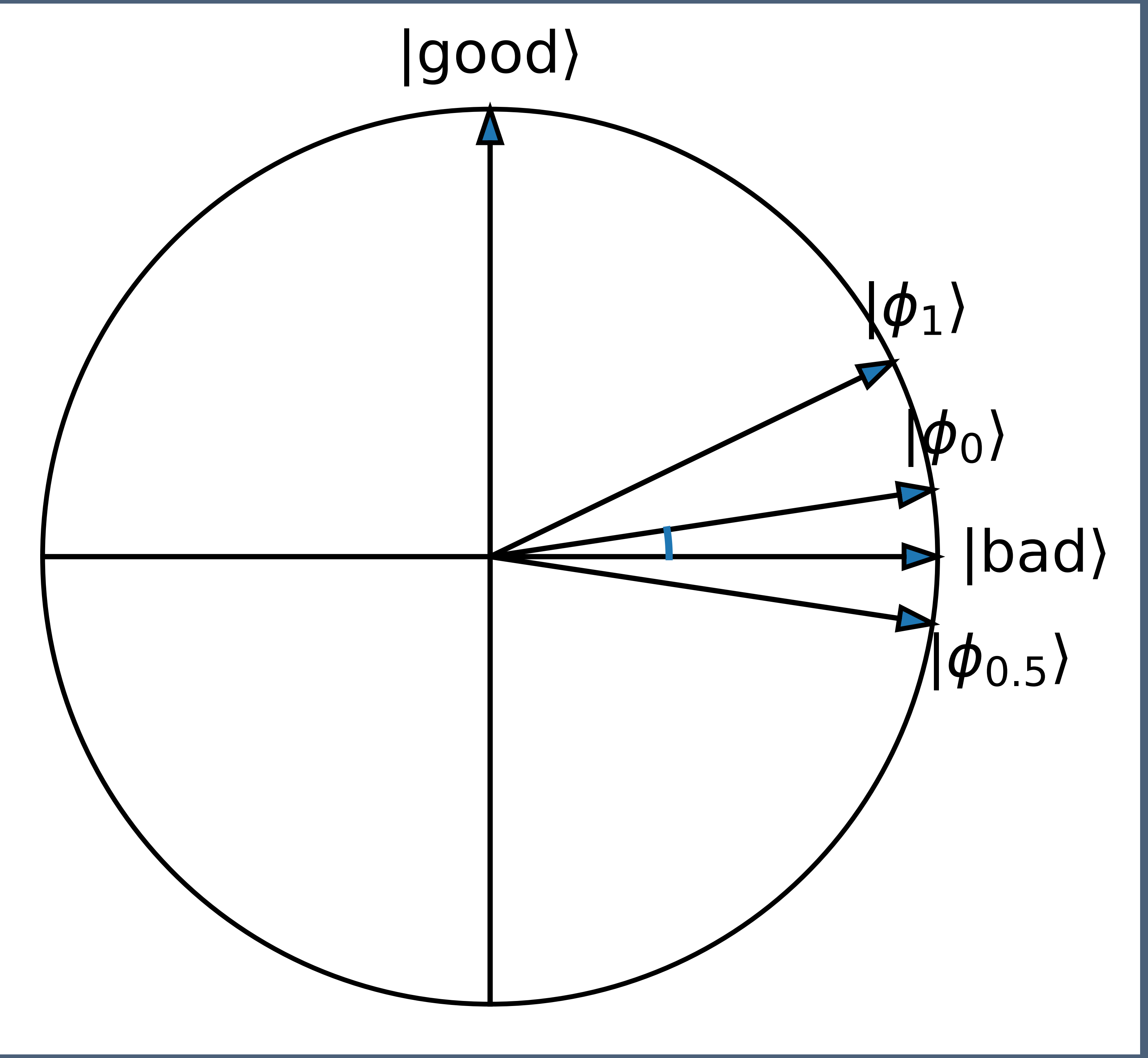
- Better idea: gently rotate towards $|\text{good}\rangle$ by reflecting over $|\text{bad}\rangle$ then $|\phi_0\rangle$



Amplitude Amplification

Often referred to as "Grover search"

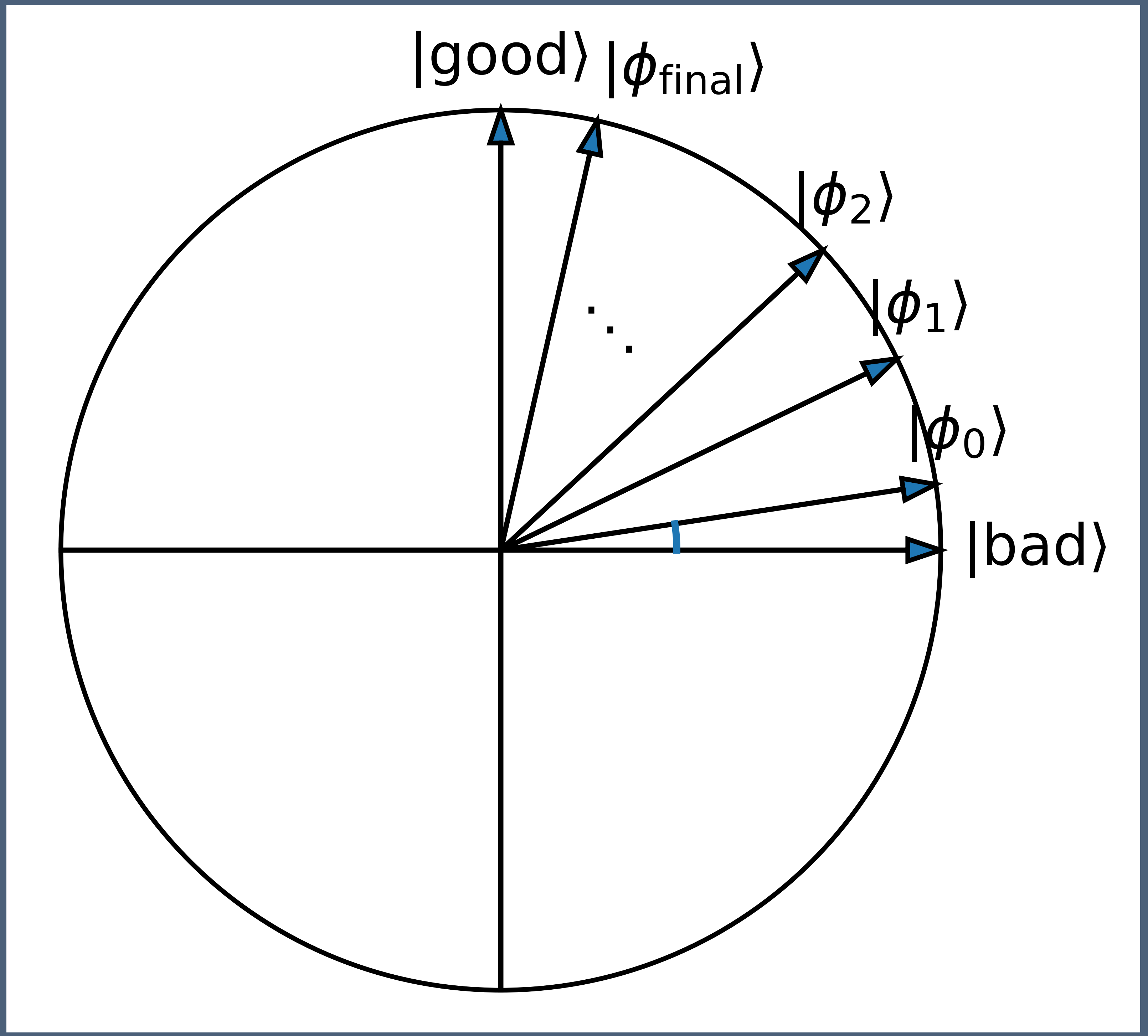
- Setting: we have (copies of) a starting state $|\phi_0\rangle = \cos\theta\,|\text{bad}\rangle + \sin\theta\,|\text{good}\rangle$, that we want to sanitise into $|\text{good}\rangle$
- Naive idea: measure $|\phi_0\rangle$ → success probability $\sin^2\theta \approx \theta^2$, so runtime is $O(1/\theta^2)$
- Better idea: gently rotate towards $|\text{good}\rangle$ by reflecting over $|\text{bad}\rangle$ then $|\phi_0\rangle$
  - Each step rotates by 2θ



Amplitude Amplification

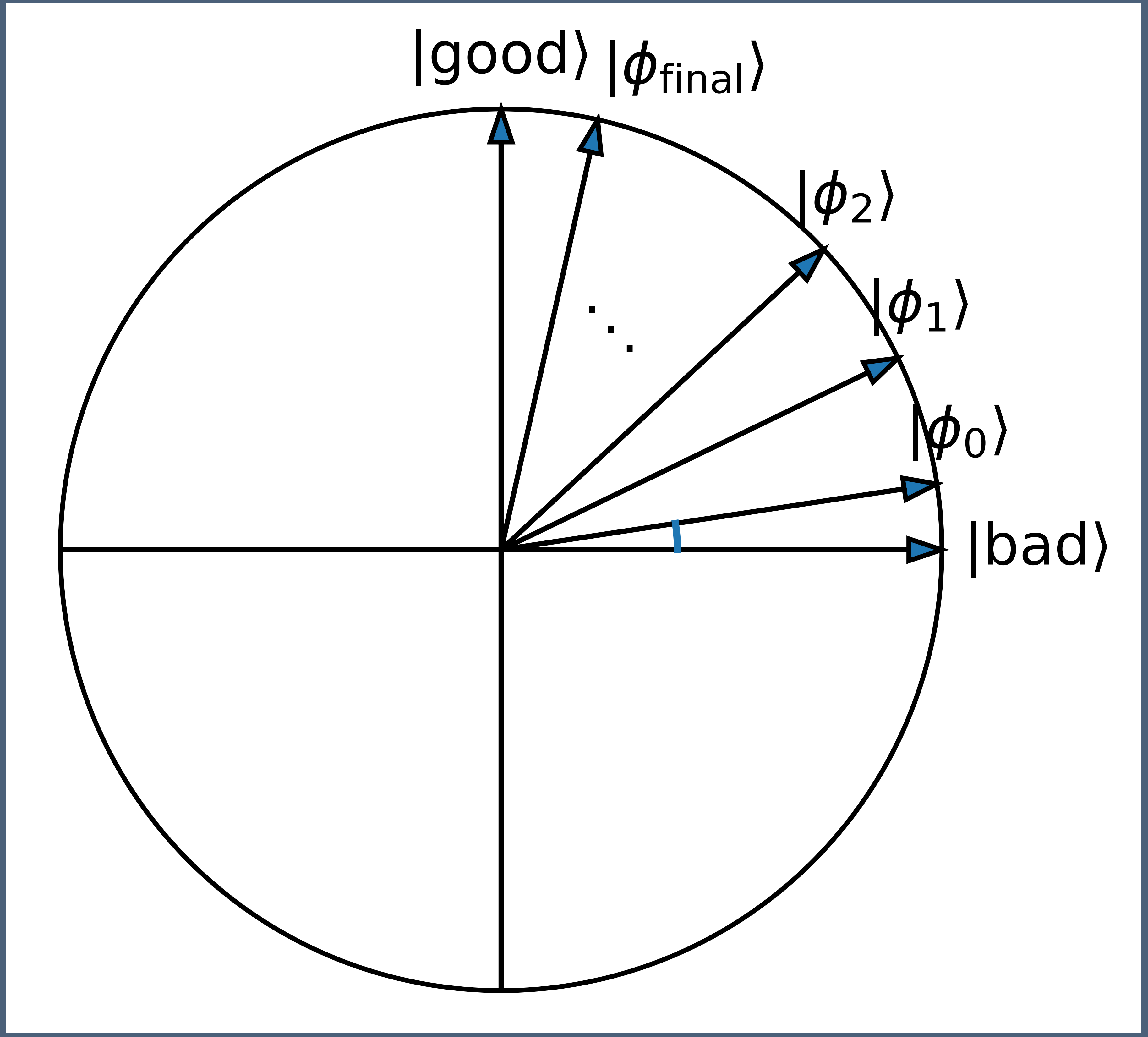
Often referred to as “Grover search”

- Setting: we have (copies of) a starting state $|\phi_0\rangle = \cos\theta|\text{bad}\rangle + \sin\theta|\text{good}\rangle$, that we want to sanitise into $|\text{good}\rangle$
- Naive idea: measure $|\phi_0\rangle$ → success probability $\sin^2\theta \approx \theta^2$, so runtime is $O(1/\theta^2)$
- Better idea: gently rotate towards $|\text{good}\rangle$ by reflecting over $|\text{bad}\rangle$ then $|\phi_0\rangle$
  - Each step rotates by 2θ
  - Runtime is $O(1/\theta)$!



Application: Quadratic Speedups

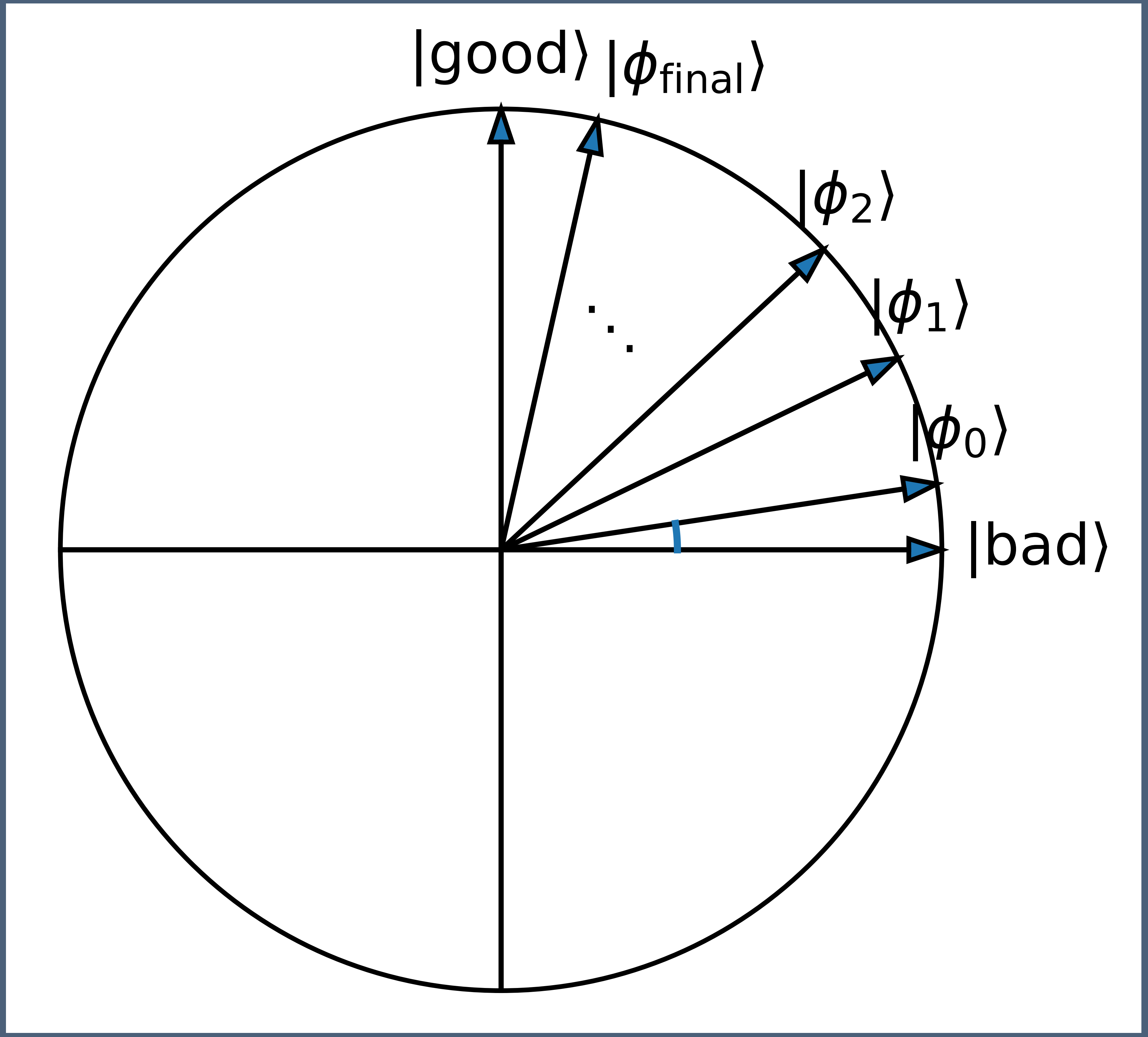
For your favourite NP search problem :)



Application: Quadratic Speedups

For your favourite NP search problem :)

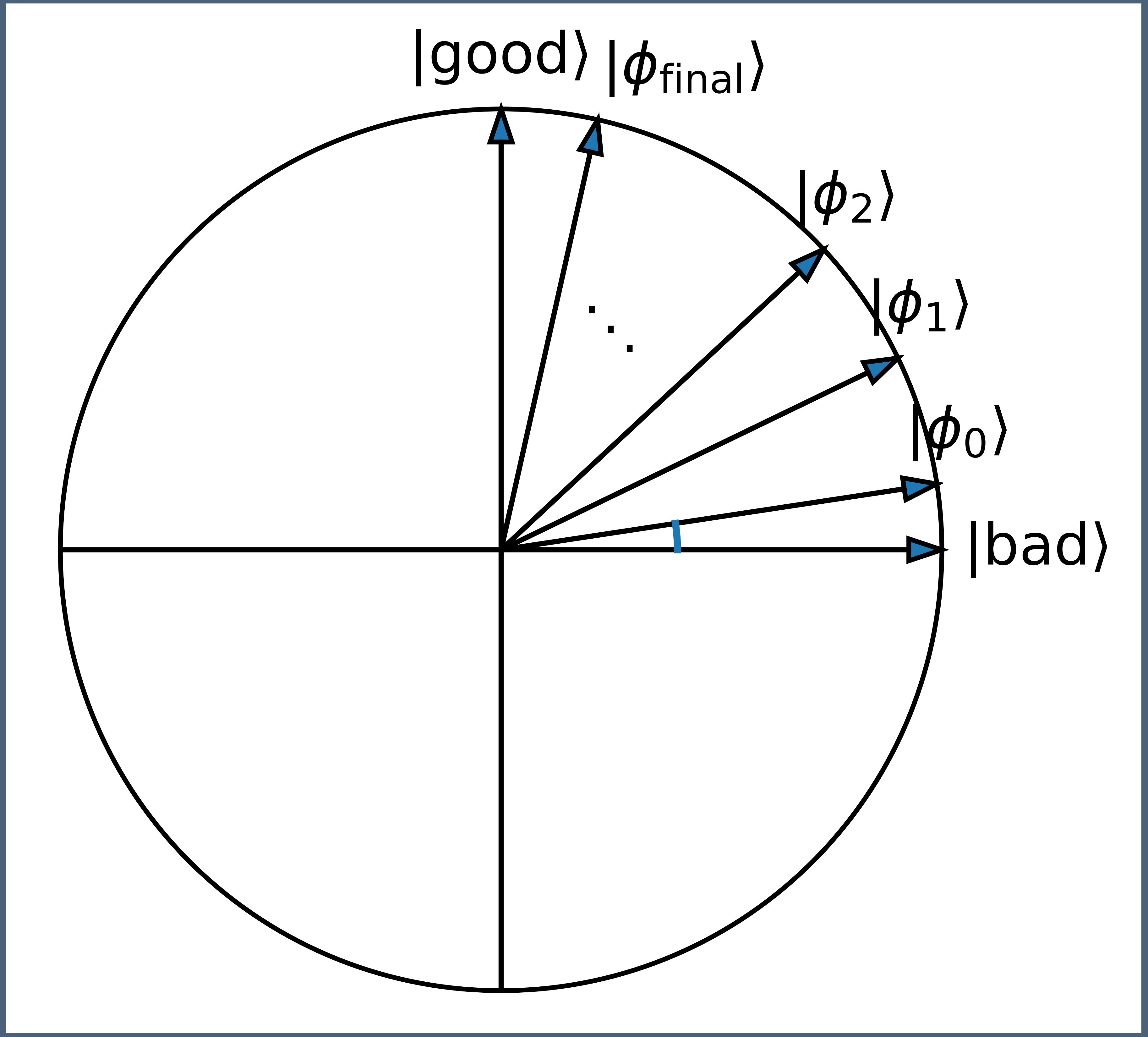
- $|\text{good}\rangle$: superposition of accepting witnesses
- $|\text{bad}\rangle$: superposition of rejecting witnesses



Application: Quadratic Speedups

For your favourite NP search problem :)

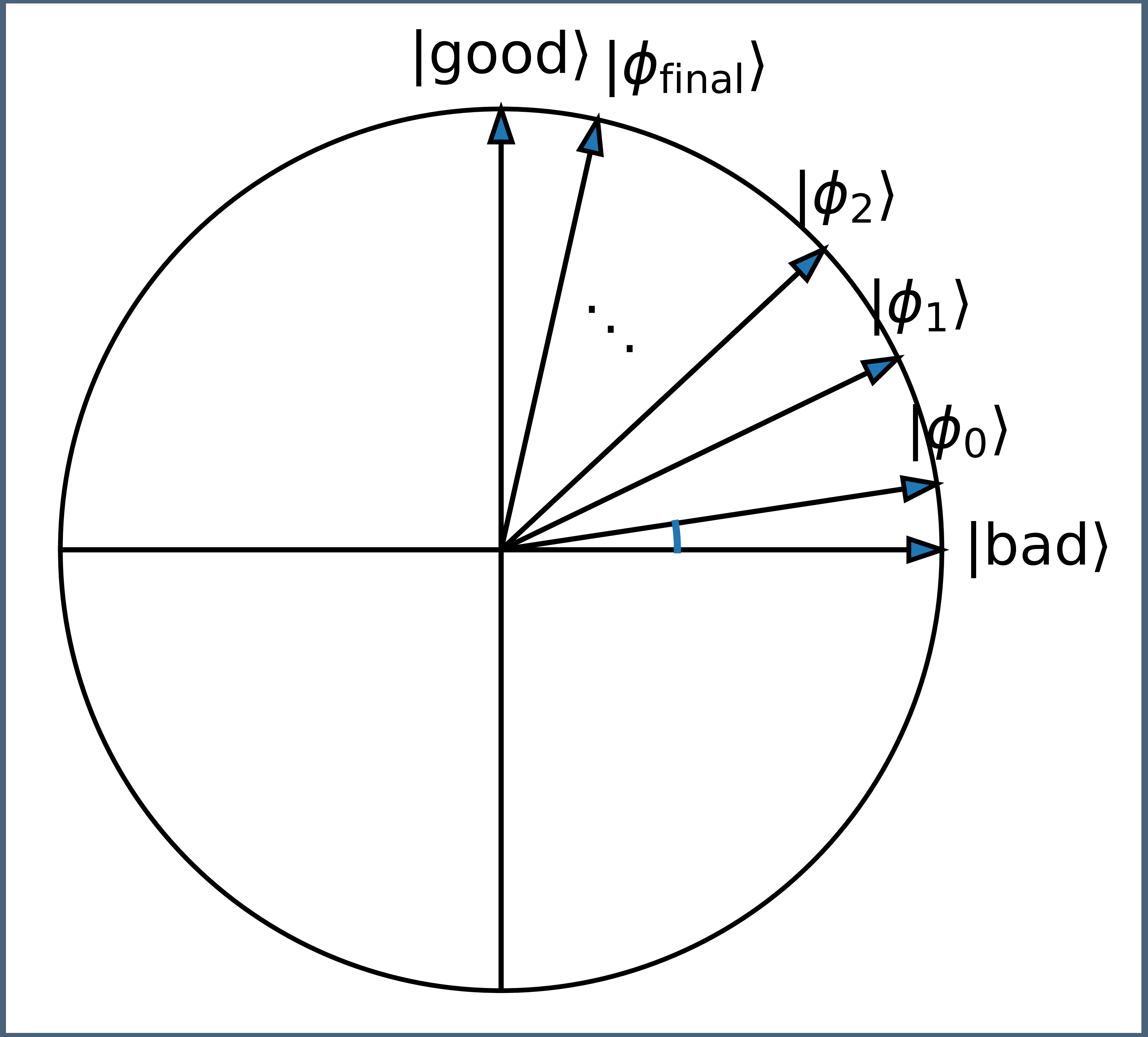
- $|\text{good}\rangle$: superposition of accepting witnesses
- $|\text{bad}\rangle$: superposition of rejecting witnesses
- $|\phi_0\rangle$: uniform superposition over all strings in $\{0,1\}^w$ ($\theta \approx 2^{-w/2}$)



Application: Quadratic Speedups

For your favourite NP search problem :)

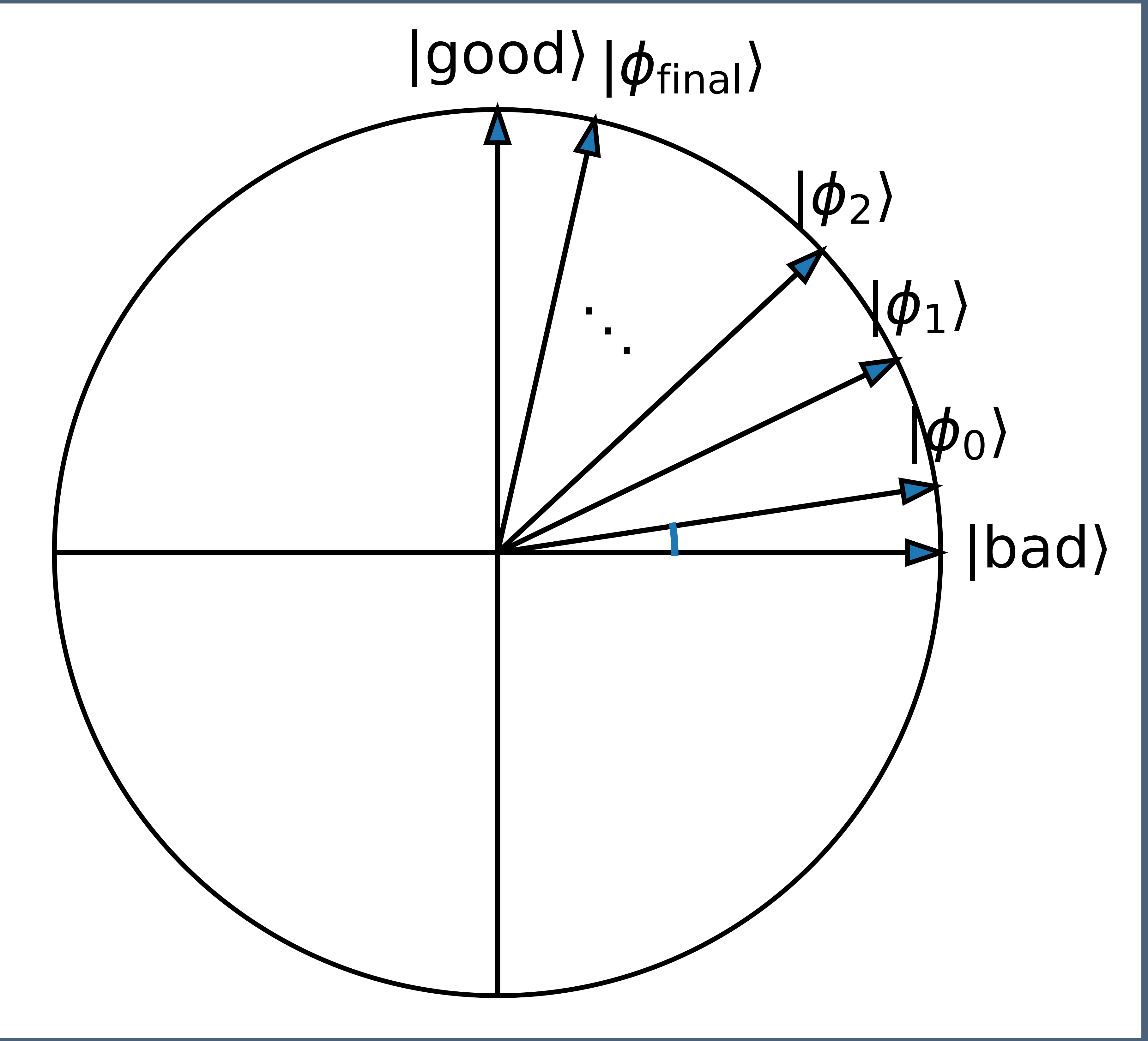
- $|\text{good}\rangle$: superposition of accepting witnesses
- $|\text{bad}\rangle$: superposition of rejecting witnesses
- $|\phi_0\rangle$: uniform superposition over all strings in $\{0,1\}^w$ ($\theta \approx 2^{-w/2}$)
- Naive approach (brute force search): runtime $O(1/\theta^2) = O(2^w)$



Application: Quadratic Speedups

For your favourite NP search problem :)

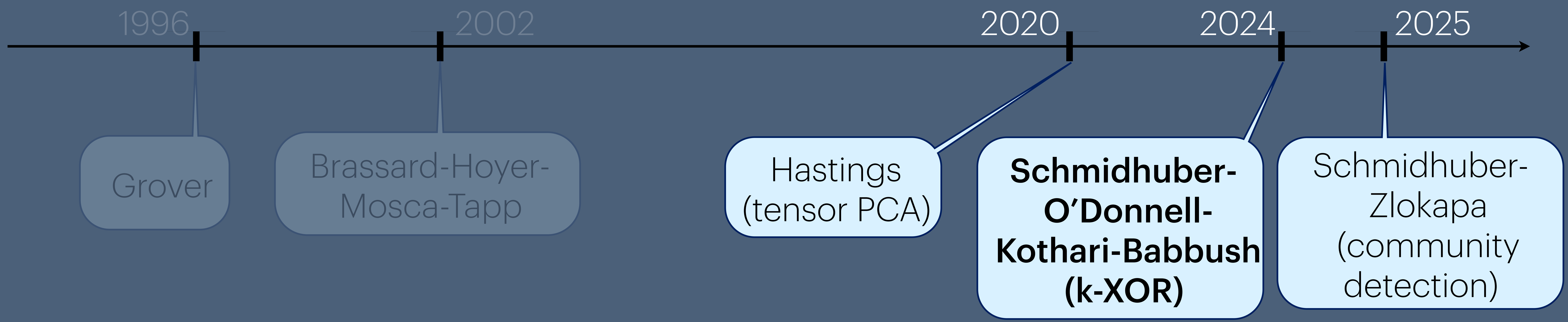
- $|\text{good}\rangle$: superposition of accepting witnesses
- $|\text{bad}\rangle$: superposition of rejecting witnesses
- $|\phi_0\rangle$: uniform superposition over all strings in $\{0,1\}^w$ ($\theta \approx 2^{-w/2}$)
- Naive approach (brute force search): runtime $O(1/\theta^2) = O(2^w)$
- Using amplitude amplification: runtime $O(1/\theta) = O(2^{w/2})$



Polynomial Speedups for Search Problems

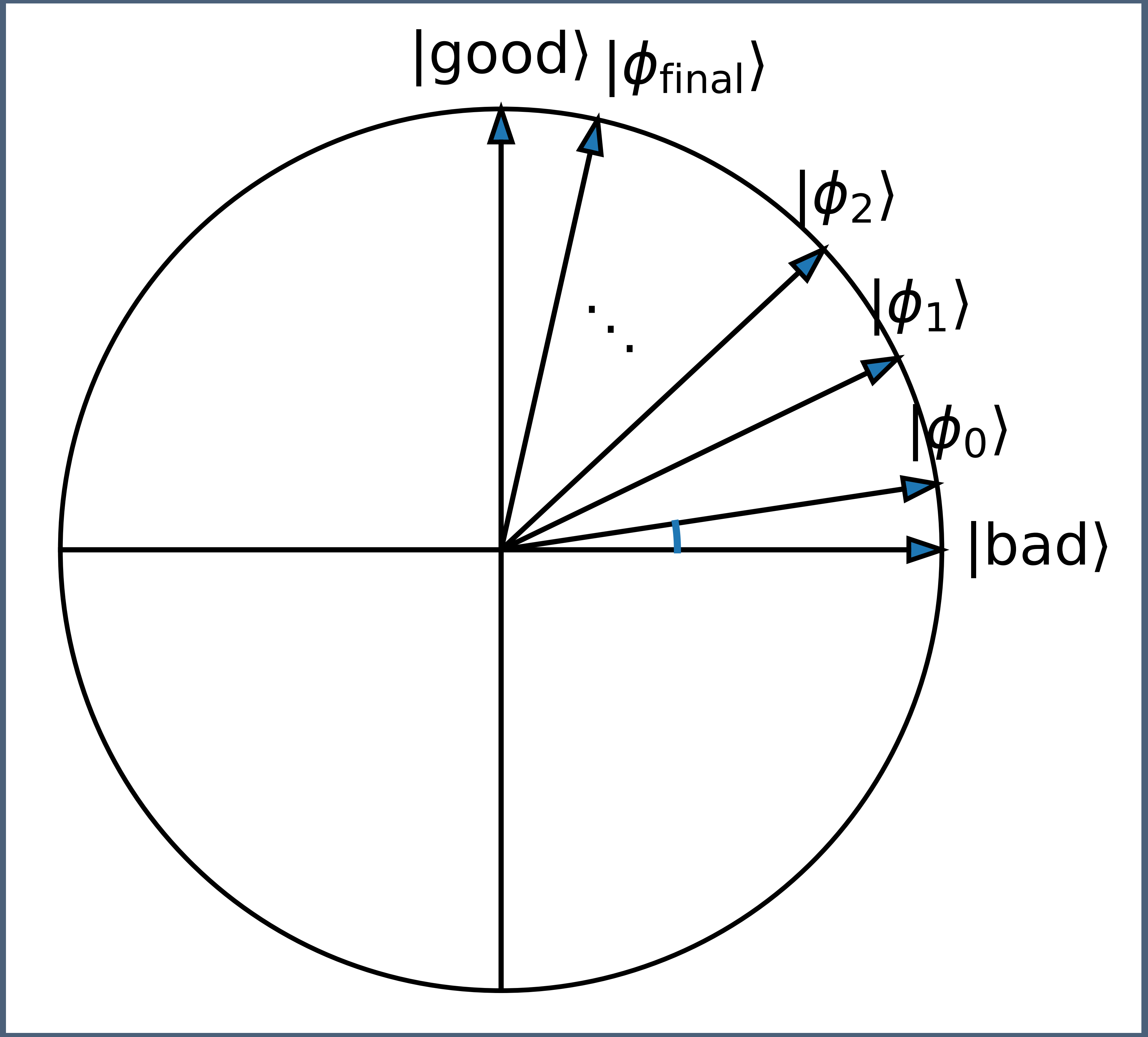
Act I: generic quadratic speedups

Act II: quartic speedups for planted inference problems



Beyond Quadratic Speedups

- $|\text{good}\rangle$: all accepting witnesses
- $|\text{bad}\rangle$: all rejecting witnesses
- $|\phi_0\rangle$: uniform superposition over all strings in $\{0,1\}^w$ ($\theta \approx 2^{-w/2}$)
- **"Guiding state" paradigm: find problems where we can select $|\phi_0\rangle$ more cleverly to ensure larger θ**



Quartic Speedup for the k-XOR Problem

Also known as Sparse Learning Parities with Noise (Sparse LPN)

Quartic Speedup for the k-XOR Problem

Also known as Sparse Learning Parities with Noise (Sparse LPN)

- Setup:
  - $\mathbf{A} \leftarrow \mathbb{F}_2^{m \times n}$ with $k = O(1)$ nonzero entries per row

Quartic Speedup for the k-XOR Problem

Also known as Sparse Learning Parities with Noise (Sparse LPN)

- Setup:
  - $\mathbf{A} \leftarrow \mathbb{F}_2^{m \times n}$ with $k = O(1)$ nonzero entries per row
  - $\mathbf{s} \leftarrow \mathbb{F}_2^n$, $\mathbf{e} \leftarrow$ sparse vector in $\mathbb{F}_2^m$

Quartic Speedup for the k-XOR Problem

Also known as Sparse Learning Parities with Noise (Sparse LPN)

- Setup:
  - $\mathbf{A} \leftarrow \mathbb{F}_2^{m \times n}$ with $k = O(1)$ nonzero entries per row
  - $\mathbf{s} \leftarrow \mathbb{F}_2^n$, $\mathbf{e} \leftarrow$ sparse vector in $\mathbb{F}_2^m$
- Task: given $\mathbf{A}, \mathbf{As} + \mathbf{e}$ as input, infer the planted secret $\mathbf{s}$

Quartic Speedup for the k-XOR Problem

Also known as Sparse Learning Parities with Noise (Sparse LPN)

- Setup:
  - $\mathbf{A} \leftarrow \mathbb{F}_2^{m \times n}$ with $k = O(1)$ nonzero entries per row
  - $\mathbf{s} \leftarrow \mathbb{F}_2^n$, $\mathbf{e} \leftarrow$ sparse vector in $\mathbb{F}_2^m$
- Task: given $\mathbf{A}, \mathbf{As} + \mathbf{e}$ as input, infer the planted secret $\mathbf{s}$
  - Naive amplitude amplification (starting with generic $|\phi_0\rangle$): $O(2^{n/2})$ time

Quartic Speedup for the k-XOR Problem

Also known as Sparse Learning Parities with Noise (Sparse LPN)

- Setup:
  - $\mathbf{A} \leftarrow \mathbb{F}_2^{m \times n}$ with $k = O(1)$ nonzero entries per row
  - $\mathbf{s} \leftarrow \mathbb{F}_2^n$, $\mathbf{e} \leftarrow$ sparse vector in $\mathbb{F}_2^m$
- Task: given $\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}$ as input, infer the planted secret $\mathbf{s}$
  - Naive amplitude amplification (starting with generic $|\phi_0\rangle$): $O(2^{n/2})$ time
  - Idea: we can prepare a special guiding state $|\phi_0\rangle$ based on $\mathbf{A}\mathbf{s} + \mathbf{e}$ such that $\theta \approx O(2^{-n/4}) \rightarrow$ an algorithm with runtime $O(2^{n/4})$!

Talk 2: Dequantising the Quartic Speedup for k-XOR

William He (Carnegie Mellon University)

- Setup:
  - $\mathbf{A} \leftarrow \mathbb{F}_2^{m\times n}$ with
  - $\mathbf{s} \leftarrow \mathbb{F}_2^n$, $\mathbf{e} \leftarrow$ sp
- Task: given $\mathbf{A}, \mathbf{As}$
  - Naive amplitud
  - Idea: we can pr $(2^{-n/4}) \to$ an algorithm with runtime $O(2^{n/4})$!

A Classical Quadratic Speedup for Planted kXOR

Meghal Gupta* William He† Ryan O'Donnell‡ Noah G. Singer§

August 14, 2025

Abstract

A recent work of Schmidhuber *et al.* (QIP, SODA, & Phys. Rev. X 2025) exhibited a quantum algorithm for the noisy planted kXOR problem running quartically faster than all known classical algorithms. In this work, we design a new classical algorithm that is quadratically faster than the best previous one, in the case of large constant k. Thus for such k, the quantum speedup of Schmidhuber *et al.* becomes only quadratic (though it retains a space advantage). Our algorithm, which also works in the semirandom case, combines tools from sublinear-time algorithms (essentially, the birthday paradox) and polynomial anticoncentration.

Next Up: Breaking Quantum Speedups

Robin Kothari (Google Quantum AI)

No exponential quantum speedup for SIS^∞ anymore

Robin Kothari* Ryan O'Donnell† Kewen Wu‡

Abstract

In 2021, Chen, Liu, and Zhandry presented an efficient quantum algorithm for the average-case ℓ_∞-Short Integer Solution (SIS^∞) problem, in a parameter range outside the normal range of cryptographic interest, but still with no known efficient classical algorithm. This was particularly exciting since SIS^∞ is a simple problem without structure, and their algorithmic techniques were different from those used in prior exponential quantum speedups.

We present efficient classical algorithms for all of the SIS^∞ and (more general) Constrained Integer Solution problems studied in their paper, showing there is no exponential quantum speedup anymore.

Next Up: Breaking Quantum Speedups

Robin Kothari (Google Quantum AI)

William He (Carnegie Mellon University)

No exponential quantum speedup for SIS^∞ anymore

Robin Kothari[*] Ryan O'Donnell[†] Kewen Wu[‡]

Abstract

In 2021, Chen, Liu, and Zhandry presented an efficient quantum algorithm for the average-case ℓ_∞-Short Integer Solution (SIS^∞) problem, in a parameter range outside the normal range of cryptographic interest, but still with no known efficient classical algorithm. This was particularly exciting since SIS^∞ is a simple problem without structure, and their algorithmic techniques were different from those used in prior exponential quantum speedups.

We present efficient classical algorithms for all of the SIS^∞ and (more general) Constrained Integer Solution problems studied in their paper, showing there is no exponential quantum speedup anymore.

A Classical Quadratic Speedup for Planted kXOR

Meghal Gupta[*] William He[†] Ryan O'Donnell[‡] Noah G. Singer[§]

August 14, 2025

Abstract

A recent work of Schmidhuber *et al.* (QIP, SODA, & Phys. Rev. X 2025) exhibited a quantum algorithm for the noisy planted kXOR problem running quartically faster than all known classical algorithms. In this work, we design a new classical algorithm that is quadratically faster than the best previous one, in the case of large constant k. Thus for such k, the quantum speedup of Schmidhuber *et al.* becomes only quadratic (though it retains a space advantage). Our algorithm, which also works in the semirandom case, combines tools from sublinear-time algorithms (essentially, the birthday paradox) and polynomial anticoncentration.

Next Up: Breaking Quantum Speedups

Robin Kothari (Google Quantum AI)

William He (Carnegie Mellon University)

No exponential quantum speedup for SIS^∞ anymore

Robin Kothari* Ryan O'Donnell† Kewen Wu‡

Abstract

In 2021, Chen, Liu, and Zhandry presented an efficient quantum algorithm for the average-case ℓ_∞-Short Integer Solution (SIS^∞) problem, in a parameter range outside the normal range of cryptographic interest, but still with no known efficient classical algorithm. This was particularly exciting since SIS^∞ is a simple problem without structure, and their algorithmic techniques were different from those used in prior exponential quantum speedups.

We present efficient classical algorithms for all of the SIS^∞ and (more general) Constrained Integer Solution problems studied in their paper, showing there is no exponential quantum speedup anymore.

A Classical Quadratic Speedup for Planted kXOR

Meghal Gupta* William He† Ryan O'Donnell‡ Noah G. Singer§

August 14, 2025

Abstract

A recent work of Schmidhuber *et al.* (QIP, SODA, & Phys. Rev. X 2025) exhibited a quantum algorithm for the noisy planted kXOR problem running quartically faster than all known classical algorithms. In this work, we design a new classical algorithm that is quadratically faster than the best previous one, in the case of large constant k. Thus for such k, the quantum speedup of Schmidhuber *et al.* becomes only quadratic (though it retains a space advantage). Our algorithm, which also works in the semirandom case, combines tools from sublinear-time algorithms (essentially, the birthday paradox) and polynomial anticoncentration.

Thank you! Questions?