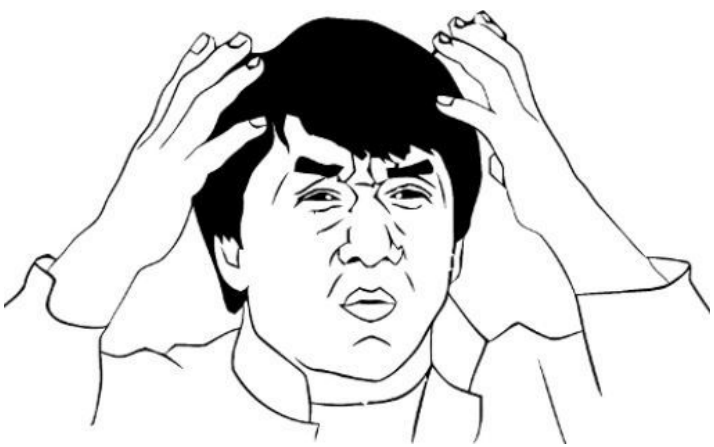
LWE with Quantum Amplitudes

Yilei Chen

Tsinghua University

Lattice problems that are conjectured hard against quantum computers:

- Short vector problems (SVP)
- Short integer solution (SIS)
- Learning with errors (LWE)



Are they really hard against quantum computers?

Is Learning with Error (LWE) problem in the complexity class BQP?

The Learning with Error (LWE) problem is believed to be in the complexity class BQP, but this has not been proven. BQP stands for "Bounded-Error Quantum Polynomial" and is the class of decision problems solvable by a quantum computer in polynomial time, with an error probability of at most 1/3 for all instances. The LWE problem is a mathematical problem in cryptography that involves solving a system of linear equations over a finite field. It is believed to be computationally hard for classical computers, but can be solved efficiently on a quantum computer.

Thus spoke ChatGPT in 2022

Plan of the talk

- Introducing the LWE problem
- Some basic ideas of quantumly solving LWE/SIS
- Quantumly solving S|LWE> for certain error amplitudes using “filtering” [Chen, Liu, Zhandry 22]
- S|LWE> for Gaussian amplitudes: algorithms and hardness [Chen, Hu, Liu, Luo, Tu 25]
- Complex Gaussian in https://eprint.iacr.org/2024/555.pdf

What is the learning with errors problem (LWE)?

What is the learning with errors problem (LWE)?

$s = [\ s_1\ , s_2\ , s_3\ , s_4\ ]$ is the secret vector.

You are given an oracle O_s( ). Over one click, returns a random linear combination of the secret, plus a small amount of noize

What is the learning with errors problem (LWE)?

$s = [\, s_1 , s_2 , s_3 , s_4 \,]$ is the secret vector.

You are given an oracle O_s( ). Over one click, returns a random linear combination of the secret, plus a small amount of noize (think of ≈ as + or - a small number)

$$34\, s_1 + 12\, s_2 + 39\, s_3 + 16\, s_4 \approx 38$$

mod 67

What is the learning with errors problem (LWE)?

$s = [\, s_1 , s_2 , s_3 , s_4 \,]$ is the secret vector.
You are given an oracle O_s( ). Over one click, returns a random linear combination of the secret, plus a small amount of noize (think of ≈ as + or - a small number)

$$34\, s_1 + 12\, s_2 + 39\, s_3 + 16\, s_4 \approx 38$$
$$63\, s_1 + 29\, s_2 + 17\, s_3 + 7\, s_4 \approx 22$$

mod 67

What is the learning with errors problem (LWE)?

$s = [\, s_1 , s_2 , s_3 , s_4 \,]$ is the secret vector.
You are given an oracle O_s( ). Over one click, returns a random linear combination of the secret, plus a small amount of noize (think of ≈ as + or - a small number)

$$
\begin{aligned}
34\, s_1 + 12\, s_2 + 39\, s_3 + 16\, s_4 &\approx 38 \\
63\, s_1 + 29\, s_2 + 17\, s_3 + 7\, s_4 &\approx 22 \\
9\, s_1 + 31\, s_2 + 52\, s_3 + 14\, s_4 &\approx 1 \\
54\, s_1 + 18\, s_2 + 43\, s_3 + 61\, s_4 &\approx 59 \\
19\, s_1 + 27\, s_2 + 53\, s_3 + 13\, s_4 &\approx 15 \\
&\;\;\cdots \\
24\, s_1 + 50\, s_2 + 3\, s_3 + 36\, s_4 &\approx 58
\end{aligned}
\qquad \text{mod } 67
$$

LWE: given the coefficients, the answers, find the secret vector.

What is learning without errors?

$$
\begin{aligned}
34\, s_1 + 12\, s_2 + 39\, s_3 + 16\, s_4 &= 38 \\
63\, s_1 + 29\, s_2 + 17\, s_3 + 7\, s_4 &= 22 \\
9\, s_1 + 31\, s_2 + 52\, s_3 + 14\, s_4 &= 1 \\
54\, s_1 + 18\, s_2 + 43\, s_3 + 61\, s_4 &= 59 \qquad \text{mod } 67 \\
19\, s_1 + 27\, s_2 + 53\, s_3 + 13\, s_4 &= 15 \\
&\;\;\vdots \\
24\, s_1 + 50\, s_2 + 3\, s_3 + 36\, s_4 &= 58
\end{aligned}
$$

$[\, s_1, s_2, s_3, s_4 \,]$ is the secret vector.

Learning without errors is easy: Gaussian elimination.

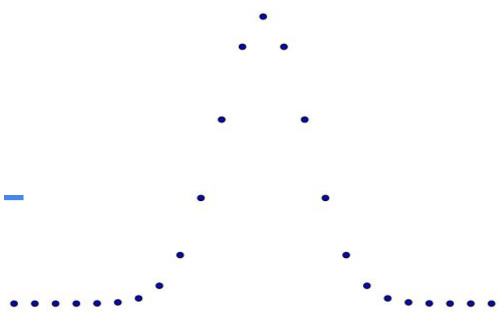
Learning with errors [Regev 2009]

$s = [\ s_1\ ,\ s_2\ , \ldots ,\ s_n\ ]$ is the secret vector.

Given samples of the form

$$a_1\ ,\quad y_1 = s \cdot a_1 + e_1 \mod q$$

e <---



$$\ldots$$

exp(- x^2/s^2)

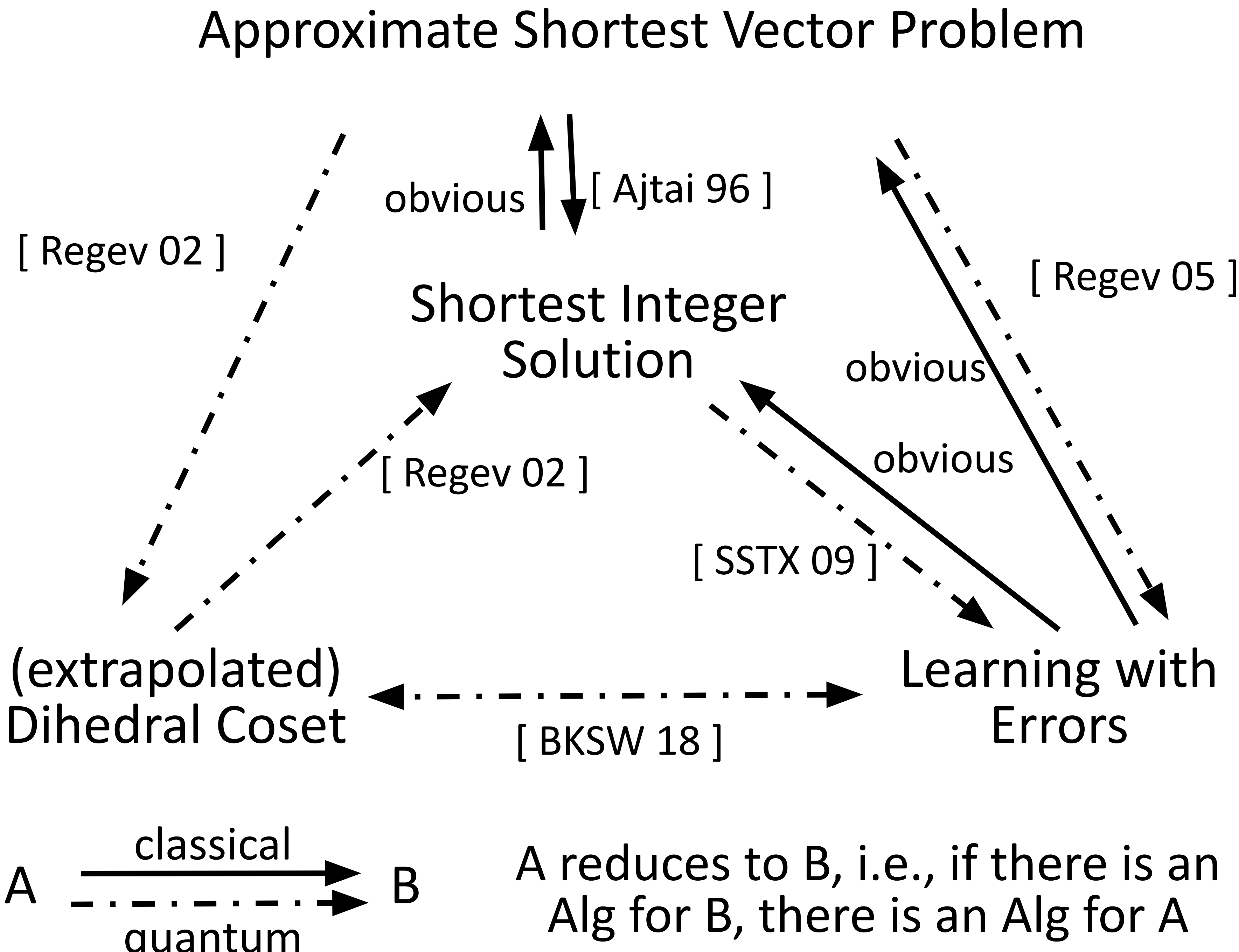
$$a_m\ ,\quad y_m = s \cdot a_m + e_m \mod q$$

Goal: find the secret vector (or the error vector).

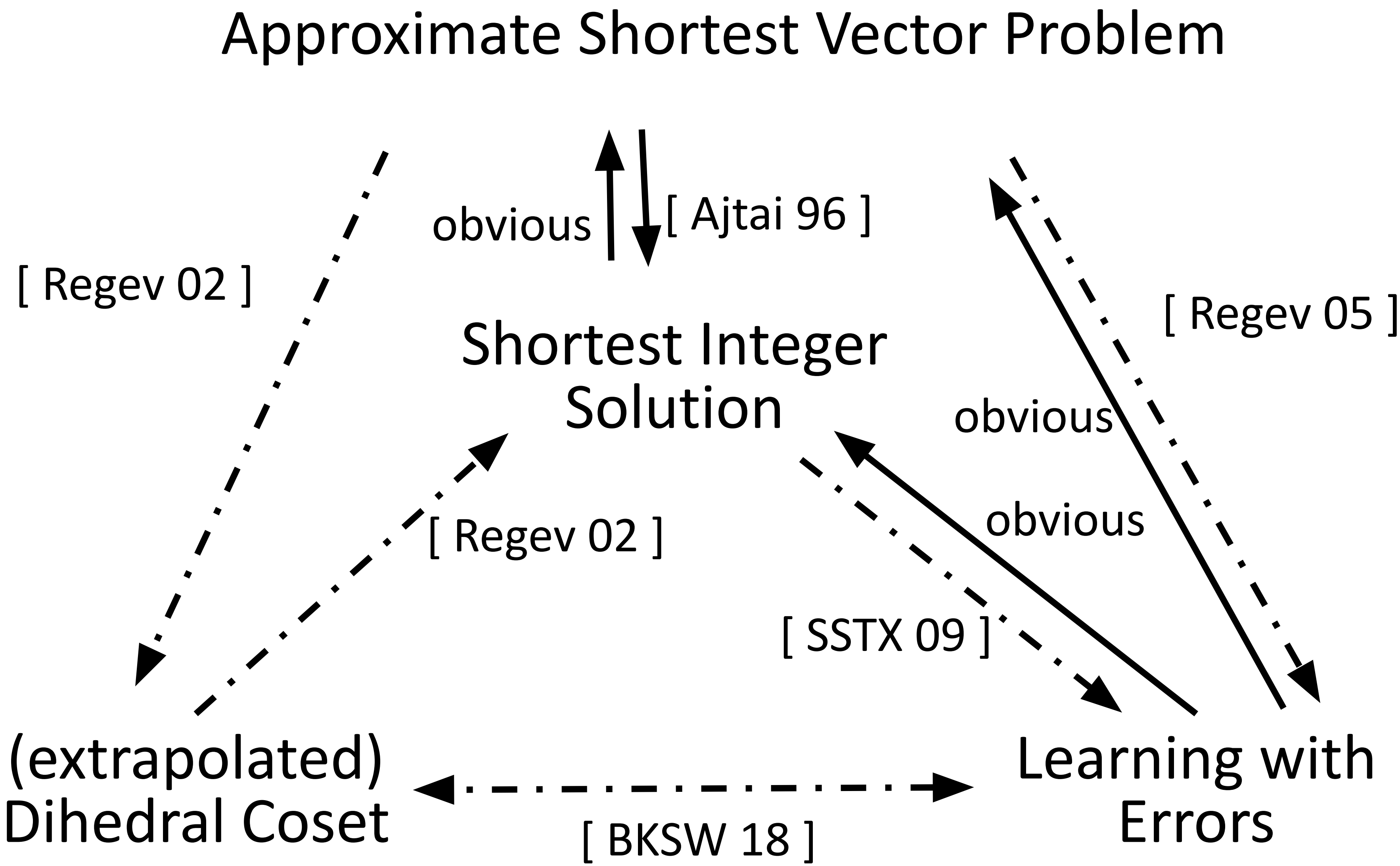
Typical parameters: $q = O(n^2)$, $m = \mathrm{poly}(n)$, $s >= 2*\mathrm{sqrt}(n)$

Plan of the talk

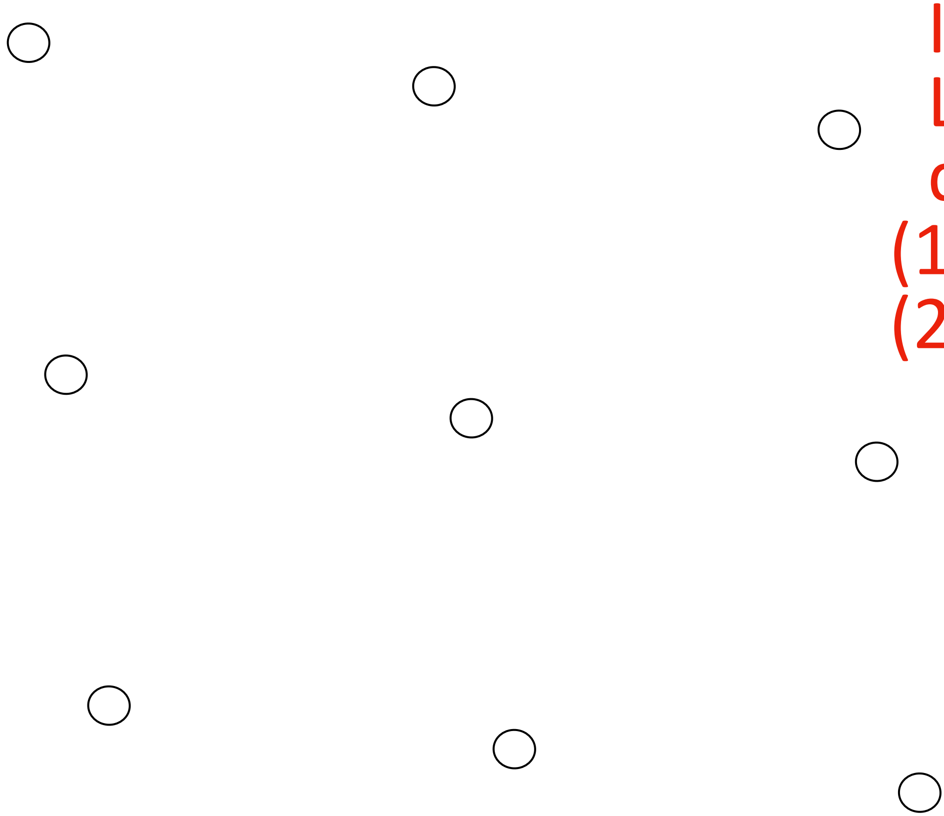
- Introducing the LWE problem
- Some basic ideas of quantumly solving LWE/SIS
- Quantumly solving S|LWE> for certain error amplitudes using “filtering” [Chen, Liu, Zhandry 22]
- S|LWE> for Gaussian amplitudes: algorithms and hardness [Chen, Hu, Liu, Luo, Tu 25]
- Complex Gaussian in https://eprint.iacr.org/2024/555.pdf



Approximate Shortest Vector Problem
obvious
[Ajtai 96]
[Regev 02]
[Regev 05]
Shortest Integer Solution
obvious
obvious
[Regev 02]
[SSTX 09]
(extrapolated) Dihedral Coset
Learning with Errors
[BKSW 18]
A
classical
quantum
B
A reduces to B, i.e., if there is an Alg for B, there is an Alg for A



Idea 0: if you solve one of the LWE-complete problems, you solve all of them.

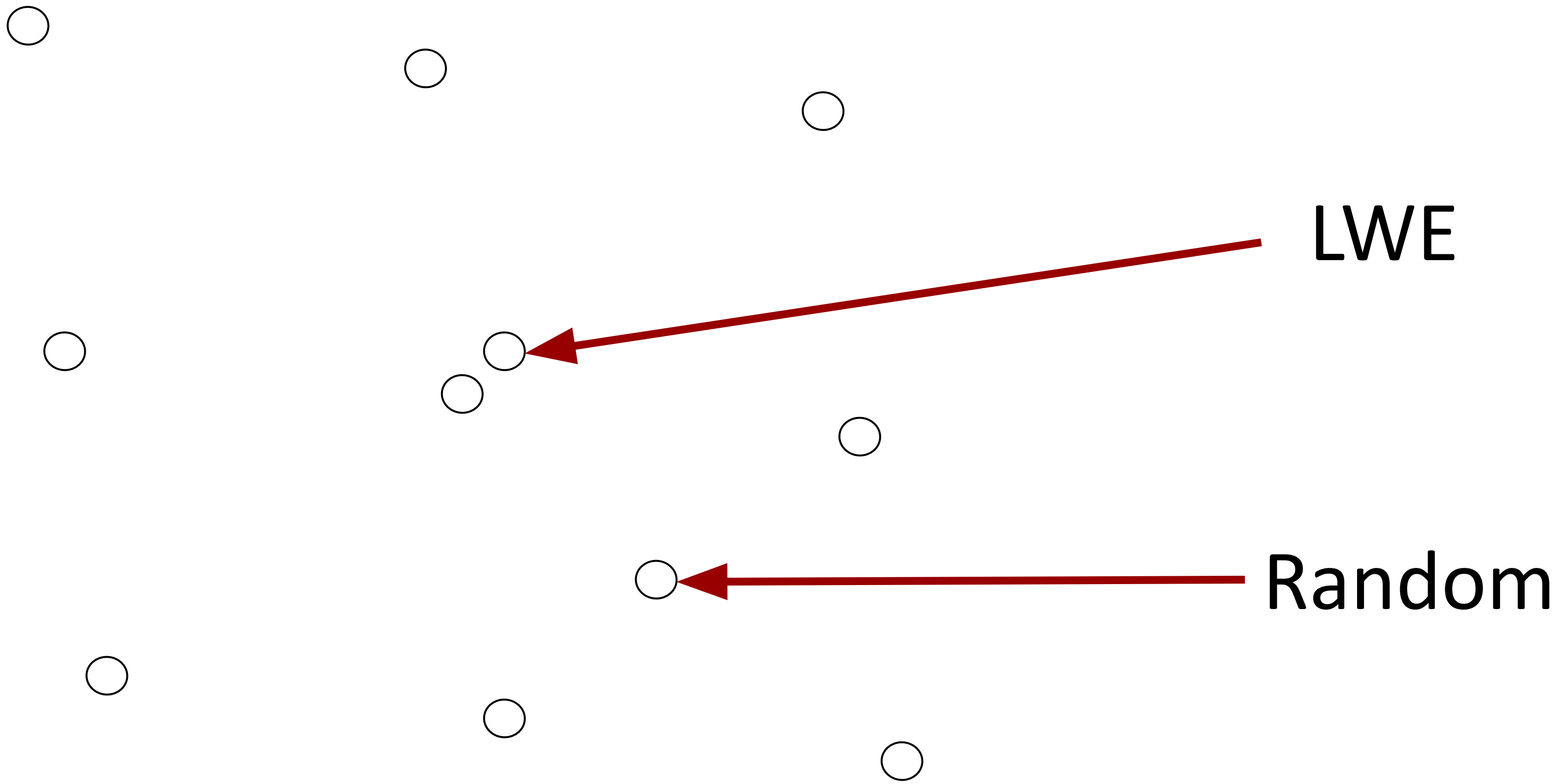


Idea 1: Solving decisional LWE: given A, y, distinguish whether

(1) y is like sA+e, or
(2) y is random

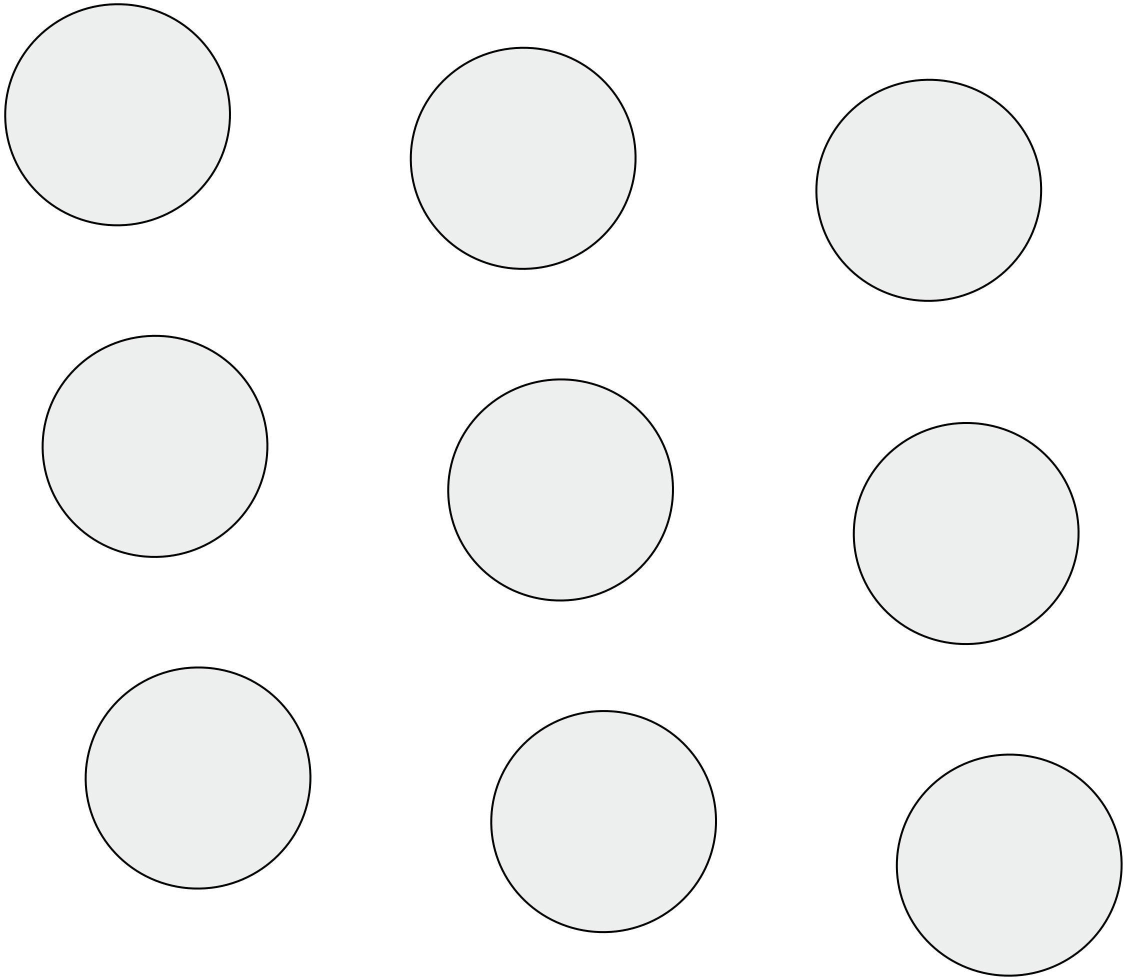
L = { z = As mod q for some s }

A very intuitive quantum idea of solving decisional LWE (that is not intuitively working)



$L = \{ z = As \bmod q \text{ for some } s \}$

A very intuitive quantum idea of solving decisional LWE (that is not intuitively working)

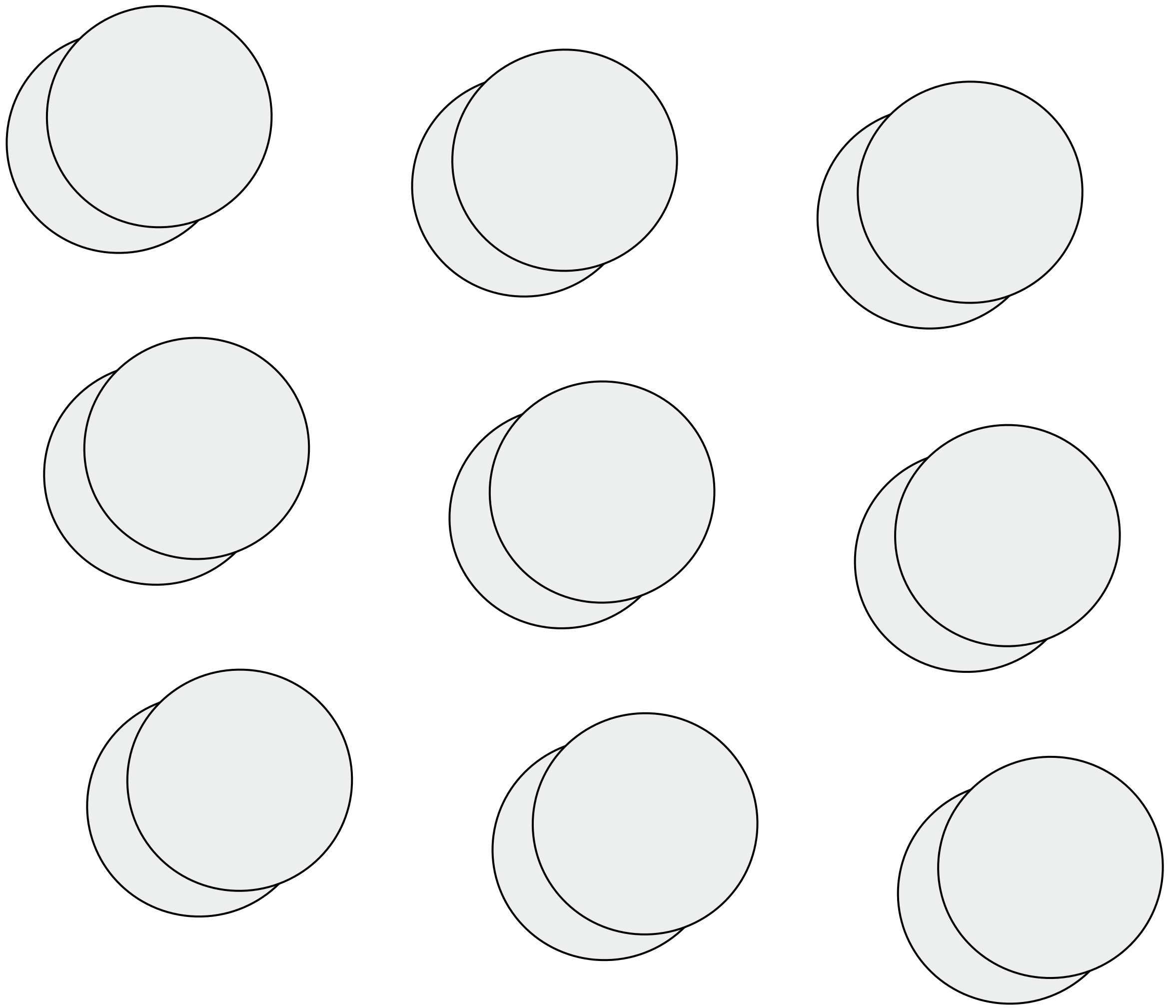


Idea:

1. prepare a uniform superposition of balls around L

$L = \{ z = As \bmod q \text{ for some } s \}$

A very intuitive quantum idea of solving decisional LWE (that is not intuitively working)



Idea:

1. prepare a uniform superposition of balls around L
2. Shift all balls by y

If y = As+e, then the overlap is large;
If y is random, then the overlap is small.

L = { z = As mod q for some s }

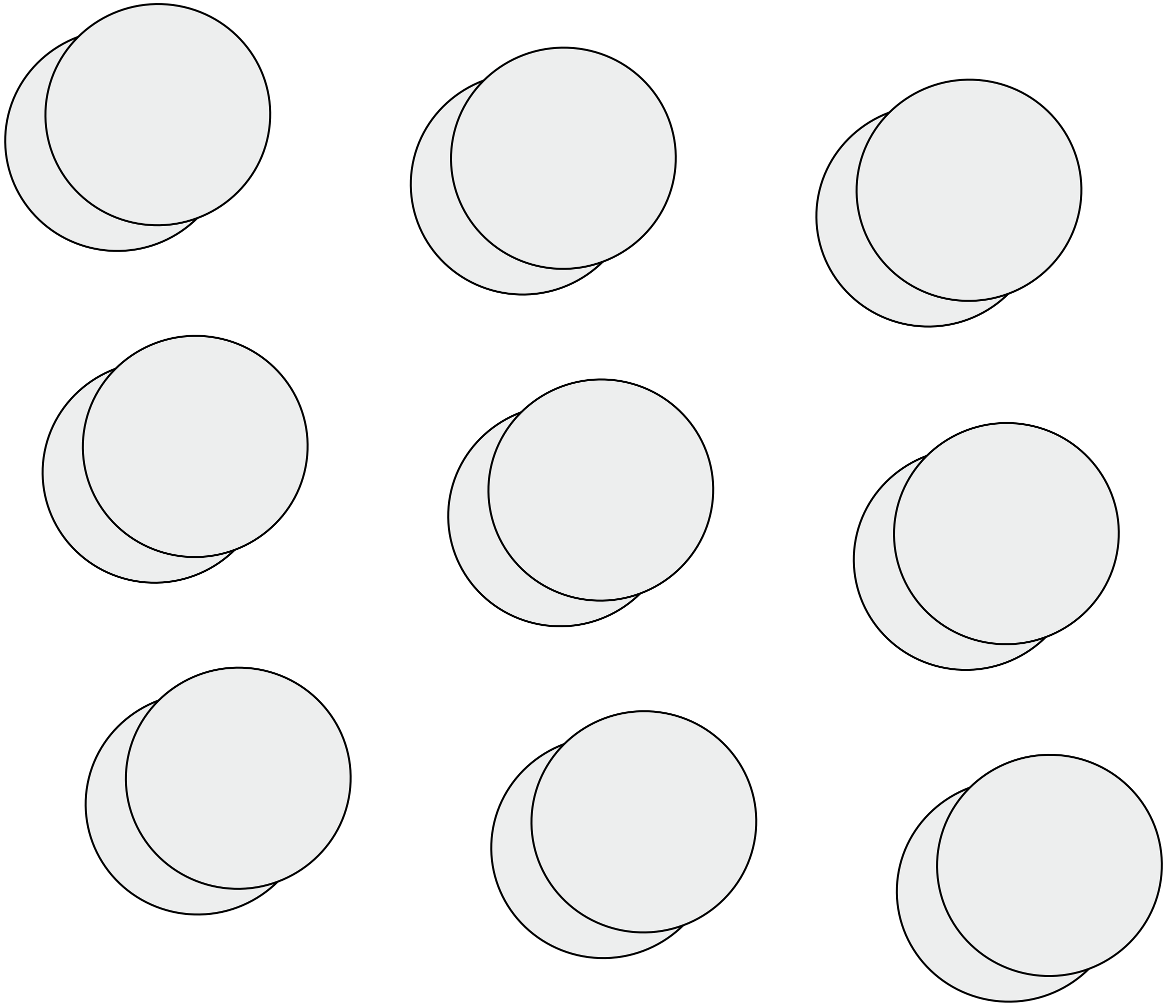
A very intuitive quantum idea of solving decisional LWE (that is not intuitively working)

Idea:

1. prepare a uniform superposition of balls around L
2. Shift all balls by y

If y = As+e, then the overlap is large;
If y is random, then the overlap is small.

Problem: don’t know how to do Step 1.



L = { z = As mod q for some s }

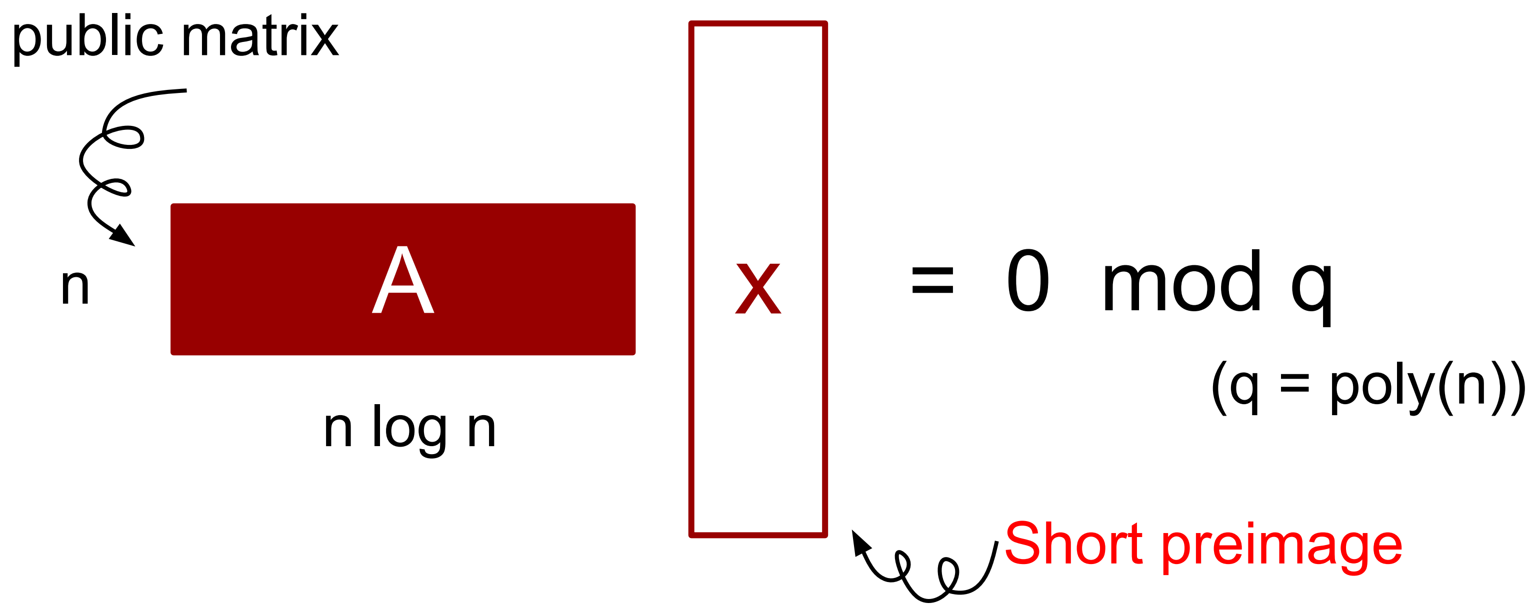
A very intuitive quantum ide
decisional LWE (that is not intu

Basic idea 2: If there is a quantum algorithm that solves LWE, then there is a quantum algorithm that solves SIS.

(Solving SIS also implies solving approximate lattice problems in general [Ajtai 96])

Basic idea 2 was initially due to [Regev 09], and later used by
(1) Stehle et al. [SSTX 09], Chen et al. [CLZ 22], Debris-Alazard et al. [DFS 24] in different lattice reductions/algorithms;
(2) [Poremba 23], [Bartusek, Khurana, Poremba 23], … for proof of deletion from lattices
(3) Extended to coding problems [Yamakawa, Zhandry 22], [Debris-Alazard, Remaud, Tillich 24], [Jordan et al 25], [Chailloux, Tillich 25], …, promising for showing quantum advantages.

Short integer solution (SIS)



Short integer solution [Ajtai 96]:

Given a random matrix A, find a non-zero vector x such that

$$Ax = 0 \bmod q \quad \& \quad |x|_2 < B \quad \text{for some } B < q$$

Basic idea 2: If there is a quantum algorithm that solves LWE, then there is a quantum algorithm that solves SIS

0: $\sum_s |s> \quad \sum_e f(e)|e>$ (think of f as Gaussian)

Basic idea 2: If there is a quantum algorithm that solves LWE, then there is a quantum algorithm that solves SIS

0: $\sum_s |s>\ \ \sum_e f(e)|e>$

Compute +sA in the second register:

1: $\sum_s |s>\ \ \sum_e f(e)|sA + e>$

Basic idea 2: If there is a quantum algorithm that solves LWE, then there is a quantum algorithm that solves SIS

$0: \sum_s |s> \quad \sum_e f(e)|e>$

Compute +sA in the second register:

$1: \sum_s |s> \quad \sum_e f(e)|sA + e>$

Uncompute the first register by solving LWE:

$2: \sum_s |0> \quad \sum_e f(e)|sA + e>$

$= \sum_s \sum_e f(e)|sA + e>$

Basic idea 2: If there is a quantum algorithm that solves LWE, then there is a quantum algorithm that solves SIS

0: $\sum_s |s> \ \ \sum_e f(e)|e>$

Compute +sA in the second register:

1: $\sum_s |s> \ \ \sum_e f(e)|sA + e>$

Uncompute the first register by solving LWE:

2: $\sum_s |0> \ \ \sum_e f(e)|sA + e>$

$= \sum_s \sum_e f(e)|sA + e>$

Take quantum Fourier transform:

3: $\sum_z \sum_s \sum_e f(e)\exp(<sA+e, z>/q \ )|z>$

$= \sum_z \sum_e f(e)\exp(<e, z>/q \ ) \sum_s \exp(<sAz>/q \ )|z>$

$= \sum_{z \text{ s.t. } Az = 0} FT(f)(z/q) \ |z>$

Basic idea 2: If there is a quantum algorithm that solves LWE, then there is a quantum algorithm that solves SIS

0: $\sum_s |s> \;\; \sum_e f(e)|e>$

Compute +sA in the second register:

1: $\sum_s |s> \;\; \sum_e f(e)|sA + e>$

Uncompute the first register by solving LWE:

2: $\sum_s |0> \;\; \sum_e f(e)|sA + e>$

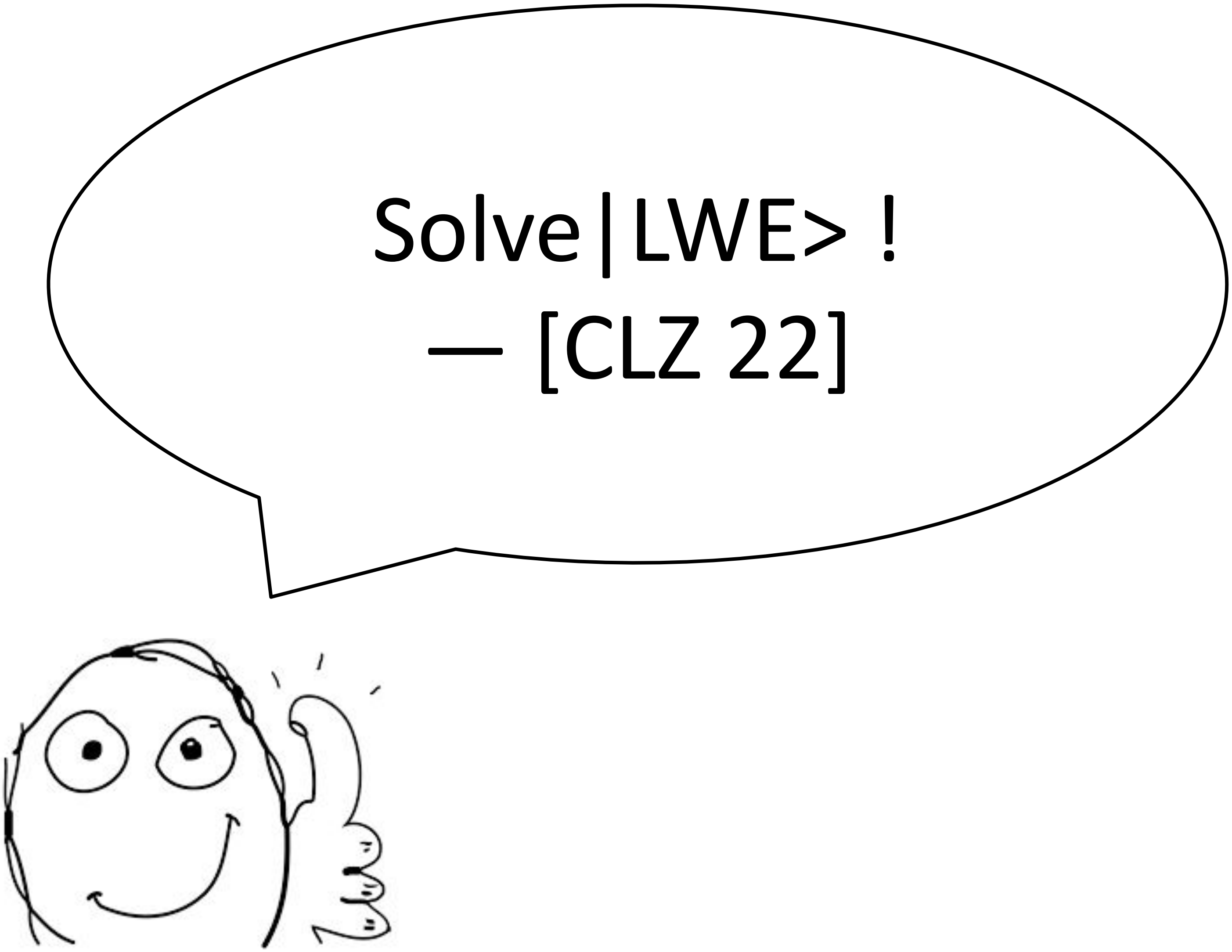
$= \sum_s \sum_e f(e)|sA + e>$

Take quantum Fourier transform:

3: $\sum_z \sum_s \sum_e f(e)\exp(<sA+e, z>/q \;)|z>$

$= \sum_z \sum_e f(e)\exp(<e, z>/q \;) \sum_s \exp(<sAz>/q \;)|z>$

$= \sum_{z \text{ s.t. } Az = 0} FT(f)(z/q) \; |z>$


Solve|LWE> !
— [CLZ 22]

Solve | Learning with errors > (S|LWE>)

$s = [s_1 , s_2 , \ldots , s_n]$ is the secret vector.

Given quantum samples of the form

$$a_1 , \quad | y_1 > = \sum_{e1 \in [0 \ldots q-1]} f(e_1) \ | s \cdot a_1 + e_1 \ \bmod q >$$

$$\ldots$$

$$a_m , \quad | y_m > = \sum_{em \in [0 \ldots q-1]} f(e_m) \ | s \cdot a_m + e_m \ \bmod q >$$

This is all we need in

$1: \sum_s |s > \sum_e f(e)|sA + e > \ \rightarrow 2: \sum_s |0 > \sum_e f(e)|sA + e >$

Solve | Learning with errors > (S|LWE>)

$s = [\, s_1 , s_2 , \ldots , s_n \,]$ is the secret vector.

Given quantum samples of the form

$$a_1 , \quad | y_1 \rangle = \sum_{e1 \in [0 \ldots q-1]} f(e_1) \; | s \cdot a_1 + e_1 \mod q \rangle$$

$$\cdots$$

$$a_m , \quad | y_m \rangle = \sum_{em \in [0 \ldots q-1]} f(e_m) \; | s \cdot a_m + e_m \mod q \rangle$$

Questions:

1. What can we say about algorithms for S|LWE>?
2. What can we say about the hardness of S|LWE>?

Solve | Learning with errors > (S|LWE>)

$s = [s_1 , s_2 , \ldots , s_n]$ is the secret vector.

Given quantum samples of the form

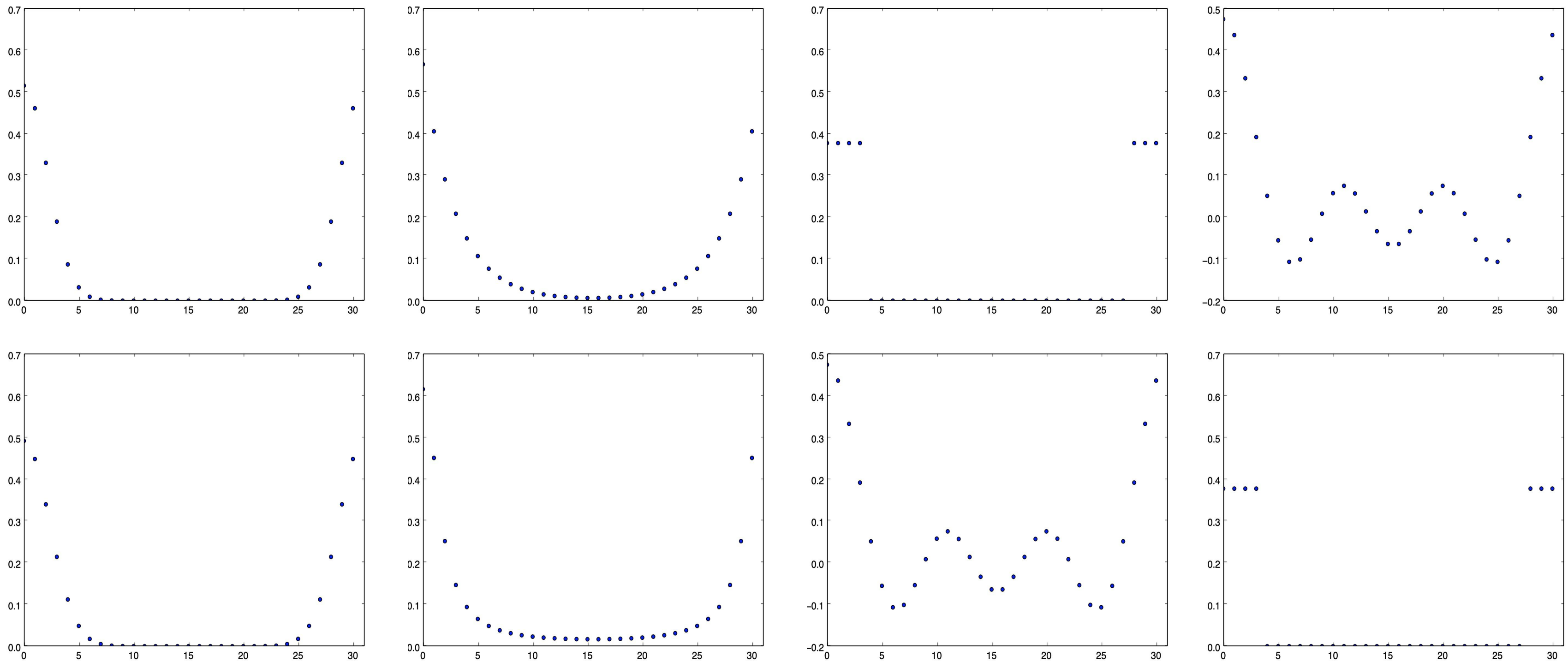
$$a_1 , \quad | y_1 > = \sum_{e1 \in [0 \ldots q-1]} f(e_1) \; | s \cdot a_1 + e_1 \mod q >$$

$$\cdots$$

$$a_m , \quad | y_m > = \sum_{em \in [0 \ldots q-1]} f(e_m) \; | s \cdot a_m + e_m \mod q >$$

[CLZ 22] A poly time quantum algorithm that finds the secret vector if the DFT of f is non-negligible over Zq and m is a sufficiently large polynomial. (E.g., when f is the bounded uniform distribution) f

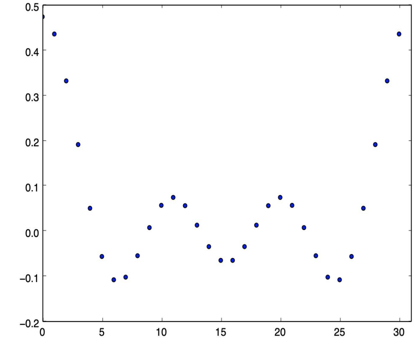
DFT(f)



Gaussian Laplacian Bounded uniform sin(x)/x

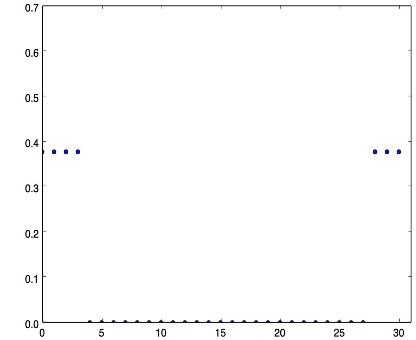
[CLZ 22] A poly time quantum algorithm that finds the secret vector if the DFT of f is non-negligible over Zq., or the DFT of f is non-negligible over Zq except for constantly many positions.

f



Application: solve a variant of SIS with infinity norm bound for some parameters.

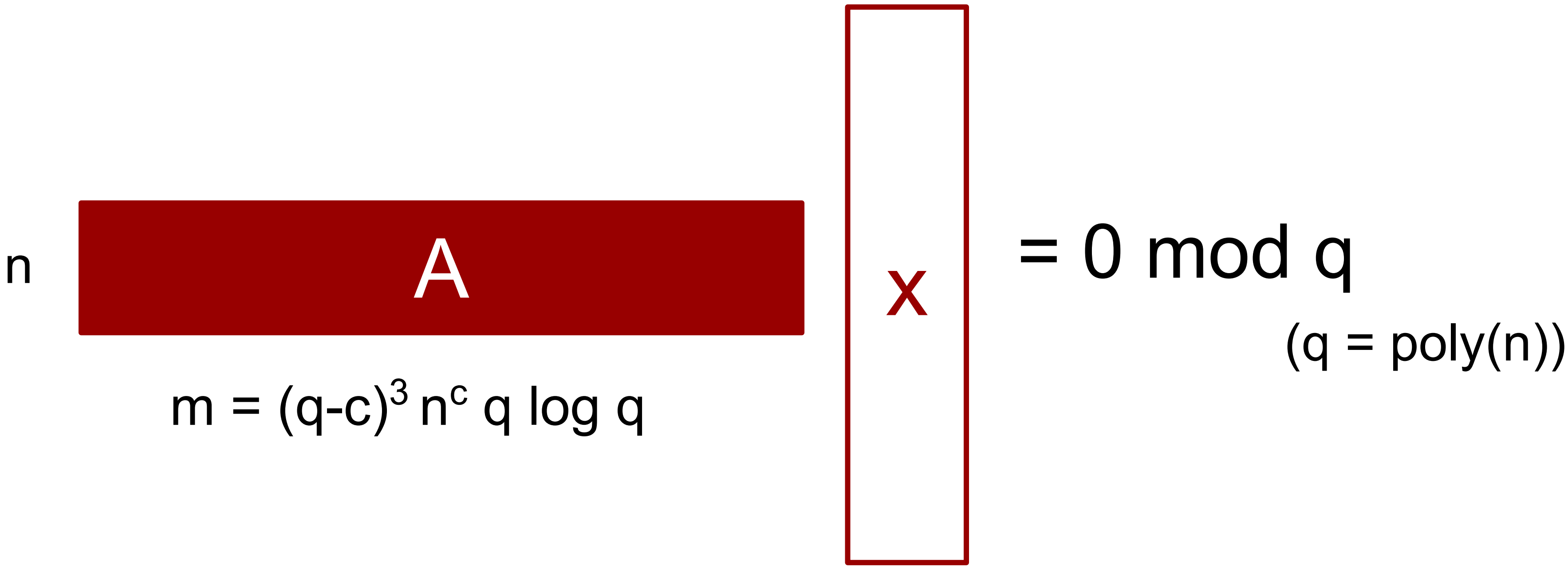
DFT(f)



sin(x)/x

[CLZ 22] A poly time quantum algorithm that finds the secret vector if the DFT of f is non-negligible over Zq., or the DFT of f is non-negligible over Zq except for constantly many positions.

Short integer solution (where x is measured by its infinity norm)



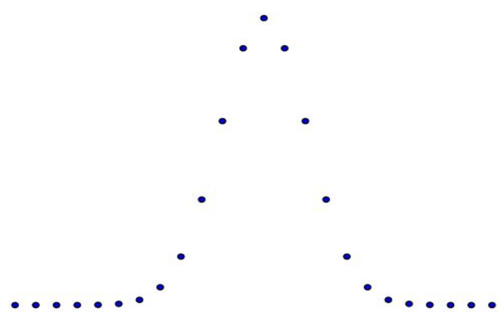
<u>CLZ22:</u> When A is very wide, can find an x with a non-trivial infinite norm in quantum polynomial time.

$$Ax = 0 \bmod q \quad \& \quad |x|_{\infty} < (q-c)/2$$

Recent: SIS^{∞} with parameters above is actually solvable classically [Imran, Ivanyos 24], [Kothari, O'Donnell, Wu 25].

How to understand an S|LWE> sample?

$$a\,, \quad |\, y > \; = \; \sum_{e \in [0 \ldots q-1]} f(e) \;\; |\, s \cdot a + e \;\; \mathrm{mod}\; q >$$



$s \cdot a$

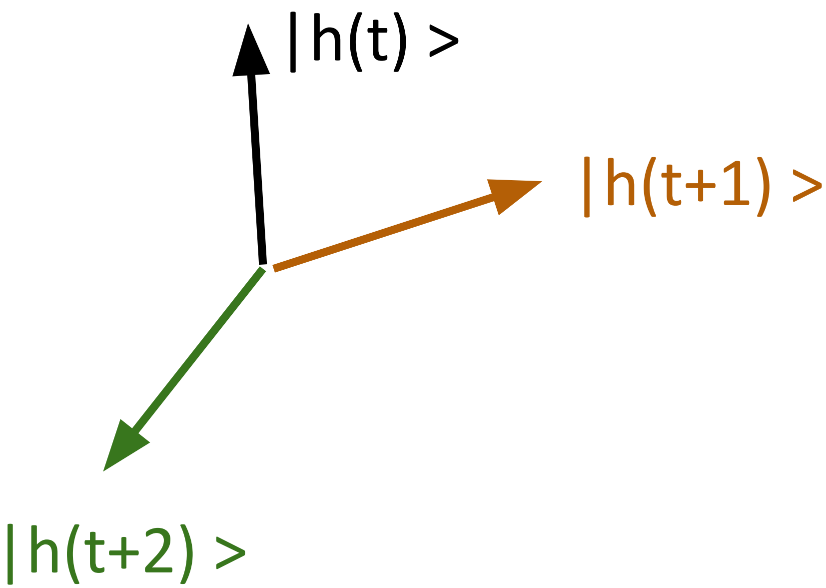
$|\, y >$ is a vector in C^q centered at $s \cdot a$

Idea of guessing $s \cdot a$

$a, \quad | y > = \sum_{e \in [0...q-1]} f(e) \; | s \cdot a + e \mod q >$

For any $t \in [0...q-1]$,
denote $|h(t) > := \sum_{e \in [0...q-1]} f(e) \; | t + e \mod q >$

In a q-dimensional space:



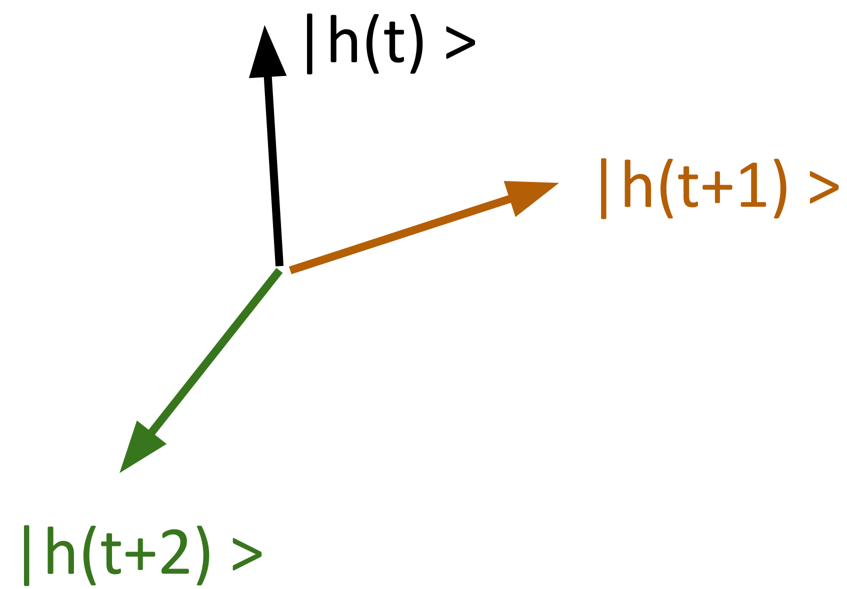
Idea of guessing $s \cdot a$

a, $|y\rangle = \sum_{e \in [0...q-1]} f(e) \; | s \cdot a + e \mod q \rangle$

For any $t \in [0...q-1]$,
denote $|h(t)\rangle := \sum_{e \in [0...q-1]} f(e) \; | t + e \mod q \rangle$

Define a matrix

[---------- h(t) ----------]
[--------- h(t+1) ---------]
[--------- h(t+2) ---------]
...
[-------- h(t+q-1)) ------]



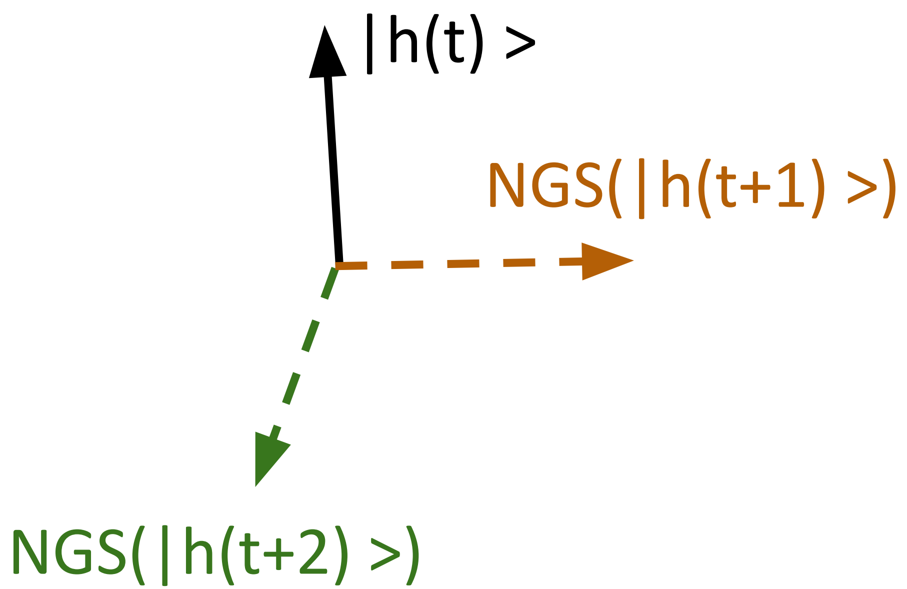
Idea of guessing $s \cdot a$

a , $| y \rangle = \sum_{e \in [0...q-1]} f(e) \; | s \cdot a + e \mod q \rangle$

For any $t \in [0...q-1]$,
denote $|h(t)\rangle := \sum_{e \in [0...q-1]} f(e) \; | t + e \mod q \rangle$

Take normalized gram-schmidt to make it unitary

[---------- h(t) ----------]
[---- NGS(h(t+1)) -----]
[---- NGS(h(t+2)) -----]
...
[---- NGS(h(t+q-1)) --]



Idea of guessing $s \cdot a$

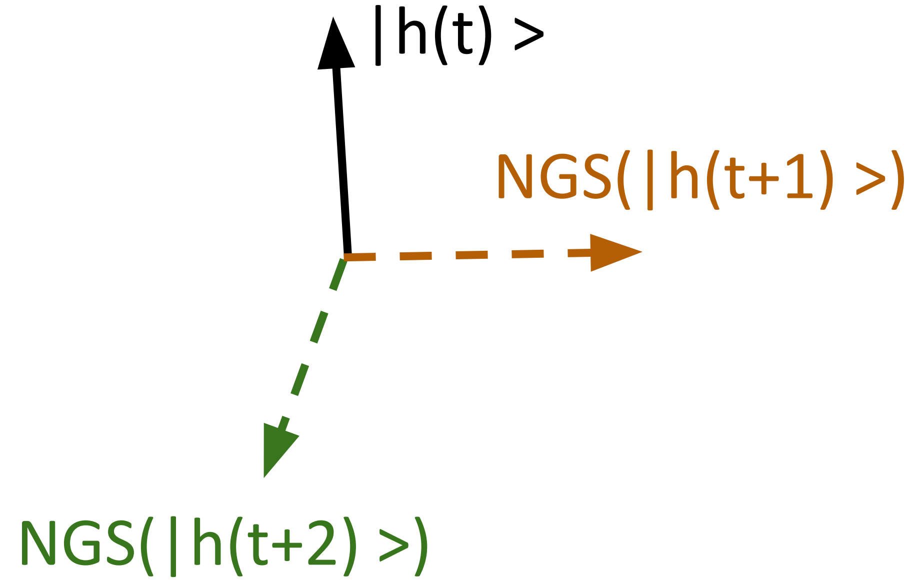
a, $|y\rangle = \sum_{e \in [0...q-1]} f(e) \; | s \cdot a + e \mod q \rangle$

1. Pick a random $t \in [0...q-1]$,

Denote $|h(t)\rangle := \sum_{e \in [0...q-1]} f(e) \; | t + e \mod q \rangle$

2. Define a unitary matrix

$U_t = \sum_{i \in [0...q-1]} | i \rangle \langle NGS(h(t+i)) |$ (NGS = Normalized Gram-Schmidt)



Idea of guessing $s \cdot a$

a, $\quad | y \rangle = \sum_{e \in [0 \ldots q-1]} f(e) \; | s \cdot a + e \bmod q \rangle$

1. Pick a random $t \in [0 \ldots q-1]$,
Denote $|h(t)\rangle := \sum_{e \in [0 \ldots q-1]} f(e) \; | t + e \bmod q \rangle$

2. Define a unitary matrix
$U_t = \sum_{i \in [0 \ldots q-1]} | i \rangle \langle NGS(h(t+i)) |$ (NGS = Normalized Gram-Schmidt)

3. (filtering) Apply U_t on $| y \rangle$, measure and get the result z
If $z=0$, we learned nothing.
If $z=1$, we know $s \cdot a \mathrel{!}= t$, since if $s \cdot a = t$, z must $=0$.
If $z=2$, we know $s \cdot a \mathrel{!}= t$ and $s \cdot a \mathrel{!}= t+1$.
...
If $z=q-1$, we know $s \cdot a = t+q-1 = t-1 \bmod q$!!!!!

3. (filtering) Apply U_t on | y >, measure and get the result z

If z=0, we learned nothing.

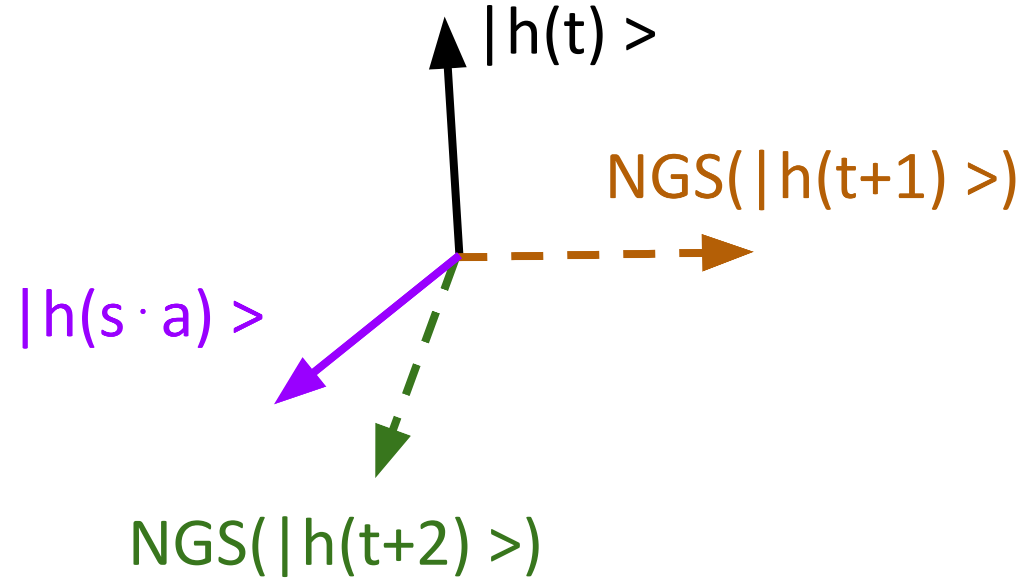
If z=1, we know s · a != t, since if s · a = t, z must =0.

If z=2, we know s · a != t and s · a != t+1.

...

If z=q-1, we know s · a = t+q-1 = t-1 mod q!!!!!

4. If z = q-1, then we guess one of s · a_i correctly. With n correct guess, we can recover s by Gaussian elimination.



3. (filtering) Apply U_t on | y >, measure and get the result z

If z=0, we learned nothing.

If z=1, we know $s \cdot a$!= t, since if $s \cdot a$ = t, z must =0.

If z=2, we know $s \cdot a$!= t and $s \cdot a$!= t+1.

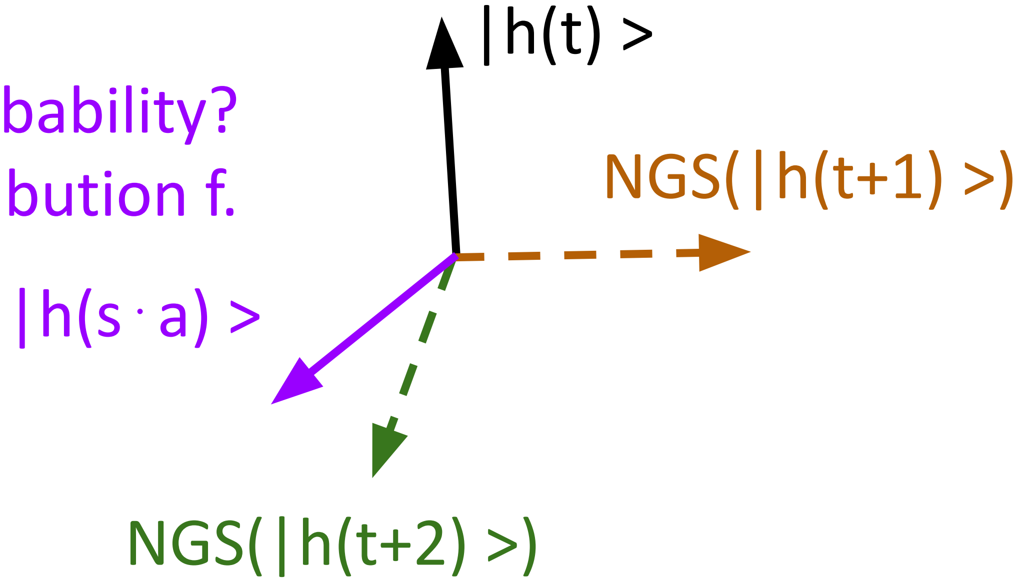
...

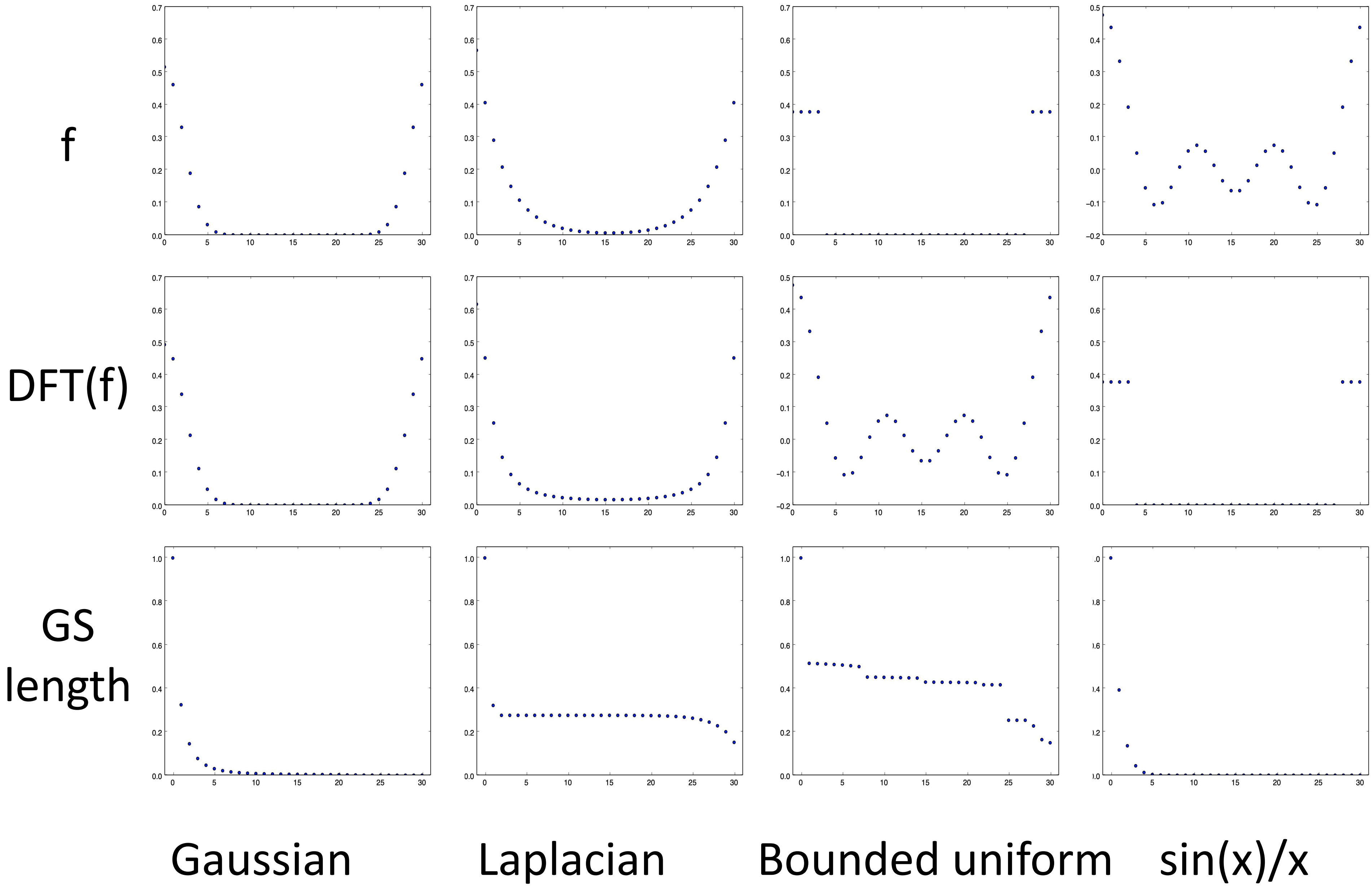
If z=q-1, we know $s \cdot a$ = t+q-1 = t-1 mod q!!!!!

4. If z = q-1, then we guess one of $s \cdot a_i$ correctly. With n correct guess, we can recover s by Gaussian elimination.

Q: How about the success probability?

A: Depends on the noise distribution f.




f
DFT(f)
GS length
Gaussian
Laplacian
Bounded uniform
sin(x)/x

Solve | Learning with errors > (S|LWE>)

$s = [\, s_1 , s_2 , \ldots , s_n \,]$ is the secret vector.

Given quantum samples of the form

$$a_j , \quad | y_j \rangle = \sum_{e_j \in [0 \ldots q-1]} f(e_j) \; | \, s \cdot a_j + e_j \mod q \rangle$$

[CLZ 22] A poly time quantum algorithm that finds the secret vector if the DFT of f is non-negligible over Zq and m is a sufficiently large polynomial. (E.g., when f is the bounded uniform distribution).

[Debris-Alazard, Fallahpour, Stehlé 24]:
A better poly time quantum algorithm for the setting above, i.e., when the DFT of f is non-negligible over Zq.

Plan of the talk

- Introducing the LWE problem
- Some basic ideas of quantumly solving LWE/SIS
- Quantumly solving S|LWE> for certain error amplitudes using “filtering” [Chen, Liu, Zhandry 22]
- S|LWE> for Gaussian amplitudes: algorithms and hardness [Chen, Hu, Liu, Luo, Tu 25]
- Complex Gaussian in https://eprint.iacr.org/2024/555.pdf

Subexponential time algorithms for S|LWE>:

$s = [\, s_1 , s_2 , \ldots , s_n \,]$ is the secret vector.

Given quantum samples of the form

$$a_j , \quad | y_j > = \sum_{e_j \in [0 \ldots q-1]} f(e_j) \; | s \cdot a_j + e_j \mod q >$$

[CHLLT 25] A subexponential time quantum algorithm for solving S|LWE> with *completely known* amplitudes.

(the amplitude f can be anything as long as DFT(f) has more than one non-negligible points, including Gaussian)

Subexponential time algorithms for S|LWE>:

$s = [\, s_1 , s_2 , \ldots , s_n \,]$ is the secret vector.

Given quantum samples of the form

$$a_j , \quad | y_j \rangle = \sum_{e_j \in [0 \ldots q-1]} f(e_j) \; | s \cdot a_j + e_j \mod q \rangle$$

[CHLLT 25] A subexponential time quantum algorithm for solving S|LWE> with *completely known* amplitudes.

Idea: Apply QFT on the S|LWE> samples

-> $\sum_k \; DFT(f)(k) e^{2\pi i k\langle a,s\rangle/q} \, |k\rangle$

-> Apply quantum rejection sampling to get $|0\rangle + e^{2\pi i\langle a,s\rangle/q} \, |1\rangle$

-> Use Kuperberg sieve: given a, $|0\rangle + e^{2\pi i\langle a,s\rangle/q} \, |1\rangle$, find s

(needs exp(sqrt{n}) many samples)

Summary of [CHLLT 25]:

S|LWE> with *completely known* amplitudes (Gaussian or others): solvable by subexponential time quantum algorithms.

S|LWE> with Gaussian amplitudes with *unknown* phases: quantumly as hard as standard LWE or GapSVP.

An improvement of Bai, Jangir, Kirshanova, Ngo, Youmans. [BJKNY25]:

S|LWE> with *completely known* Gaussian amplitudes is solvable by quasipolynomial time quantum algorithms, when *the modulus is a power of two*.

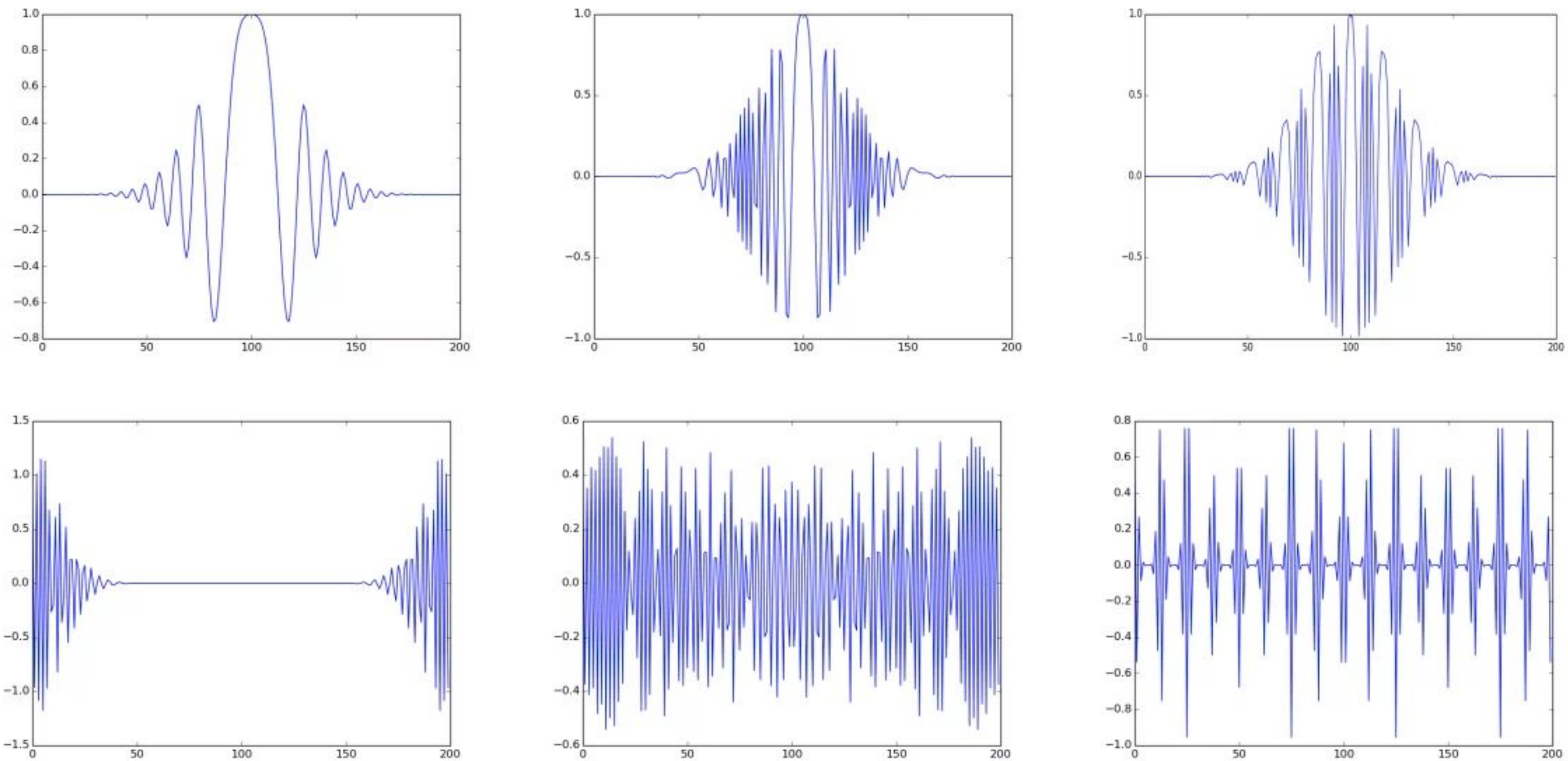
Plan of the talk

- Introducing the LWE problem
- Some basic ideas of quantumly solving LWE/SIS
- Quantumly solving S|LWE> for certain error amplitudes using "filtering" [Chen, Liu, Zhandry 22]
- S|LWE> for Gaussian amplitudes: algorithms and hardness [Chen, Hu, Liu, Luo, Tu 25]
- Complex Gaussian in https://eprint.iacr.org/2024/555.pdf


Experience from before: we need a good amplitude function!

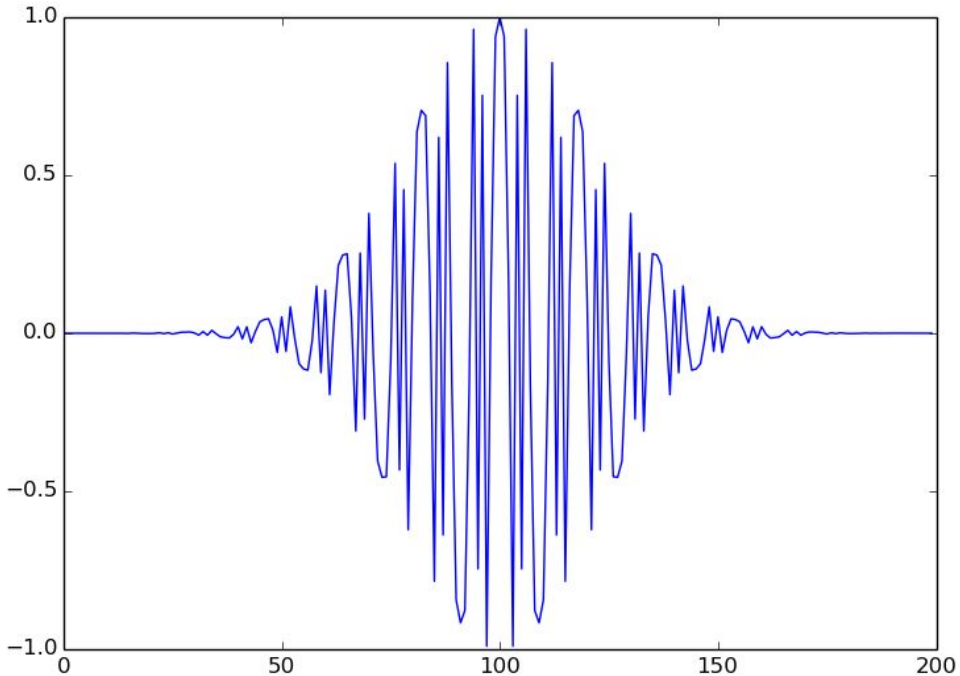
Gaussian with complex variance

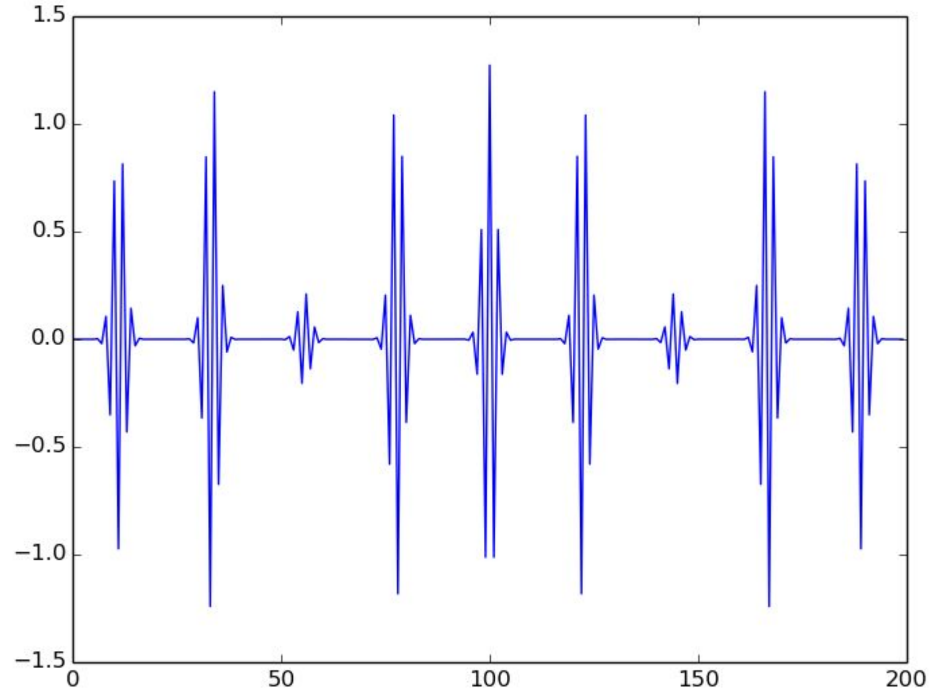
$$f(x) = \exp(\, -\pi\, (a+bi)\, x^2\,)$$



Complex Gaussian

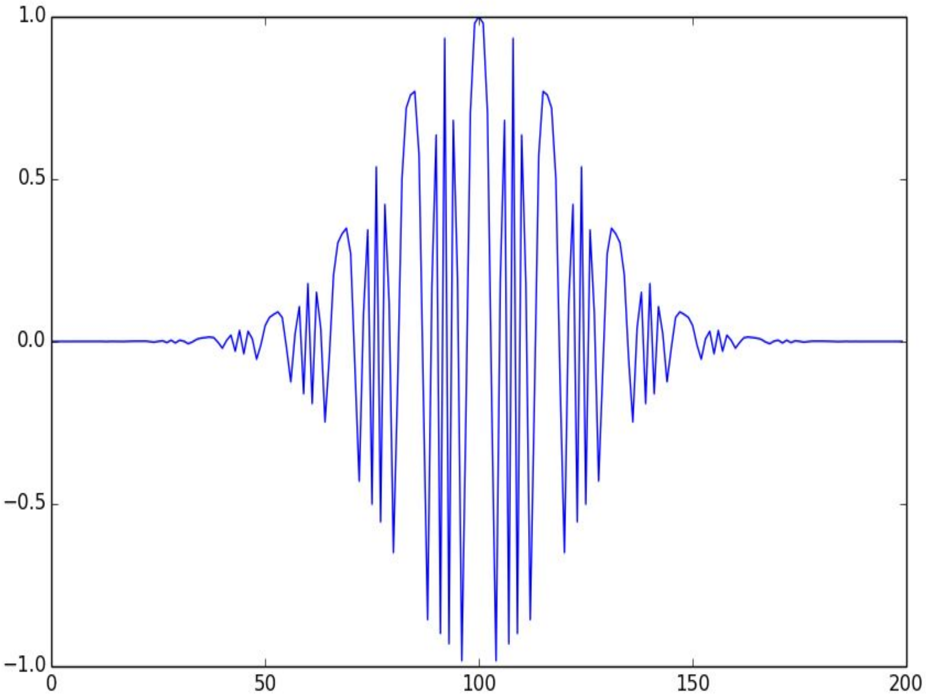
('cg, with r, s, c, q = ', 54, 3.00001, 100, 200)
('s^2 r^4/(s^4+r^4) = ', 8.999974265319997)

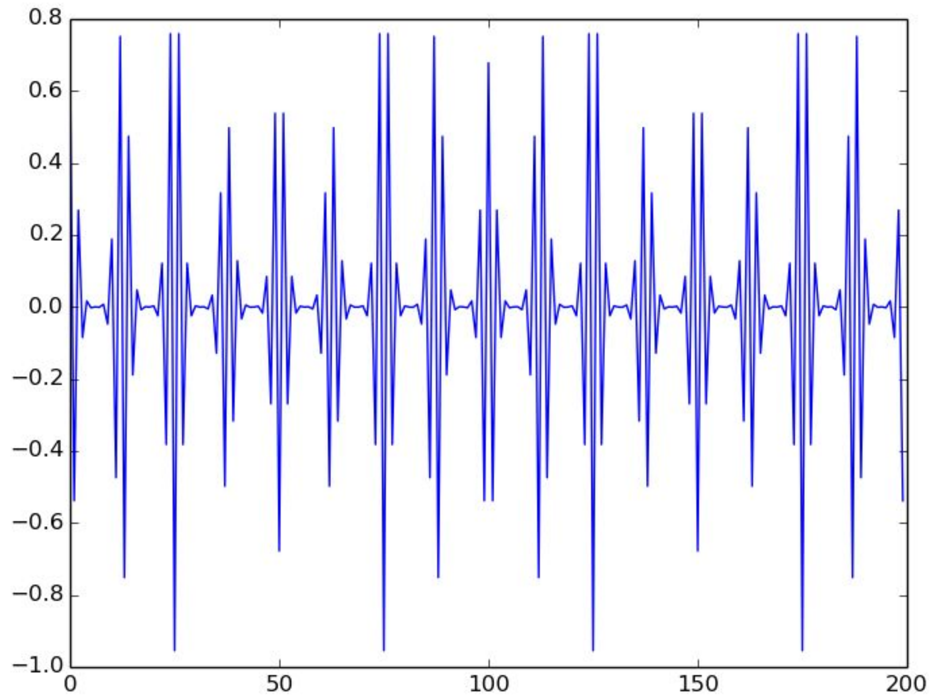




Complex Gaussian

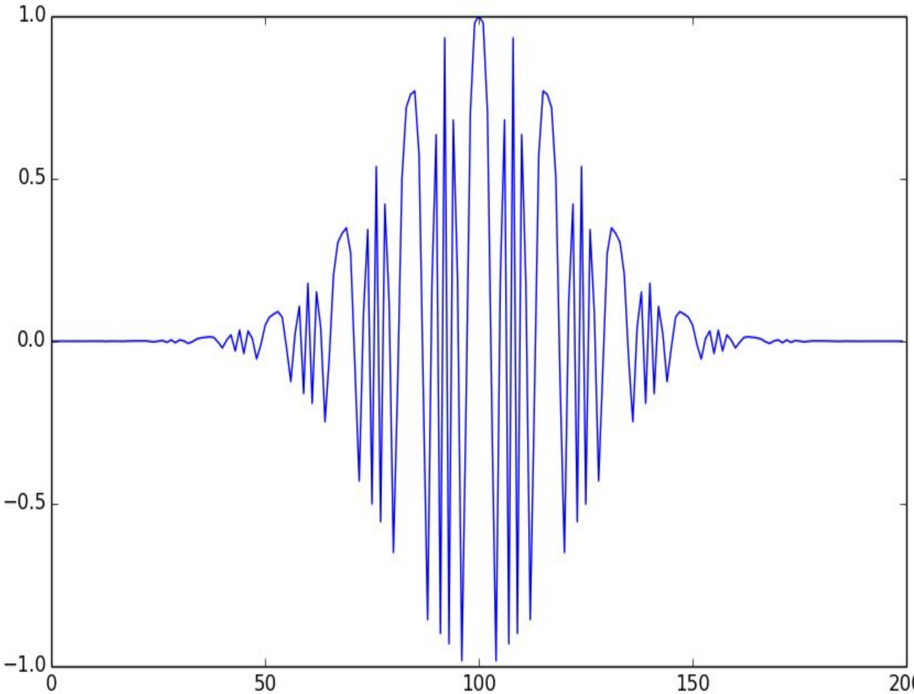
('cg, with r, s, c, q = ', 54, 4.0001, 100, 200)
('s^2 r^4/(s^4+r^4) = ', 16.0003182430807)

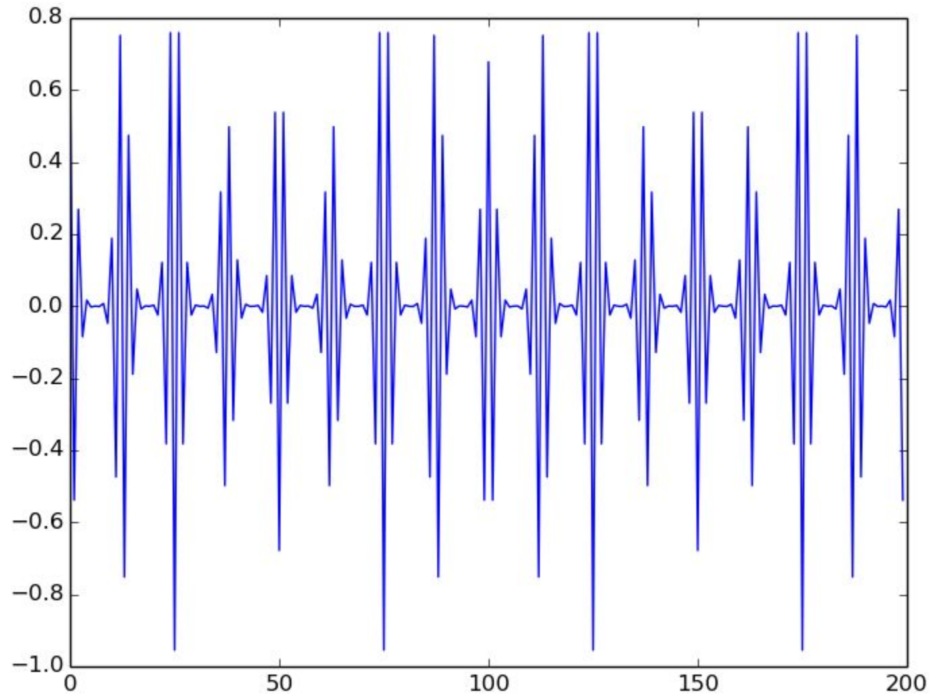




Takeaway from Complex Gaussian:

- For f(x) = exp(-π (1/r^2+i/T) x^2), it is easy to find the center of the state mod T. [CHLLT 25]
- The complex Gaussian amplitude is useful for reducing LWE from a large modulus to a smaller modulus.
- How to use it for solving standard LWE: still don't know.





LWE with Quantum Amplitudes

Yilei Chen

Thanks for your time!