On Lattices, Learning with Errors, Random Linear Codes, and Cryptography

Oded Regev
Tel-Aviv University

Outline

- Introduction to lattices
- Main theorem: a hard learning problem
- Application: a stronger and more efficient public key cryptosystem
- Proof of main theorem
  - Overview
  - Part I: Quantum
  - Part II: Classical

Lattices

Basis:

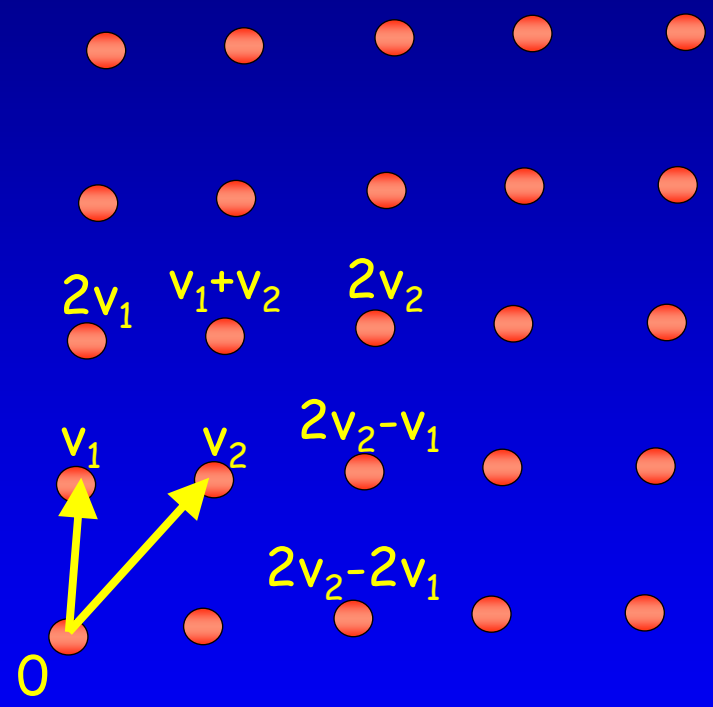
$v_1,\ldots,v_n$ vectors in R^n

The lattice L is

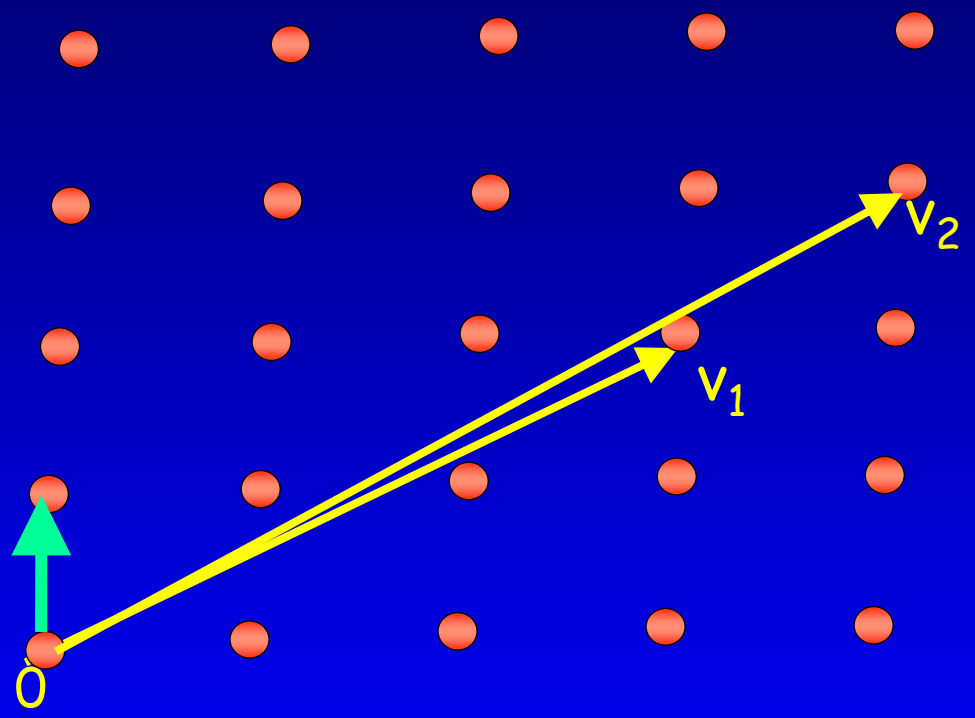
$L=\{a_1v_1+\ldots+a_nv_n \mid a_i \text{ integers}\}$

The dual lattice of L is

$L^*=\{x \mid 8\ y2L,\ hx,yi\ 2\ Z\}$

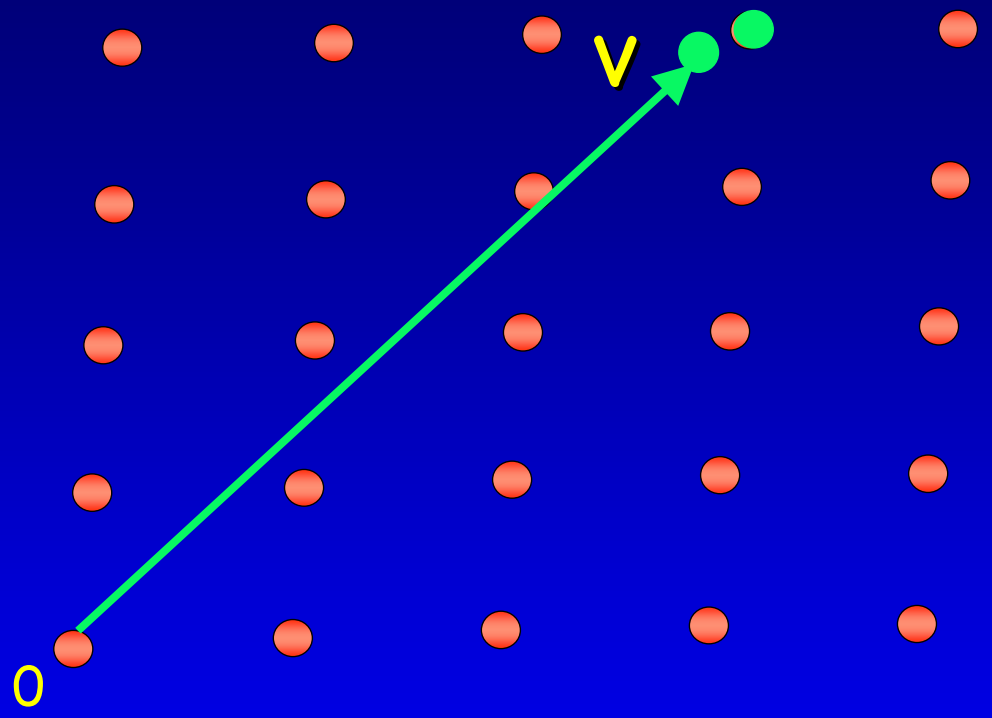


Shortest Vector Problem (SVP)



- SVP: Given a lattice, find an approximately shortest vector

Closest Vector Problem (CVP_d)



- CVP_d: Given a lattice and a target vector within distance d, find the closest lattice point

Main Theorem

Hardness of Learning

Learning from parity with error

- Let $s \in Z_2^n$ be a secret
- We have random equations modulo 2 with error (everything independent):

$$
\begin{aligned}
s_2+s_3+s_4+\ s_6+\ldots+s_n &\approx 0\\
s_1+s_2+\ \ s_4+\ \ s_6+\ldots+s_n &\approx 1\\
s_1+\ \ s_3+s_4+s_5+\ \ \ldots+s_n &\approx 1\\
\ \ s_2+s_3+s_4+\ \ \ s_6+\ldots+s_n &\approx 0\\
&\ \ \vdots
\end{aligned}
$$

- Without error, it's easy!

Learning from parity with error

- More formally, we need to learn s from samples of the form $(t, st+e)$ where t is chosen uniformly from Z_2^n and e is a bit that is 1 with probability 10%.
- Easy algorithms need $2^{O(n)}$ equations/time
- Best algorithm needs $2^{O(n/\log n)}$ equations/time [BlumKalaiWasserman'00]

- Open question: why is this problem so hard?

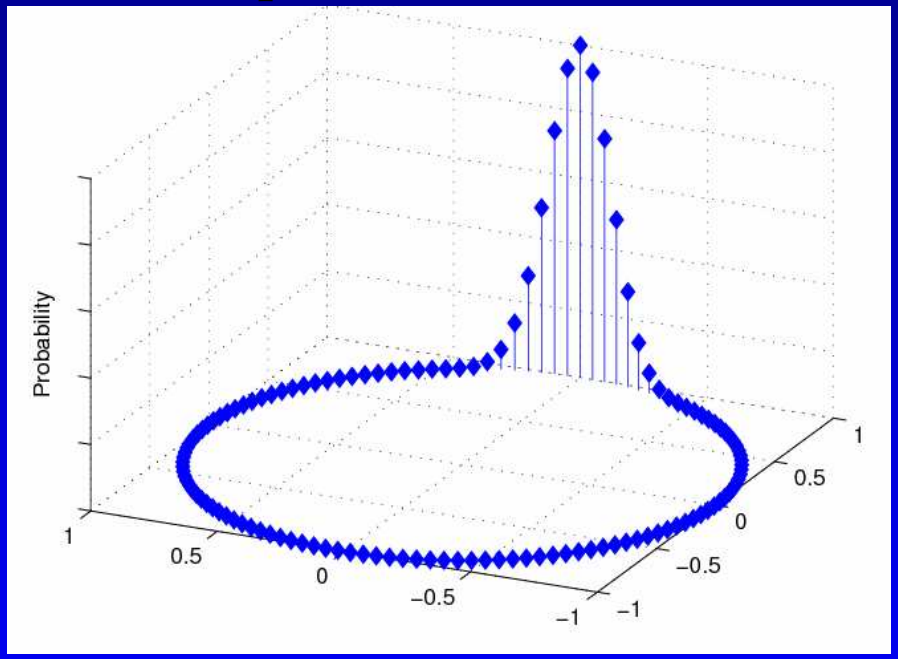
Learning modulo p

- Fix some $p<poly(n)$
- Let $s \in Z_p^n$ be a secret
- We have random equations modulo p with error:

$$2s_1+0s_2+2s_3+1s_4+2s_5+4s_6+\ldots+4s_n \approx 2$$
$$0s_1+1s_2+5s_3+0s_4+6s_5+6s_6+\ldots+2s_n \approx 4$$
$$6s_1+5s_2+2s_3+0s_4+5s_5+2s_6+\ldots+0s_n \approx 2$$
$$6s_1+4s_2+4s_3+4s_4+3s_5+3s_6+\ldots+1s_n \approx 5$$
$$\vdots$$

Learning modulo p

- More formally, we need to learn s from samples of the form $(t, st+e)$ where t is chosen uniformly from Z_p^n and e is chosen from Z_p



- Easy algorithms need $2^{O(n \log n)}$ equations/time
- Best algorithm needs $2^{O(n)}$ equations/time [BlumKalaiWasserman'00]

Main Theorem

Learning modulo p is as hard as worst-case lattice problems using a quantum reduction

- In other words: solving the problem implies an efficient quantum algorithm for lattices

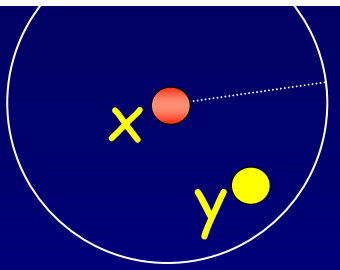
Equivalent formulation

- For m=poly(n), let C be a random m£n matrix with elements in Z_p. Given $Cs+e$ for some $s \in Z_p^n$ and some noise vector $e \in Z_p^m$, recover s.
- This is the problem of decoding from a random linear code

Why Quantum?

- As part of the reduction, we need to perform a certain algorithmic task on lattices
- We do not know how to do it classically, only quantumly!

Why Quantum?



- We are given an oracle that solves CVP_d for some small d
- As far as I can see, the only way to generate inputs to this oracle is:
  - Somehow choose $x \in L$
  - Let y be some random vector within dist d of x
  - Call the oracle with y
- The answer is x. But we already know the answer !!
- Quantumly, being able to compute x from y is very useful: it allows us to transform the state $|y,x\rangle$ to the state $|y,0\rangle$ reversibly (and then we can apply the quantum Fourier transform)

Application:

New Public Key Encryption Scheme

Previous lattice-based PKES

[AjtaiDwork96,GoldreichGoldwasserHalevi97,R'03]

- Main advantages:
  - Based on a lattice problem
  - Worst-case hardness
- Main disadvantages:
  - Based only on unique-SVP
  - Impractical (think of n as 100):
    - Public key size $O(n^4)$
    - Encryption expands by $O(n^2)$

Ajtai's recent PKES [Ajtai05]

- Main advantages:
  - Practical (think of n as 100):
    - Public key size O(n)
    - Encryption expands by O(n)
- Main disadvantages:
  - Not based on lattice problem
  - No worst-case hardness

New lattice-based PKES
[This work]

- Main advantages:
  - quantum
    - Worst-case hardness
    - Based on the main lattice problems (SVP, SIVP)
    - Practical (think of n as 100):
      - Public key size O(n)
      - Encryption expands by O(n)
- Breaking the cryptosystem implies an efficient quantum algorithm for lattices
- In fact, security is based on the learning problem (no quantum needed here)

The Cryptosystem

- Everything modulo 4
- Private key: 4 random numbers

  1 2 0 3

- Public key: a 6x4 matrix and approximate inner product

  → $2\cdot 1 + 0\cdot 2 + 1\cdot 0 + 2\cdot 3 \simeq 0$

  $1\cdot 1 + 2\cdot 2 + 2\cdot 0 + 3\cdot 3 \simeq 2$

  $0\cdot 1 + 2\cdot 2 + 0\cdot 0 + 3\cdot 3 \simeq 1$

  → $1\cdot 1 + 2\cdot 2 + 0\cdot 0 + 2\cdot 3 \simeq 0$

  $0\cdot 1 + 3\cdot 2 + 1\cdot 0 + 3\cdot 3 \simeq 3$

  $3\cdot 1 + 3\cdot 2 + 0\cdot 0 + 2\cdot 3 \simeq 2$

- Encrypt the bit 0:

  $3\cdot ? + 2\cdot ? + 1\cdot ? + 0\cdot ? \approx 1$

- Encrypt the bit 1:

  $3\cdot ? + 2\cdot ? + 1\cdot ? + 0\cdot ? \approx 3$

Proof of the Main Theorem

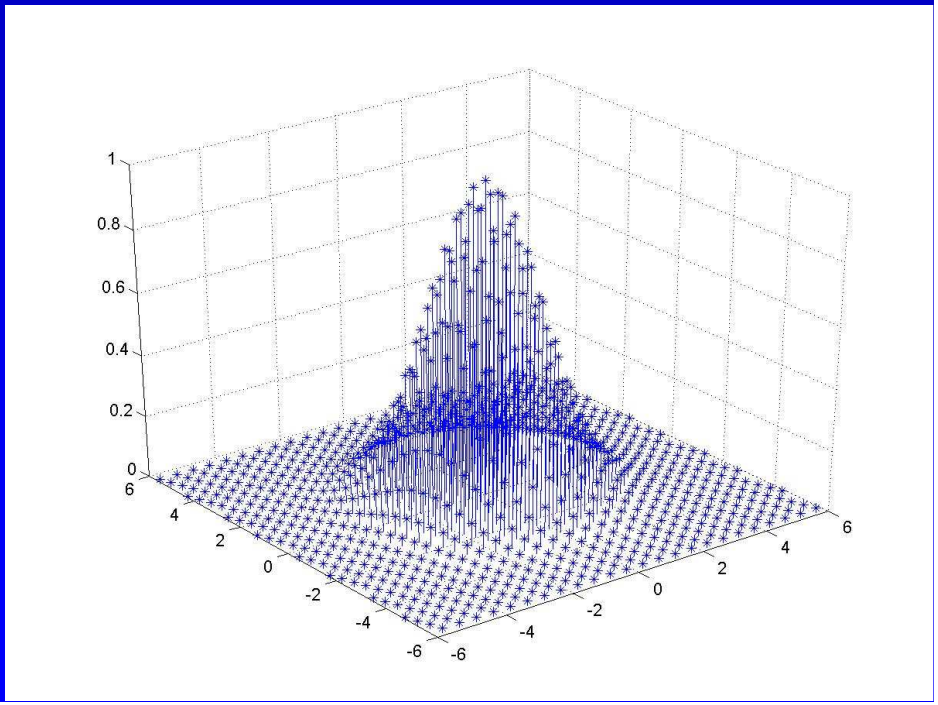
Overview

Gaussian Distribution

- Define a Gaussian distribution on a lattice (normalization omitted)

$$\forall x \in L,\ D_r(x) = e^{-\|x/r\|^2}$$

- We can efficiently sample from D_r for large $r=2^n$

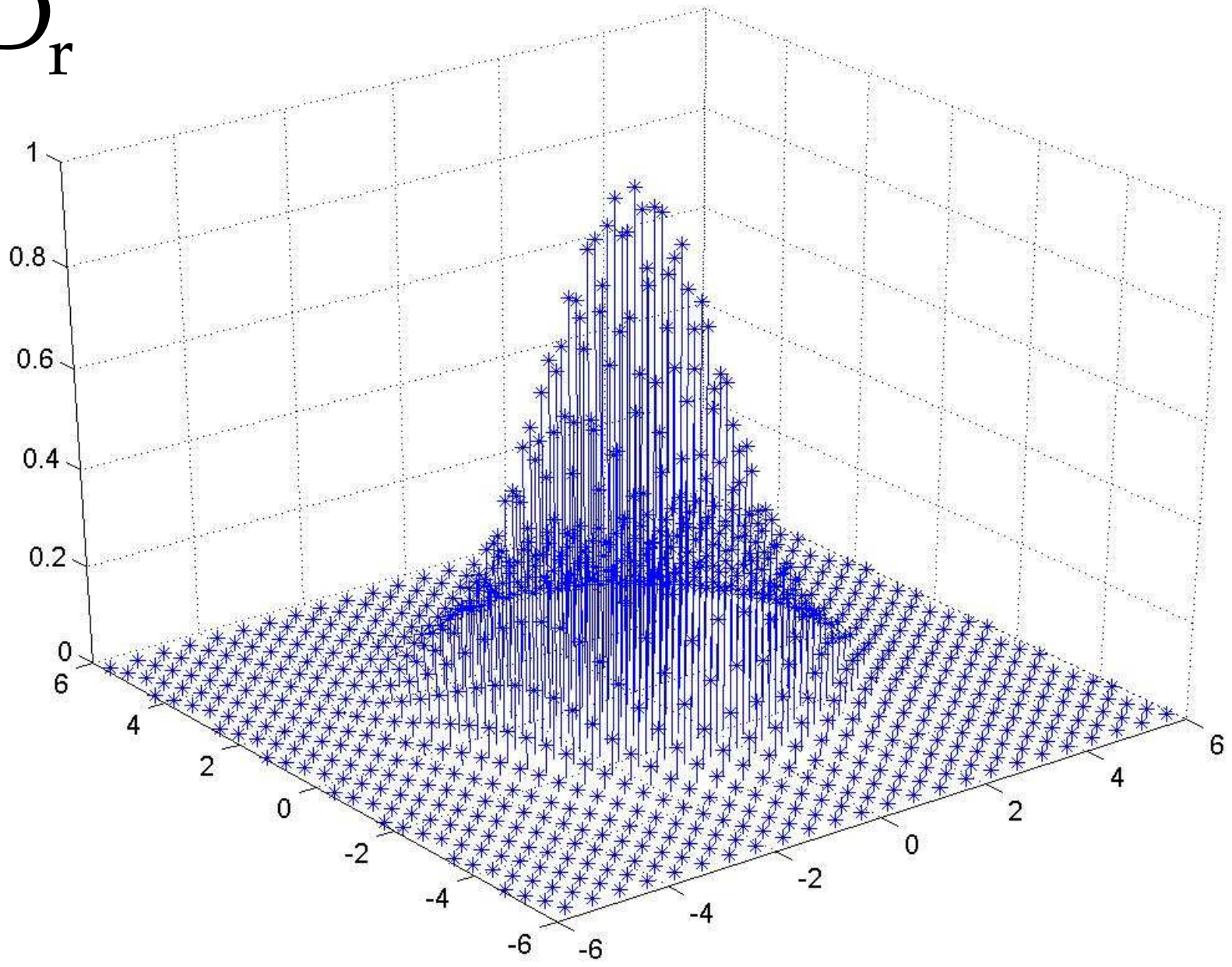


The Reduction

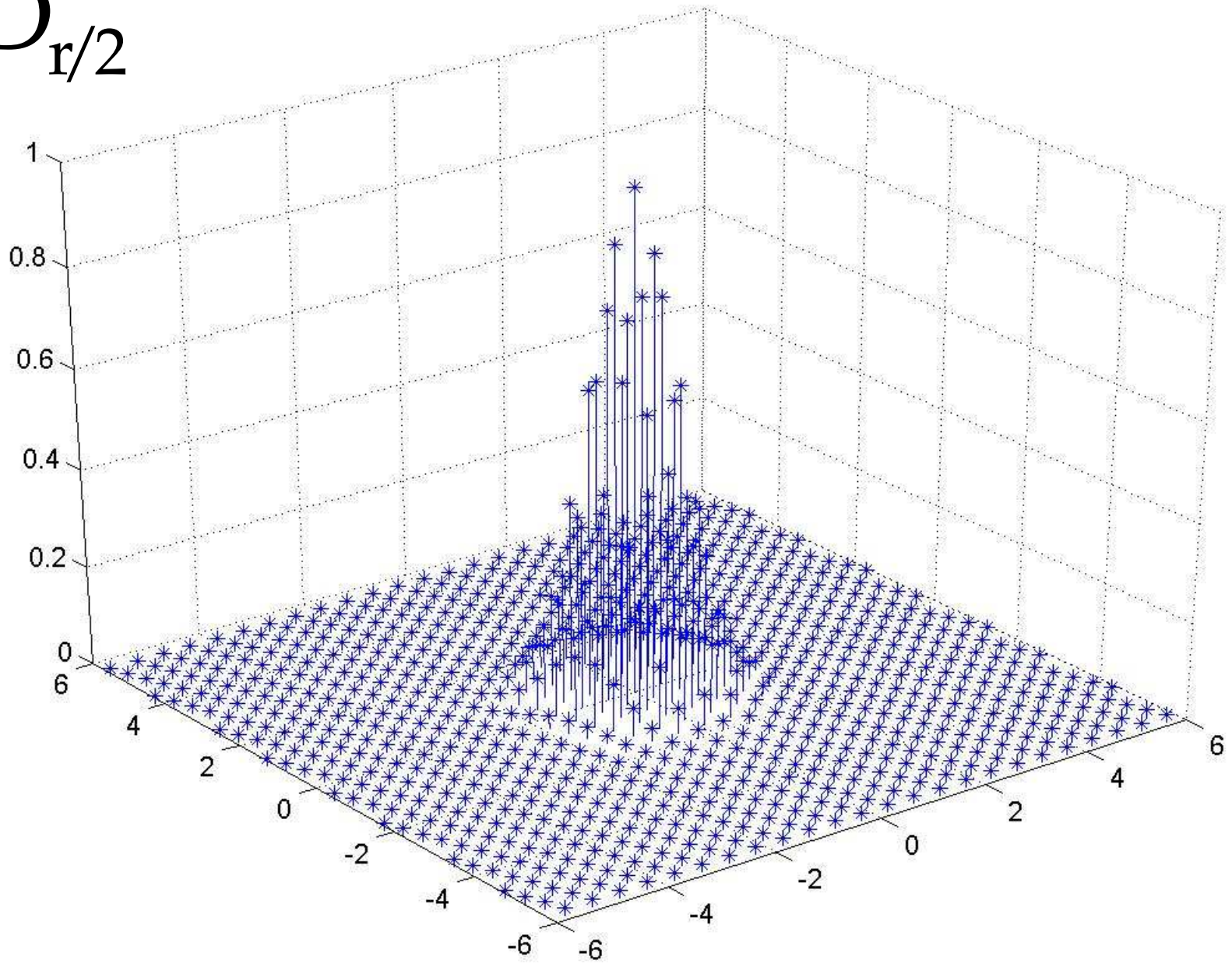
- Assume the existence of an algorithm for the learning modulo p problem for $p=2\sqrt{n}$

- Our lattice algorithm:
  - $r=2^n$
  - Take poly(n) samples from D_r
  - Repeat:
    - Given poly(n) samples from D_r compute poly(n) samples from $D_{r/2}$
    - Set $r \leftarrow r/2$
  - When r is small, output a short vector

D_r



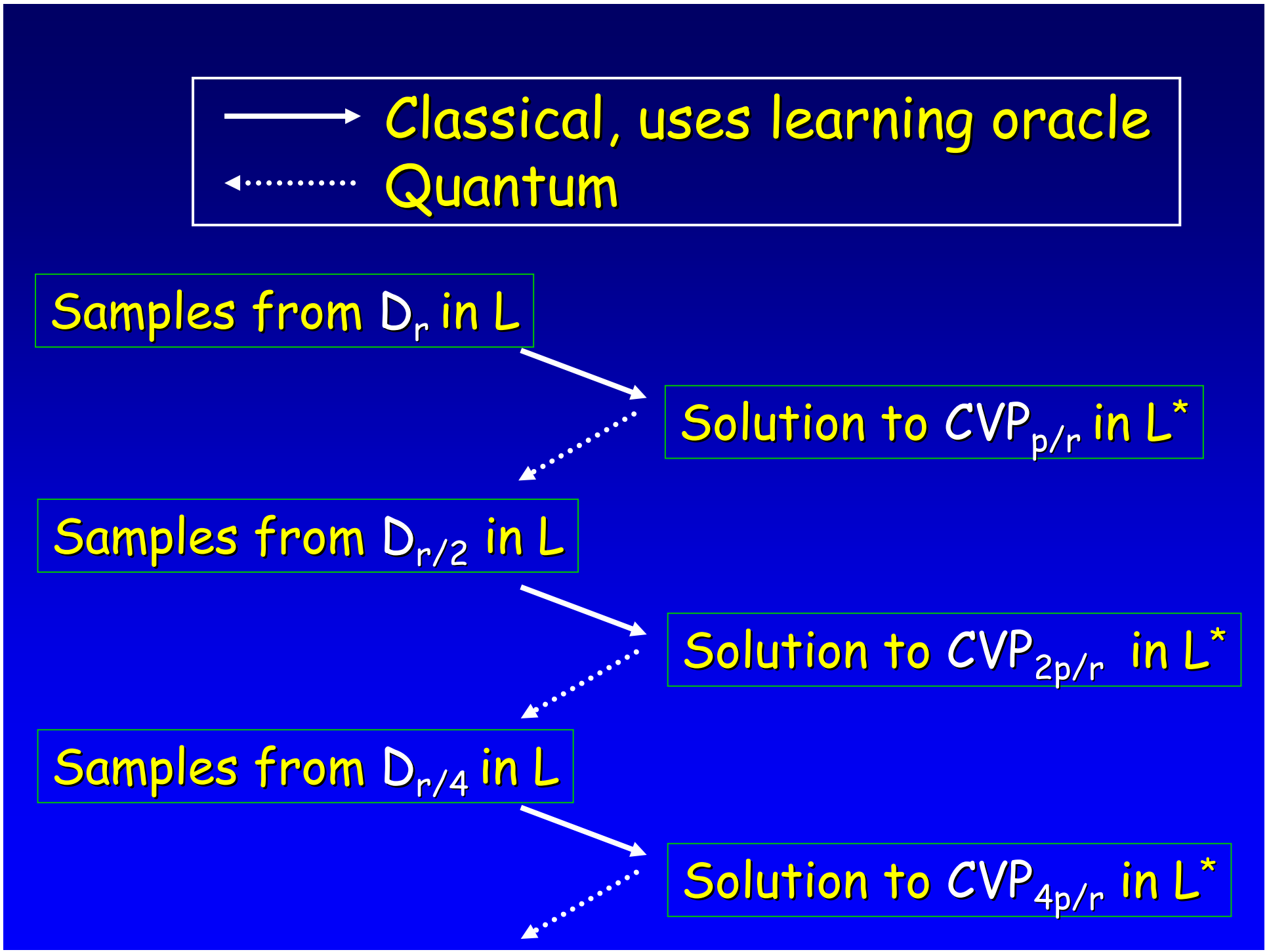
$D_{r/2}$



Obtaining $D_{r/2}$ from D_r

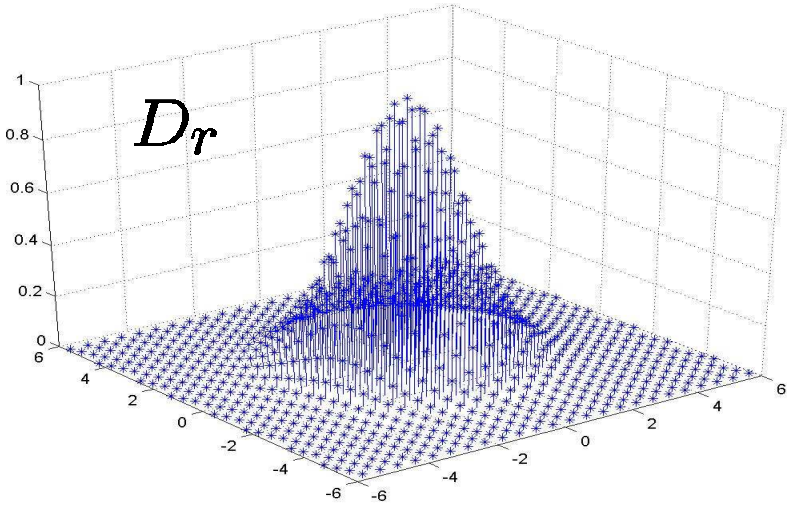
$p=2\sqrt{n}$

- Lemma 1:
  Given poly(n) samples from D_r, and an oracle for 'learning modulo p', we can solve $CVP_{p/r}$ in L^*
  - No quantum here J
- Lemma 2:
  Given a solution to CVP_d in L^*, we can obtain samples from $D_{\sqrt{n}/d}$
  - Quantum K
  - Based on the quantum Fourier transform

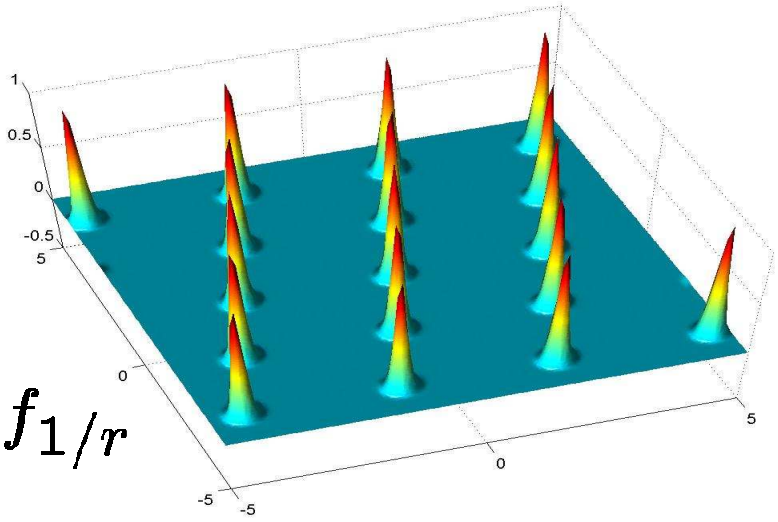

Classical, uses learning oracle
Quantum
Samples from D_r in L
Solution to $CVP_{p/r}$ in L^*
Samples from $D_{r/2}$ in L
Solution to $CVP_{2p/r}$ in L^*
Samples from $D_{r/4}$ in L
Solution to $CVP_{4p/r}$ in L^*

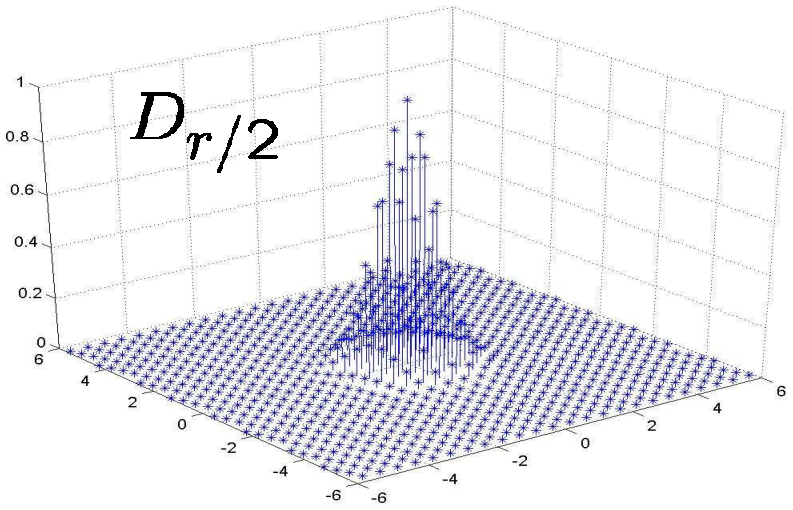
Fourier Transform

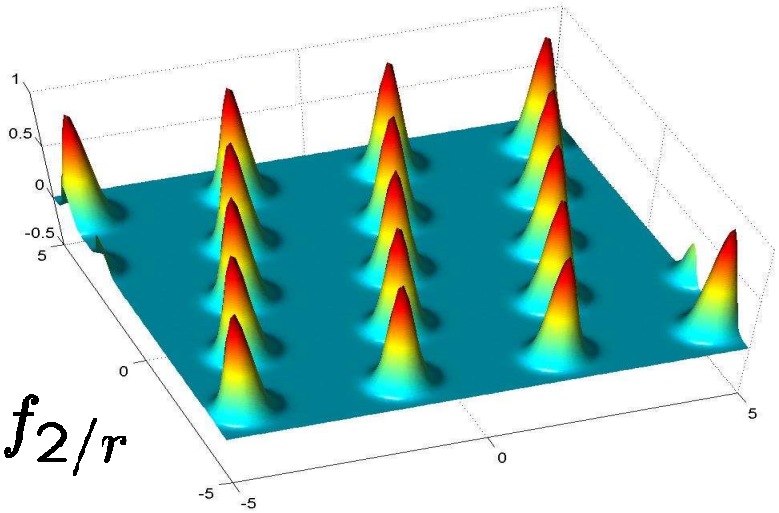
Primal world (L)



Dual world (L^*)







Fourier Transform

- The Fourier transform of D_r is given by

$$f_{1/r}(x) \approx e^{-\|r \cdot \mathrm{dist}(x, L^*)\|^2}$$

- Its value is
  - 1 for x in L^*,
  - e^{-1} at points of distance 1/r from L^*,
  - ¼0 at points far away from L^*.

Proof of the Main Theorem

Lemma 2: Obtaining $D_{\sqrt{n}/d}$ from CVP_d

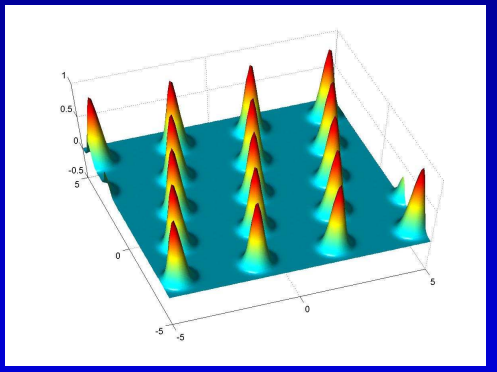
From CVP_d to $D_{\sqrt{n}/d}$

- Assume we can solve CVP_d; we'll show how to obtain samples from $D_{\sqrt{n}/d}$

- Step 1:

  Create the quantum state

$$\sum_{x \in \mathbb{R}^n} f_{d/\sqrt{n}}(x)|x\rangle$$



by adding a Gaussian to each lattice point and uncomputing the lattice point by using the CVP algorithm

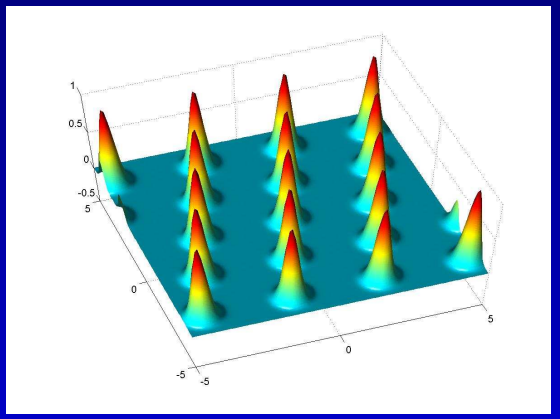
From CVP_d to $D_{\sqrt{n}/d}$

- Step 2:

  Compute the quantum Fourier transform of

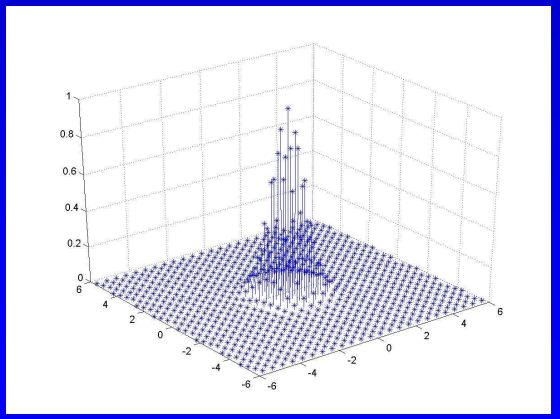
$$\sum_{x \in \mathbb{R}^n} f_{d/\sqrt{n}}(x)|x\rangle$$

  It is exactly $D_{\sqrt{n}/d}$!!



- Step 3:

  Measure and obtain one sample from $D_{\sqrt{n}/d}$



- By repeating this process, we can obtain poly(n) samples

From CVP_d to $D_{\sqrt{n}/d}$

- More precisely, create the state $\sum_{y \in L^*} |y\rangle$
- And the state $\sum_{x \in \mathbb{R}^n} e^{-\|(\sqrt{n}/d)x\|^2} |x\rangle$
- Tensor them together and add first to second

$$\sum_{y \in L^*, x \in \mathbb{R}^n} e^{-\|(\sqrt{n}/d)x\|^2} |y, x+y\rangle$$

- Uncompute first register by solving $CVP_{p/r}$

$$\sum_{x \in \mathbb{R}^n} e^{-\|(\sqrt{n}/d)\cdot \mathrm{dist}(x,L^*)\|^2} |x\rangle \approx \sum_{x \in \mathbb{R}^n} f_{d/\sqrt{n}}(x) |x\rangle$$

Proof of the Main Theorem

Lemma 1: Solving $CVP_{p/r}$ given samples from D_r and an oracle for learning mod p

It's enough to approximate $f_{p/r}$

- Lemma: being able to approximate $f_{p/r}$ implies a solution to $CVP_{p/r}$
- Proof Idea – walk uphill:
  - $f_{p/r}(x) > \frac{1}{4}$ for points x of distance < p/r
  - Keep making small modifications to x as long as $f_{p/r}(x)$ increases
  - Stop when $f_{p/r}(x)=1$ (then we are on a lattice point)

What's ahead in this part

- For warm-up, we show how to approximate $f_{1/r}$ given samples from D_r
  - No need for learning
  - This is main idea in [AharonovR'04]
- Then we show how to approximate $f_{2/r}$ given samples from D_r and an oracle for the learning problem
- Approximating $f_{p/r}$ is similar

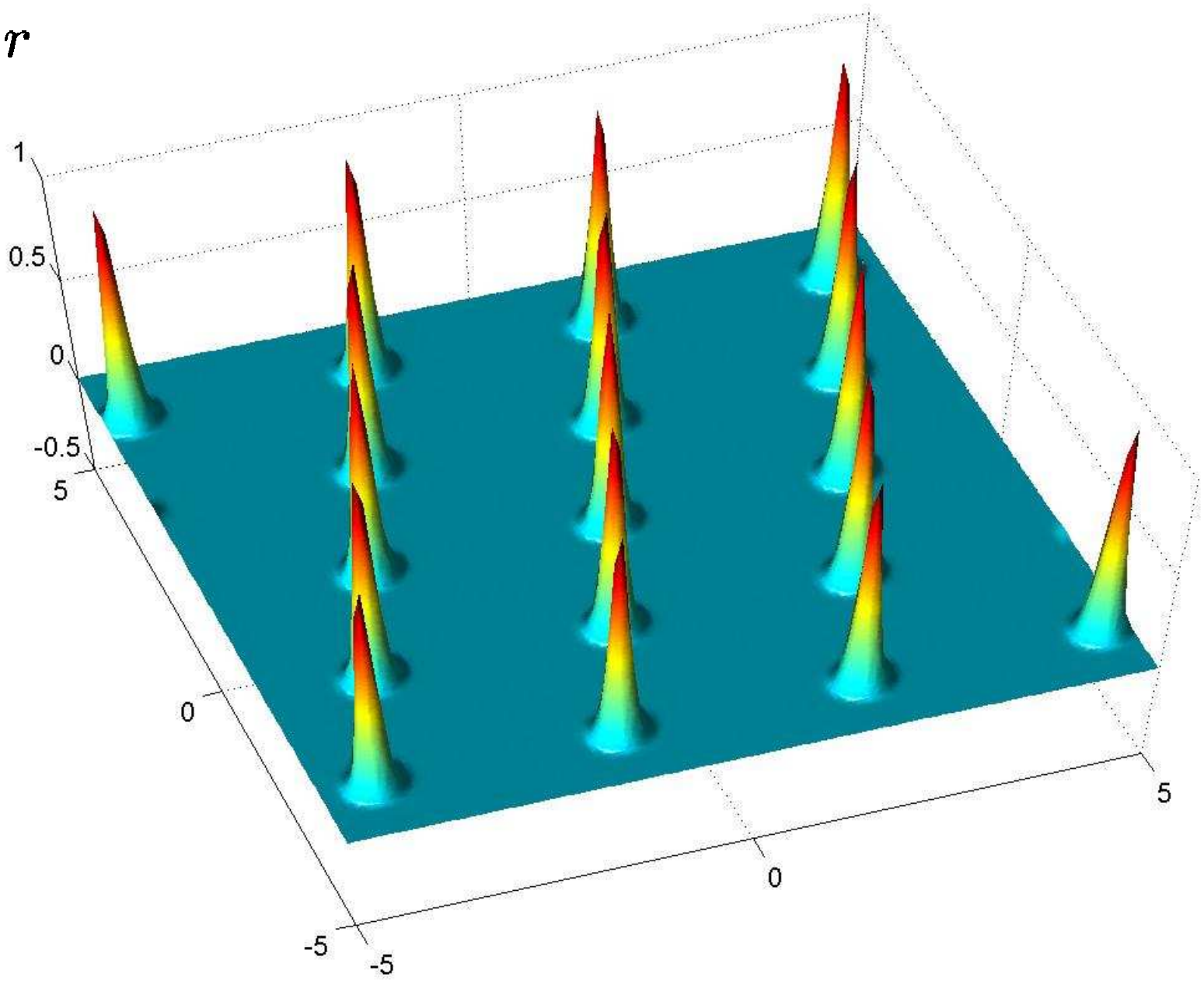
Warm-up: approximating $f_{1/r}$

- Let's write $f_{1/r}$ in its Fourier representation:

$$f_{1/r}(x) = \sum_{w \in L} \widehat{f_{1/r}}(w) \cos(2\pi\langle w, x\rangle)$$
$$= \sum_{w \in L} D_r(w) \cos(2\pi\langle w, x\rangle)$$
$$= E_{w \sim D_r}[\cos(2\pi\langle w, x\rangle)]$$

- Using samples from D_r, we can compute a good approximation to $f_{1/r}$ (this is the main idea in [AharonovR'04])

$f_{1/r}$



Fourier Transform

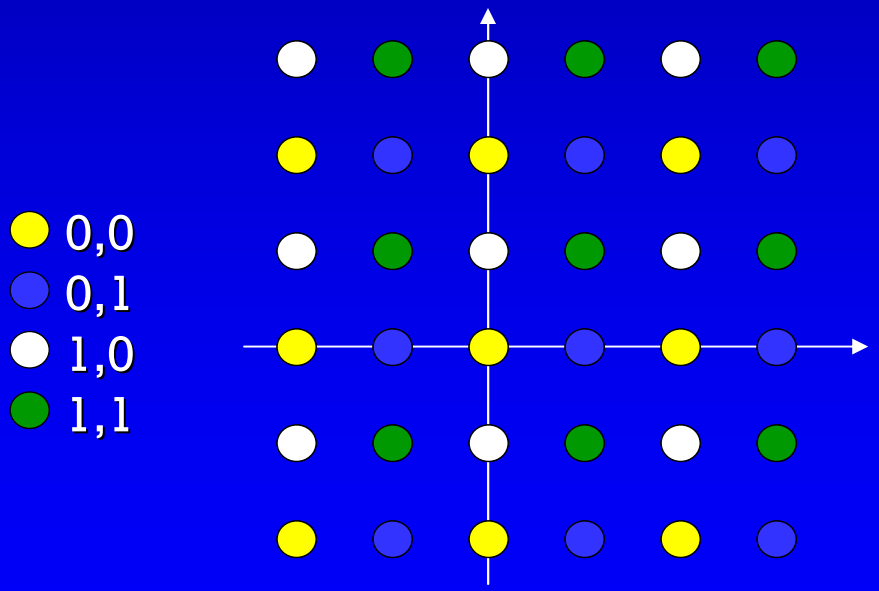
- Consider the Fourier representation again:

$$f_{1/r}(x) = E_{w \sim D_r}\left[\cos(2\pi\langle w, x\rangle)\right]$$

- For $x \in L^*$, $\langle w, x\rangle$ is integer for all w in L and therefore we get $f_{1/r}(x)=1$
- For x that is close to L^*, $\langle w, x\rangle$ is distributed around an integer. Its standard deviation can be (say) 1.

Approximating $f_{2/r}$

- Main idea: partition D_r into 2^n distributions
- For $t \in (Z_2)^n$, denote the translate t by D^t_r
- Given a lattice point we can compute its t
- The probability on $(Z_2)^n$ obtained by sampling from D_r and outputting t is close to uniform

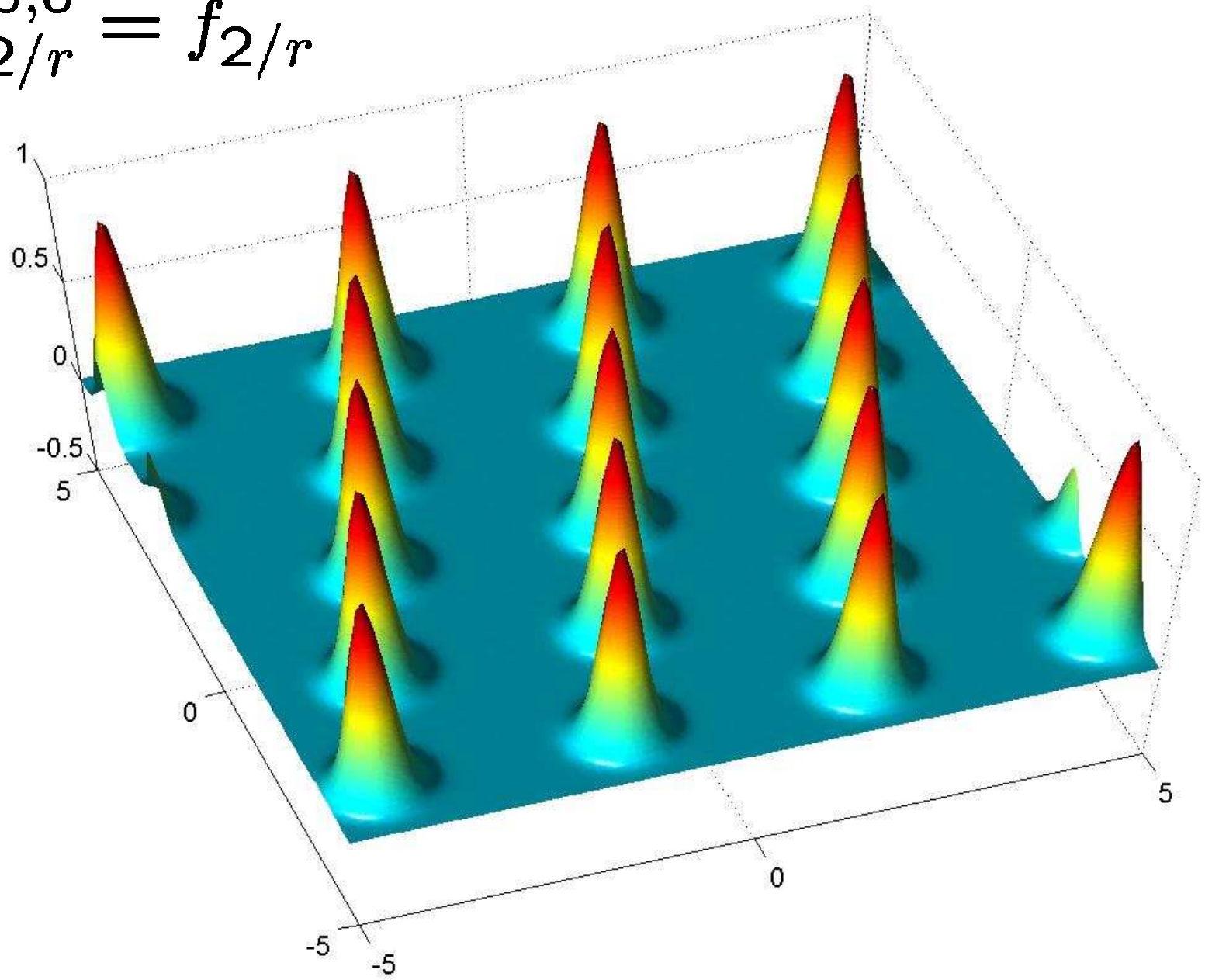


Approximating $f_{2/r}$

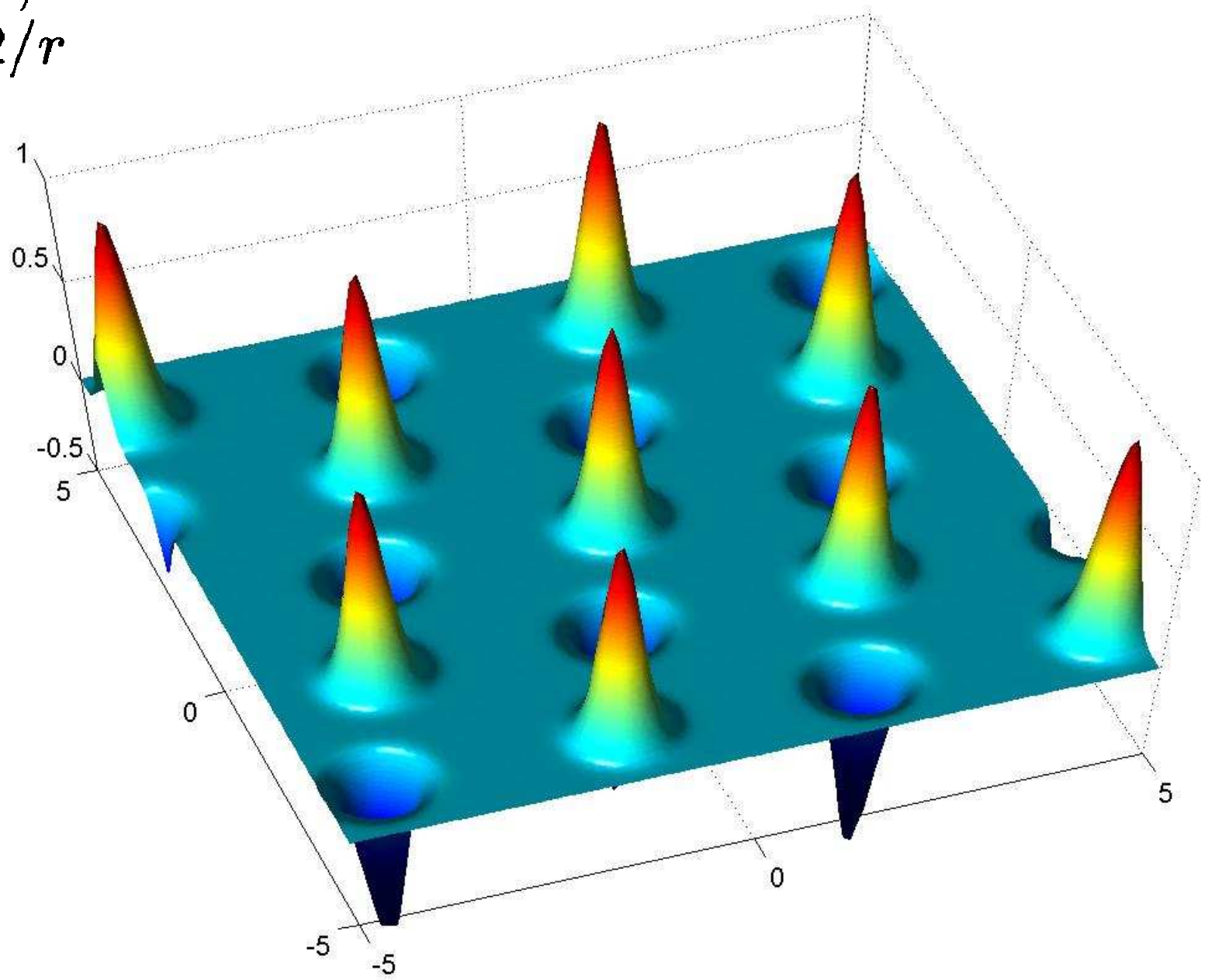
- Hence, by using samples from D_r we can produce samples from the following distribution on pairs (t,w):
  - Sample $t \in (Z_2)^n$ uniformly at random
  - Sample w from D^t_r
- Consider the Fourier transform of D^t_r

$$f^t_{2/r}(x) = E_{w \sim D^t_r}[\cos(\pi \langle w, x \rangle)]$$

$$f_{2/r}^{0,0} = f_{2/r}$$



$f_{2/r}^{1,1}$



Approximating $f_{2/r}$

- The functions $f^{t}_{2/r}$ look almost like $f_{2/r}$
- Only difference is that some Gaussians have their sign flipped
- Approximating $f^{t}_{2/r}$ is enough: we can easily take the absolute value and obtain $f_{2/r}$
- For this, however, we need to obtain several pairs (t,w) for the same t

- The problem is that each sample (t,w) has a different t !

Approximating $f_{2/r}$

- Fix x close to L^*
- The sign of its Gaussian is ±1 depending on hs,ti mod 2 for $s \in (Z_2)^n$ that depends only on x
- The distribution of $\langle x,w \rangle$ mod 2 when w is sampled from $D^{\dagger}_r$ is centred around $\langle s,t \rangle$ mod 2
- Hence, we obtain equations modulo 2 with error:

$$\begin{aligned} \text{hs,t}_1\text{i} &\text{ ¼dhx,w}_1\text{ic mod 2} \\ \text{hs,t}_2\text{i} &\text{ ¼dhx,w}_2\text{ic mod 2} \\ \text{hs,t}_3\text{i} &\text{ ¼dhx,w}_3\text{ic mod 2} \\ &\vdots \end{aligned}$$

Approximating $f_{2/r}$

- Using the learning algorithm, we solve these equations and obtain s
- Knowing s, we can cancel the sign
- Averaging over enough samples gives us an approximation to $f_{2/r}$

Open Problems 1/4

- Dequantize the reduction:
  - This would lead to the 'ultimate' lattice-based cryptosystem (based on SVP, efficient)
  - Main obstacle: what can one do classically with a solution to CVP_d?
- Construct even more efficient schemes based on special classes of lattices such as cyclic lattices
  - For hash functions this was done by Micciancio

Open Problems 2/4

- Extend to learning from parity (i.e., p=2) or even some constant p
  - Is there something inherently different about the case of constant p?
- Use the 'learning mod p' problem to derive other lattice-based hardness results
  - Recently, used by Klivans and Sherstov to derive hardness of learning problems

Open Problems 3/4

- Cryptanalysis
  - Current attacks limited to low dimension [NguyenStern98]
  - New systems [Ajtai05,R05] are efficient and can be easily used with dimension 100+
- Security against chosen-ciphertext attacks
  - Known lattice-based cryptosystems are not secure against CCA

Open Problems 4/4

- Comparison with number theoretic cryptography
  - E.g., can one factor integers using an oracle for n-approximate SVP?
- Signature schemes
  - Can one construct provably secure lattice-based signature schemes?