Parameterized hardness of coding and lattice problems

João Ribeiro
CMU → NOVA LINCS & FCT-UNL

Huck Bennett
Oregon State University

Mahdi Cheraghchi
University of Michigan

Venkatesan Guruswami
UC Berkeley

Why codes and lattices?

- Fundamental objects in mathematics and computer science.

- Computational problems on such objects are basis of post-quantum cryptography.

NIST

PROJECTS/PROGRAMS

Post-Quantum Cryptography

Summary

Post-Quantum Cryptography (PQC) - An area of cryptography that researches and advances the use of quantum-resistant primitives, with the goal of keeping existing public key infrastructure intact in a future era of quantum computing. Intended to be secure against both quantum and classical computers and deployable without drastic changes to existing communication protocols and networks.

What is a (linear) code?

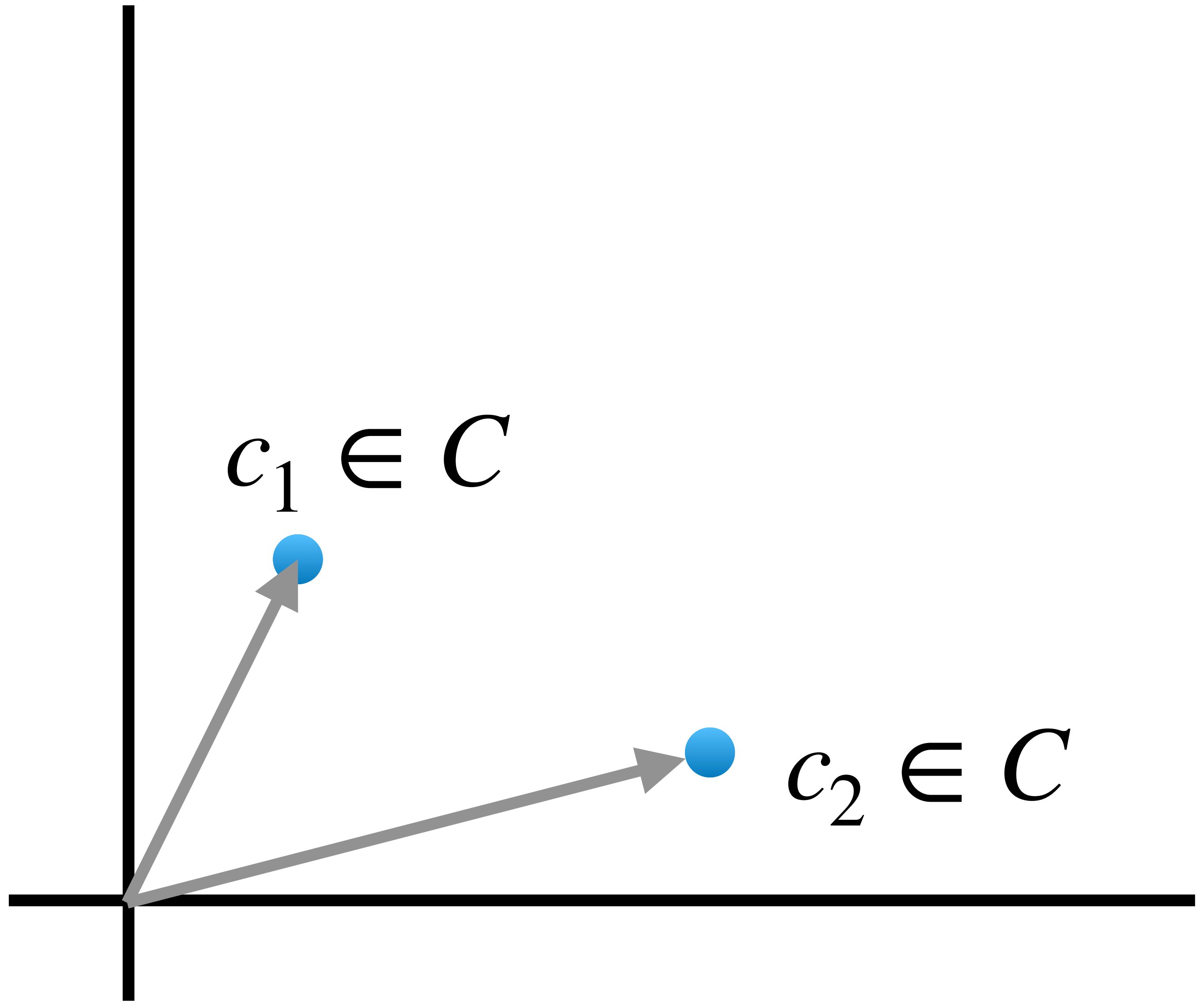
$C \subseteq \mathbb{F}_q^n$ is a linear code if it is a vector subspace of $\mathbb{F}_q^n$.

There exists a **generator matrix** $G \in \mathbb{F}^{n \times k}$ such that $C = \{Gv : v \in \mathbb{F}_q^k\}$.

What is a (linear) code?

$C \subseteq \mathbb{F}_q^n$ is a linear code if it is a vector subspace of $\mathbb{F}_q^n$.

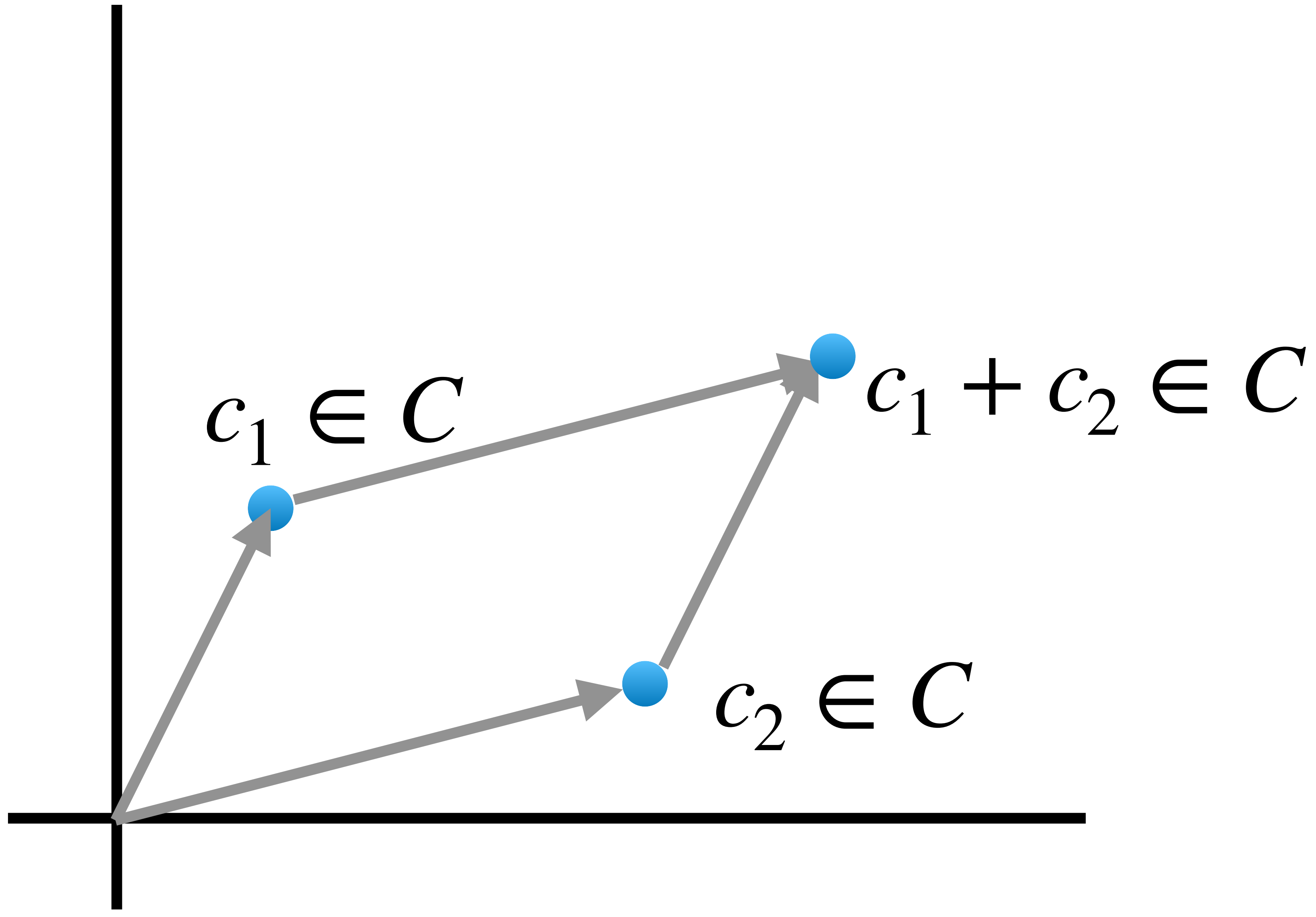
There exists a **generator matrix** $G \in \mathbb{F}^{n\times k}$ such that $C = \{Gv : v \in \mathbb{F}_q^k\}$.



What is a (linear) code?

$C \subseteq \mathbb{F}_q^n$ is a linear code if it is a vector subspace of $\mathbb{F}_q^n$.

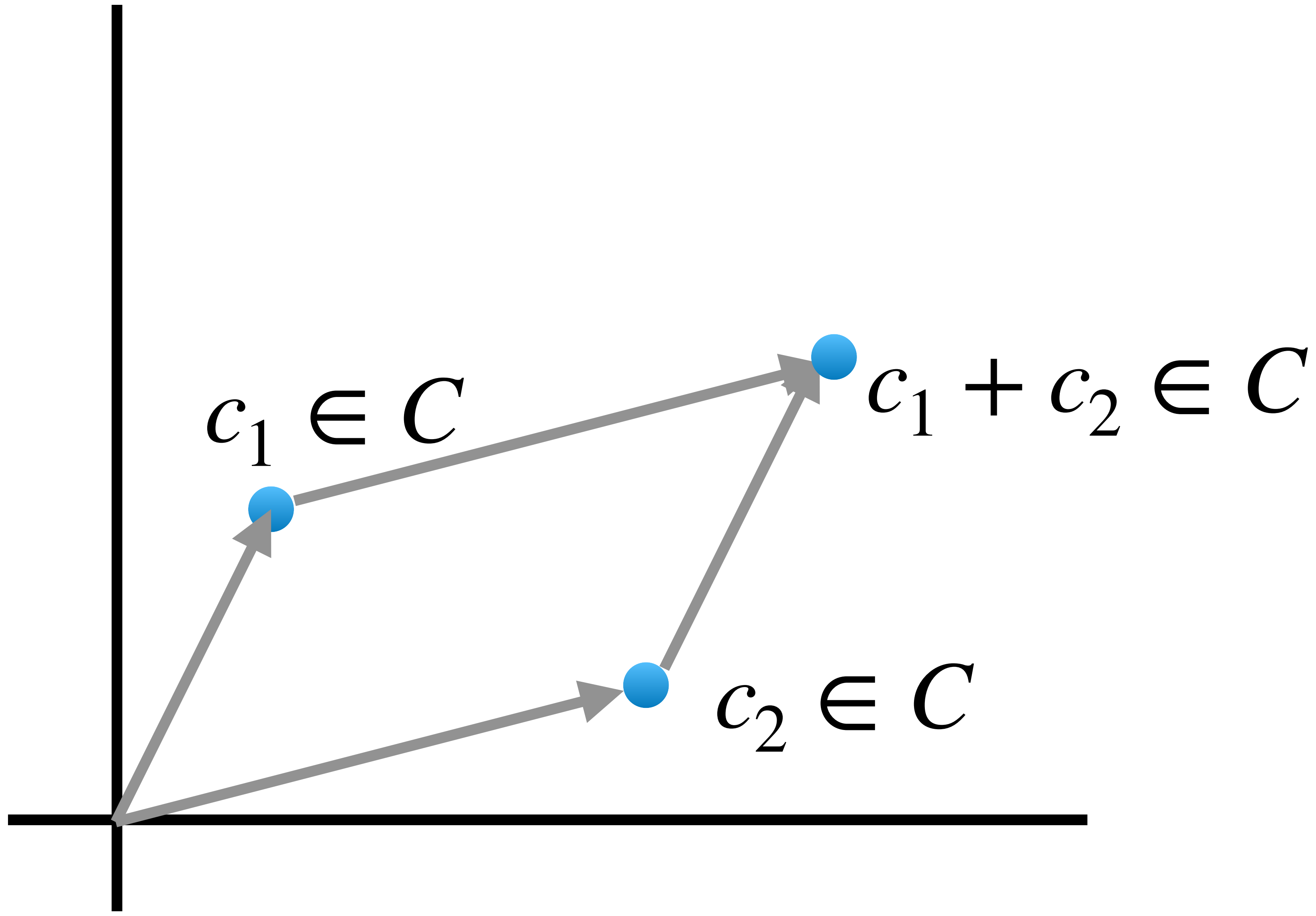
There exists a **generator matrix** $G \in \mathbb{F}^{n \times k}$ such that $C = \{Gv : v \in \mathbb{F}_q^k\}$.



What is a (linear) code?

$C \subseteq \mathbb{F}_q^n$ is a linear code if it is a vector subspace of $\mathbb{F}_q^n$.

There exists a **generator matrix** $G \in \mathbb{F}^{n \times k}$ such that $C = \{Gv : v \in \mathbb{F}_q^k\}$.

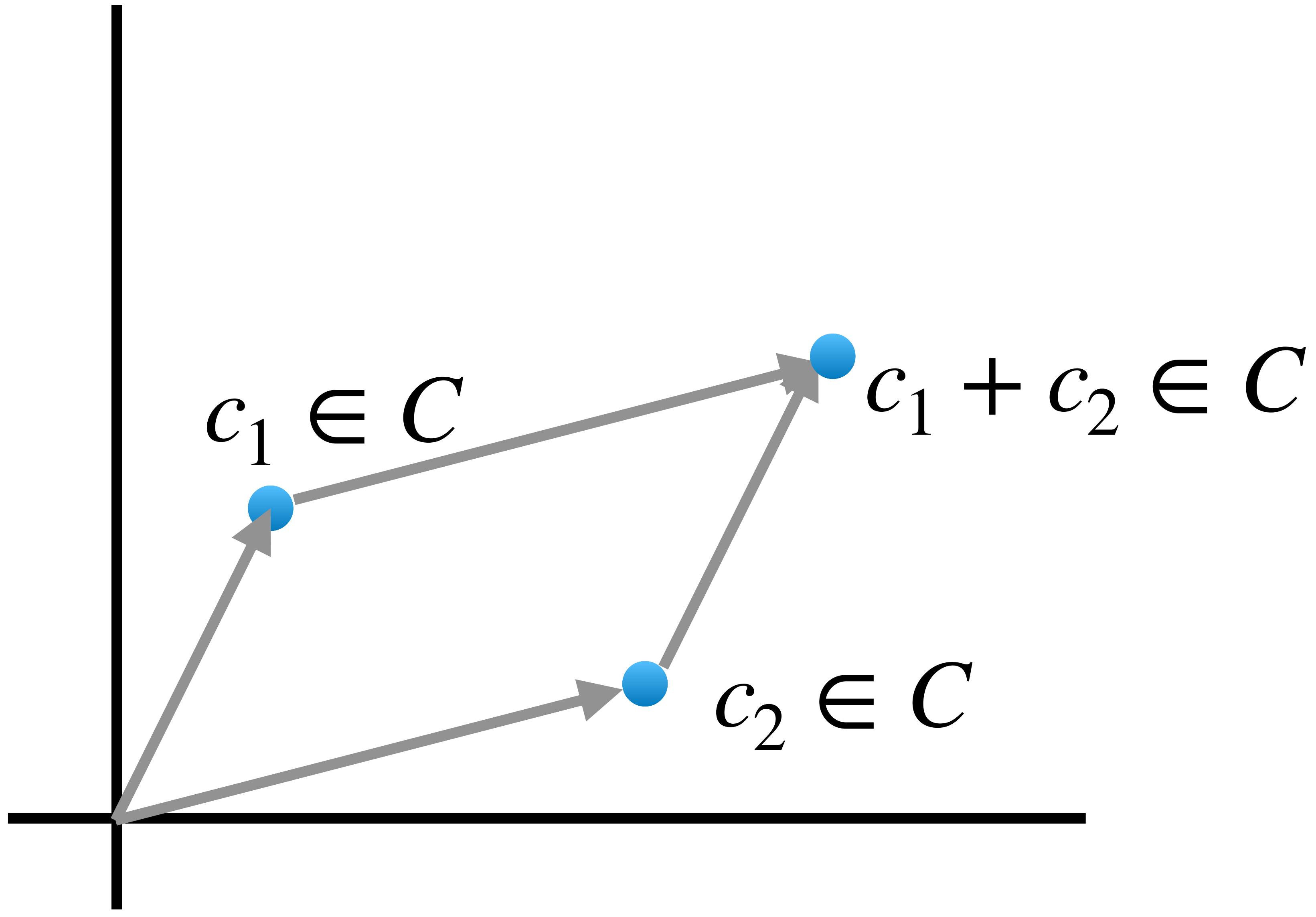


$\|v\|_0 = \#\{i : v_i \neq 0\}$ (Hamming weight of v)

What is a (linear) code?

$C \subseteq \mathbb{F}_q^n$ is a linear code if it is a vector subspace of $\mathbb{F}_q^n$.

There exists a **generator matrix** $G \in \mathbb{F}^{n\times k}$ such that $C = \{Gv : v \in \mathbb{F}_q^k\}$.



$\|v\|_0 = \#\{i : v_i \neq 0\}$ (Hamming weight of v)

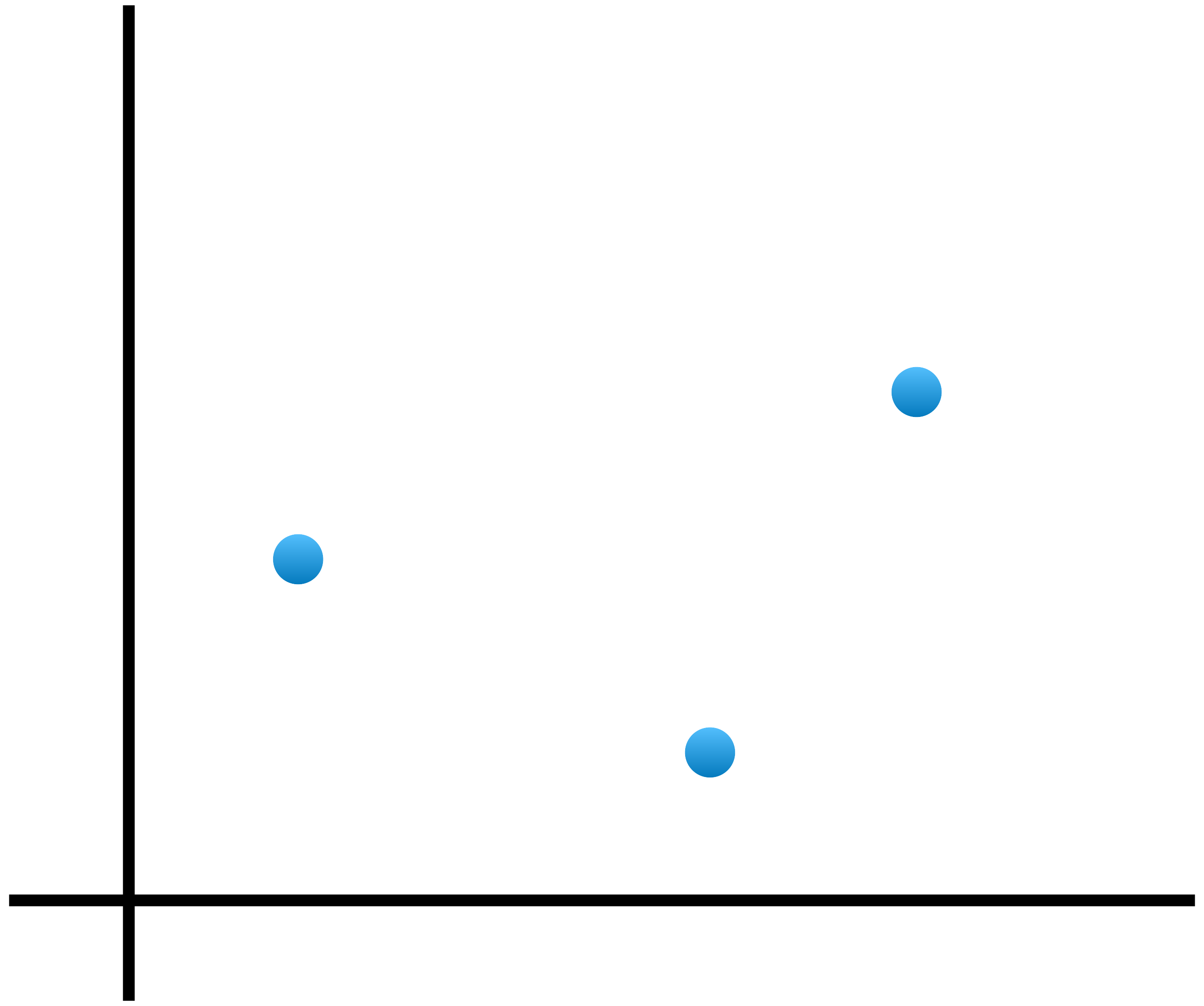
Minimum distance of C:

$$d(C) = \min_{c,c' \in C, c \neq c'} \|c - c'\|_0$$

What is a (linear) code?

$C \subseteq \mathbb{F}_q^n$ is a linear code if it is a vector subspace of $\mathbb{F}_q^n$.

There exists a **generator matrix** $G \in \mathbb{F}^{n \times k}$ such that $C = \{Gv : v \in \mathbb{F}_q^k\}$.



min dist $\geq d$

$\|v\|_0 = \#\{i : c_i \neq 0\}$ (Hamming weight of v)

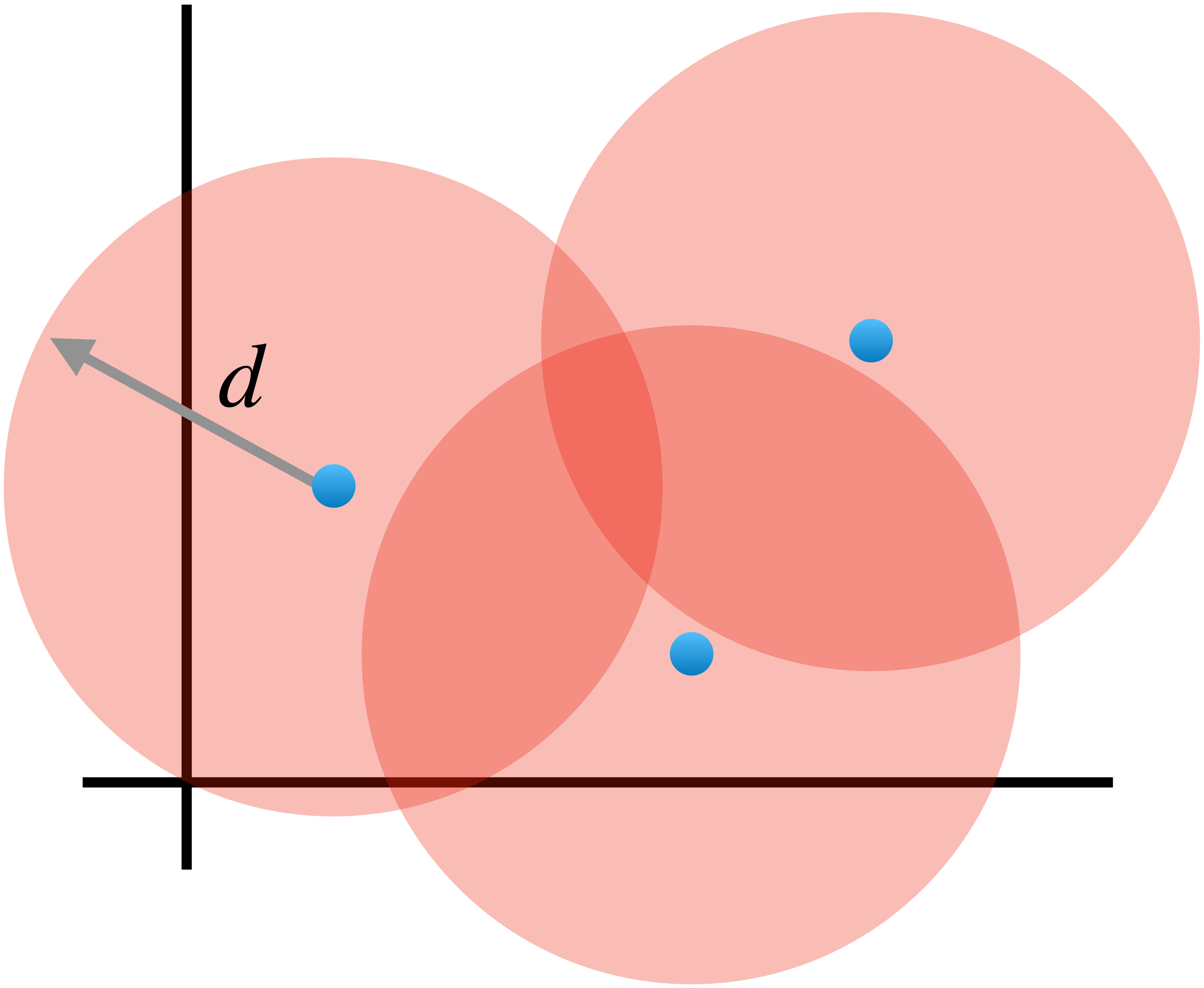
Minimum distance of C:

$$d(C) = \min_{c, c' \in C, c \neq c'} \|c - c'\|_0$$

What is a (linear) code?

$C \subseteq \mathbb{F}_q^n$ is a linear code if it is a vector subspace of $\mathbb{F}_q^n$.

There exists a **generator matrix** $G \in \mathbb{F}^{n \times k}$ such that $C = \{Gv : v \in \mathbb{F}_q^k\}$.



min dist $\geq d$

$\|v\|_0 = \#\{i : c_i \neq 0\}$ (Hamming weight of v)

Minimum distance of C:

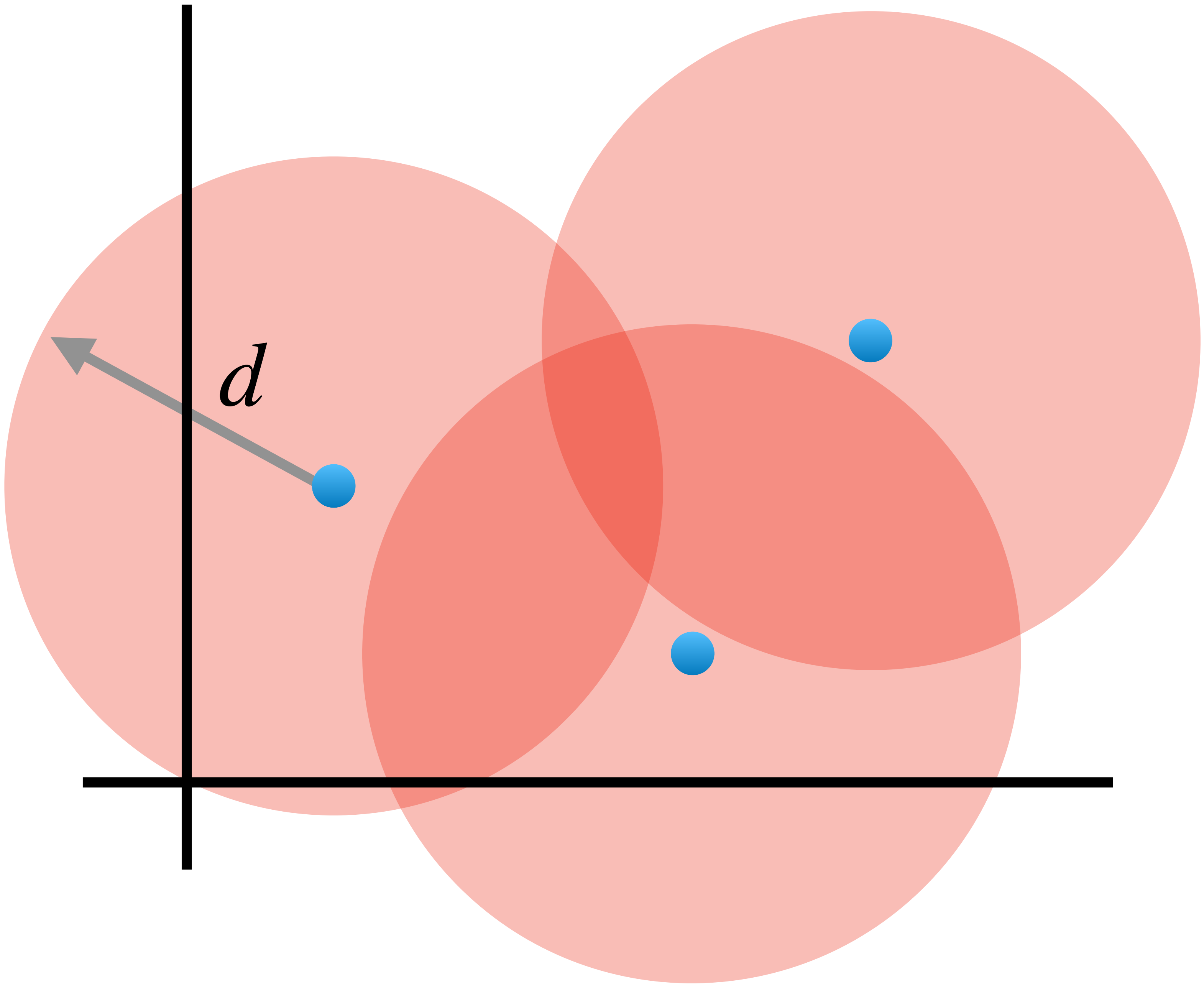
$$d(C) = \min_{c, c' \in C, c \neq c'} \|c - c'\|_0$$

Minimum distance and error-correction

$$\|v\|_0 = \#\{i : c_i \neq 0\}$$

Minimum distance of C:

$$d(C) = \min_{c,c' \in C, c \neq c'} \|c - c'\|_0$$



min dist $\geq d$

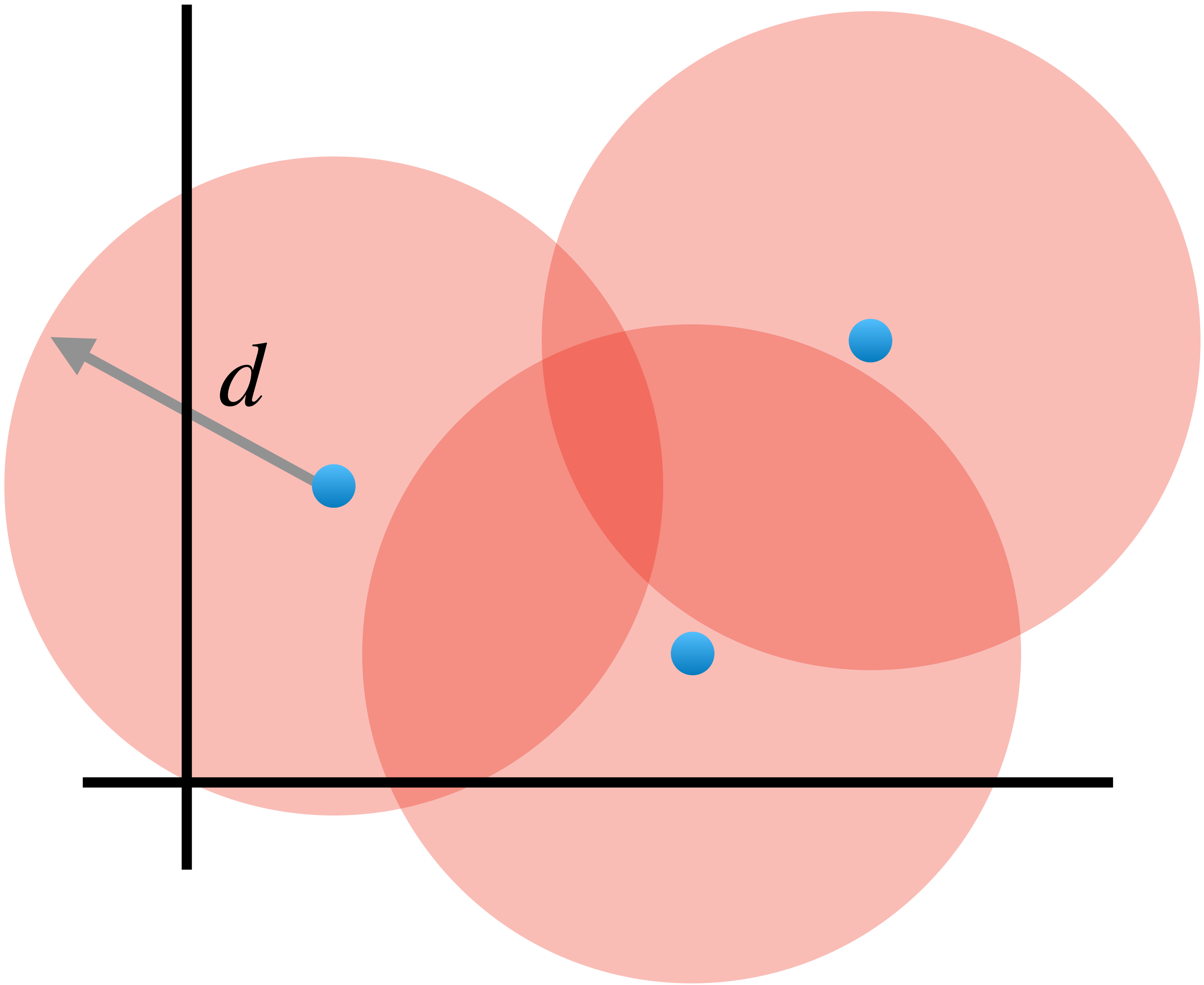
Minimum distance captures error-correction properties of C

Minimum distance and error-correction

$$\|v\|_0 = \#\{i : c_i \neq 0\}$$

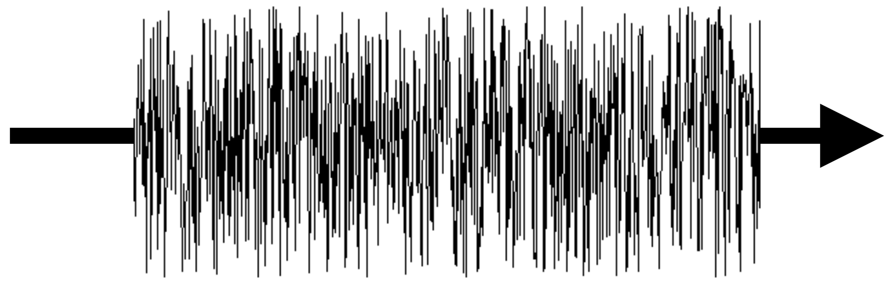
Minimum distance of C:

$$d(C) = \min_{c,c' \in C, c \neq c'} \|c - c'\|_0$$



min dist $\geq d$

Minimum distance captures error-correction properties of C

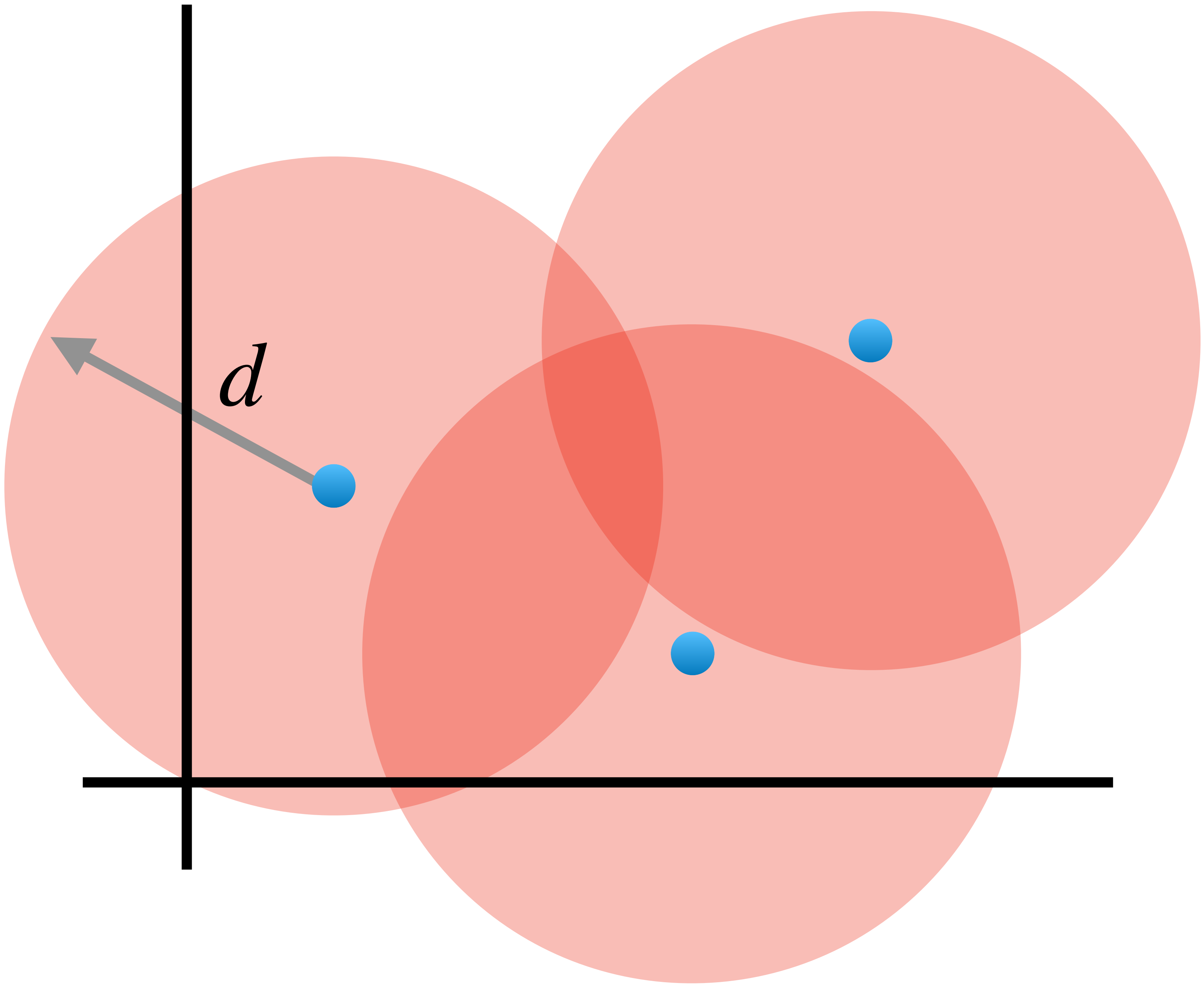
$c \in C$  z

Minimum distance and error-correction

$$\|v\|_0 = \#\{i : c_i \neq 0\}$$

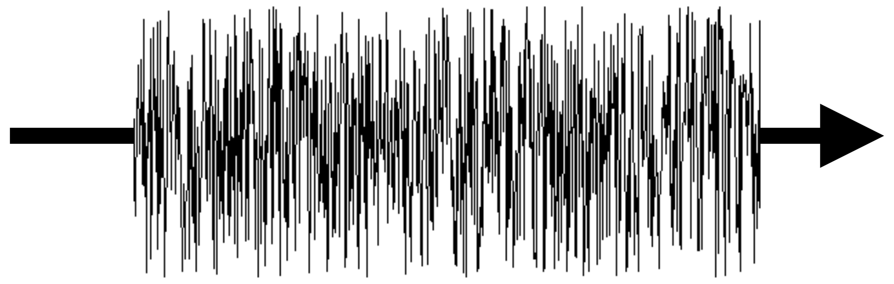
Minimum distance of C:

$$d(C) = \min_{c,c' \in C, c \neq c'} \|c - c'\|_0$$



min dist $\geq d$

Minimum distance captures error-correction properties of C

$c \in C$  z

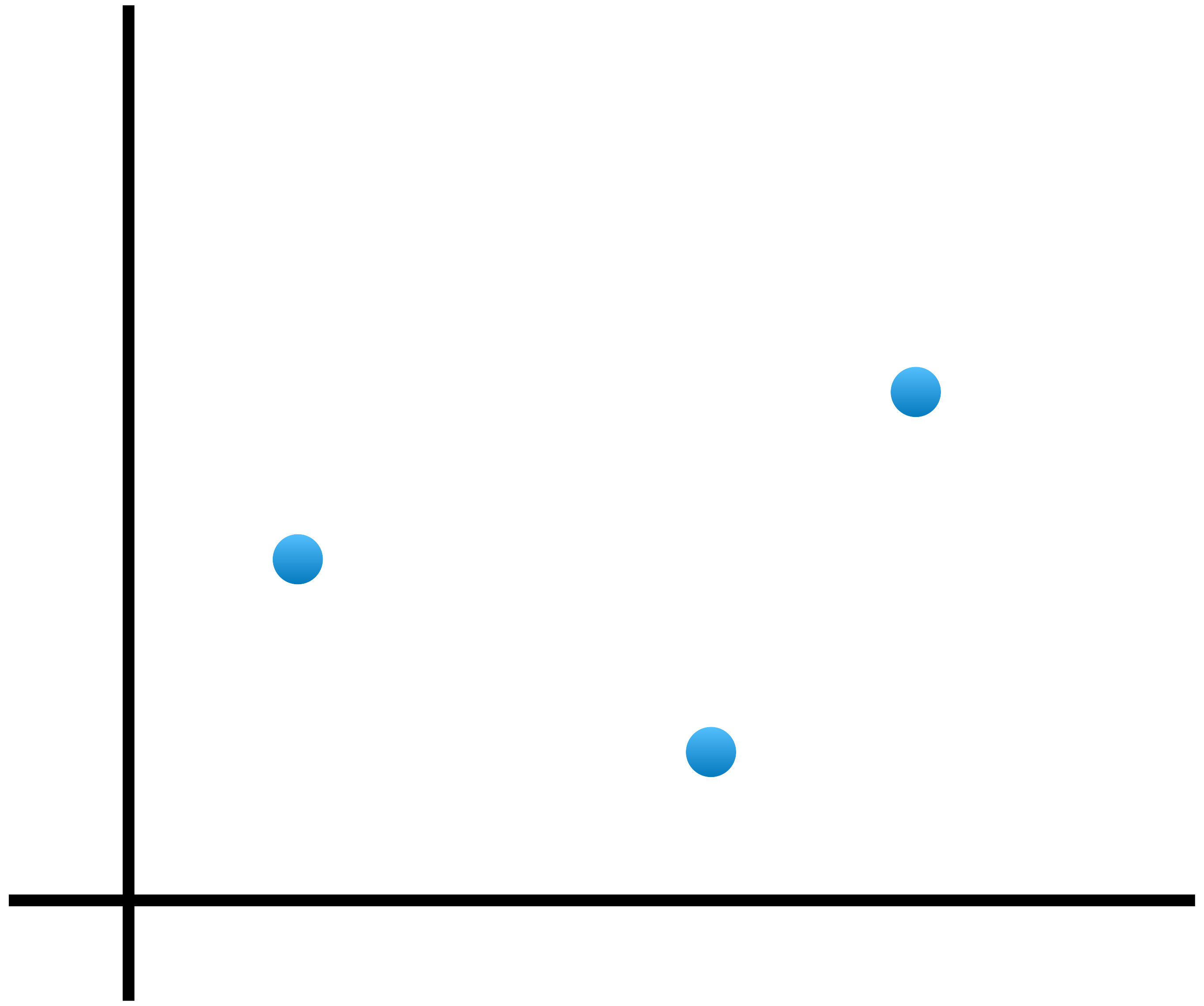
If $\leq \dfrac{d-1}{2}$ errors introduced, guaranteed to recover c from z.

Minimum distance and error-correction

$$\|v\|_0 = \#\{i : c_i \neq 0\}$$

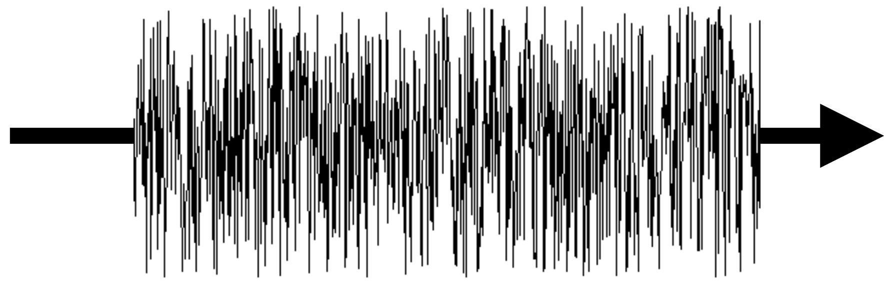
Minimum distance of C:

$$d(C) = \min_{c,c' \in C, c \neq c'} \|c - c'\|_0$$



min dist $\geq d$

Minimum distance captures error-correction properties of C

$c \in C$  z

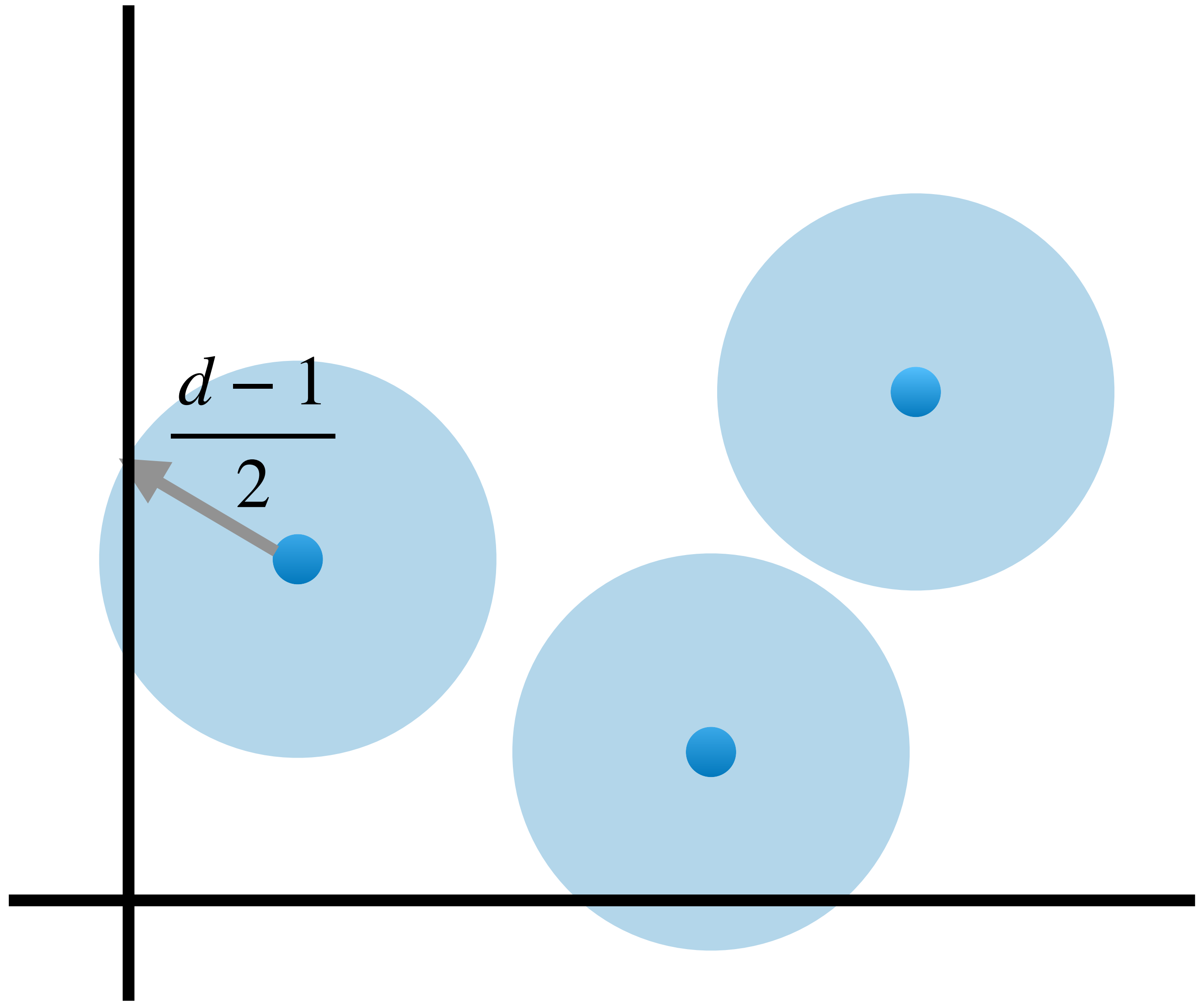
If $\leq \frac{d-1}{2}$ errors introduced, guaranteed to recover c from z.

Minimum distance and error-correction

$$\|v\|_0 = \#\{i : c_i \neq 0\}$$

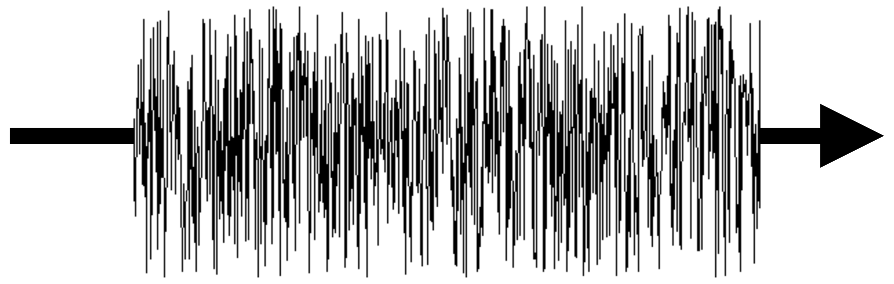
Minimum distance of C:

$$d(C) = \min_{c,c' \in C, c \neq c'} \|c - c'\|_0$$



min dist $\geq d$

Minimum distance captures error-correction properties of C

$c \in C$  z

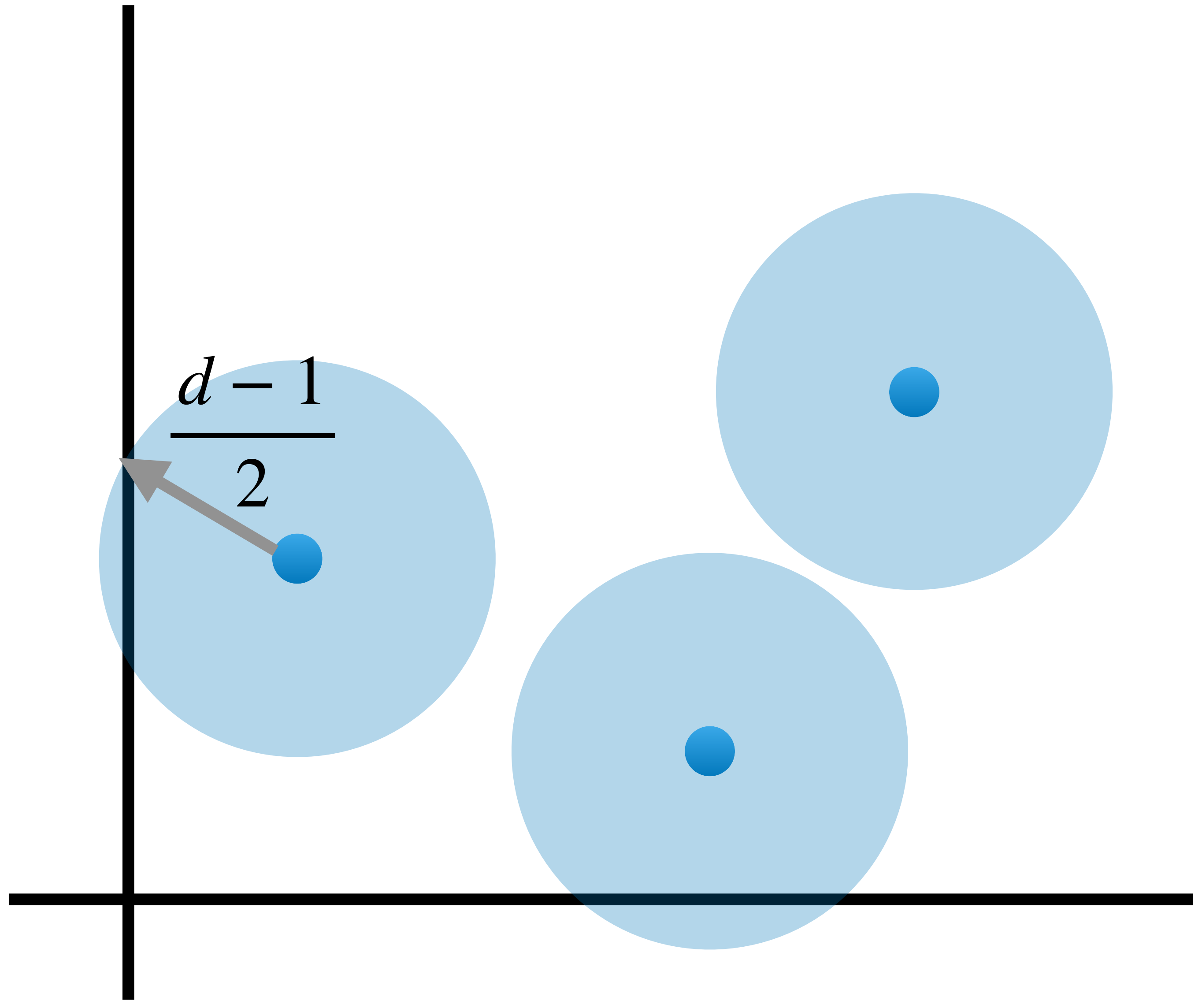
If $\leq \dfrac{d-1}{2}$ errors introduced, guaranteed to recover c from z.

Minimum distance and error-correction

$$\|v\|_0 = \#\{i : c_i \neq 0\}$$

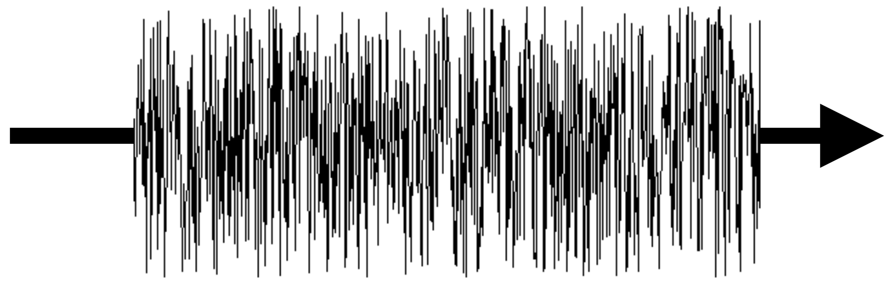
Minimum distance of C:

$$d(C) = \min_{c,c' \in C, c \neq c'} \|c - c'\|_0$$



min dist $\geq d$

Minimum distance captures error-correction properties of C

$c \in C$  z

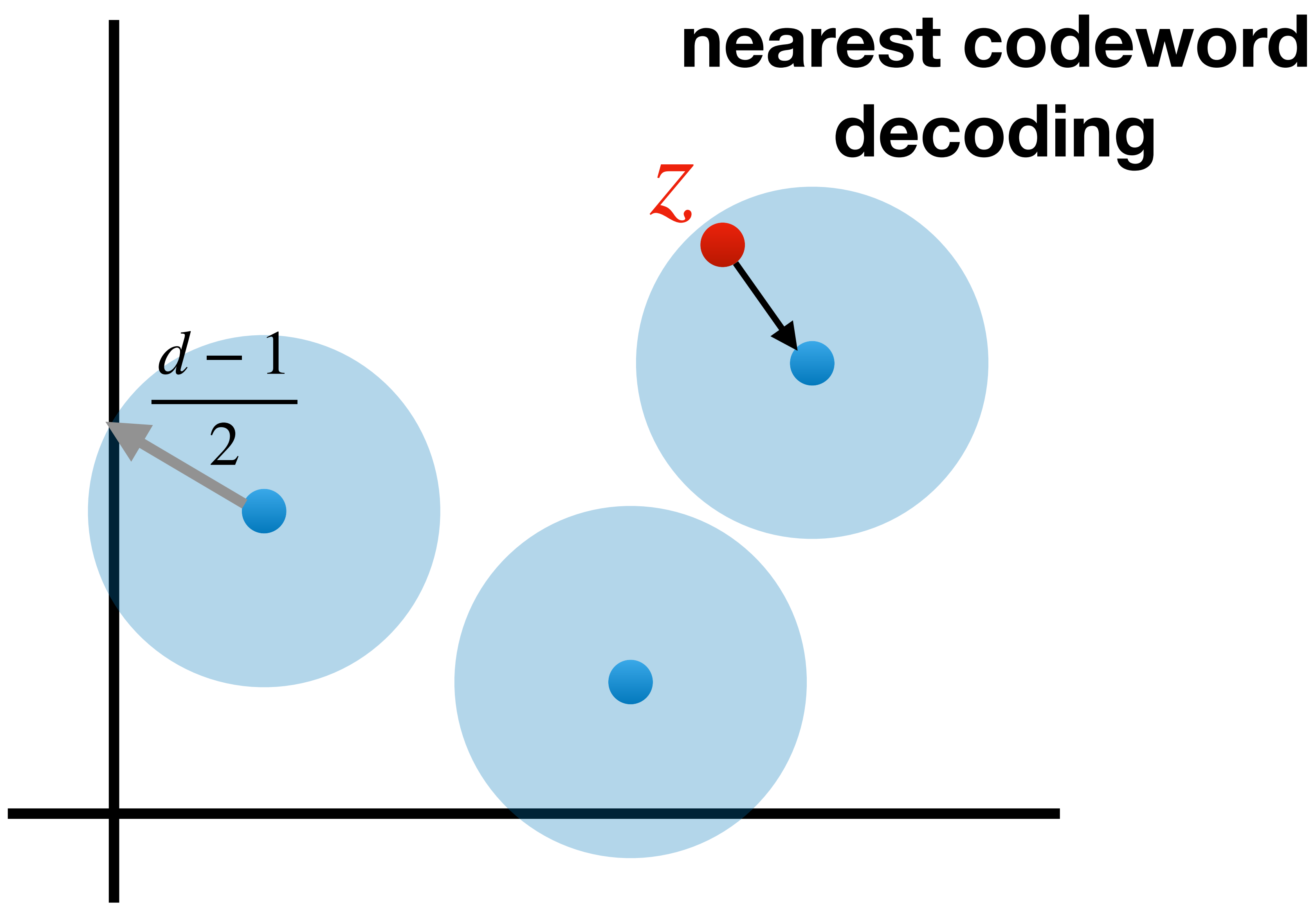
If $\leq \frac{d-1}{2}$ errors introduced, guaranteed to recover c from z.

Minimum distance and error-correction

$$\|v\|_0 = \#\{i : c_i \neq 0\}$$

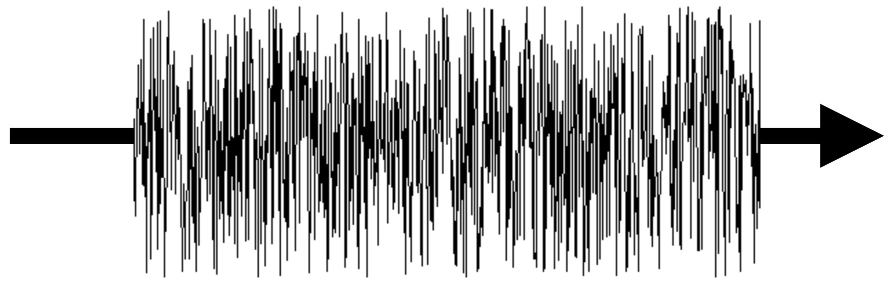
Minimum distance of C:

$$d(C) = \min_{c,c' \in C, c \neq c'} \|c - c'\|_0$$



min dist $\geq d$

Minimum distance captures error-correction properties of C

$c \in C$  z

If $\leq \frac{d-1}{2}$ errors introduced, guaranteed to recover c from z.

Fundamental coding problems

Nearest Codeword Problem over $\mathbb{F}_q$ (NCP$_q$)

Input: A generator matrix $G \in \mathbb{F}_q^{n \times k}$, a distance bound $d \geq 0$, and a target vector $t \in \mathbb{F}_q^n$

(YES) There is $c \in C(G)$ such that $\|c - t\|_0 \leq d$

(NO) For every $c \in C(G)$ it holds that $\|c - t\|_0 > d$

Fundamental coding problems

Minimum Distance Problem over $\mathbb{F}_q$ (MDP$_q$)

Input: A generator matrix $G \in \mathbb{F}_q^{n \times k}$ and a distance bound $d \geq 0$

(YES) There is $c \in C(G)$ such that $\|c\|_0 \leq d$ (⟺ C has minimum distance $\leq d$)

(NO) For every $c \in C(G)$ it holds that $\|c\|_0 > d$ (⟺ C has minimum distance $> d$)

MDP$_q$ is NCP$_q$ with target $t = 0$

What is a lattice?

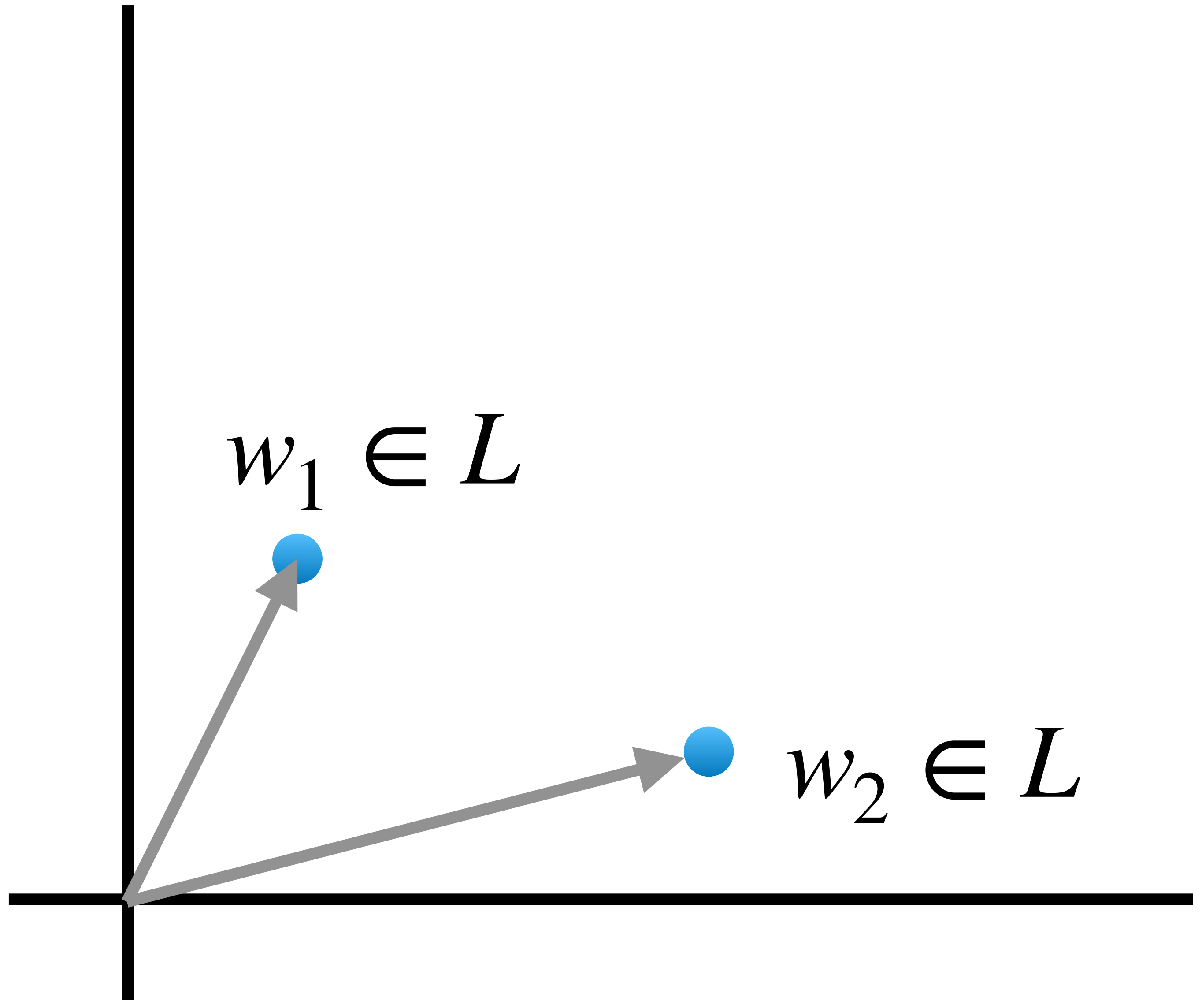
$L \subseteq \mathbb{R}^n$ is a lattice if it is a discrete subgroup of $\mathbb{R}^n$.

There exists a **basis** $B \in \mathbb{R}^{n \times k}$ such that $L = \{Bv : v \in \mathbb{Z}^k\}$.

What is a lattice?

$L \subseteq \mathbb{R}^n$ is a lattice if it is a discrete subgroup of $\mathbb{R}^n$.

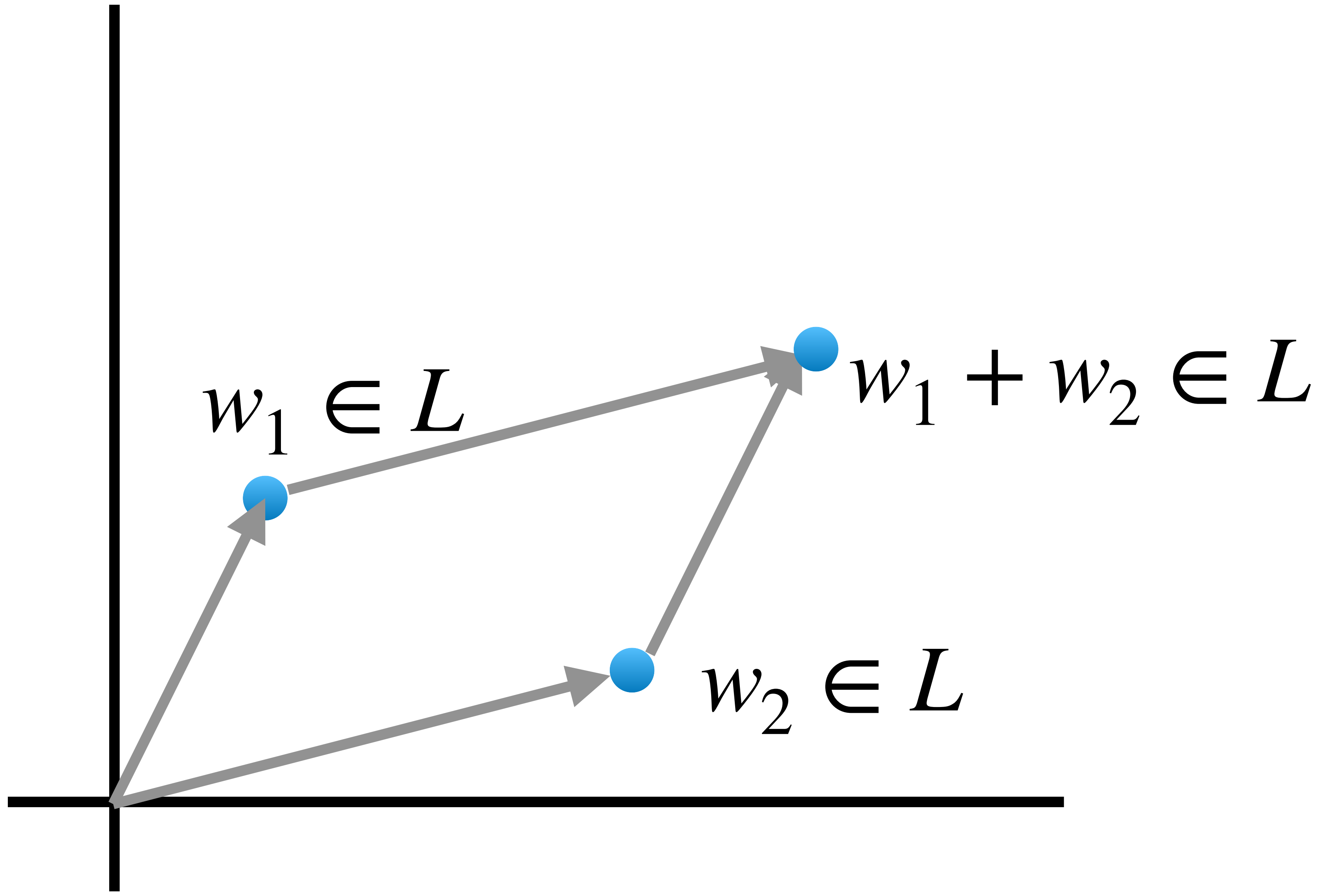
There exists a **basis** $B \in \mathbb{R}^{n \times k}$ such that $L = \{Bv : v \in \mathbb{Z}^k\}$.



What is a lattice?

$L \subseteq \mathbb{R}^n$ is a lattice if it is a discrete subgroup of $\mathbb{R}^n$.

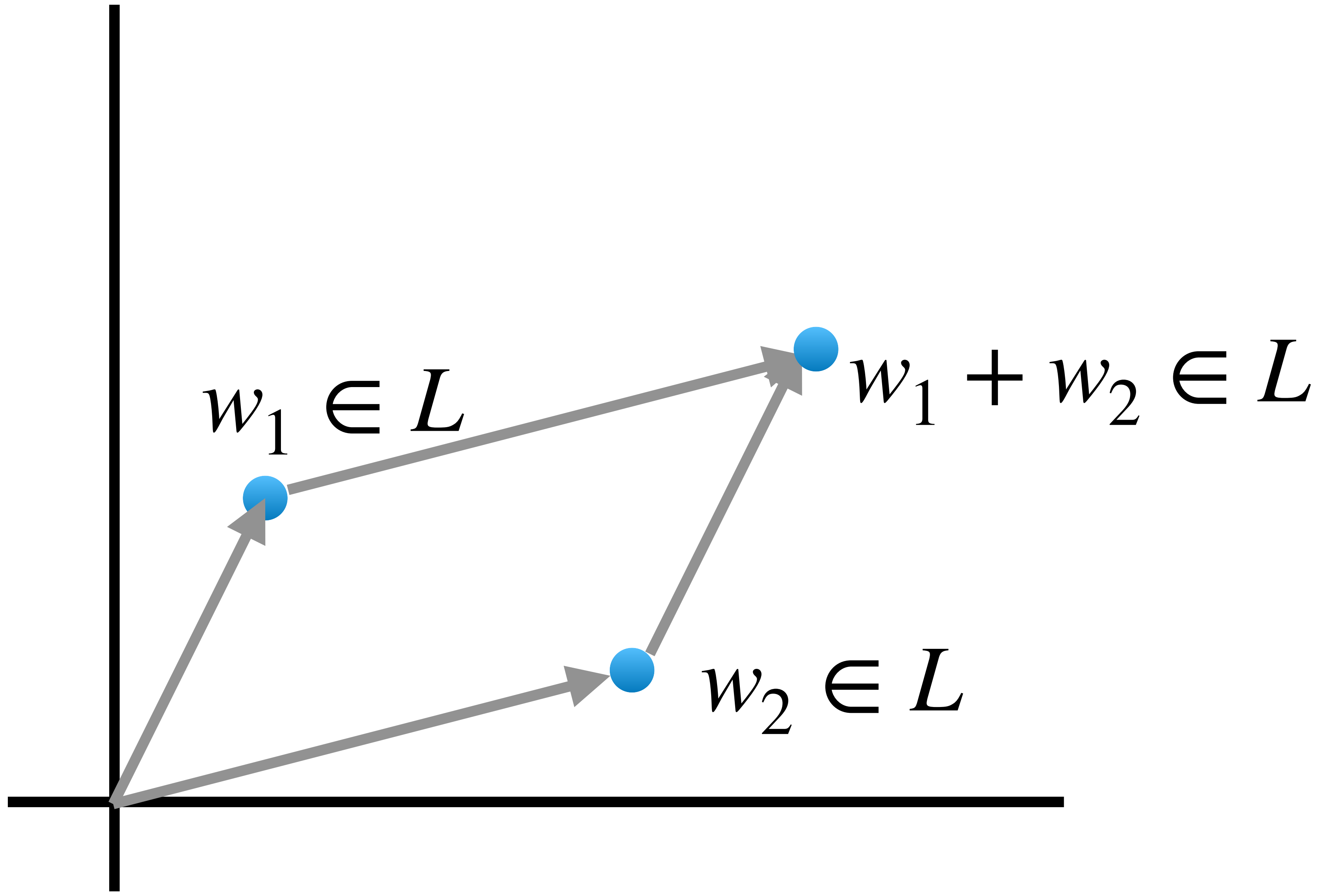
There exists a **basis** $B \in \mathbb{R}^{n \times k}$ such that $L = \{Bv : v \in \mathbb{Z}^k\}$.



What is a lattice?

$L \subseteq \mathbb{R}^n$ is a lattice if it is a discrete subgroup of $\mathbb{R}^n$.

There exists a **basis** $B \in \mathbb{R}^{n \times k}$ such that $L = \{Bv : v \in \mathbb{Z}^k\}$.

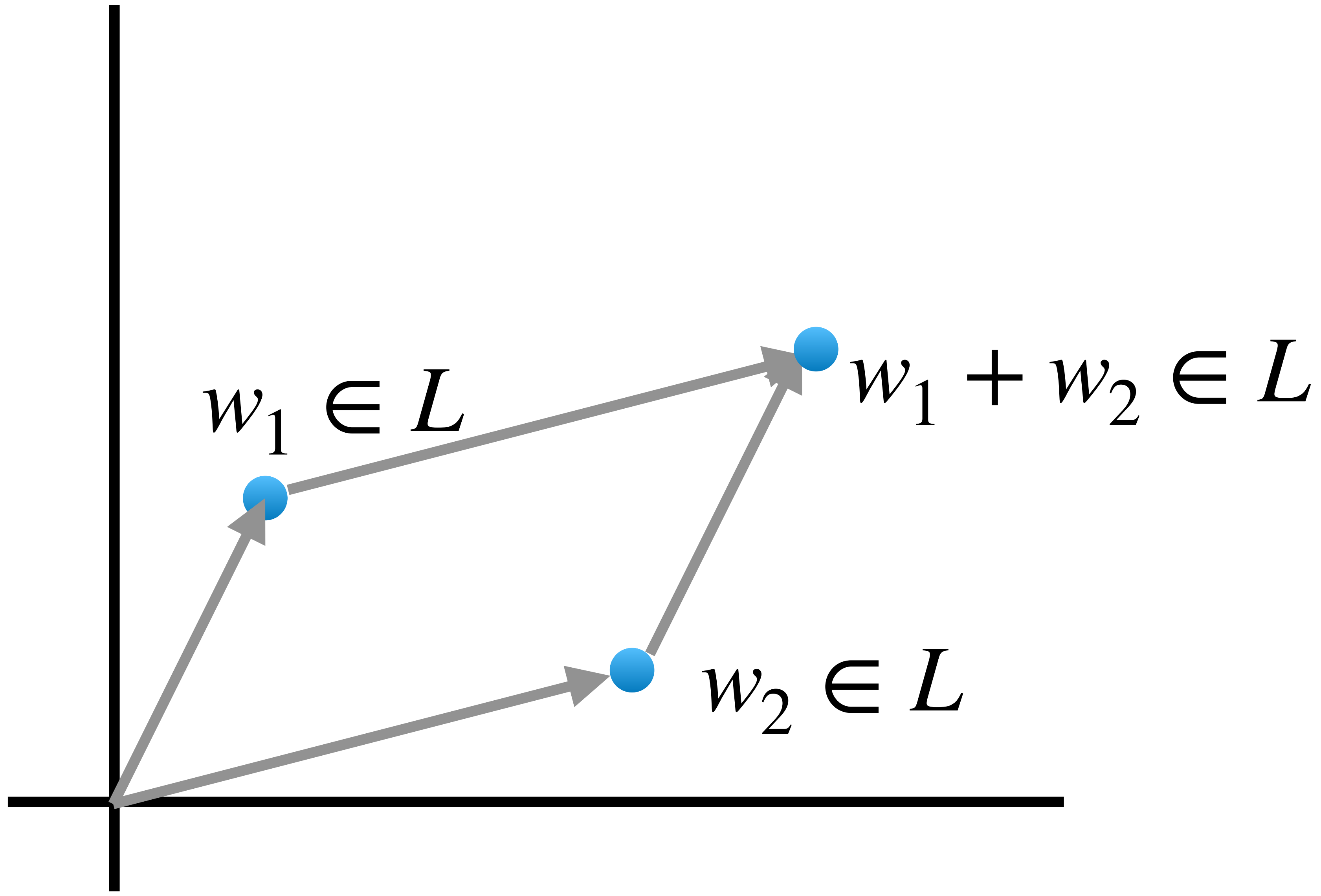


$$\|v\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}$$ (ℓ_p-norm of v, $p \geq 1$)

What is a lattice?

$L \subseteq \mathbb{R}^n$ is a lattice if it is a discrete subgroup of $\mathbb{R}^n$.

There exists a **basis** $B \in \mathbb{R}^{n \times k}$ such that $L = \{Bv : v \in \mathbb{Z}^k\}$.



$$\|v\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}$$ (ℓ_p-norm of v, $p \geq 1$)

"Minimum distance" of L:

$$\min_{v \in L \setminus \{0\}} \|v\|_p$$

Fundamental lattice problems

Closest Vector Problem in the ℓ_p norm (CVP$_p$)

Input: A basis $B \in \mathbb{Z}^{n \times k}$, a distance bound $d \geq 0$, and a target vector $t \in \mathbb{Z}^n$

(YES) There is $v \in L(B)$ such that $\|v - t\|_p \leq d$

(NO) For every $v \in L(B)$ it holds that $\|v - t\|_p > d$

Fundamental lattice problems

Shortest Vector Problem in the ℓ_p norm (SVP$_p$)

Input: A basis $B \in \mathbb{Z}^{n \times k}$ and a distance bound $d \geq 0$

(YES) There is $v \in L(B)$ such that $\|v\|_p \leq d$

(NO) For every $v \in L(B)$ it holds that $\|v\|_p > d$

SVP$_p$ is CVP$_p$ with target $t = 0$

How hard are all these problems?

Pretty hard!

All **NP-hard**:

- NCP: Berlekamp, McEliece, van Tilborg ’78
- MDP: Vardy ’97
- CVP: van Emde Boas ’81 ($p = 2$)
- SVP: Ajtai ’98 ($p = 2$)

What if we only want approximate solutions?

For example, γ-approximate SVP$_p$ for approximation factor $\gamma \geq 1$:

Input: A basis $B \in \mathbb{Z}^{n \times k}$ and a distance bound d

(YES) There exists $v \in L(B)$ such that $\|v\|_p \leq d$

(NO) Every $v \in L(B)$ satisfies $\|v\|_p > \gamma d$

What if we only want approximate solutions?

For example, γ-approximate SVP$_p$ for approximation factor $\gamma \geq 1$:

Input: A basis $B \in \mathbb{Z}^{n \times k}$ and a distance bound d

(YES) There exists $v \in L(B)$ such that $\|v\|_p \leq d$

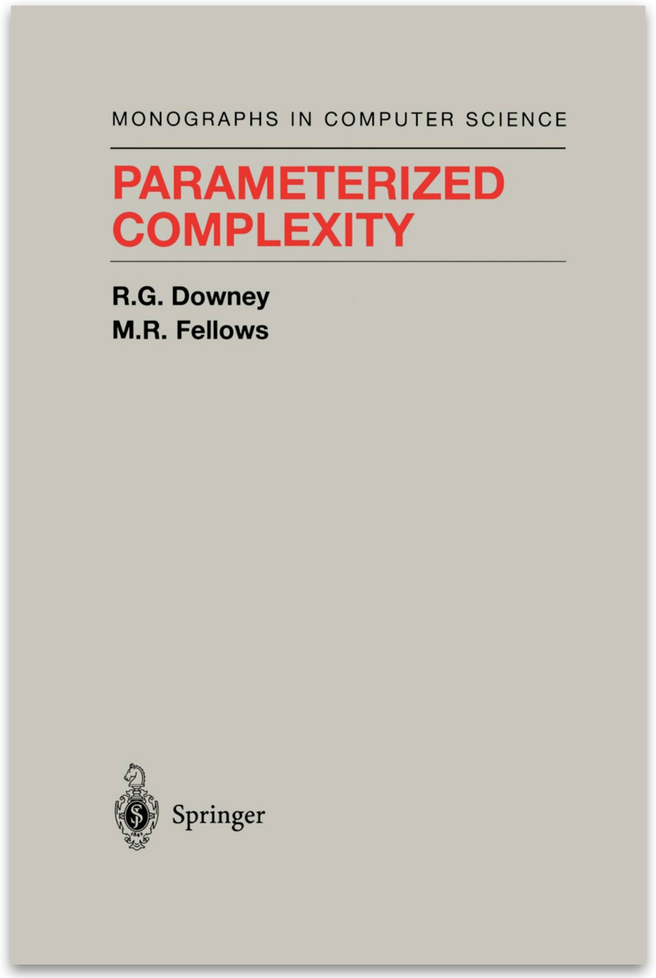
(NO) Every $v \in L(B)$ satisfies $\|v\|_p > \gamma d$

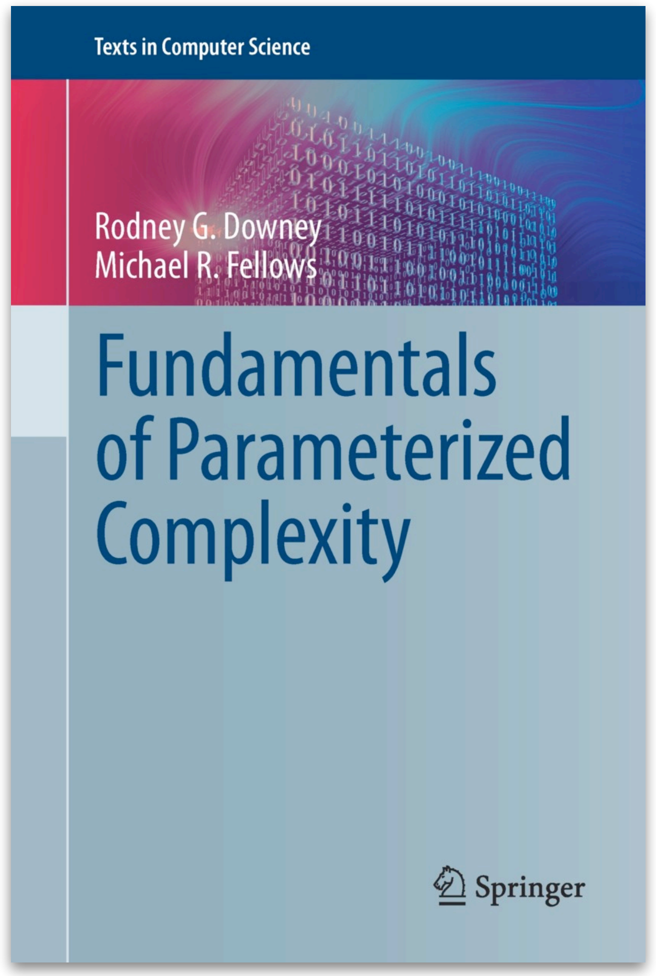
Still pretty hard!

All **NP-hard** for arbitrary constant approximation factor γ (and beyond):

- NCP: Håstad ’01
- MDP: Dumer, Micciancio, Sudan ’03
- CVP/SVP: Micciancio ’00; Khot ’05; Haviv, Regev ’12 ($p \geq 1$)

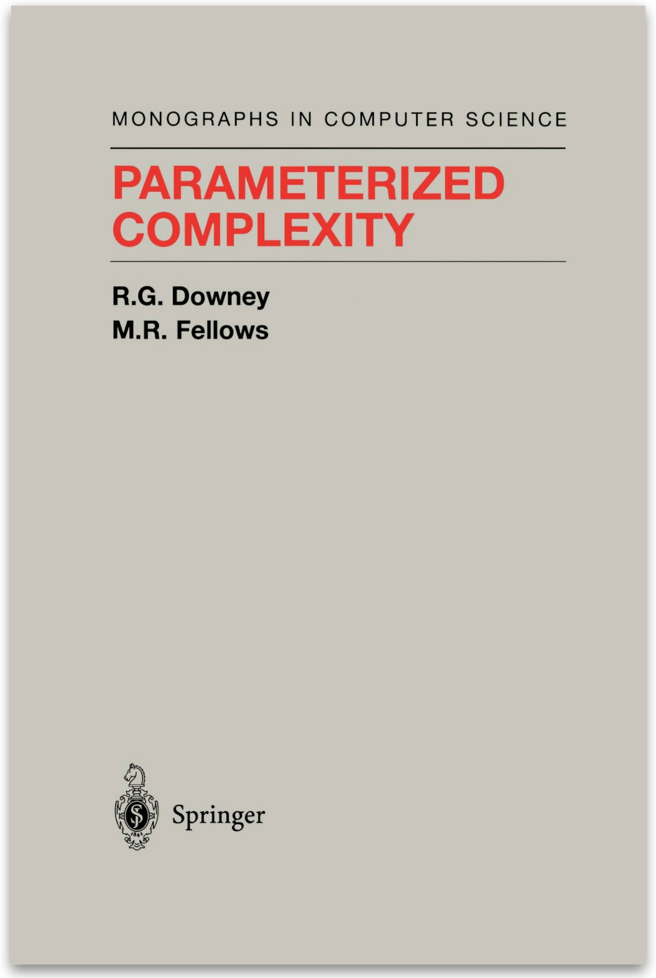
Parameterized complexity

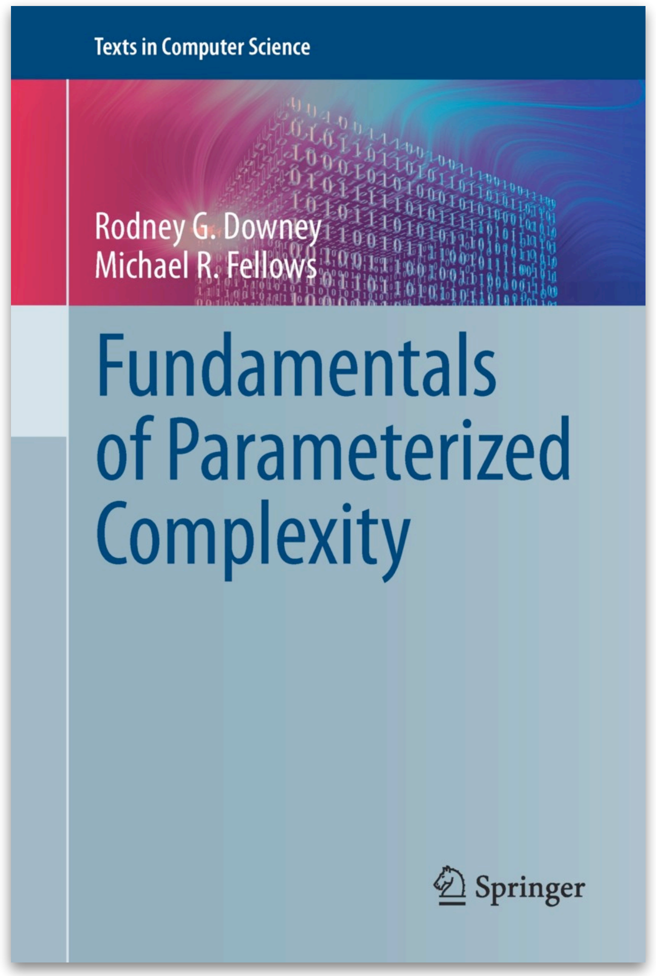




Problem Π is NP-hard. Does this mean that “real-world” instances of Π are computationally intractable? **Not necessarily…**

Parameterized complexity





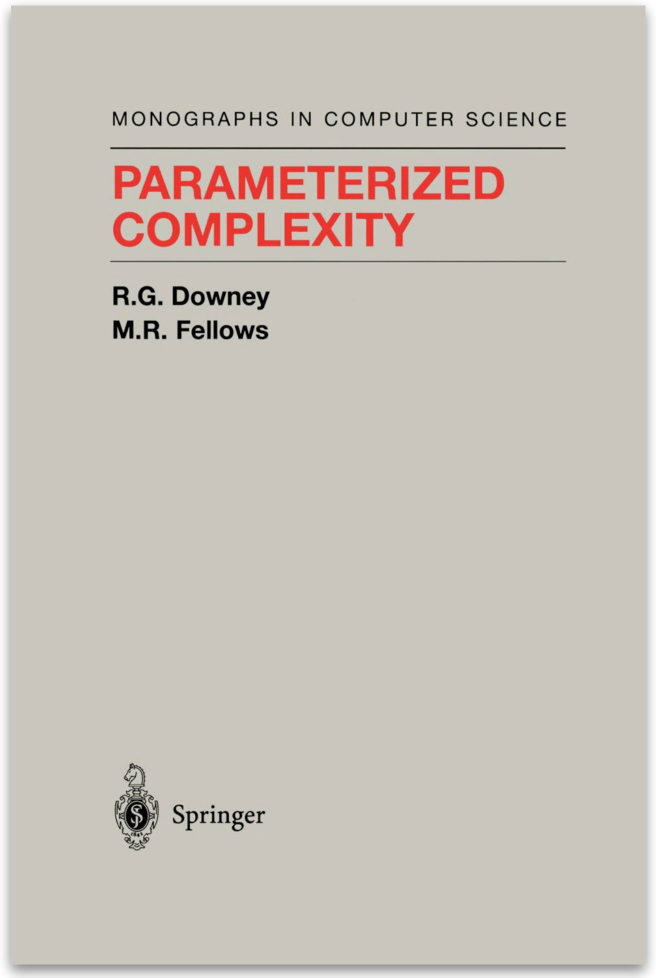
Problem Π is NP-hard. Does this mean that “real-world” instances of Π are computationally intractable? **Not necessarily…**

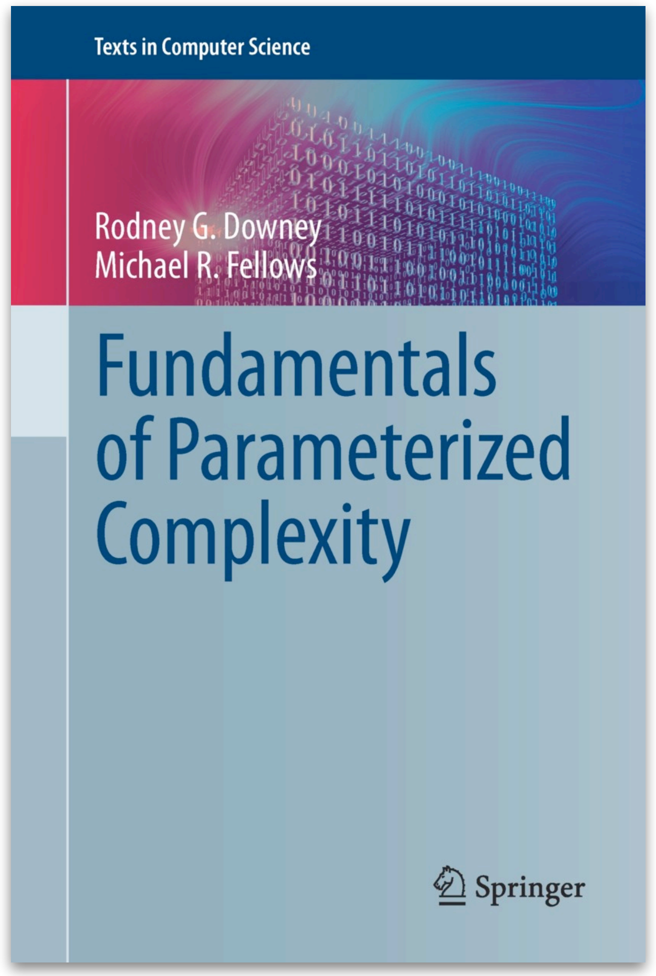
Example: Vertex Cover

Input: n-vertex graph G and parameter k

YES if G has vertex cover of size $\leq k$, **NO** otherwise.

Parameterized complexity





Problem Π is NP-hard. Does this mean that "real-world" instances of Π are computationally intractable? **Not necessarily…**

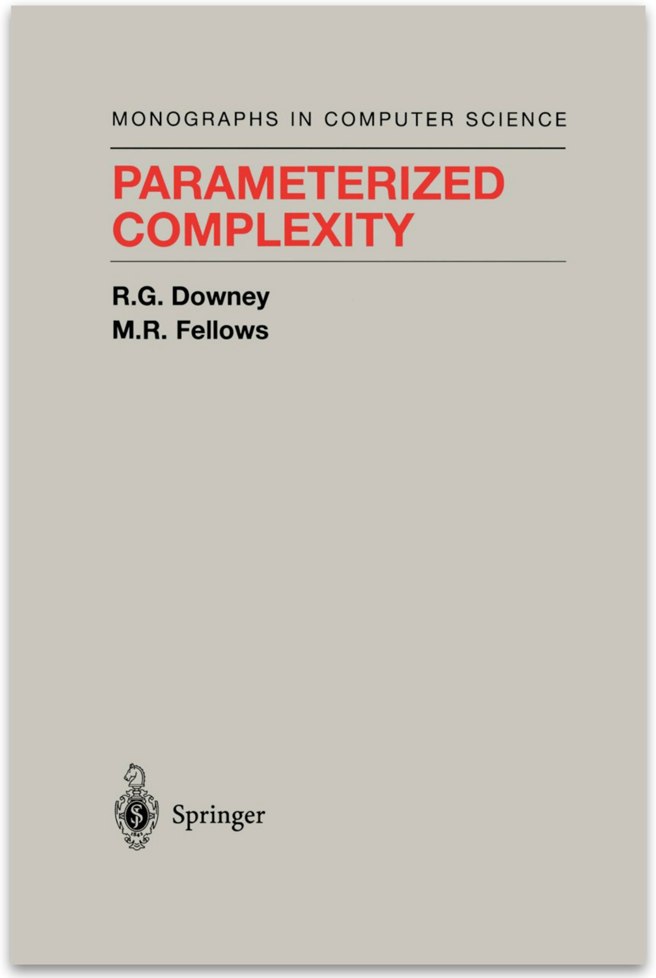
Example: Vertex Cover

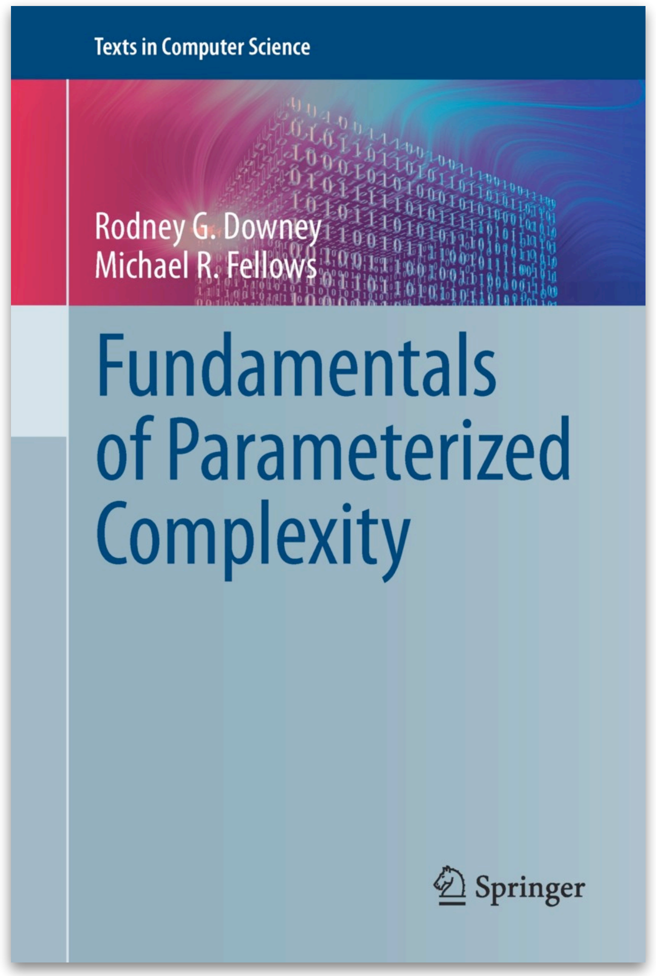
Input: n-vertex graph G and parameter k

YES if G has vertex cover of size $\leq k$, **NO** otherwise.

- Vertex Cover is NP-hard (Karp '72)
- There's an algorithm running in time $O(2^k n)$ (Fellows '88)

Parameterized complexity





Problem Π is NP-hard. Does this mean that "real-world" instances of Π are computationally intractable? **Not necessarily…**

Example: Vertex Cover

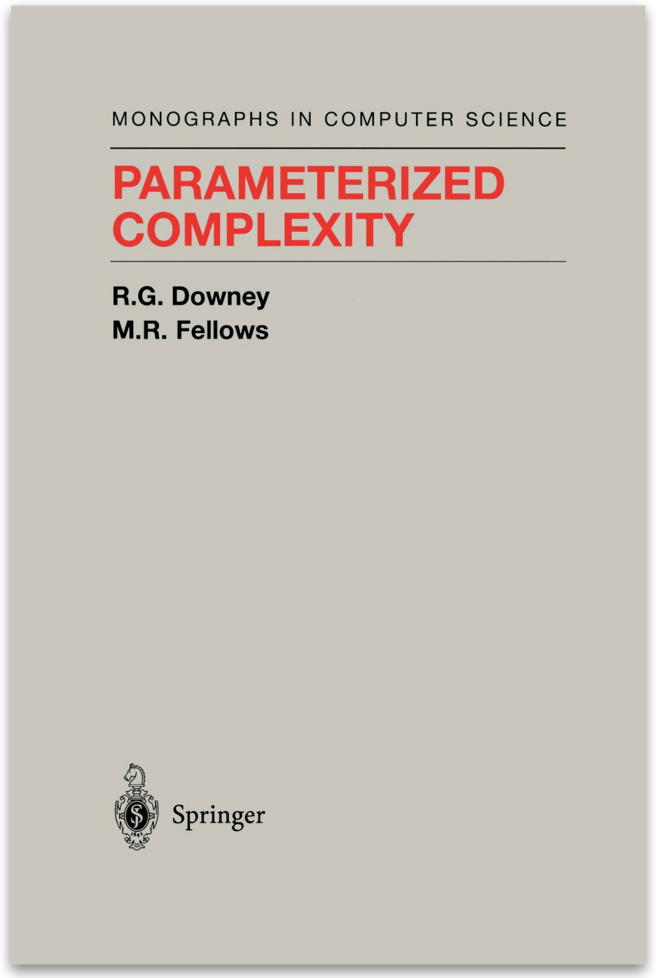
Input: n-vertex graph G and parameter k

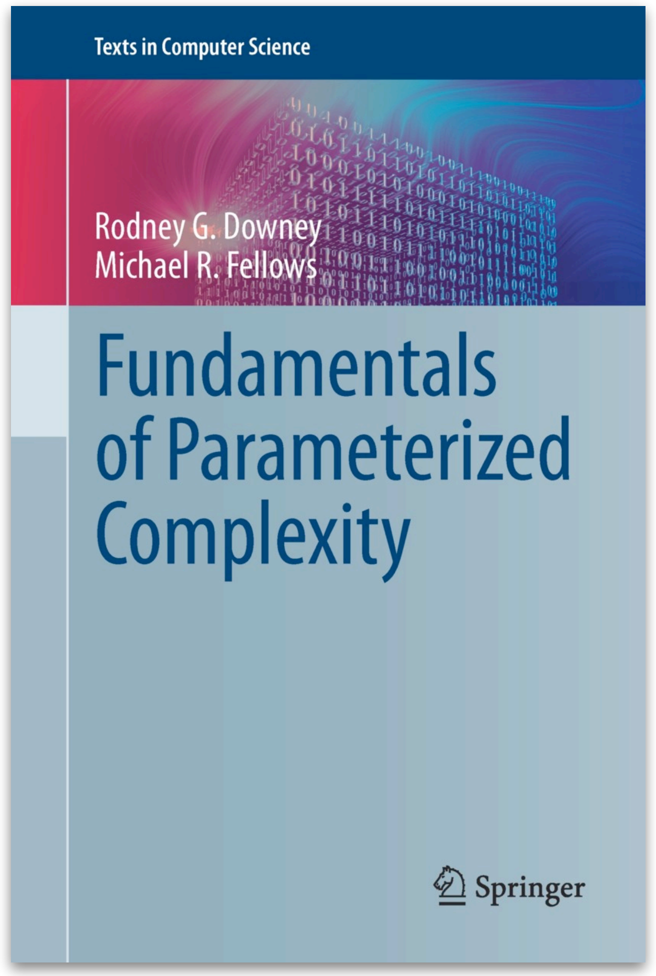
YES if G has vertex cover of size $\leq k$, **NO** otherwise.

- Vertex Cover is NP-hard (Karp '72)
- There's an algorithm running in time $O(2^k n)$ (Fellows '88)

$\Longrightarrow$ **Practical algo for Vertex Cover instances with small k!**

Parameterized complexity





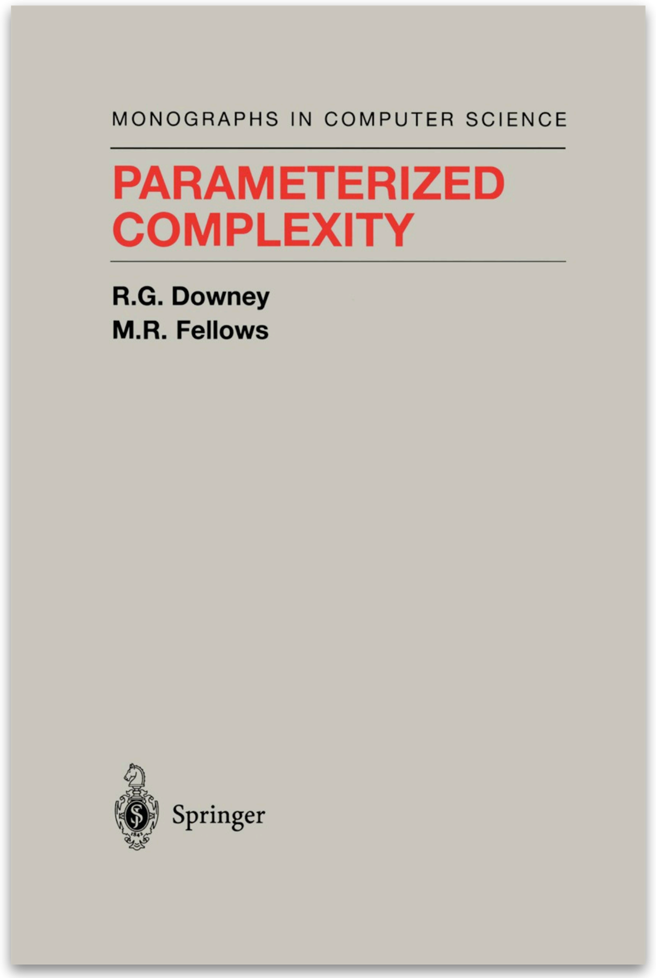
See input to problem Π as (x, k), where k is the **parameter of interest.**

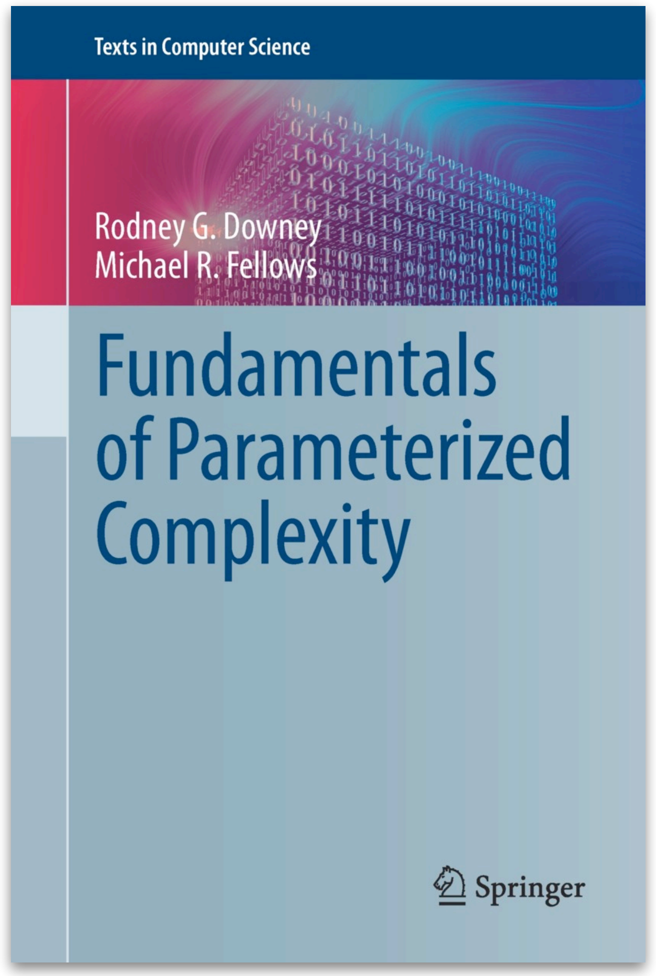
A problem Π parameterized by k is Fixed-Parameter Tractable (FPT) if there is an associated algorithm running in time

$$f(k) \cdot |x|^{O(1)}$$

for **some** function f.

Parameterized complexity





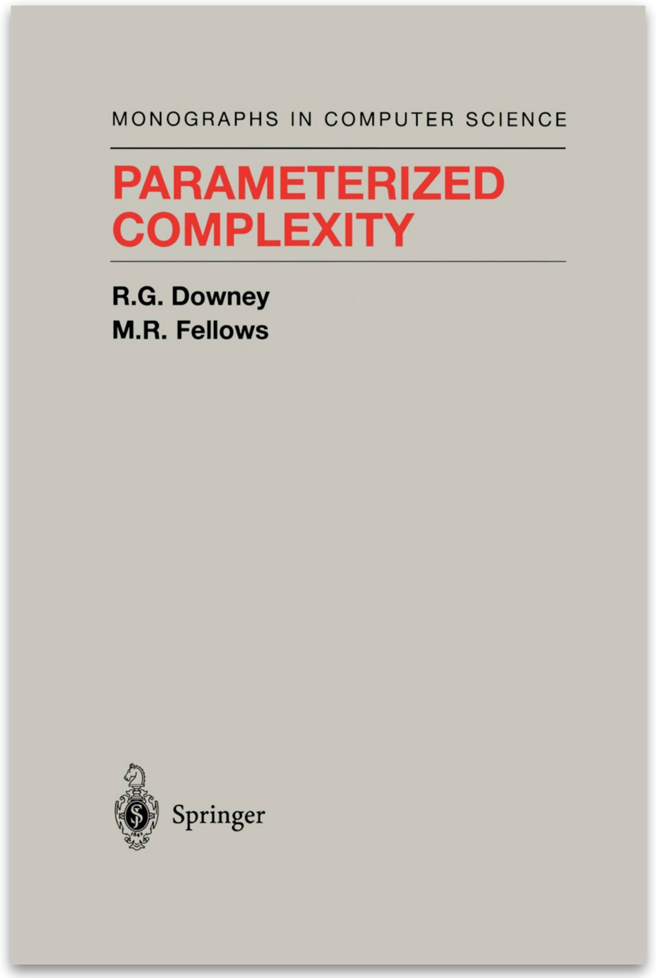
See input to problem Π as (x, k), where k is the **parameter of interest.**

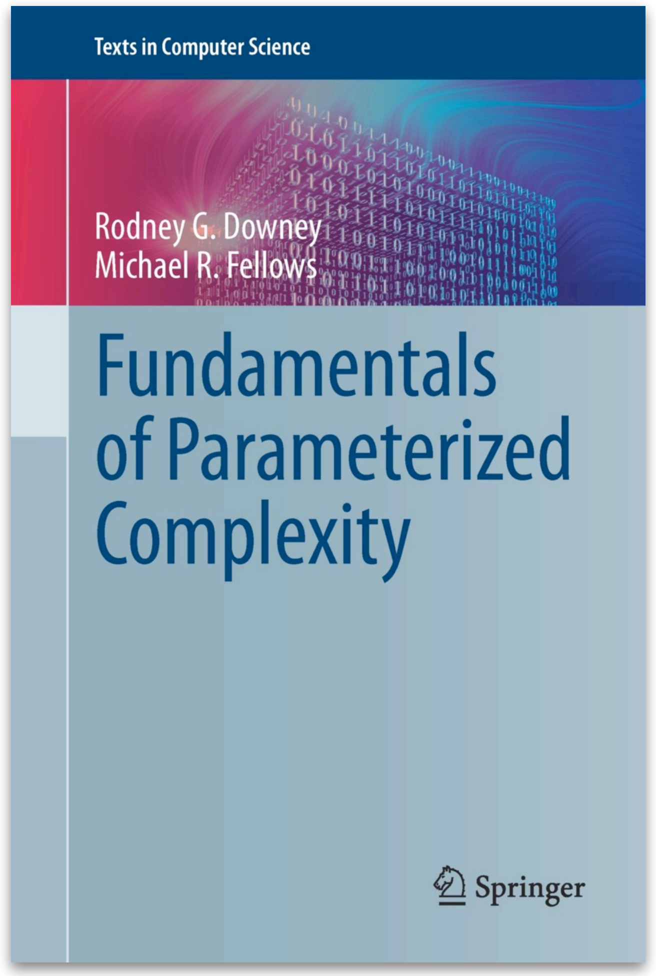
A problem Π parameterized by k is Fixed-Parameter Tractable (FPT) if there is an associated algorithm running in time

$$\underbrace{f(k) \cdot |x|^{O(1)}}_{\text{“decouples” input parts } x \text{ and } k}$$

for **some** function f.

Parameterized complexity





See input to problem Π as (x, k), where k is the **parameter of interest.**

A problem Π parameterized by k is Fixed-Parameter Tractable (FPT) if there is an associated algorithm running in time

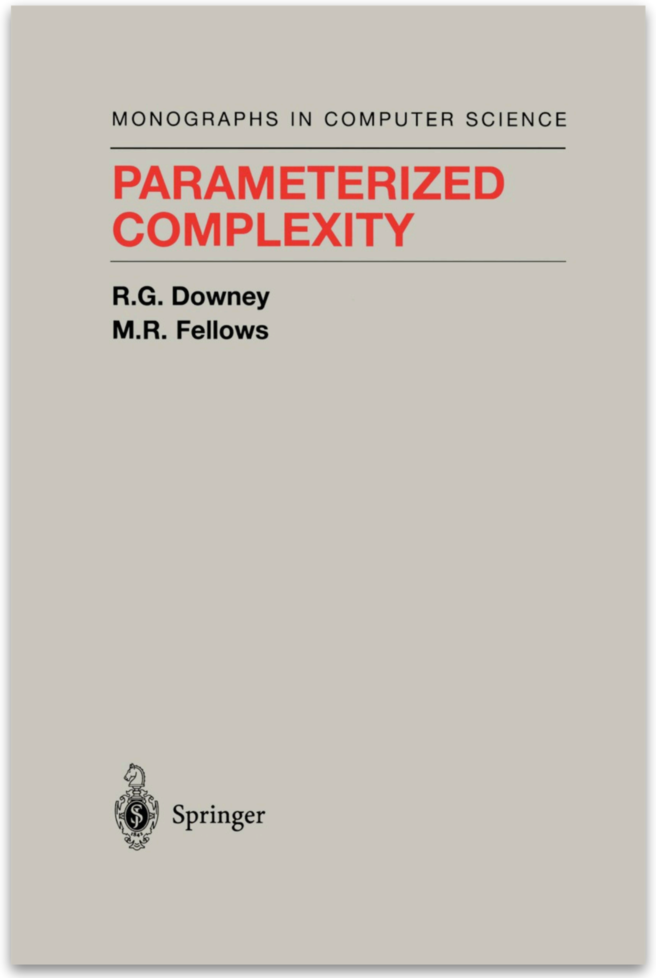
$$\underline{f(k) \cdot |x|^{O(1)}}$$

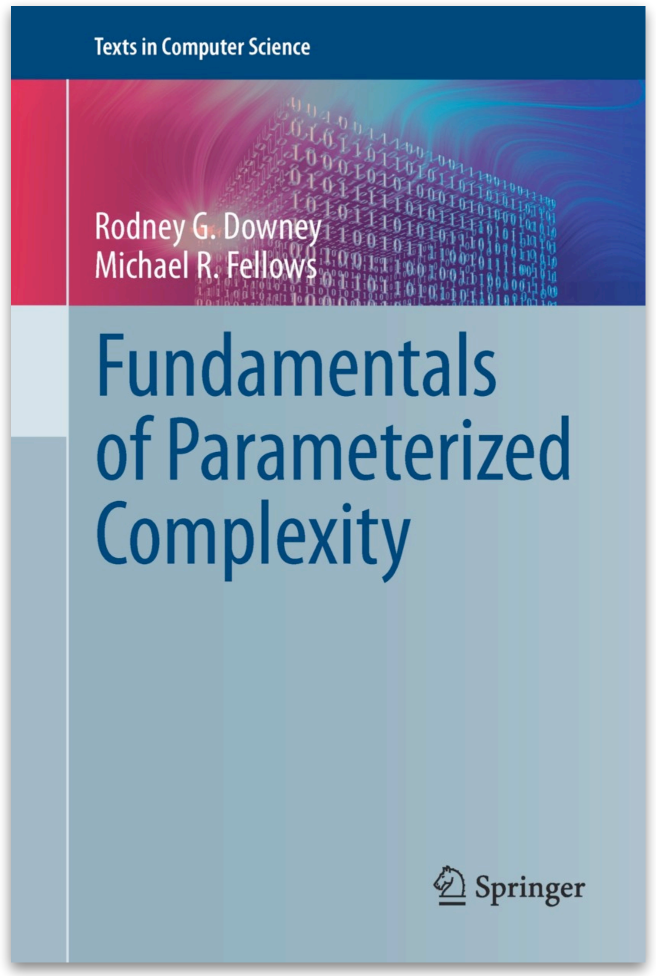
"decouples" input parts x and k

for **some** function f.

Example: Vertex Cover parameterized by cover size is FPT!

Parameterized complexity





See input to problem Π as (x, k), where k is the **parameter of interest.**

A problem Π parameterized by k is Fixed-Parameter Tractable (FPT) if there is an associated algorithm running in time

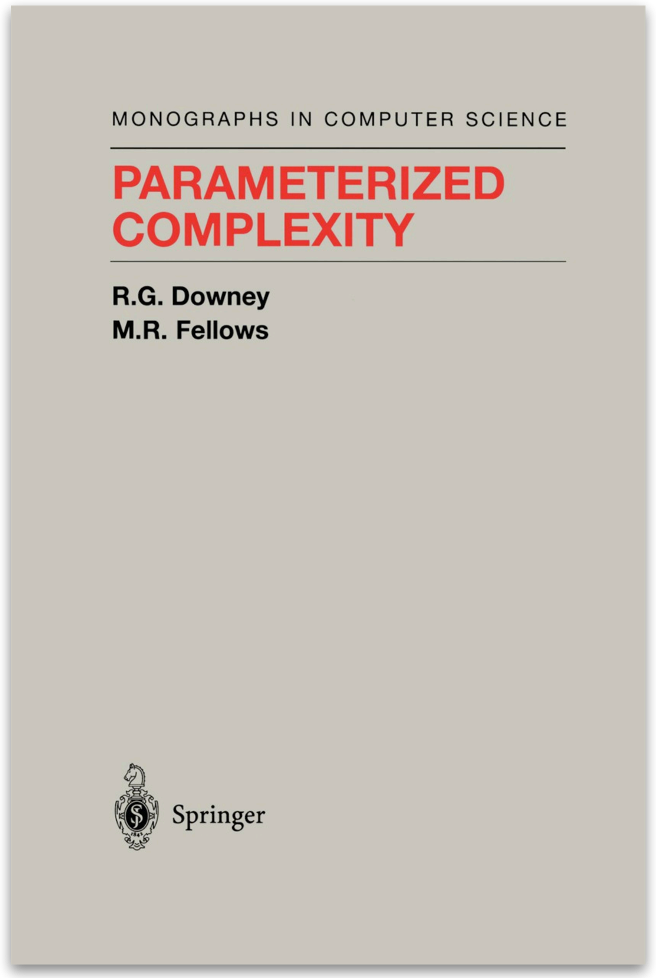
$$\underbrace{f(k) \cdot |x|^{O(1)}}_{\text{“decouples” input parts } x \text{ and } k}$$

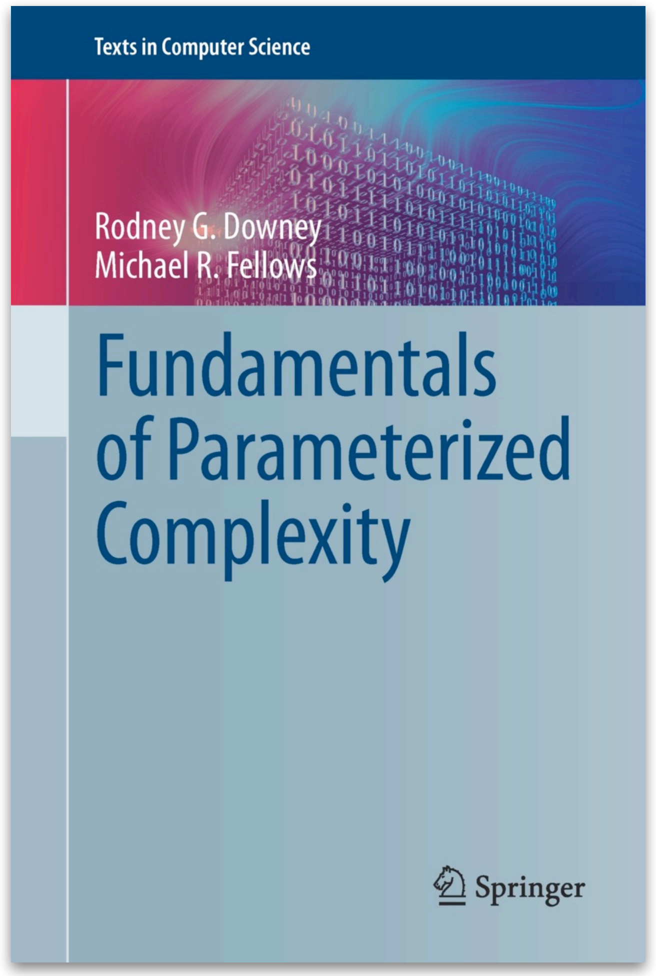
for **some** function f.

Example: Vertex Cover parameterized by cover size is FPT!

Are there interesting problems believed not to be FPT?

Parameterized complexity





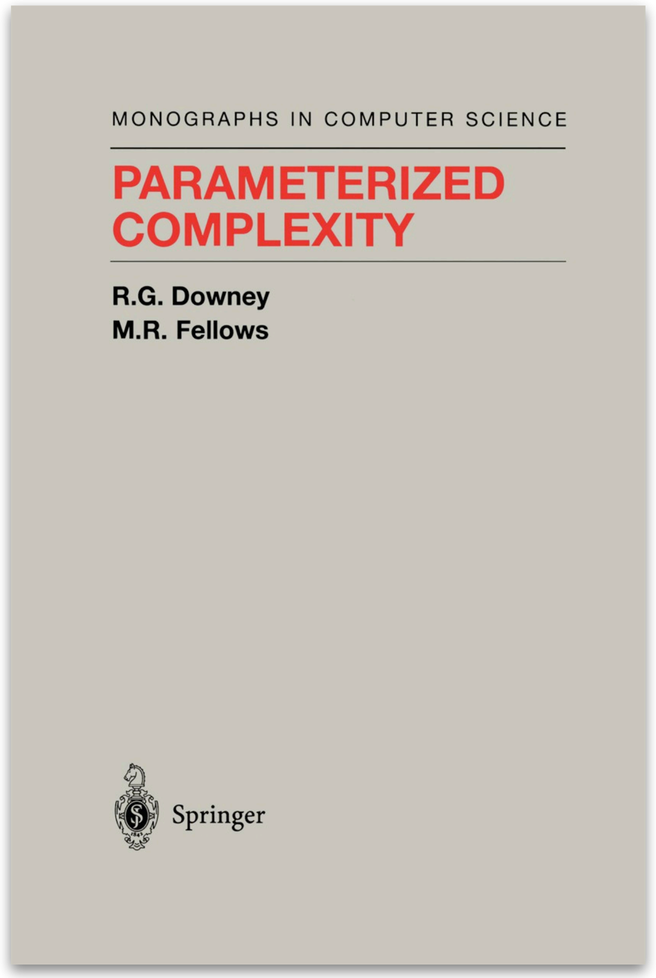
Clique

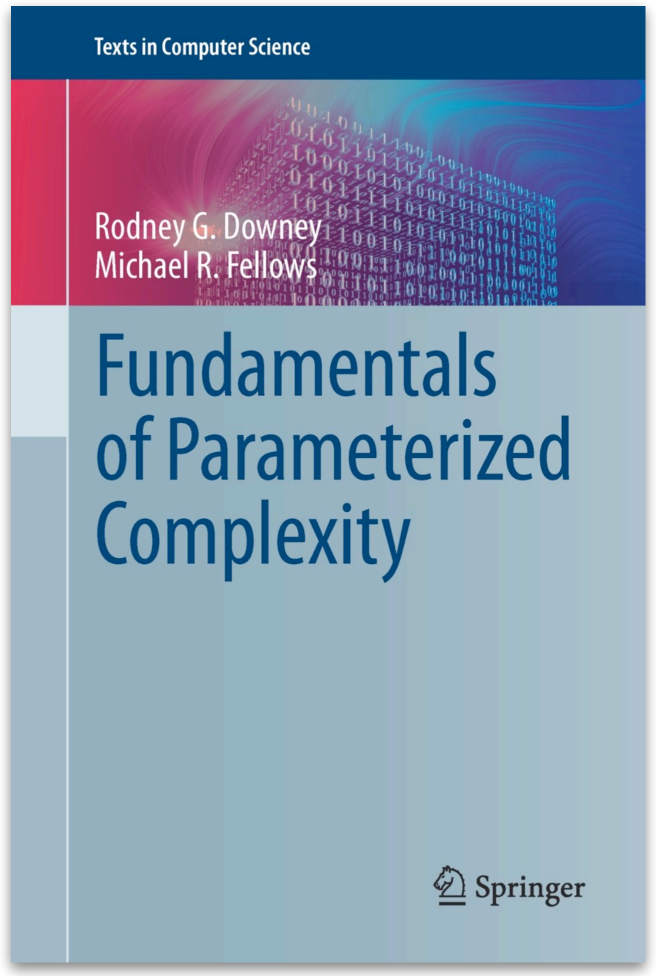
Input: n-vertex graph G and **parameter of interest** k

YES if G has clique of size k, **NO** otherwise.

Clique is the **canonical hard** parameterized problem. A parameterized problem Π is "hard" if there is an "FPT reduction" from Clique to Π.

Parameterized complexity





Clique

Input: n-vertex graph G and **parameter of interest** k

YES if G has clique of size k, **NO** otherwise.

Clique is the **canonical hard** parameterized problem. A parameterized problem Π is "hard" if there is an "FPT reduction" from Clique to Π.

FPT reduction:

(G, k) Clique instance $\xrightarrow{\text{algorithm running in time } f(k) \cdot n^c}$ (x, k') Π instance

YES/**NO** instances mapped to **YES**/**NO** instances & $k' = g(k)$

Dictionary

Unparameterized world

- The class P
- The class NP
- Π is NP-hard if there is a polynomial-time reduction from 3SAT to Π

Parameterized world

- The class FPT
- The class $W[1]$
- Π is $W[1]$-hard if there is an **FPT-reduction** from Clique to Π

FPT reduction:

(G, k)
Clique instance

$\xrightarrow{\text{algorithm running in time } f(k) \cdot n^c}$

(x, k')
Π instance

YES/NO instances mapped to YES/NO instances & $k' = g(k)$

Parameterized hardness of coding and lattice problems

Downey-Fellows ’13: $W[1]$-hardness of MDP_2 one of the “most infamous” open problems in parameterized complexity.

Parameterized hardness of coding and lattice problems

Downey-Fellows ’13: $W[1]$-hardness of MDP_2 one of the “most infamous” open problems in parameterized complexity.

2021: Enter Bhattacharyya, Bonnet, Egri, Ghoshal, Karthik C. S., Lin, Manurangsi, and Marx!

Parameterized hardness of coding and lattice problems

Downey-Fellows '13: $W[1]$-hardness of MDP$_2$ one of the "most infamous" open problems in parameterized complexity.

2021: Enter Bhattacharyya, Bonnet, Egri, Ghoshal, Karthik C. S., Lin, Manurangsi, and Marx!

Codes:

γ-NCP$_q$ is $W[1]$-hard for all q and γ

γ-MDP$_2$ is $W[1]$-hard for all γ

Lattices:

γ-CVP$_p$ is $W[1]$-hard for all $p \geq 1$ and γ

γ-SVP$_p$ is $W[1]$-hard for all $p > 1$ and some (small) $\gamma(p) > 1$

Parameterized hardness of coding and lattice problems

Downey-Fellows '13: $W[1]$-hardness of MDP_2 one of the "most infamous" open problems in parameterized complexity.

2021: Enter Bhattacharyya, Bonnet, Egri, Ghoshal, Karthik C. S., Lin, Manurangsi, and Marx!

Codes:

γ-NCP_q is $W[1]$-hard for all q and γ

γ-MDP_2 is $W[1]$-hard for all γ

what about γ-MDP_q for $q > 2$?

Lattices:

γ-CVP_p is $W[1]$-hard for all $p \geq 1$ and γ

γ-SVP_p is $W[1]$-hard for all $p > 1$ and some (small) $\gamma(p) > 1$

what about γ-SVP_p for large γ?

and SVP_1?

Our results

Codes:

γ-MDP$_q$ is $W[1]$-hard for **all q and** all γ

Lattices:

γ-SVP$_1$ is $W[1]$**-hard for any** $\gamma < 2$

γ-SVP$_p$ is $W[1]$-hard for $p > 1$ and **all** $\gamma > 1$

NP-hardness of approximating SVP

A pretty cool approach of Khot

Let's take for granted that γ-CVP$_p$ is NP-hard, and reduce it to γ'-SVP$_p$.

Want: Transform a γ-CVP$_p$ instance (B, d, t) into a γ'-SVP$_p$ instance (B', d').

NP-hardness of approximating SVP

A pretty cool approach of Khot

Let's take for granted that γ-CVP$_p$ is NP-hard, and reduce it to γ'-SVP$_p$.

Want: Transform a γ-CVP$_p$ instance (B, d, t) into a γ'-SVP$_p$ instance (B', d').

The obvious approach:

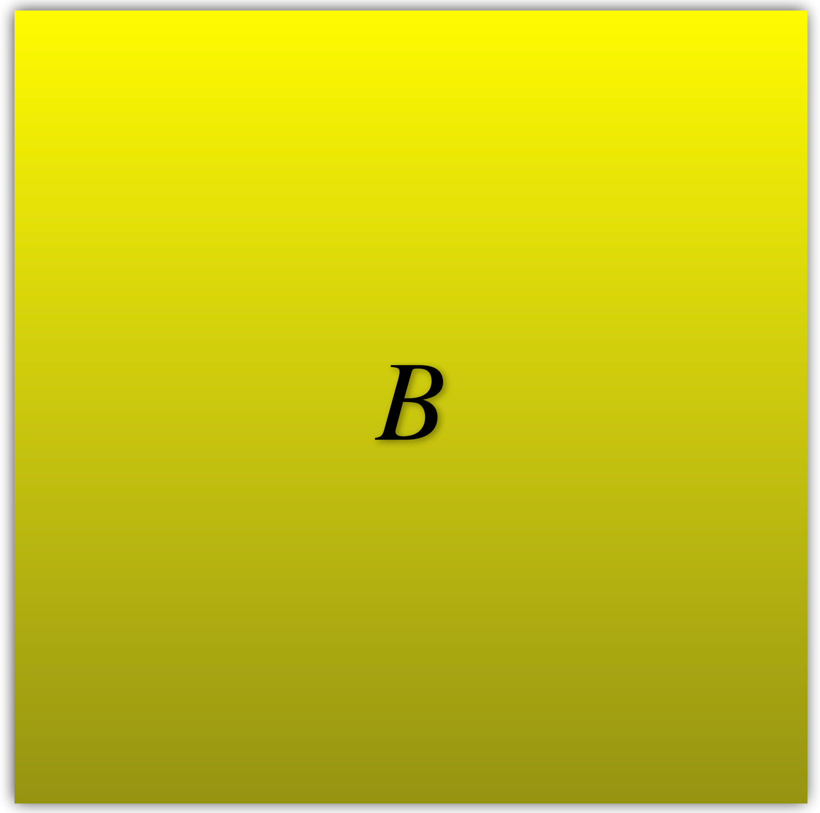
NP-hardness of approximating SVP

A pretty cool approach of Khot

Let's take for granted that γ-CVP$_p$ is NP-hard, and reduce it to γ'-SVP$_p$.

Want: Transform a γ-CVP$_p$ instance (B, d, t) into a γ'-SVP$_p$ instance (B', d').

The obvious approach:



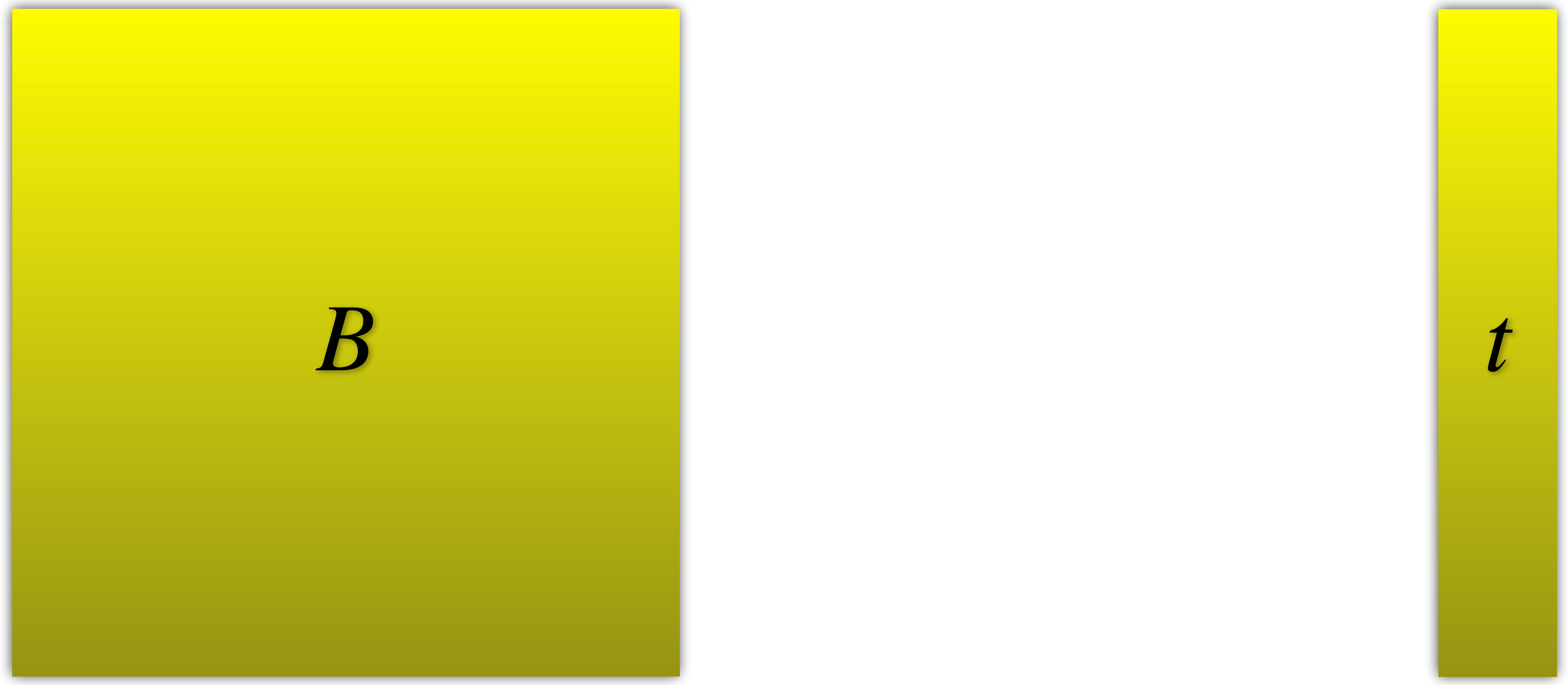
NP-hardness of approximating SVP

A pretty cool approach of Khot

Let's take for granted that γ-CVP$_p$ is NP-hard, and reduce it to γ'-SVP$_p$.

Want: Transform a γ-CVP$_p$ instance (B, d, t) into a γ'-SVP$_p$ instance (B', d').

The obvious approach:



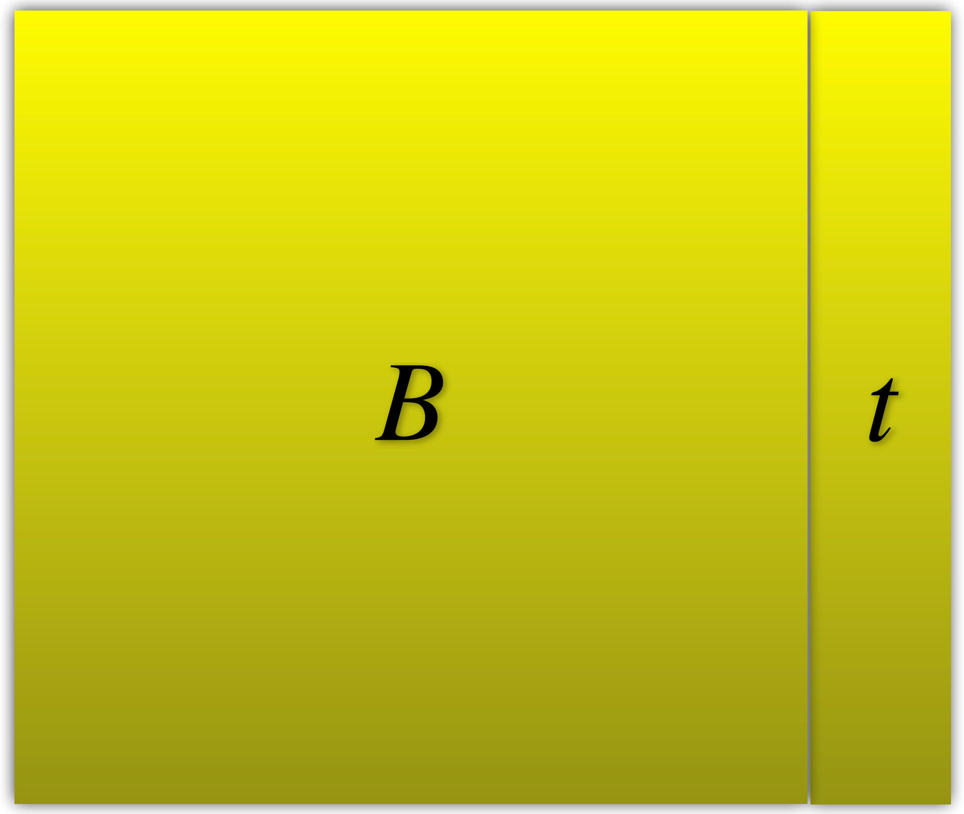
NP-hardness of approximating SVP

A pretty cool approach of Khot

Let's take for granted that γ-CVP$_p$ is NP-hard, and reduce it to γ'-SVP$_p$.

Want: Transform a γ-CVP$_p$ instance (B, d, t) into a γ'-SVP$_p$ instance (B', d').

The obvious approach:



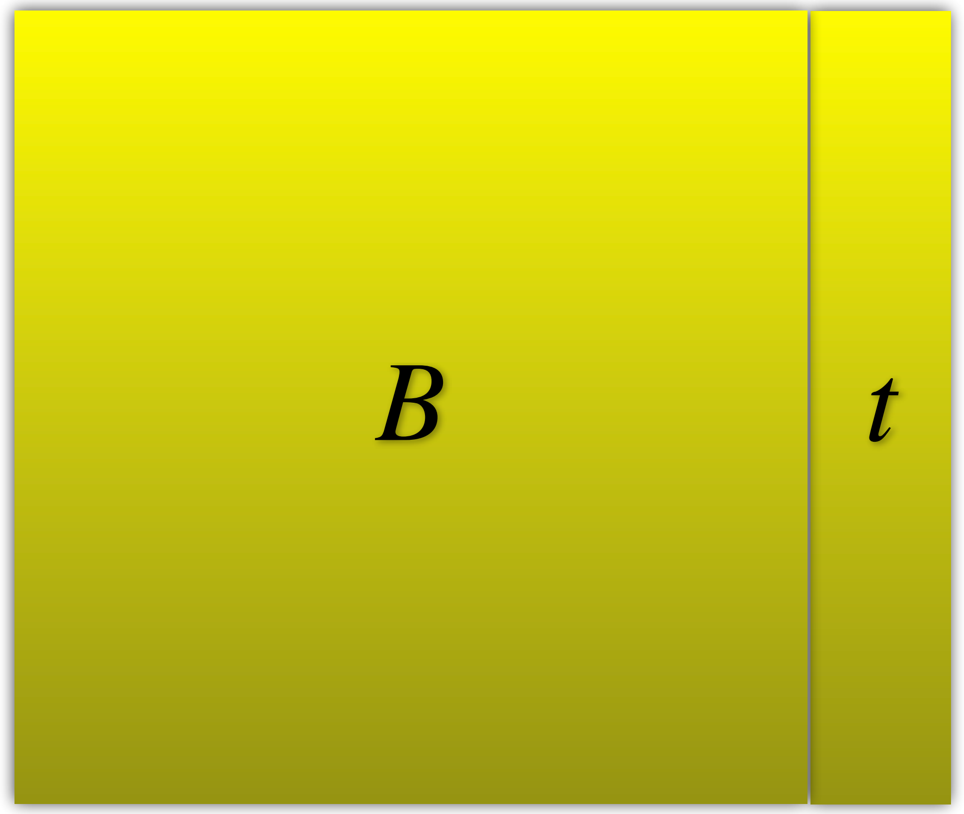
NP-hardness of approximating SVP

A pretty cool approach of Khot

Let’s take for granted that γ-CVP$_p$ is NP-hard, and reduce it to γ'-SVP$_p$.

Want: Transform a γ-CVP$_p$ instance (B, d, t) into a γ'-SVP$_p$ instance (B', d').

The obvious approach:



$$B' = [B \mid t]$$

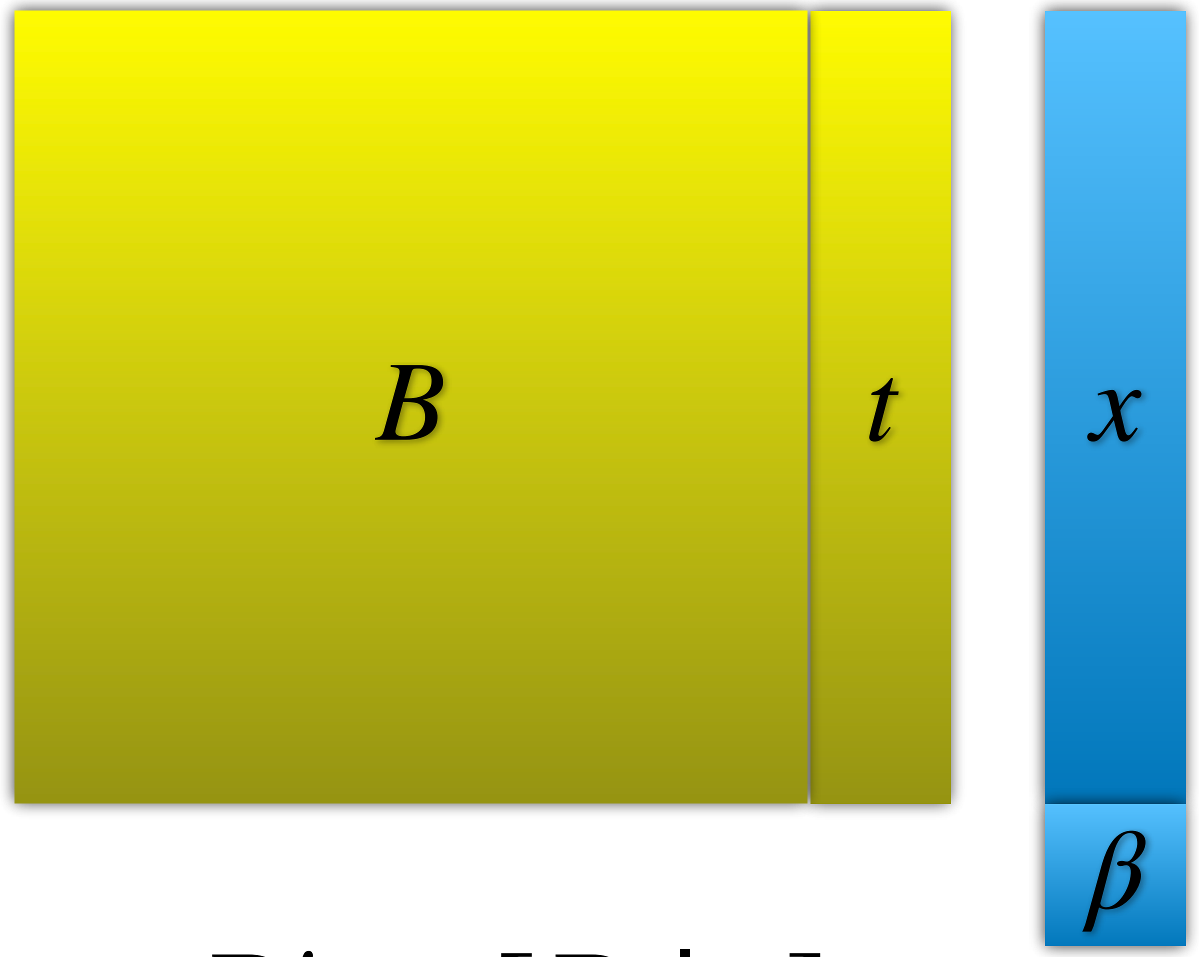
NP-hardness of approximating SVP

A pretty cool approach of Khot

Let's take for granted that γ-CVP$_p$ is NP-hard, and reduce it to γ'-SVP$_p$.

Want: Transform a γ-CVP$_p$ instance (B, d, t) into a γ'-SVP$_p$ instance (B', d').

The obvious approach:



$$B' = [B \mid t]$$

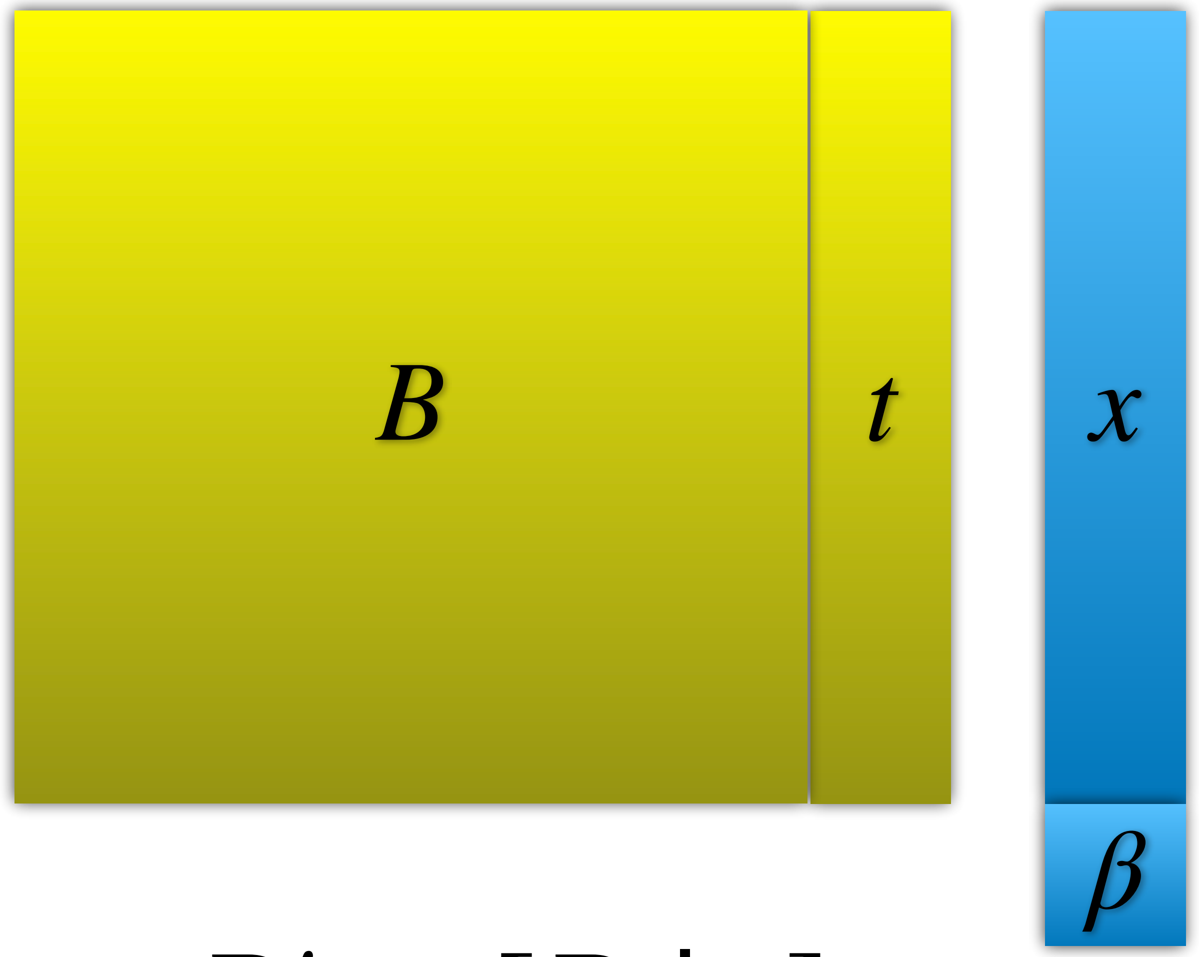
NP-hardness of approximating SVP

A pretty cool approach of Khot

Let’s take for granted that γ-CVP$_p$ is NP-hard, and reduce it to γ'-SVP$_p$.

Want: Transform a γ-CVP$_p$ instance (B, d, t) into a γ'-SVP$_p$ instance (B', d').

The obvious approach:



$$= Bx + \beta t$$

$$B' = [B \mid t]$$

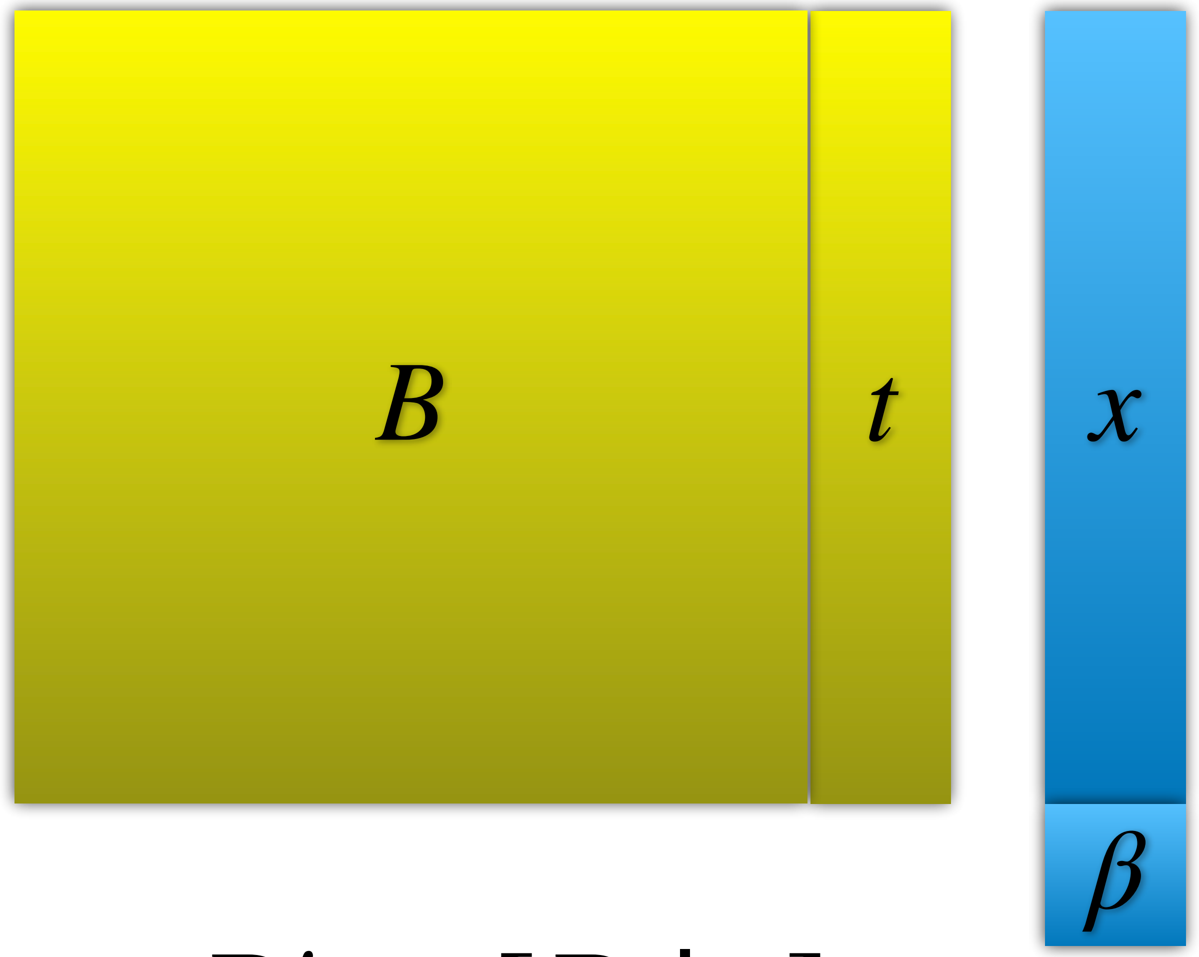
NP-hardness of approximating SVP

A pretty cool approach of Khot

Let's take for granted that γ-CVP$_p$ is NP-hard, and reduce it to γ'-SVP$_p$.

Want: Transform a γ-CVP$_p$ instance (B, d, t) into a γ'-SVP$_p$ instance (B', d').

The obvious approach:



$= Bx + \beta t$

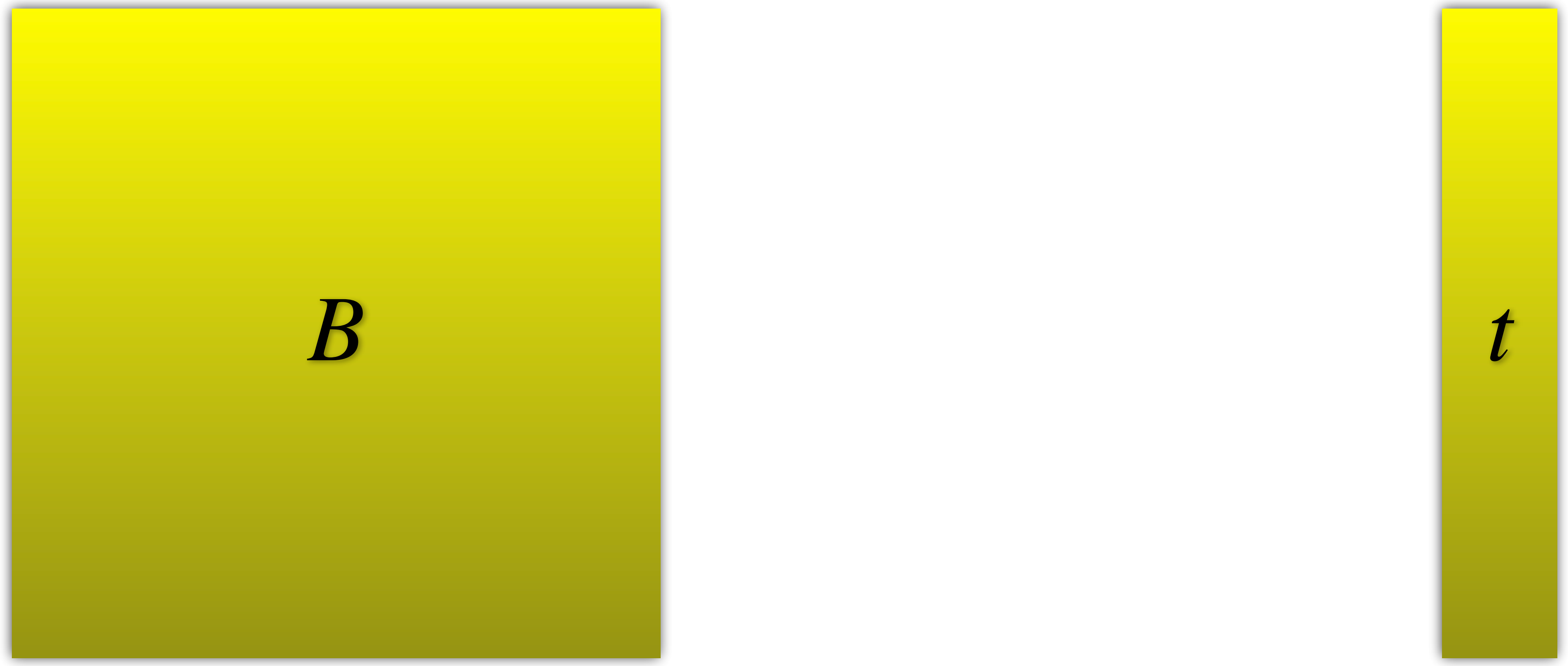
$B' = [B \mid t]$

If (B, d, t) is NO instance of γ-CVP$_p$ …

- $\beta \neq 0$: Then $||Bx + \beta t||_p$ is large
- $\beta = 0$: No guarantees…

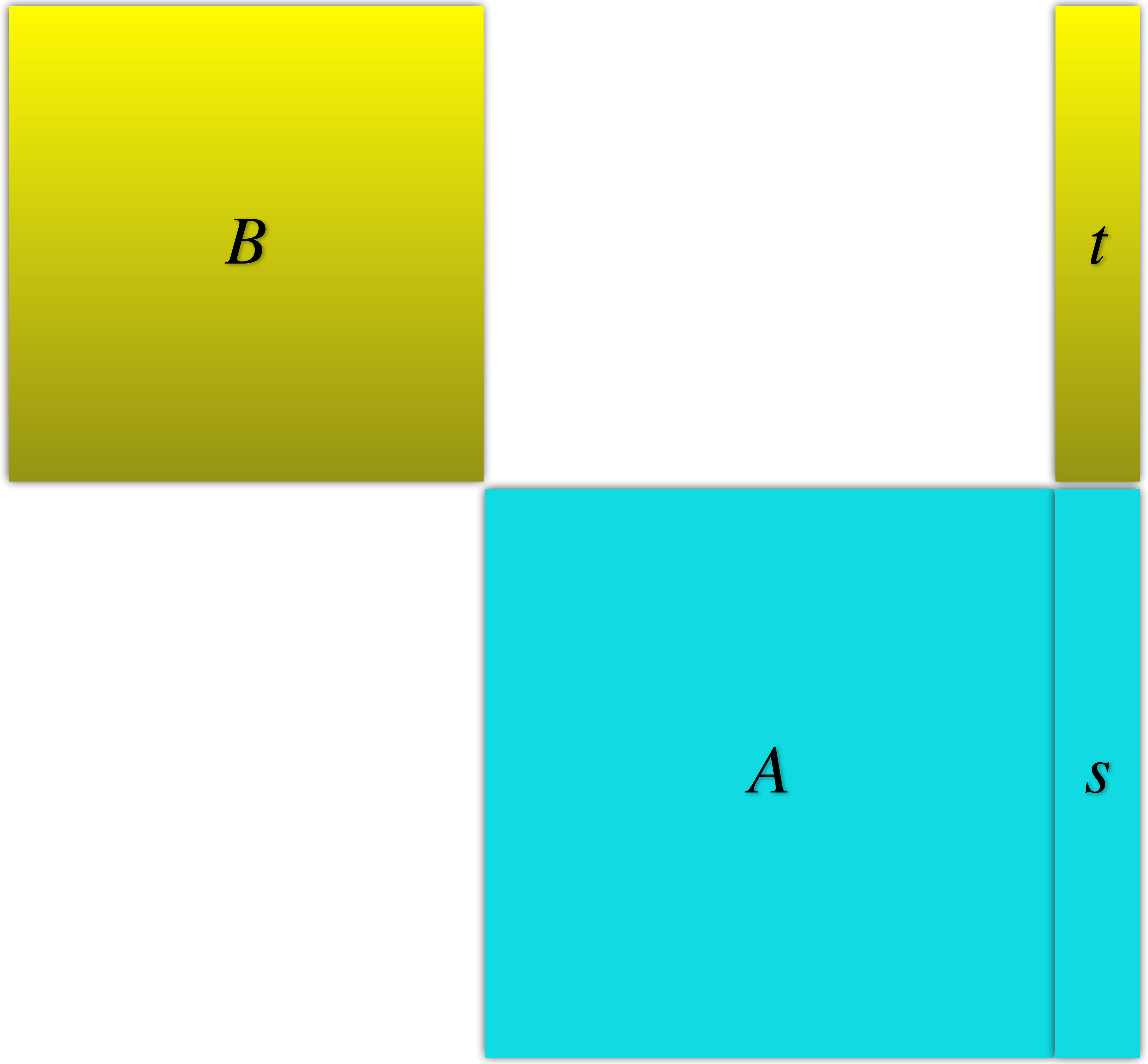
NP-hardness of approximating SVP

A pretty cool approach of Khot



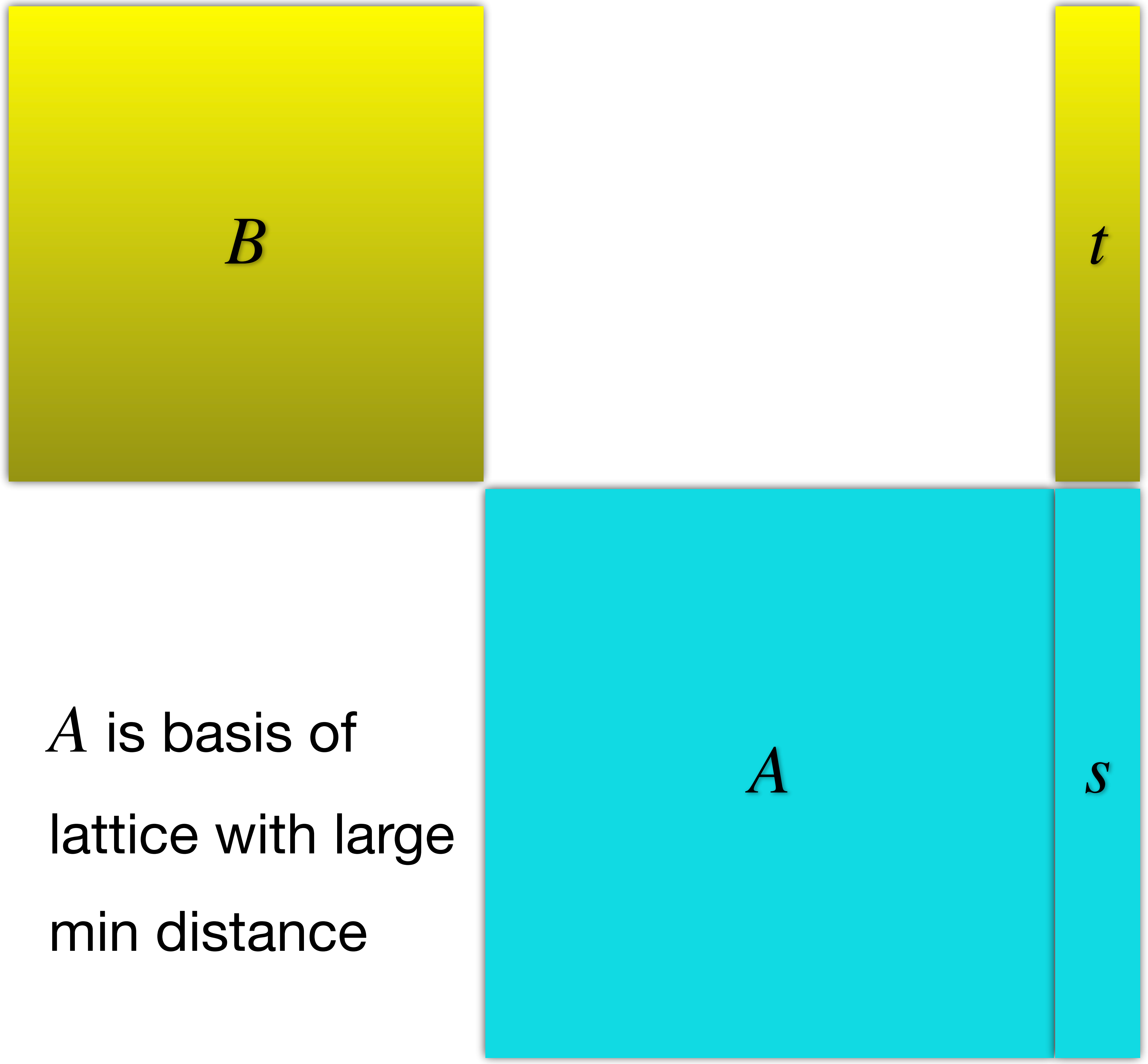
NP-hardness of approximating SVP

A pretty cool approach of Khot



NP-hardness of approximating SVP

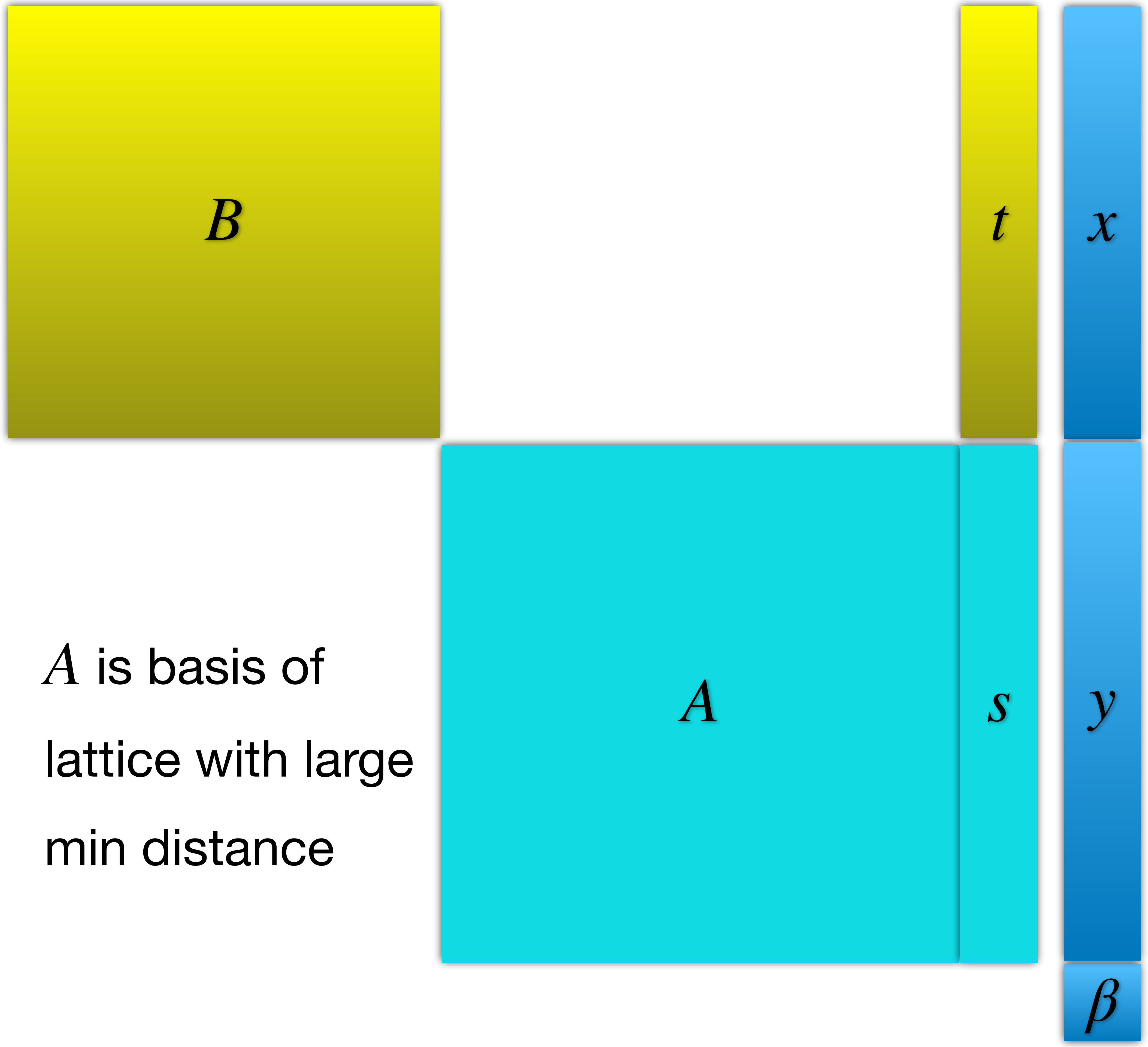
A pretty cool approach of Khot



A is basis of lattice with large min distance

NP-hardness of approximating SVP

A pretty cool approach of Khot



A is basis of lattice with large min distance

NP-hardness of approximating SVP

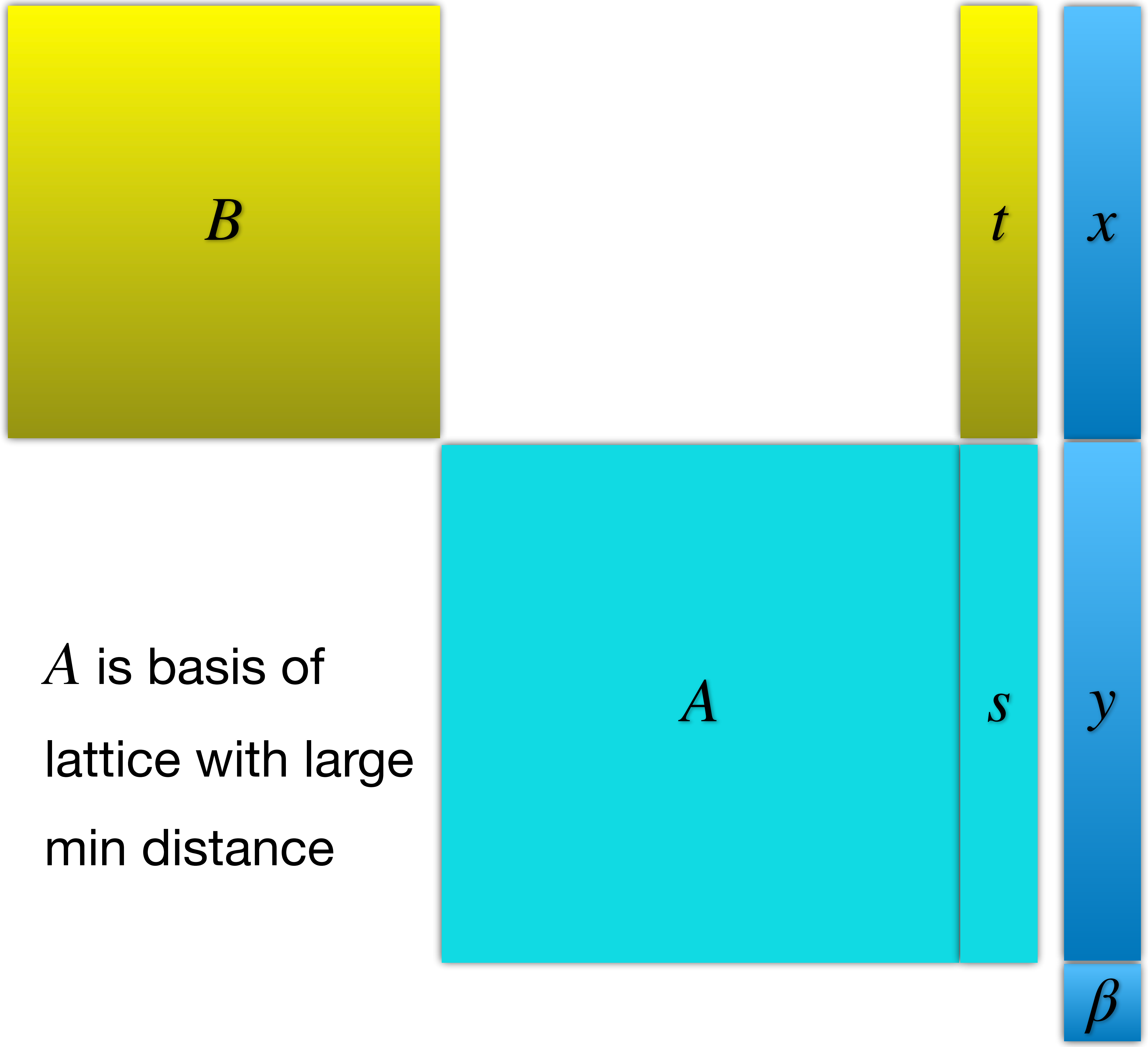
A pretty cool approach of Khot

$$\begin{bmatrix} B & 0 & t \\ 0 & A & s \end{bmatrix} \begin{bmatrix} x \\ y \\ \beta \end{bmatrix} = \begin{bmatrix} Bx + \beta t \\ Ay + \beta s \end{bmatrix}$$

A is basis of lattice with large min distance

NP-hardness of approximating SVP

A pretty cool approach of Khot



$$= \begin{bmatrix} Bx + \beta t \\ Ay + \beta s \end{bmatrix}$$

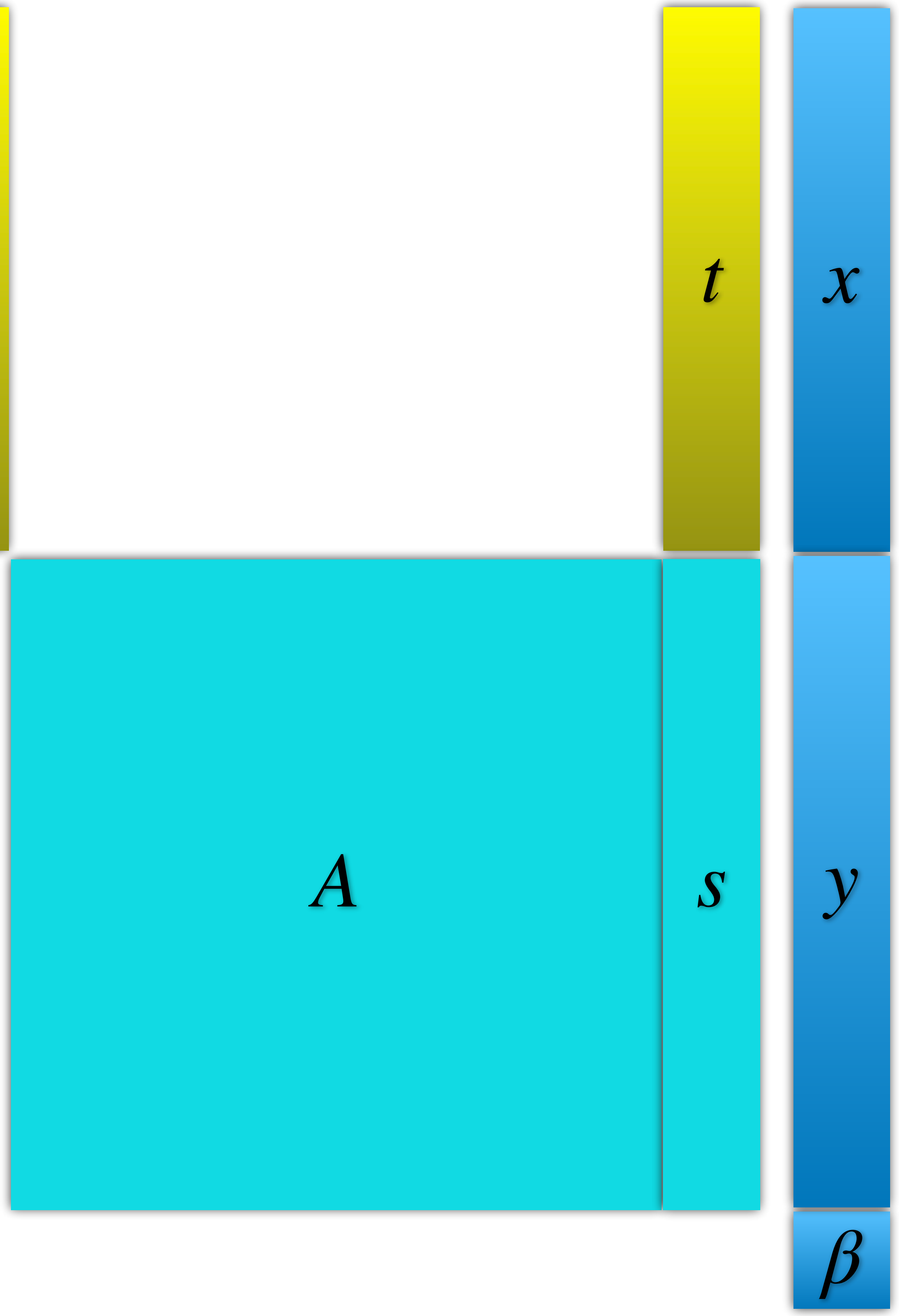
A is basis of lattice with large min distance

- If (B, d, t) is a **NO** CVP$_p$ instance, number of short vectors is **at most**

$$N_{\text{bad}} = |\{v \in \mathbb{Z}^n : \|v\|_p < \gamma d\}|$$

NP-hardness of approximating SVP

A pretty cool approach of Khot



$$= \begin{bmatrix} Bx + \beta t \\ Ay + \beta s \end{bmatrix}$$

A is basis of lattice with large min distance

- If (B, d, t) is a **NO** CVP$_p$ instance, number of short vectors is **at most**
  $$N_{\text{bad}} = |\{v \in \mathbb{Z}^n : \|v\|_p < \gamma d\}|$$
- If (B, d, t) is a **YES** CVP$_p$ instance, number of short vectors is **at least**
  $$N_{\text{good}} = |\{w \in L(A) : \|w - s\|_p \ll \gamma d\}|$$

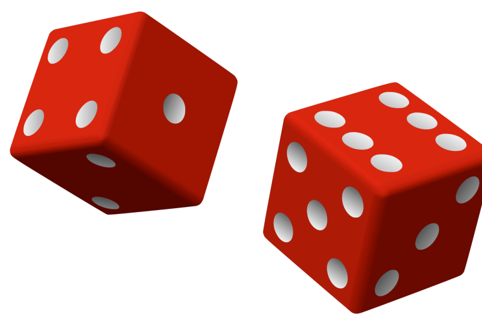
NP-hardness of approximating SVP

A pretty cool approach of Khot

$$\begin{bmatrix} B & 0 & t \\ 0 & A & s \end{bmatrix} \begin{bmatrix} x \\ y \\ \beta \end{bmatrix} = \begin{bmatrix} Bx + \beta t \\ Ay + \beta s \end{bmatrix}$$

A is basis of lattice with large min distance

- If (B, d, t) is a **NO** CVP$_p$ instance, number of short vectors is **at most**
  $$N_{\text{bad}} = |\{v \in \mathbb{Z}^n : \|v\|_p < \gamma d\}|$$
- If (B, d, t) is a **YES** CVP$_p$ instance, number of short vectors is **at least**
  $$N_{\text{good}} = |\{w \in L(A) : \|w - s\|_p \ll \gamma d\}|$$
- Provided that $N_{\text{good}} \gg N_{\text{bad}}$, obtain the desired SVP instance (B', d') by **randomly sparsifying** this intermediate lattice.



Locally dense lattices

Khot's reduction works if

$$N_{\text{good}} = |\{w \in L(A) : \|w - s\|_p \ll \gamma d\}| \quad \gg \quad N_{\text{bad}} = |\{v \in \mathbb{Z}^n : \|v\|_p < \gamma d\}|$$

Locally dense lattices

Khot's reduction works if

$$N_{\text{good}} = |\{w \in L(A) : \|w - s\|_p \ll \gamma d\}| \quad \gg \quad N_{\text{bad}} = |\{v \in \mathbb{Z}^n : \|v\|_p < \gamma d\}|$$

Locally dense lattice $L(A)$***:***

- $L(A)$ has minimum distance $\gg \gamma d$;
- With high probability over random sampling of s there are many vectors in $L(A)$ close to s.

Locally dense lattices

Khot's reduction works if

$$N_{\text{good}} = |\{w \in L(A) : \|w - s\|_p \ll \gamma d\}| \quad \gg \quad N_{\text{bad}} = |\{v \in \mathbb{Z}^n : \|v\|_p < \gamma d\}|$$

Locally dense lattice $L(A)$:

- $L(A)$ has minimum distance $\gg \gamma d$;
- With high probability over random sampling of s there are many vectors in $L(A)$ close to s.

A construction based on linear codes:

G generator matrix of binary BCH code with minimum distance $> \gamma d$:

$$L(A) = C(G) + 2\mathbb{Z}^m$$

with $m = \text{poly}(n)$.

Pros and cons of Khot's reduction

Pro: Khot's reduction from CVP to SVP is an FPT-reduction!

Pros and cons of Khot's reduction

Pro: Khot's reduction from CVP to SVP is an FPT-reduction!

Con 1: It doesn't work for $p = 1$...

Metric embeddings take care of **Con1** in the non-FPT setting, but they don't work in the FPT setting.

Pros and cons of Khot's reduction

Pro: Khot's reduction from CVP to SVP is an FPT-reduction!

Con 1: It doesn't work for $p = 1$...

Metric embeddings take care of **Con1** in the non-FPT setting, but they don't work in the FPT setting.

Con2: Not easy to amplify approximation factor (we'll see more about this)

Insight 1: Khot for codes

Khot's approach also reduces γ-NCP$_q$ to γ'-MDP$_q$ for some $\gamma(q), \gamma'(q) > 1$!

Insight 1: Khot for codes

Khot's approach also reduces γ-NCP$_q$ to γ'-MDP$_q$ for some $\gamma(q), \gamma'(q) > 1$!

"locally dense lattice"	$\longrightarrow$	"locally dense code"
"lattice sparsification"	$\longrightarrow$	"intersection with random linear code"

Insight 1: Khot for codes

Khot's approach also reduces γ-NCP$_q$ to γ'-MDP$_q$ for some $\gamma(q), \gamma'(q) > 1$!

"locally dense lattice"	$\longrightarrow$	"locally dense code"
"lattice sparsification"	$\longrightarrow$	"intersection with random linear code"

Conclusion: γ-MDP$_q$ is $W[1]$-hard for **all** q and some $\gamma'(q) > 1$

Insight 1: Khot for codes

Khot's approach also reduces γ-NCP$_q$ to γ'-MDP$_q$ for some $\gamma(q), \gamma'(q) > 1$!

"locally dense lattice"	$\longrightarrow$	"locally dense code"
"lattice sparsification"	$\longrightarrow$	"intersection with random linear code"

Conclusion: γ-MDP$_q$ is $W[1]$-hard for **all** q and some $\gamma'(q) > 1$

How can we get $W[1]$-hardness for **all** $\gamma' > 1$?

Amplifying the approximation factor in coding problems

γ-MDP$_q$ is $W[1]$-hard for **some** approximation factor $\gamma' > 1$ $\implies$ γ-MDP$_q$ also $W[1]$-hard for $\gamma' > \gamma$

Amplifying the approximation factor in coding problems

γ-MDP$_q$ is $W[1]$-hard for **some** approximation factor $\gamma' > 1$

$\implies$

γ-MDP$_q$ also $W[1]$-hard for $\gamma' > \gamma$

How? Tensor product of codes!

Amplifying the approximation factor in coding problems

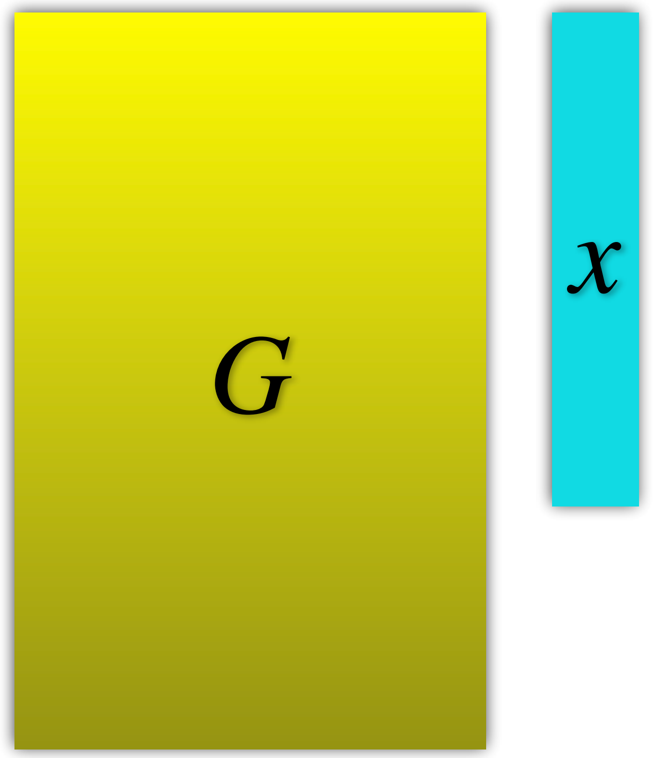
γ-MDP$_q$ is $W[1]$-hard for **some** approximation factor $\gamma' > 1$ $\Longrightarrow$ γ-MDP$_q$ also $W[1]$-hard for $\gamma' > \gamma$

How? Tensor product of codes!

Amplifying the approximation factor in coding problems

γ-MDP$_q$ is $W[1]$-hard for **some** approximation factor $\gamma' > 1$ $\Longrightarrow$ γ-MDP$_q$ also $W[1]$-hard for $\gamma' > \gamma$

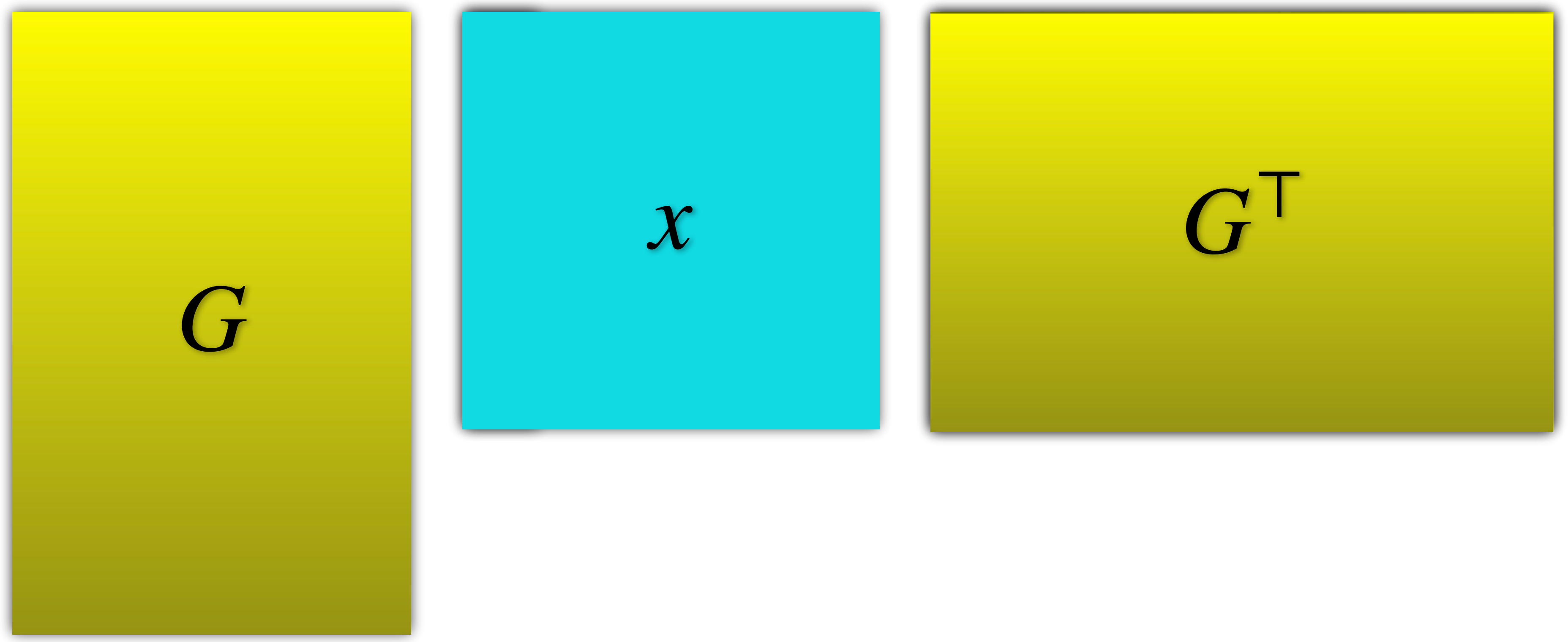
How? Tensor product of codes!



Amplifying the approximation factor in coding problems

γ-MDP$_q$ is $W[1]$-hard for **some** approximation factor $\gamma' > 1$ $\implies$ γ-MDP$_q$ also $W[1]$-hard for $\gamma' > \gamma$

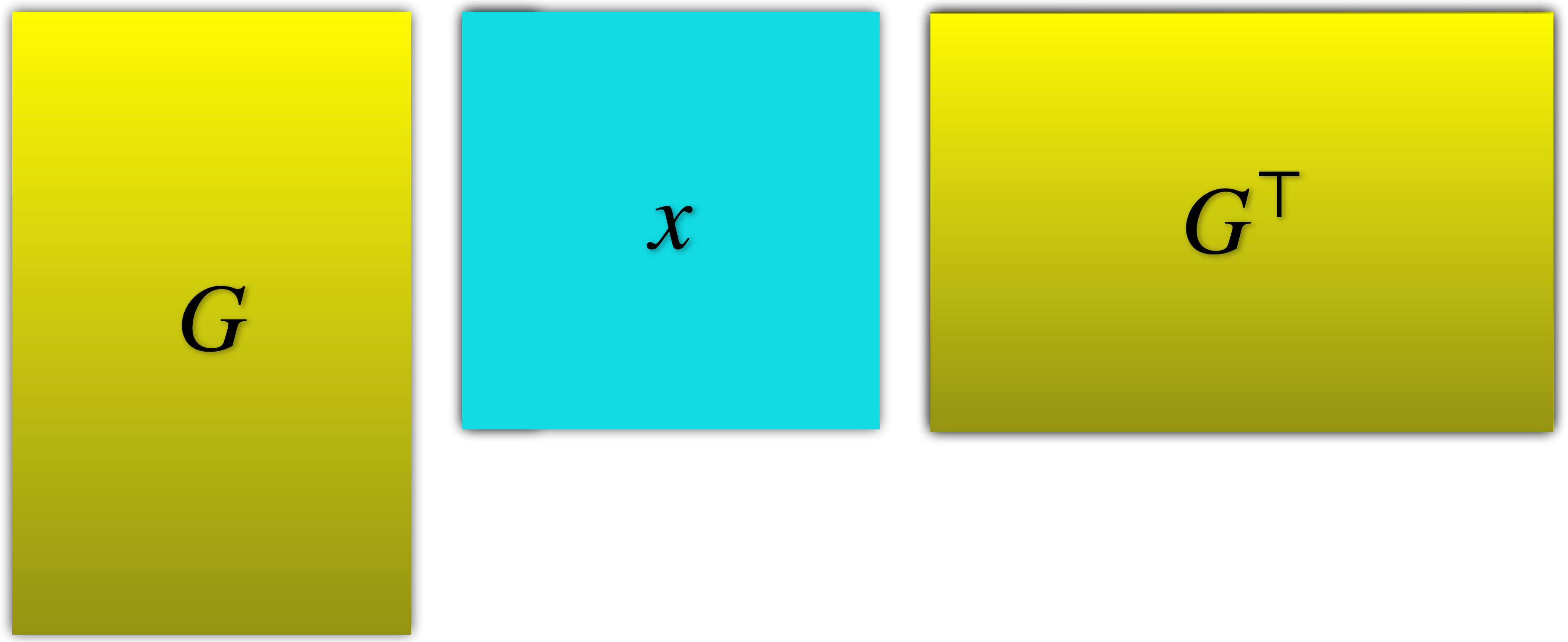
How? Tensor product of codes!



Amplifying the approximation factor in coding problems

γ-MDP$_q$ is $W[1]$-hard for **some** approximation factor $\gamma' > 1$ $\implies$ γ-MDP$_q$ also $W[1]$-hard for $\gamma' > \gamma$

How? Tensor product of codes!



Use the fact that $d(C \otimes C) = d(C)^2$:

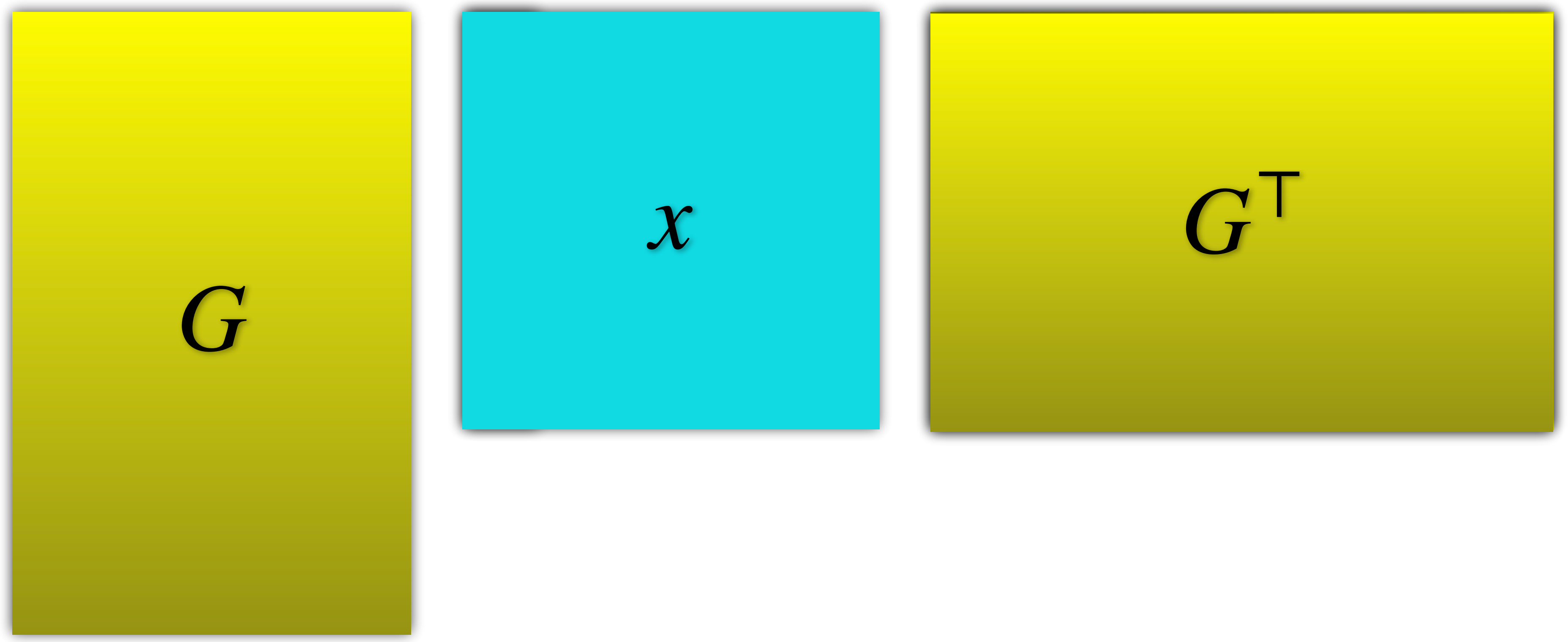
(G, d) is γ-MDP$_q$ YES/NO instance iff

$(G \otimes G, d^2)$ is YES/NO $(\gamma' = \gamma^2)$-MDP$_q$ instance

Amplifying the approximation factor in coding problems

γ-MDP$_q$ is $W[1]$-hard for **some** approximation factor $\gamma' > 1$ $\implies$ γ-MDP$_q$ also $W[1]$-hard for $\gamma' > \gamma$

How? Tensor product of codes!



Use the fact that $d(C \otimes C) = d(C)^2$:

(G, d) is γ-MDP$_q$ YES/NO instance iff

$(G \otimes G, d^2)$ is YES/NO $(\gamma' = \gamma^2)$-MDP$_q$ instance

Conclusion: γ-MDP$_q$ is $W[1]$-hard for **all q and all $\gamma > 1$**

Insight 2: If we don't care about approx factor…

Khot's approach uses binary BCH codes because they're useful for amplifying the approximation factor in the non-FPT regime.

But they don't work when $p = 1$…

Insight 2: If we don't care about approx factor…

Khot's approach uses binary BCH codes because they're useful for amplifying the approximation factor in the non-FPT regime.

But they don't work when $p = 1$…

"BCH-based locally dense lattice" $\longrightarrow$ "Reed-Solomon-based locally dense lattice"

(Bennett-Peikert '22)

Insight 2: If we don't care about approx factor…

Khot's approach uses binary BCH codes because they're useful for amplifying the approximation factor in the non-FPT regime.

But they don't work when $p = 1$…

"BCH-based locally dense lattice" $\longrightarrow$ "Reed-Solomon-based locally dense lattice"

(Bennett-Peikert '22)

Conclusion: γ-SVP$_1$ is $W[1]$**-hard** for any approximation factor $\gamma < 2$.

Amplifying the approximation factor for lattices

This was easy for codes, since $d(C \otimes C) = d(C)^2$.

But for lattices it may happen that $d(L \otimes L) \ll d(L)^2$ …

Amplifying the approximation factor for lattices

This was easy for codes, since $d(C \otimes C) = d(C)^2$.

But for lattices it may happen that $d(L \otimes L) \ll d(L)^2$ …

Amplifying the approximation factor for lattices

This was easy for codes, since $d(C \otimes C) = d(C)^2$.

But for lattices it may happen that $d(L \otimes L) \ll d(L)^2$ …

“Haviv-Regev showed that Khot’s SVP instances tensor nicely, so it looks like we’re done…”

Amplifying the approximation factor for lattices

This was easy for codes, since $d(C \otimes C) = d(C)^2$.

But for lattices it may happen that $d(L \otimes L) \ll d(L)^2$ …

"Haviv-Regev showed that Khot's SVP instances tensor nicely, so it looks like we're done…"

"Wait… Khot's SVP instances come from special CVP instances based on Exact Set Cover."

Amplifying the approximation factor for lattices

This was easy for codes, since $d(C \otimes C) = d(C)^2$.

But for lattices it may happen that $d(L \otimes L) \ll d(L)^2$ …

“Haviv-Regev showed that Khot’s SVP instances tensor nicely, so it looks like we’re done…”

“Wait… Khot’s SVP instances come from special CVP instances based on Exact Set Cover.”

“Oh! And we don’t know whether approximating Exact Set Cover is $W[1]$-hard…”

Amplifying the approximation factor for lattices

This was easy for codes, since $d(C \otimes C) = d(C)^2$.

But for lattices it may happen that $d(L \otimes L) \ll d(L)^2$ …

"Haviv-Regev showed that Khot's SVP instances tensor nicely, so it looks like we're done…"

"Wait… Khot's SVP instances come from special CVP instances based on Exact Set Cover."

"Oh! And we don't know whether approximating Exact Set Cover is $W[1]$-hard…"

"Hmm… Cool problem!"

Insight 3: Turning codes into lattices

Need a different approach to amplify the approximation factor of SVP!

Insight 3: Turning codes into lattices

Need a different approach to amplify the approximation factor of SVP!

Idea: Apply Khot's approach to reduce from NCP_2 to SVP_p !

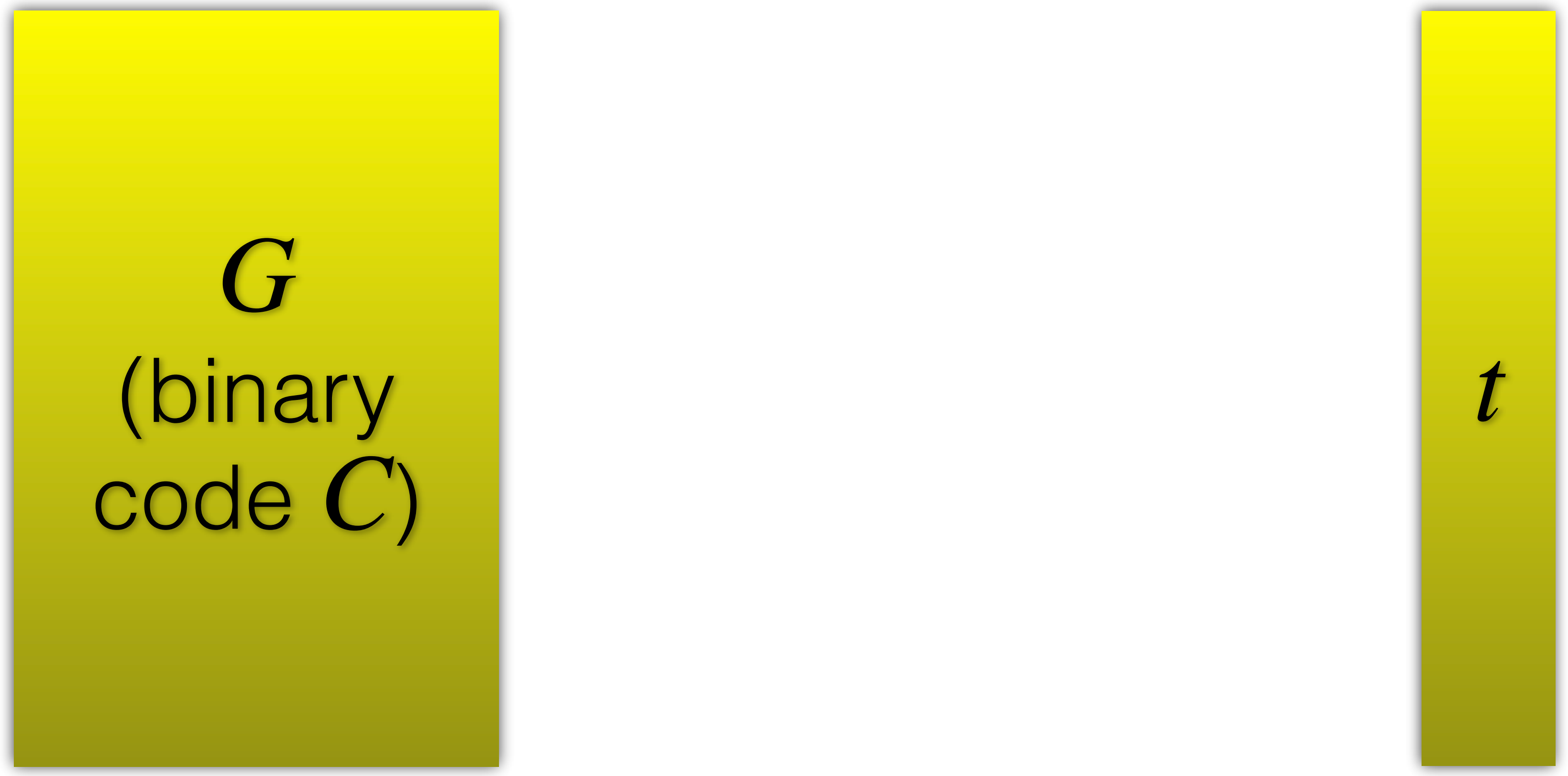
Can ensure Haviv-Regev tensoring conditions are satisfied in this case.

Insight 3: Turning codes into lattices

Need a different approach to amplify the approximation factor of SVP!

Idea: Apply Khot's approach to reduce from NCP_2 to SVP_p !

Can ensure Haviv-Regev tensoring conditions are satisfied in this case.



Insight 3: Turning codes into lattices

Need a different approach to amplify the approximation factor of SVP!

Idea: Apply Khot's approach to reduce from NCP_2 to SVP_p !

Can ensure Haviv-Regev tensoring conditions are satisfied in this case.

$L = 2(C(G) + 2\mathbb{Z}^n)$
(construction A)

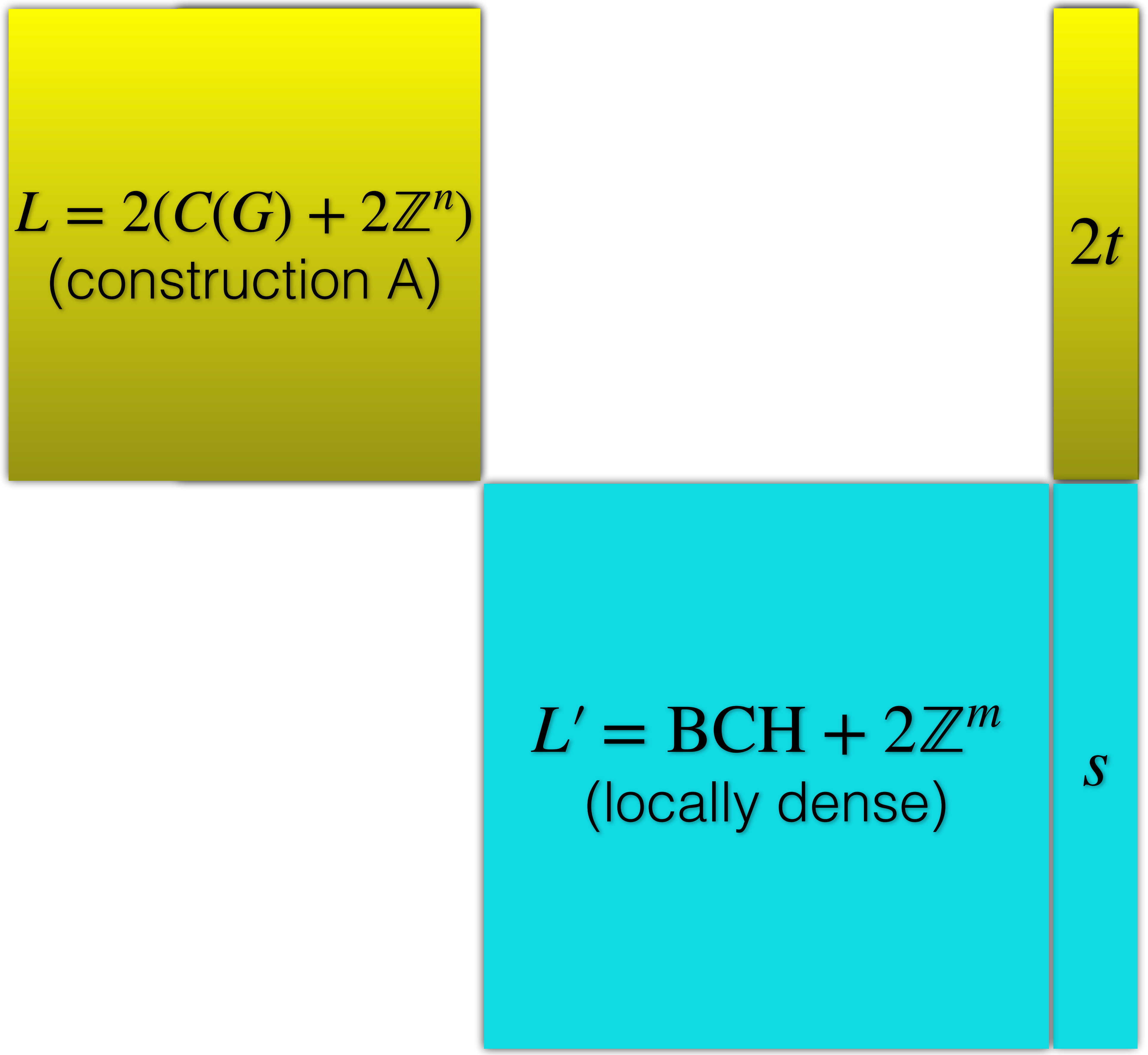
$2t$

Insight 3: Turning codes into lattices

Need a different approach to amplify the approximation factor of SVP!

Idea: Apply Khot's approach to reduce from NCP_2 to SVP_p !

Can ensure Haviv-Regev tensoring conditions are satisfied in this case.



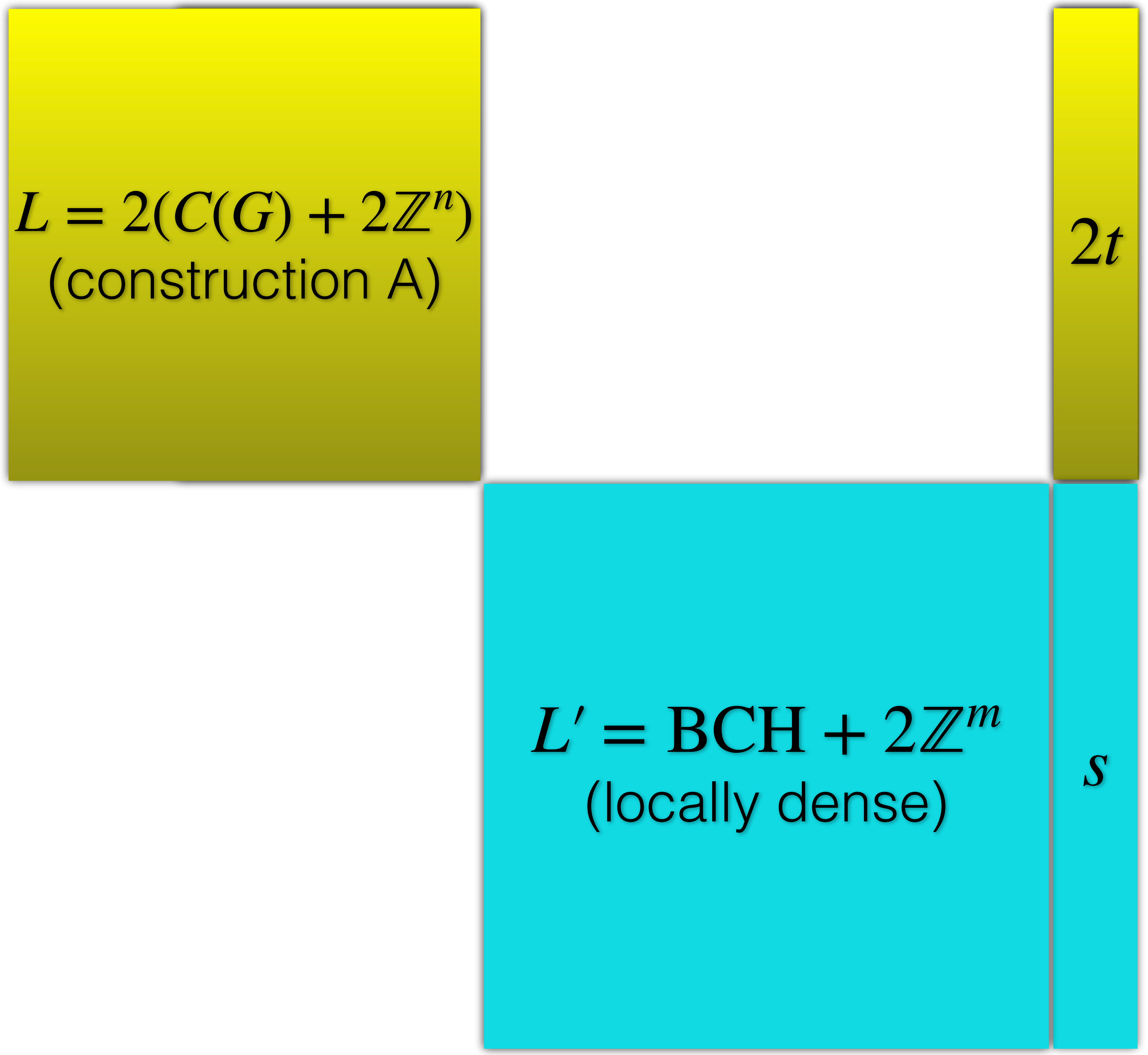
Insight 3: Turning codes into lattices

Need a different approach to amplify the approximation factor of SVP!

Idea: Apply Khot's approach to reduce from NCP_2 to SVP_p !

Can ensure Haviv-Regev tensoring conditions are satisfied in this case.

Conclusion: γ-SVP_p is $W[1]$**-hard** for all $p > 1$ and **all approx factor** $\gamma > 1$.



Wrapping up

- Deterministic FPT reductions?
  Unknown for SVP even in the unparameterized setting!

- $W[1]$-hardness of γ-SVP$_1$ for all approximation factors $\gamma > 1$?
  The last missing puzzle piece!

Wrapping up

- Deterministic FPT reductions?
  Unknown for SVP even in the unparameterized setting!

- $W[1]$-hardness of γ-SVP$_1$ for all approximation factors $\gamma > 1$?
  The last missing puzzle piece!

Thanks!