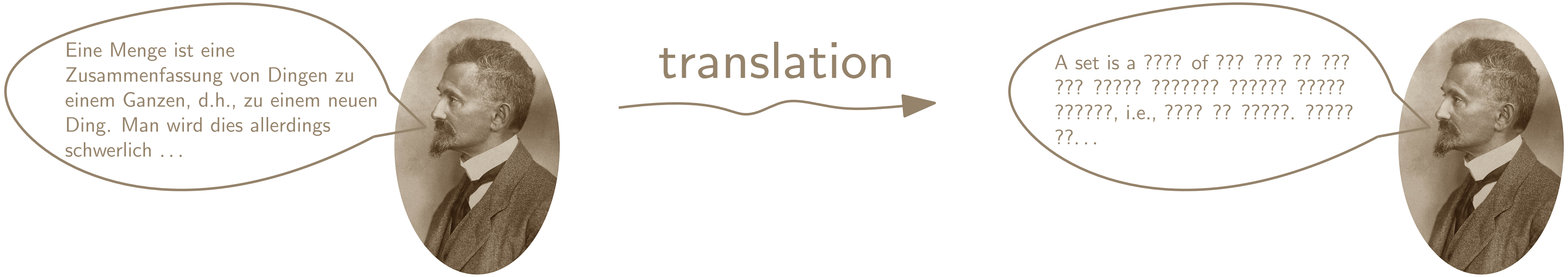
max planck institut
informatik

Translating Hausdorff Is Hard: Fine-Grained Lower Bounds for Hausdorff Distance Under Translation

Karl Bringmann and **André Nusser**



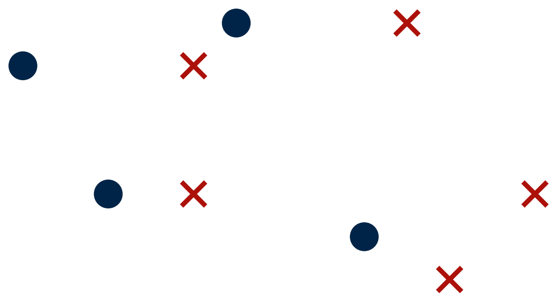
Preamble

Hausdorff Distance

Given: Point sets A (●) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance

$$\delta_{\vec{H}}(A, B) := \max_{a \in A} \min_{b \in B} \|a - b\|_p$$

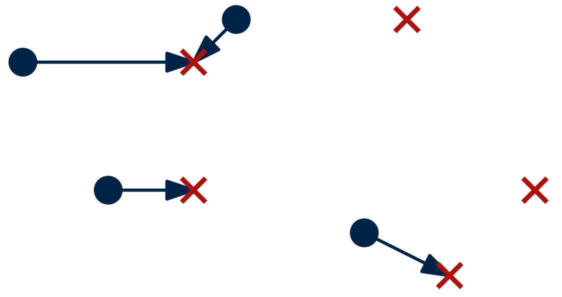


Hausdorff Distance

Given: Point sets A (●) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance

$$\delta_{\vec{H}}(A, B) := \max_{a \in A} \min_{b \in B} \|a - b\|_p$$

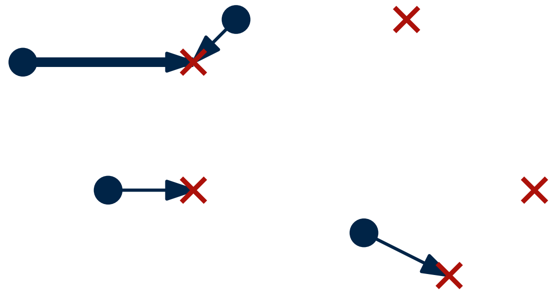


Hausdorff Distance

Given: Point sets A (●) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance

$$\delta_{\vec{H}}(A, B) := \max_{a \in A} \min_{b \in B} \|a - b\|_p$$



Hausdorff Distance

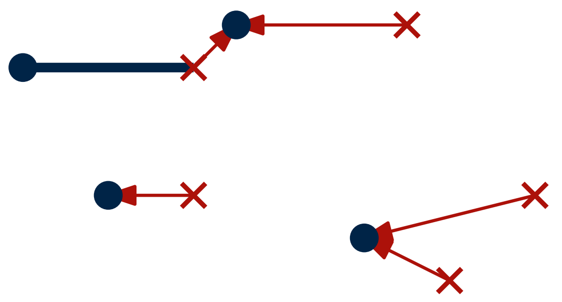
Given: Point sets A (●) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance

$$\delta_{\vec{H}}(A, B) := \max_{a \in A} \min_{b \in B} \|a - b\|_p$$

(Undirected) Hausdorff Distance

$$\delta_H(A, B) := \max\{\delta_{\vec{H}}(A, B), \delta_{\vec{H}}(B, A)\}$$



Hausdorff Distance

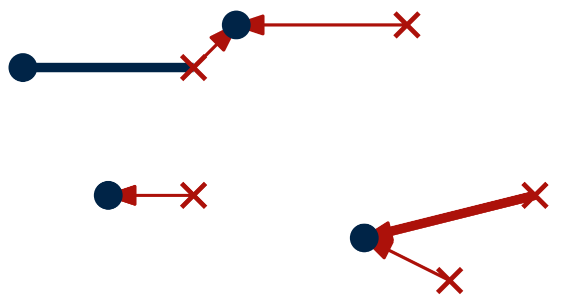
Given: Point sets A (●) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance

$$\delta_{\vec{H}}(A,B) := \max_{a\in A}\min_{b\in B}\|a-b\|_p$$

(Undirected) Hausdorff Distance

$$\delta_H(A,B) := \max\{\delta_{\vec{H}}(A,B), \delta_{\vec{H}}(B,A)\}$$



Hausdorff Distance

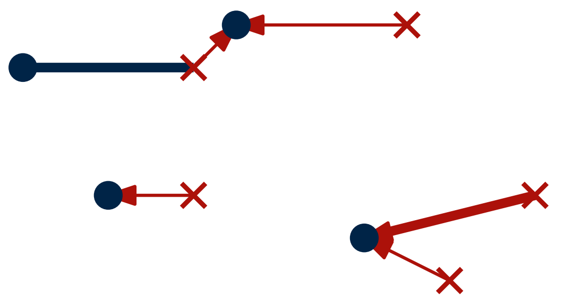
Given: Point sets A (●) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance

$$\delta_{\vec{H}}(A, B) := \max_{a \in A} \min_{b \in B} \|a - b\|_p$$

(Undirected) Hausdorff Distance

$$\delta_H(A, B) := \max\{\delta_{\vec{H}}(A, B), \delta_{\vec{H}}(B, A)\}$$



→ Computable in time $\tilde{\mathcal{O}}(n + m)$

Hausdorff Distance

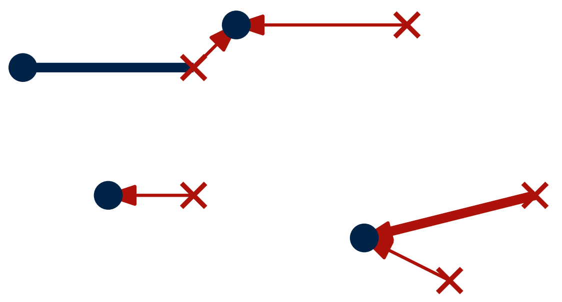
Given: Point sets A (•) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance

$$\delta_{\vec{H}}(A,B) := \max_{a \in A} \min_{b \in B} \|a-b\|_p$$

(Undirected) Hausdorff Distance

$$\delta_H(A,B) := \max\{\delta_{\vec{H}}(A,B), \delta_{\vec{H}}(B,A)\}$$



→ Computable in time $\tilde{O}(n+m)$

Hausdorff Distance Under Translation

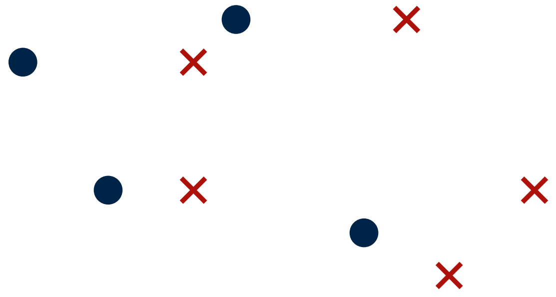
Given: Point sets A (●) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance U. Translation

$$\delta^T_{\vec{H}}(A, B) := \min_{\tau \in \mathbb{R}^2} \delta_{\vec{H}}(A, B + \tau)$$

(Undirected) Hausdorff Distance U. Translation

$$\delta^T_{H}(A, B) := \min_{\tau \in \mathbb{R}^2} \delta_{H}(A, B + \tau)$$



Hausdorff Distance Under Translation

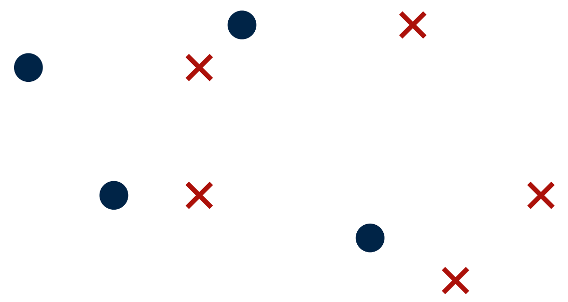
Given: Point sets A (●) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance U. Translation

$$\delta^T_{\vec{H}}(A, B) := \min_{\tau \in \mathbb{R}^2} \delta_{\vec{H}}(A, B + \tau)$$

(Undirected) Hausdorff Distance U. Translation

$$\delta^T_H(A, B) := \min_{\tau \in \mathbb{R}^2} \delta_H(A, B + \tau)$$



Hausdorff Distance Under Translation

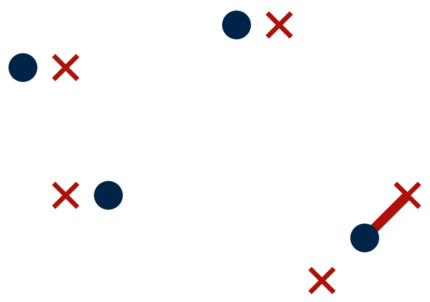
Given: Point sets A (●) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance U. Translation

$$\delta_{\vec{H}}^{T}(A,B) := \min_{\tau \in \mathbb{R}^2} \delta_{\vec{H}}(A, B+\tau)$$

(Undirected) Hausdorff Distance U. Translation

$$\delta_{H}^{T}(A,B) := \min_{\tau \in \mathbb{R}^2} \delta_{H}(A, B+\tau)$$



Hausdorff Distance Under Translation

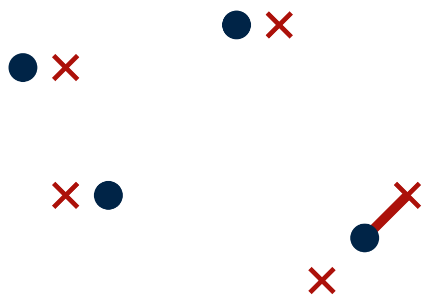
Given: Point sets A (●) and B (×) in $\mathbb{R}^2$

Directed Hausdorff Distance U. Translation

$$\delta^T_{\vec{H}}(A, B) := \min_{\tau \in \mathbb{R}^2} \delta_{\vec{H}}(A, B + \tau)$$

(Undirected) Hausdorff Distance U. Translation

$$\delta^T_{H}(A, B) := \min_{\tau \in \mathbb{R}^2} \delta_{H}(A, B + \tau)$$



→ **What's the "correct" running time?**

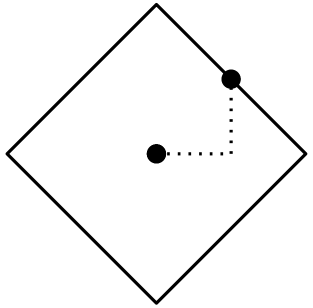
Motivation

Most basic translation invariant point set similarity measure

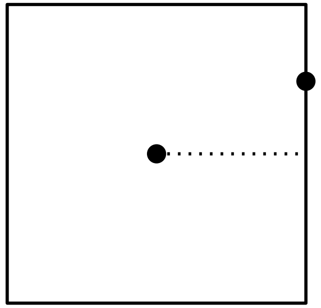
Results

Results

L_1 :

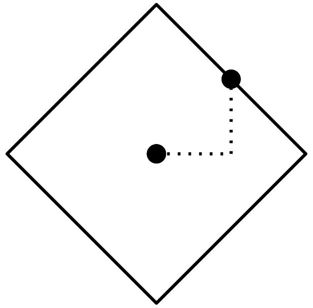


L_∞ :

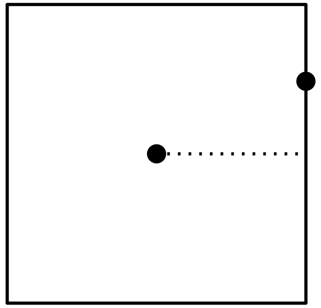


Results

L_1 :



L_∞ :

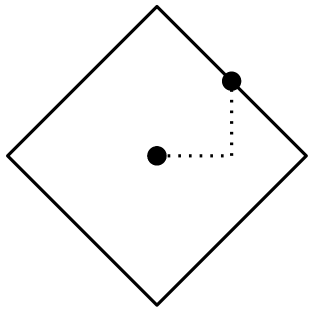


Theorem [Chew, Kedem '92]

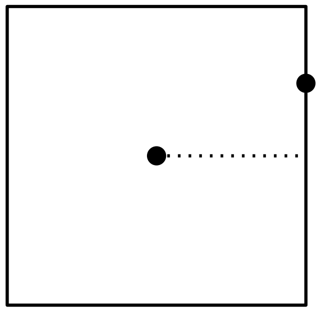
Hausdorff Distance Under Translation in L_1/L_∞ can be computed in time $\tilde{O}(nm)$.

Results

L_1 :



L_∞ :



Theorem [Chew, Kedem '92]

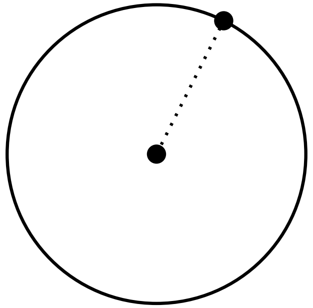
Hausdorff Distance Under Translation in L_1/L_∞ can be computed in time $\tilde{O}(nm)$.

Theorem

Hausdorff Distance Under Translation in L_1/L_∞ cannot be computed in time $(nm)^{1-\epsilon}$, unless SETH fails.

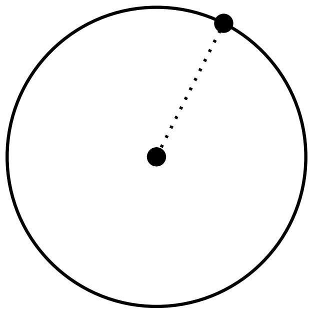
Results

$L_2:$



Results

L_2 :

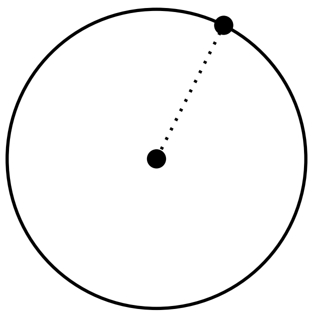


Theorem [Huttenlocher, Kedem, Sharir ’93]

Hausdorff Distance Under Translation in L_2 can be computed in $\tilde{\mathcal{O}}(nm(n+m))$.

Results

L_2 :



Theorem [Huttenlocher, Kedem, Sharir '93]

Hausdorff Distance Under Translation in L_2 can be computed in $\tilde{\mathcal{O}}(nm(n+m))$.

Theorem

Hausdorff Distance Under Translation in L_2 with $m \in \mathcal{O}(1)$ cannot be computed in time $n^{2-\epsilon}$, unless the 3SUM Hypothesis fails.

First Reduction

Theorem

Hausdorff Distance Under Translation in L_1/L_∞ cannot be computed in time $(nm)^{1-\epsilon}$, unless SETH fails.

Orthogonal Vectors

Orthogonal Vectors (OV)

Given sets $X, Y \subset \{0,1\}^d$, do there exist $x \in X$ and $y \in Y$ such that $\langle x, y \rangle = 0$?

Orthogonal Vectors

Orthogonal Vectors (OV)

Given sets $X, Y \subset \{0,1\}^d$, do there exist $x \in X$ and $y \in Y$ such that $\langle x, y \rangle = 0$?

Orthogonal Vectors Hypothesis (OVH)

Orthogonal Vectors cannot be solved in time $\mathcal{O}((|X| \cdot |Y|)^{1-\epsilon} \text{poly}(d))$.

Orthogonal Vectors

Orthogonal Vectors (OV)

Given sets $X, Y \subset \{0,1\}^d$, do there exist $x \in X$ and $y \in Y$ such that $\langle x, y \rangle = 0$?

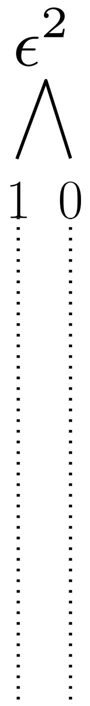
Orthogonal Vectors Hypothesis (OVH)

Orthogonal Vectors cannot be solved in time $\mathcal{O}((|X| \cdot |Y|)^{1-\epsilon} \text{poly}(d))$.

$\rightarrow$ SETH implies OVH [Williams '05]

OV $\leq \delta_H^T$ in L_1

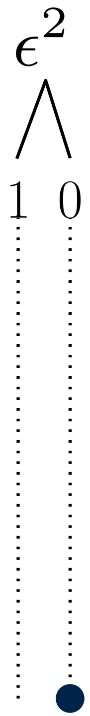
Vector Gadgets



$$x = (0, 1, 1, 0, 1, 0, 0)$$

OV $\leq \delta_H^T$ in L_1

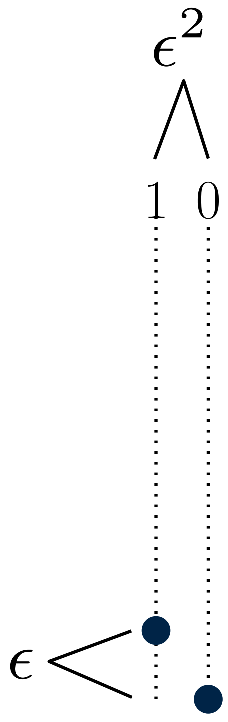
Vector Gadgets



$$x = (0, 1, 1, 0, 1, 0, 0)$$

OV $\leq \delta_H^T$ in L_1

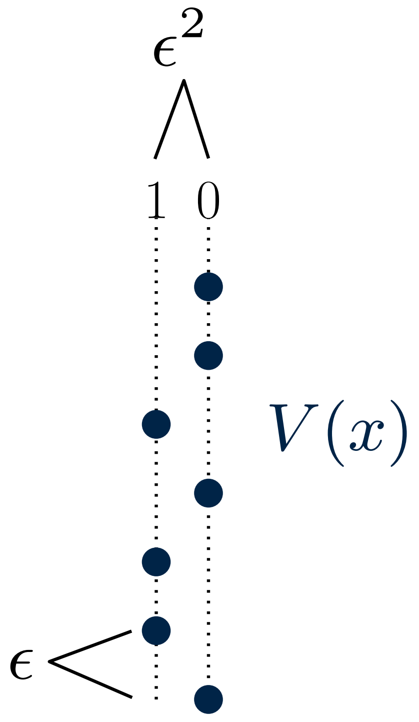
Vector Gadgets



$$x = (0, 1, 1, 0, 1, 0, 0)$$

$\mathsf{OV} \leq \delta_H^T$ in L_1

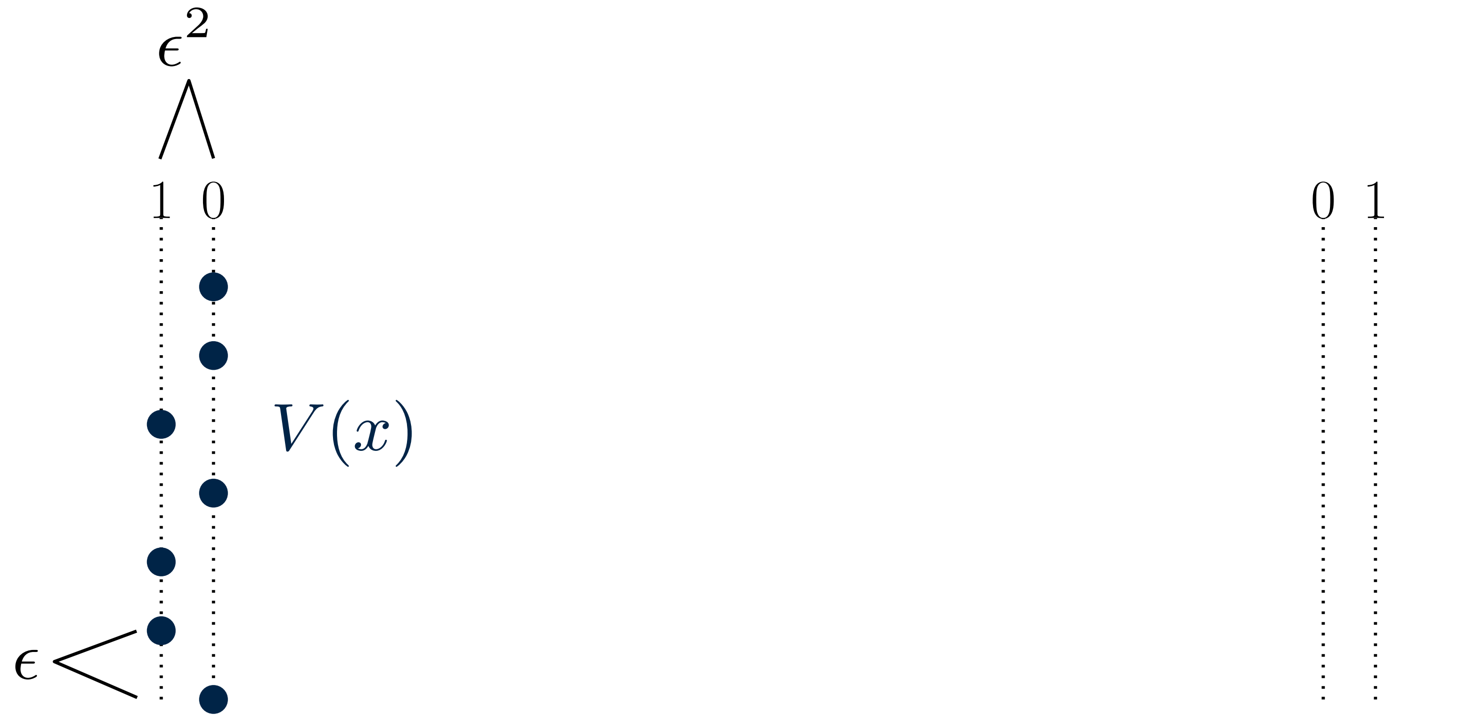
Vector Gadgets



$$x = (0, 1, 1, 0, 1, 0, 0)$$

OV $\leq \delta_H^T$ in L_1

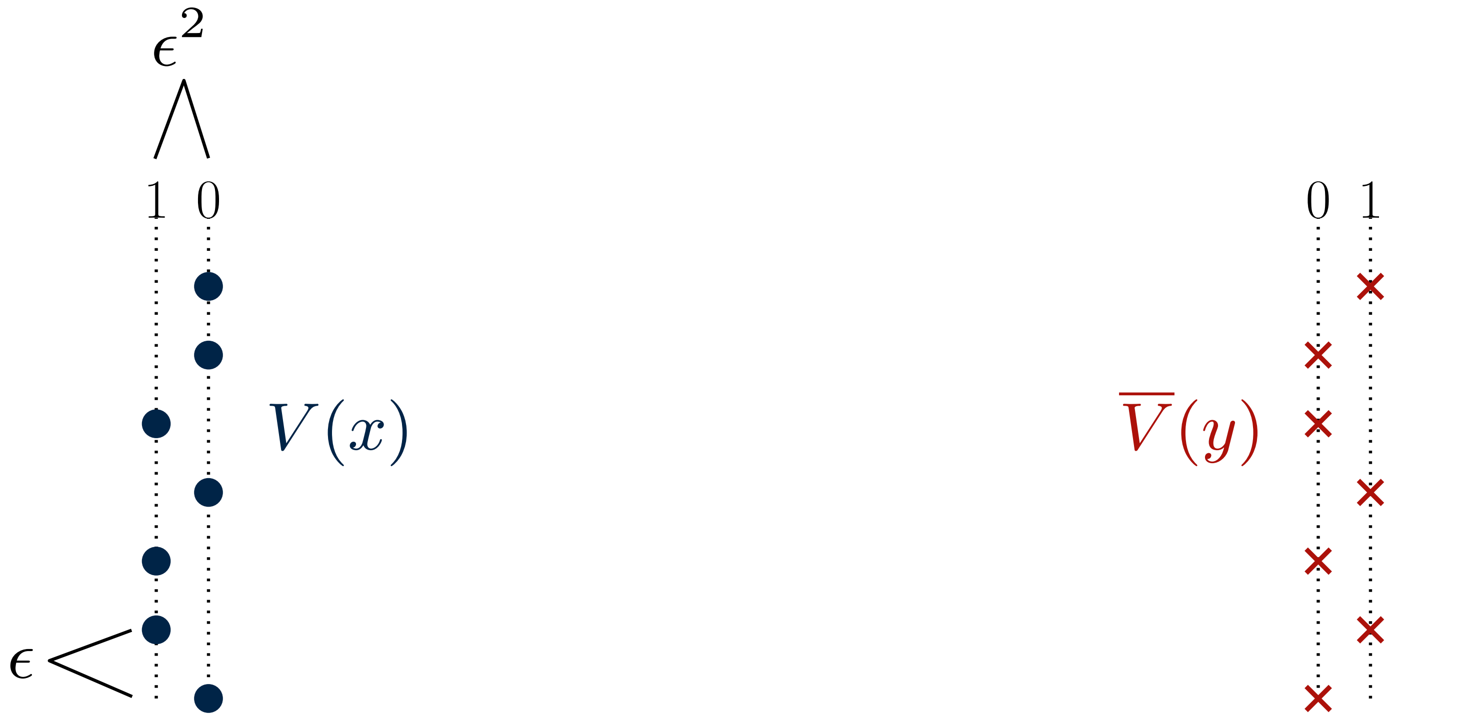
Vector Gadgets



$x = (0, 1, 1, 0, 1, 0, 0)$ $y = (0, 1, 0, 1, 0, 0, 1)$

OV $\leq \delta_H^T$ in L_1

Vector Gadgets

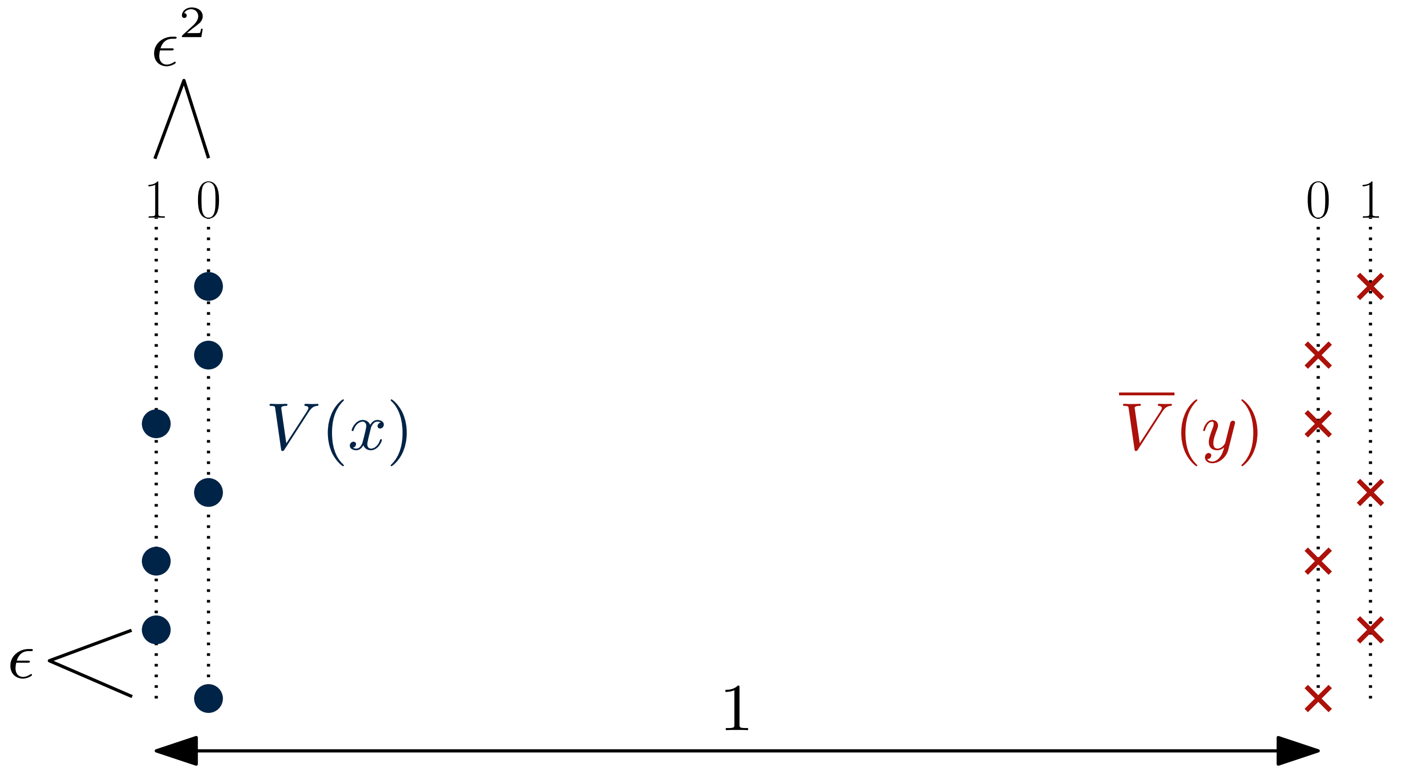


$x = (0, 1, 1, 0, 1, 0, 0)$

$y = (0, 1, 0, 1, 0, 0, 1)$

OV $\leq \delta_H^T$ in L_1

Vector Gadgets



$x = (0, 1, 1, 0, 1, 0, 0)$ $y = (0, 1, 0, 1, 0, 0, 1)$

Hausdorff distance at most $1 \iff \langle x, y \rangle = 0$

OV $\leq \delta_H^T$ in L_1

OV Input:
$X = \{x_1, \ldots, x_4\}$
$Y = \{y_1, \ldots y_3\}$

Choice Gadget

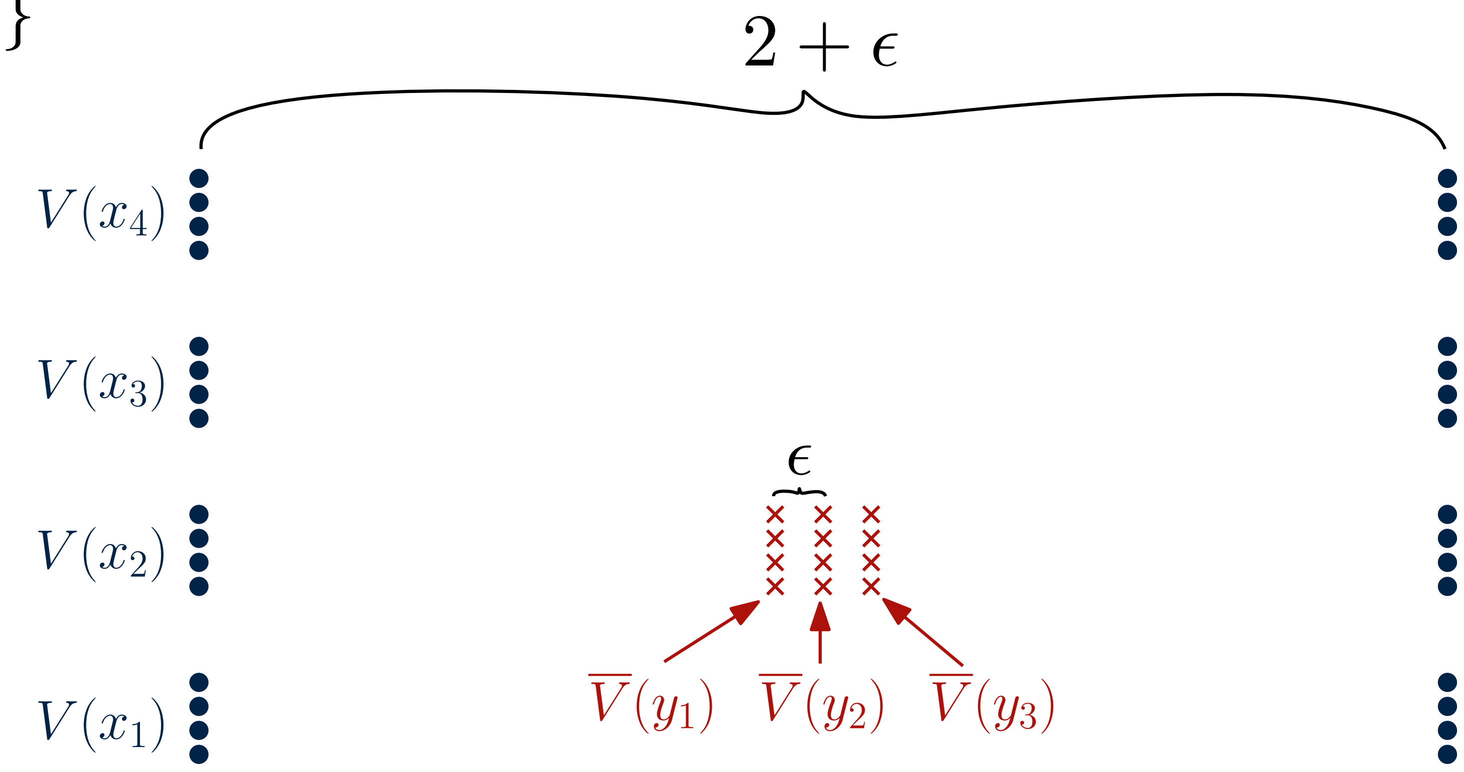
OV $\leq \delta_H^T$ in L_1

OV Input:

$X = \{x_1, \ldots, x_4\}$

$Y = \{y_1, \ldots y_3\}$

Choice Gadget



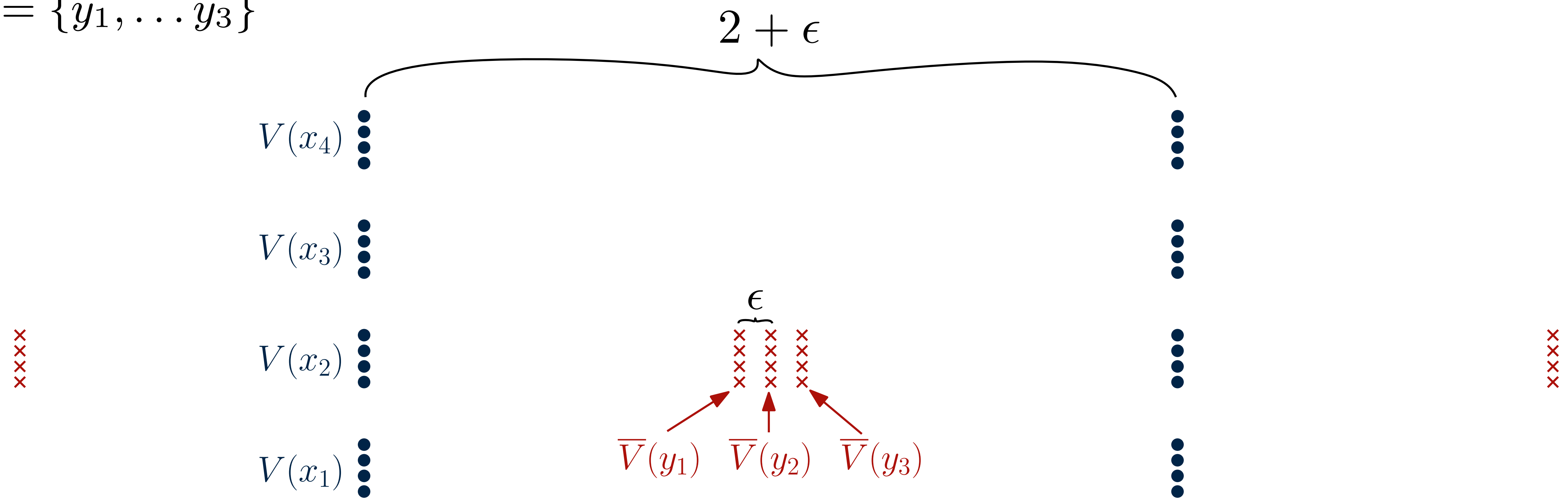
OV $\leq \delta_H^T$ in L_1

OV Input:

$X = \{x_1, \ldots, x_4\}$

$Y = \{y_1, \ldots y_3\}$

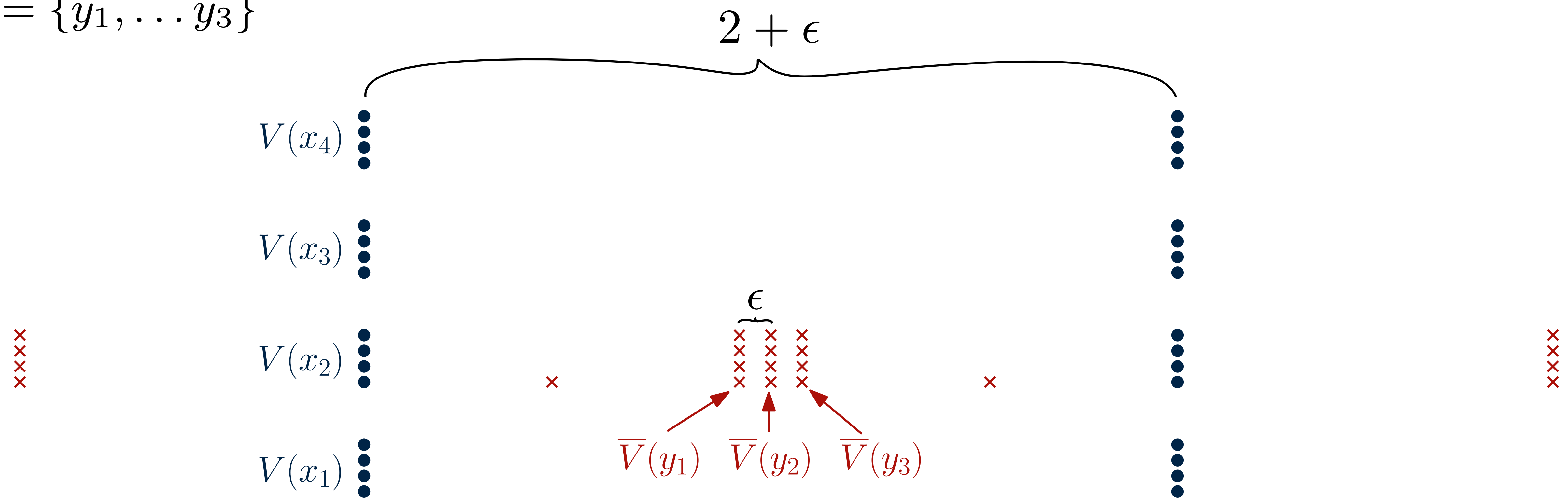
Choice Gadget



OV $\leq \delta_H^T$ in L_1

OV Input:
$X = \{x_1, \ldots, x_4\}$
$Y = \{y_1, \ldots y_3\}$

Choice Gadget



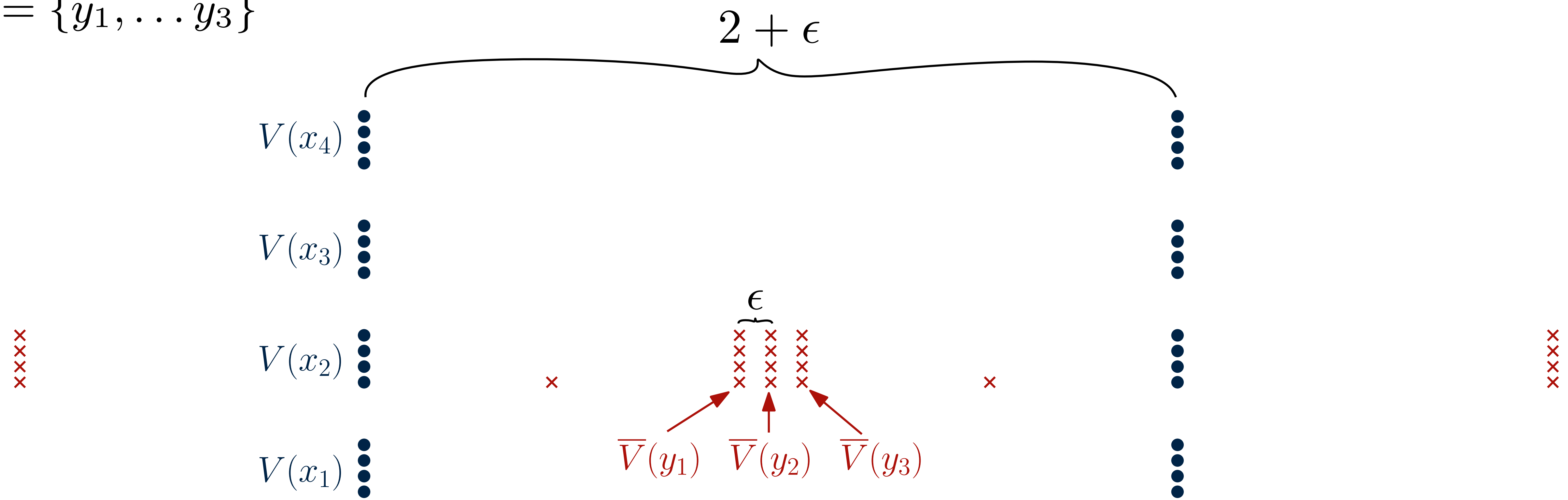
OV $\leq \delta_H^T$ in L_1

OV Input:

$X = \{x_1, \ldots, x_4\}$

$Y = \{y_1, \ldots y_3\}$

Choice Gadget



Hausdorff distance under translation at most 1 $\iff \exists i, j : \langle x_i, y_j \rangle = 0$

Second Reduction

Theorem

Hausdorff Distance Under Translation in L_2 with $m \in \mathcal{O}(1)$ cannot be computed in time $n^{2-\epsilon}$, unless the 3SUM Hypothesis fails.

3SUM

3SUM Problem

Given $X = (x_0, \ldots, x_{n-1}) \subset \{1, \ldots, \text{poly}(n)\}$, do there exist $i, j, k \in [n]$ s.t.

$$x_i + x_j = x_k$$

3SUM

3SUM Problem

Given $X = (x_0, \ldots, x_{n-1}) \subset \{1, \ldots, \text{poly}(n)\}$, do there exist $i, j, k \in [n]$ s.t.

$$x_i + x_j = x_k$$

3SUM Hypothesis

The 3SUM Problem cannot be solved in time $n^{2-\epsilon}$.

3SUM

Convolution

3SUM Problem

Given $X = (x_0, \ldots, x_{n-1}) \subset \{1, \ldots, \text{poly}(n)\}$, do there exist $i, j, \cancel{k} \in [n]$ s.t.

$$x_i + x_j = x_{\cancel{k}\, i+j}$$

3SUM Hypothesis

The 3SUM Problem cannot be solved in time $n^{2-\epsilon}$.

max planck institut informatik

3SUM

Convolution

3SUM Problem

Given $X = (x_0, \ldots, x_{n-1}) \subset \{1, \ldots, \text{poly}(n)\}$, do there exist $i, j, \cancel{k} \in [n]$ s.t.

$$x_i + x_j = x_{\cancel{k}\; i+j}$$

Convolution

3SUM Hypothesis

The Convolution 3SUM Problem cannot be solved in time $n^{2-\epsilon}$.

3SUM

Convolution

3SUM Problem

Given $X = (x_0, \dots, x_{n-1}) \subset \{1, \dots, \text{poly}(n)\}$, do there exist $i, j, \not{k} \in [n]$ s.t.

$$x_i + x_j = x_{\not{k}\, i+j}$$

Convolution

3SUM Hypothesis

The Convolution 3SUM Problem cannot be solved in time $n^{2-\epsilon}$.

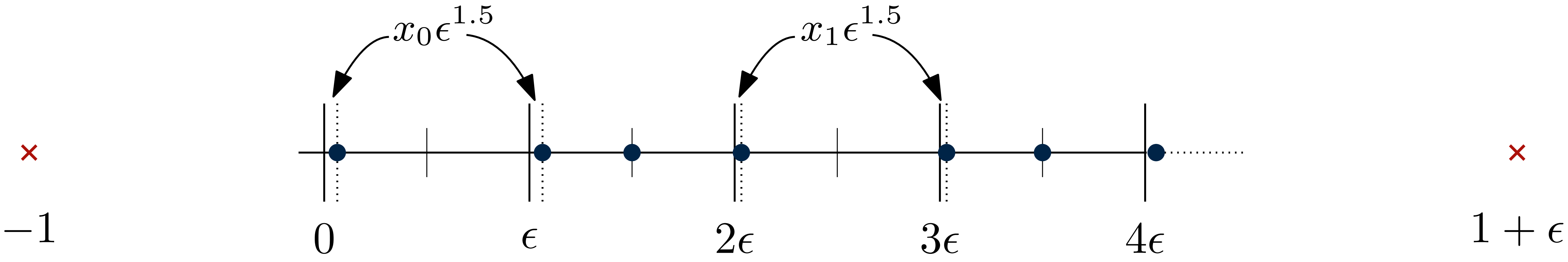
→ 3SUM and Convolution 3SUM Hypotheses are equivalent [Pătrașcu '10]

Conv3SUM $\leq \delta_H^T$ in L_2

Encode the sequence $X = (x_0, \ldots, x_{n-1})$:

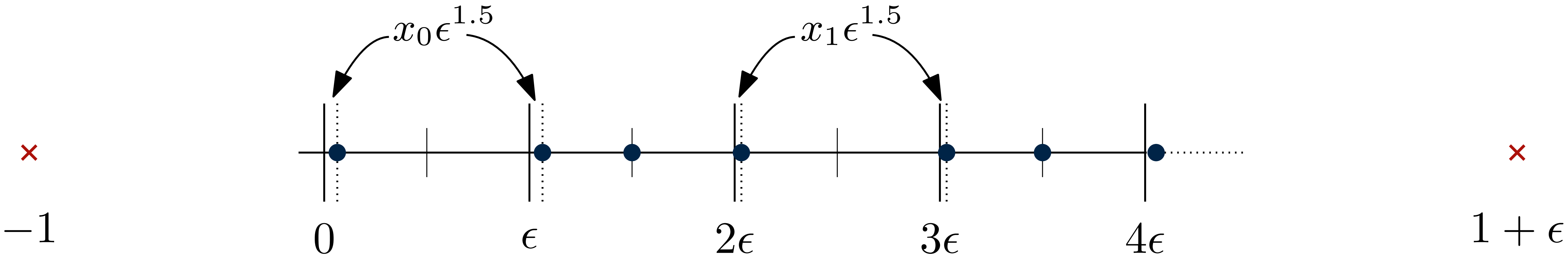
Conv3SUM $\leq \delta_H^T$ in L_2

Encode the sequence $X = (x_0, \ldots, x_{n-1})$:



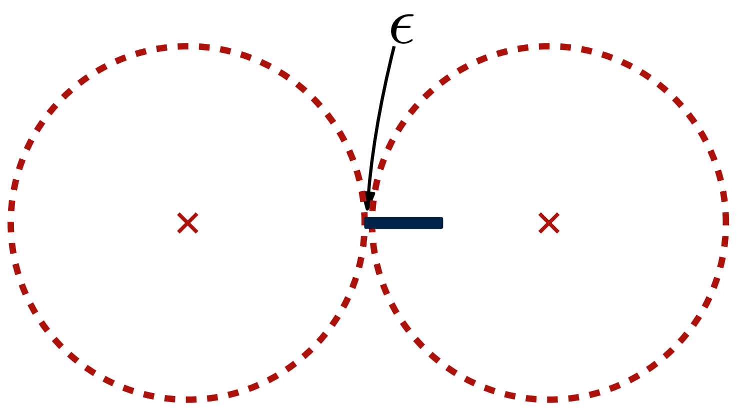
Conv3SUM $\leq \delta_H^T$ in L_2

Encode the sequence $X = (x_0, \ldots, x_{n-1})$:



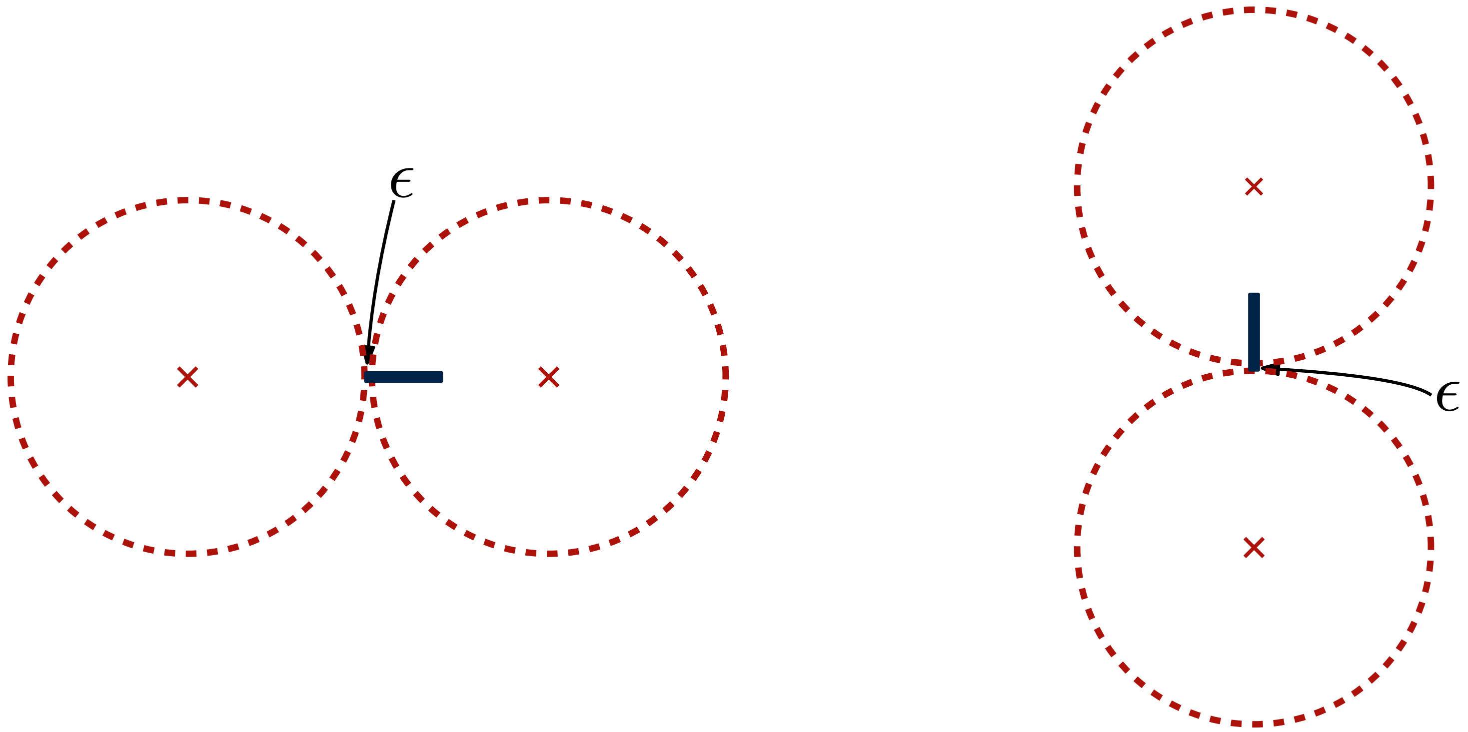
Feasible translations: $\tau_x \approx 2i\epsilon + x_i\epsilon^{1.5}$ for $i \in [n]$

Conv3SUM $\leq \delta_H^T$ in L_2



$$\tau_x \approx 2i\epsilon + x_i\epsilon^{1.5}$$

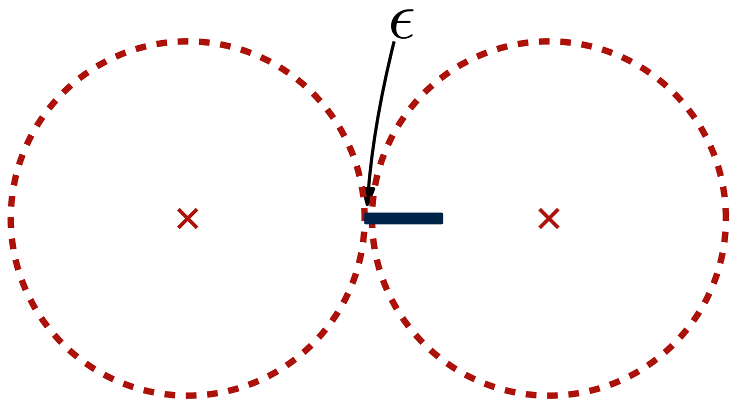
Conv3SUM $\leq \delta_H^T$ in L_2



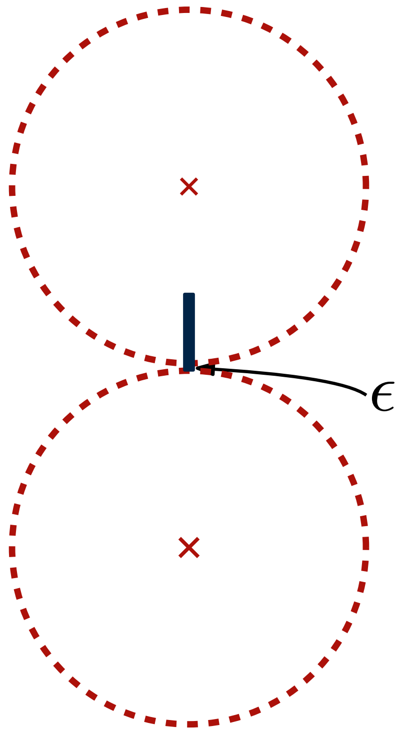
$\tau_x \approx 2i\epsilon + x_i\epsilon^{1.5}$

$\tau_y \approx 2j\epsilon + x_j\epsilon^{1.5}$

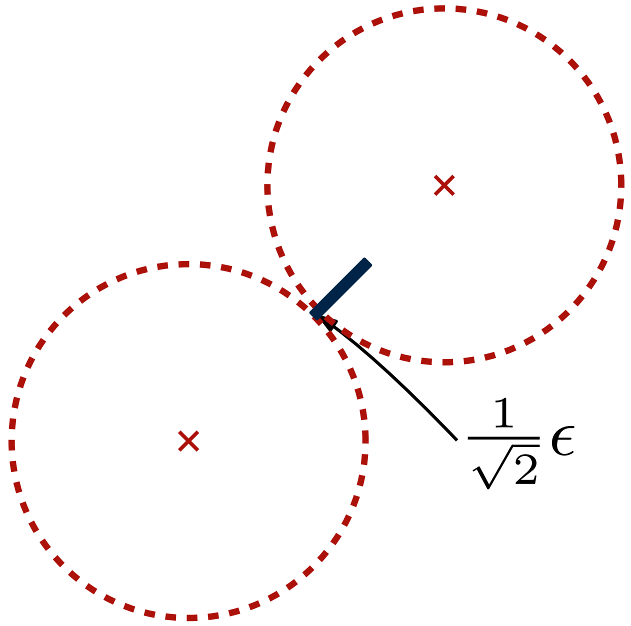
Conv3SUM $\leq \delta_H^T$ in L_2



$\tau_x \approx 2i\epsilon + x_i\epsilon^{1.5}$



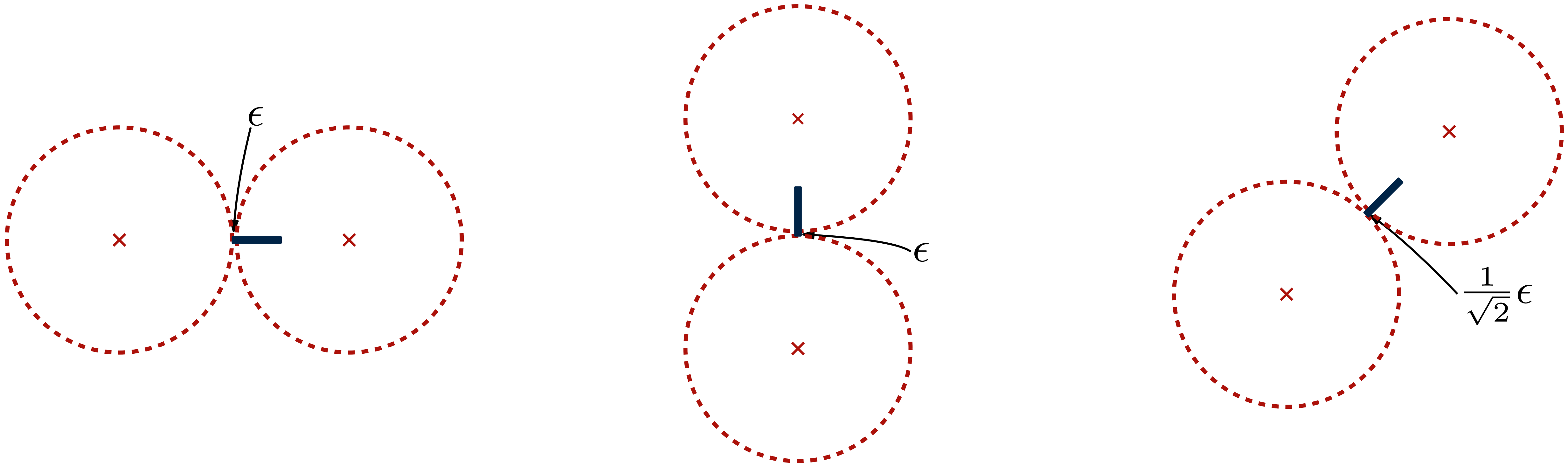
$\tau_y \approx 2j\epsilon + x_j\epsilon^{1.5}$



$\tau_x + \tau_y \approx 2k\epsilon + x_k\epsilon^{1.5}$

max planck institut informatik

Conv3SUM $\leq \delta_H^T$ in L_2



$\tau_x \approx 2i\epsilon + x_i\epsilon^{1.5}$

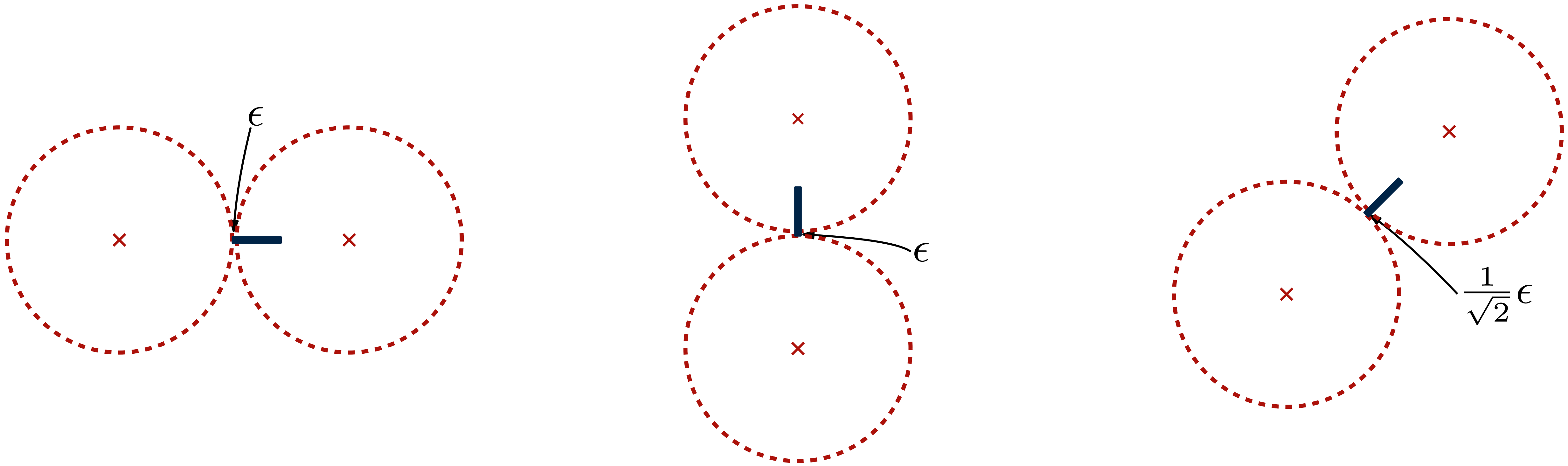
$\tau_y \approx 2j\epsilon + x_j\epsilon^{1.5}$

$\tau_x + \tau_y \approx 2k\epsilon + x_k\epsilon^{1.5}$

$$\exists i,j,k \in [n] : 2i\epsilon + 2j\epsilon - 2k\epsilon + x_i\epsilon^{1.5} + x_j\epsilon^{1.5} - x_k\epsilon^{1.5} \approx 0$$

max planck institut informatik

Conv3SUM $\leq \delta_H^T$ in L_2



$\tau_x \approx 2i\epsilon + x_i\epsilon^{1.5}$

$\tau_y \approx 2j\epsilon + x_j\epsilon^{1.5}$

$\tau_x + \tau_y \approx 2k\epsilon + x_k\epsilon^{1.5}$

$$\exists i, j, k \in [n] : 2i\epsilon + 2j\epsilon - 2k\epsilon + x_i\epsilon^{1.5} + x_j\epsilon^{1.5} - x_k\epsilon^{1.5} \approx 0$$

$$\iff \exists i, j, k \in [n] : i + j = k \text{ and } x_i + x_j = x_k$$

Summary

Summary

- Classic algorithms are conditionally optimal
  - L_2 for a special case

Summary

- Classic algorithms are conditionally optimal
  - L_2 for a special case
- Hardness from two different conjectures
  - SETH vs. 3SUM

Summary

- Classic algorithms are conditionally optimal
  - L_2 for a special case
- Hardness from two different conjectures
  - SETH vs. 3SUM

- *Note:* Using $m \in \mathcal{O}(n)$ in the L_2 result encodes 4SUM
  - 4SUM is quadratic; want reduction from 5SUM

Summary

- Classic algorithms are conditionally optimal
  - L_2 for a special case
- Hardness from two different conjectures
  - SETH vs. 3SUM

- *Note:* Using $m \in \mathcal{O}(n)$ in the L_2 result encodes 4SUM
  - 4SUM is quadratic; want reduction from 5SUM
- *Open Question:* Is the balanced version in L_2 subcubic?

Summary

- Classic algorithms are conditionally optimal
  - L_2 for a special case
- Hardness from two different conjectures
  - SETH vs. 3SUM

- *Note:* Using $m \in \mathcal{O}(n)$ in the L_2 result encodes 4SUM
  - 4SUM is quadratic; want reduction from 5SUM
- *Open Question:* Is the balanced version in L_2 subcubic?

Vielen Dank!

max planck institut informatik