RUB

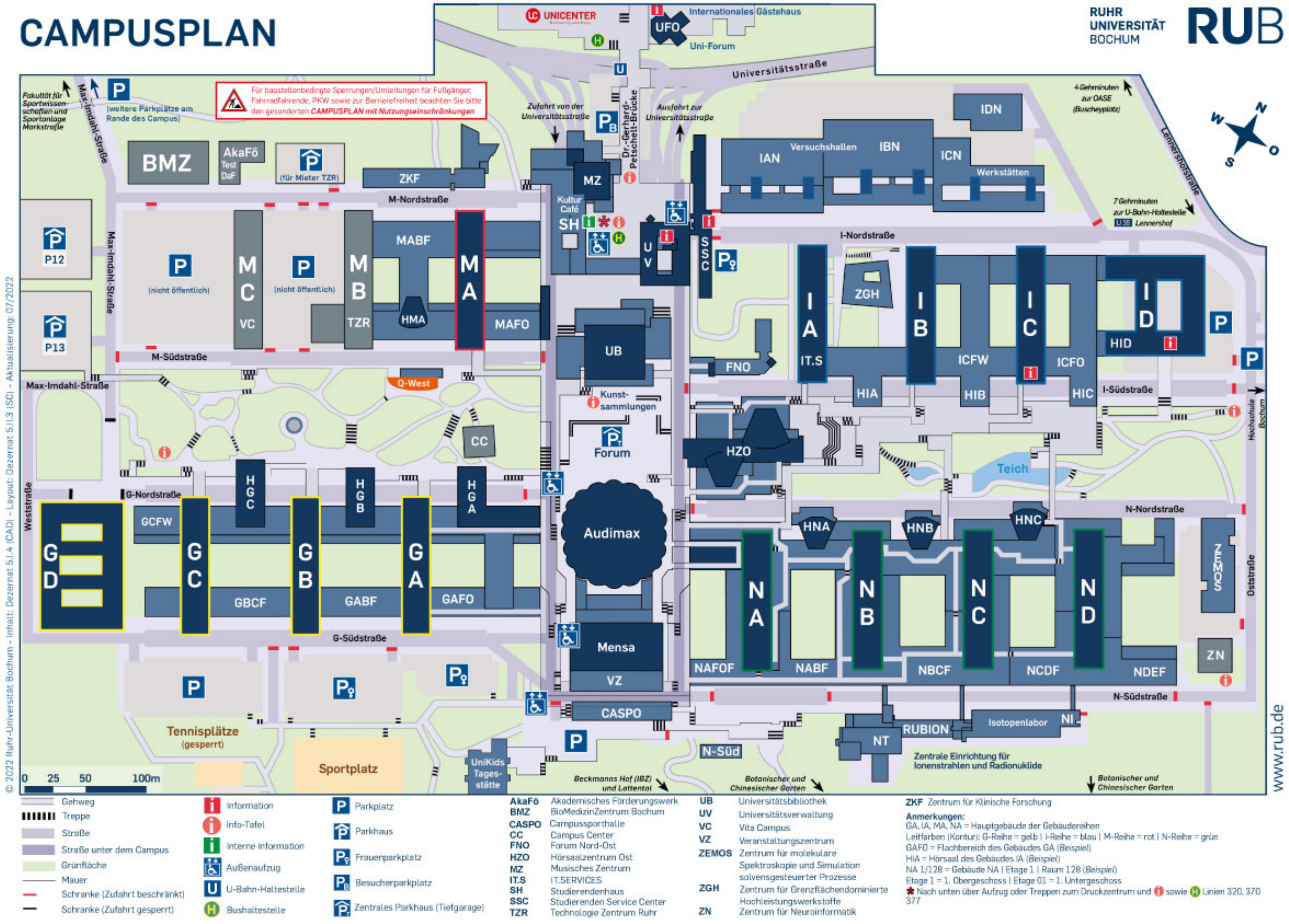
Hardness and modifications of the weak graph distance

13.03.2024

Maike Buchin, **Wolf Kißler**

Distance measures for immersed graphs

Representations of geometric networks:

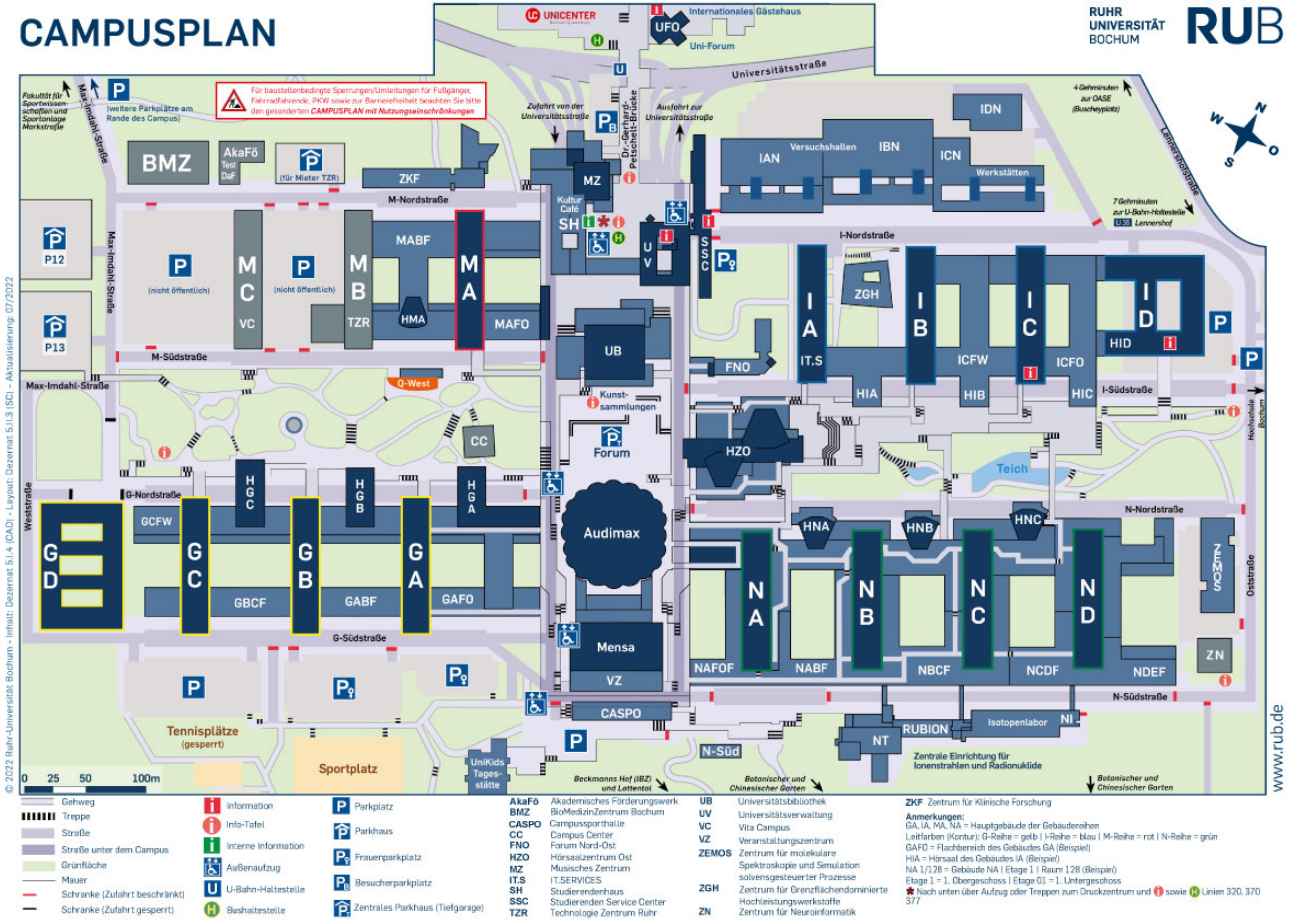


Ruhr-Universität Bochum, Dezernat 5.I.4 & Dezernat 5.II.3

Distance measures for immersed graphs

Representations of geometric networks:

- Embedded graphs: Drawings without crossings
- Immersed graphs: Drawings that may contain crossings
- Plane graphs: Graphs embedded in $\mathbb{R}^2$



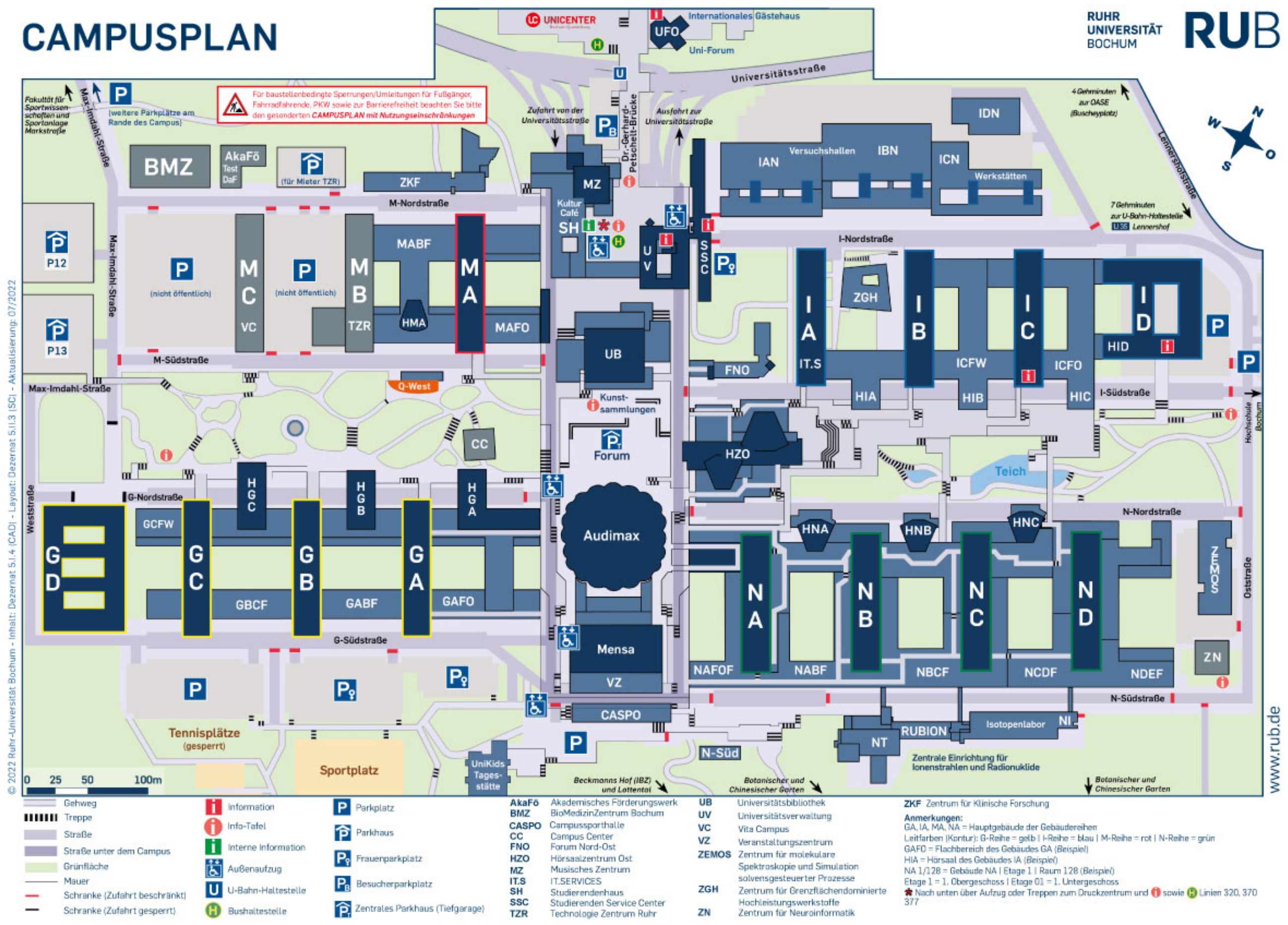
Ruhr-Universität Bochum, Dezernat 5.I.4 & Dezernat 5.II.3

Distance measures for immersed graphs

Representations of geometric networks:

- Embedded graphs: Drawings without crossings
- Immersed graphs: Drawings that may contain crossings
- Plane graphs: Graphs embedded in $\mathbb{R}^2$

We consider non-degenerate straight-line immersions



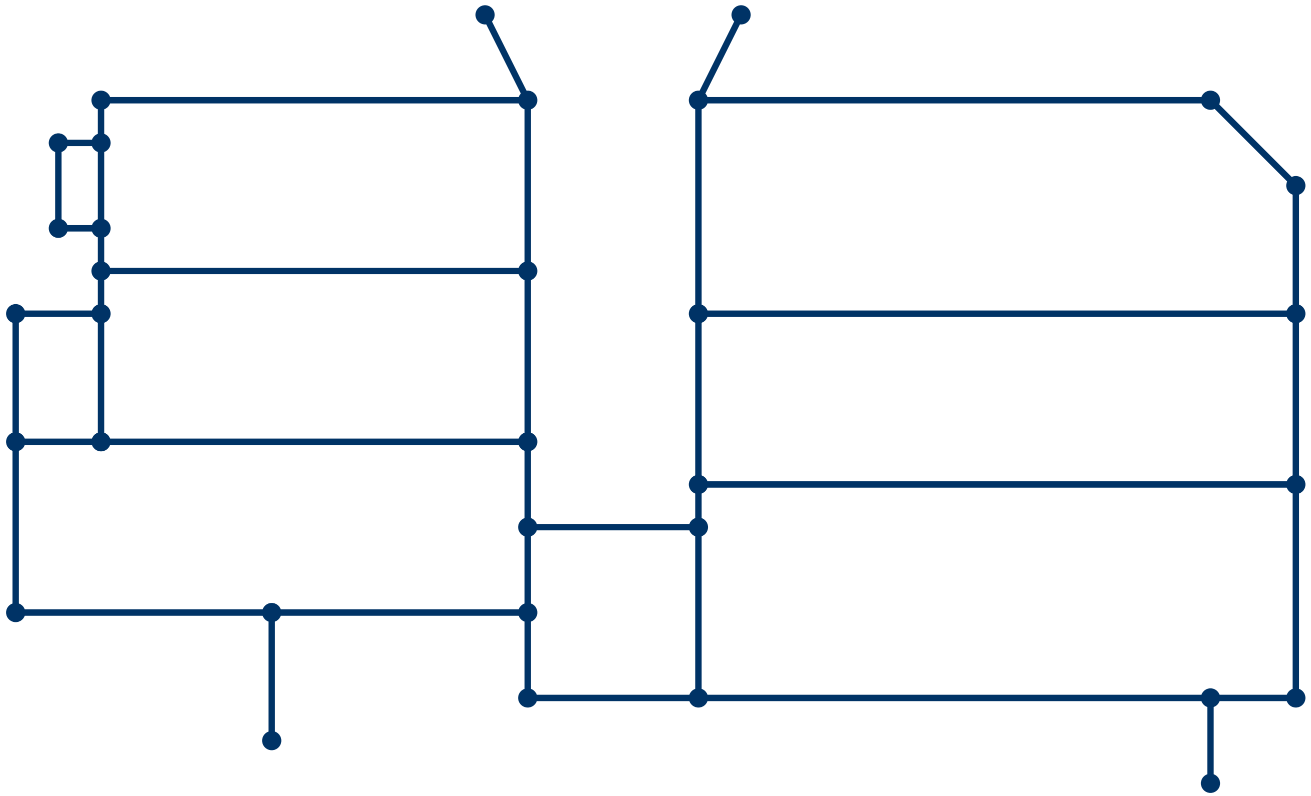
Ruhr-Universität Bochum, Dezernat 5.I.4 & Dezernat 5.II.3

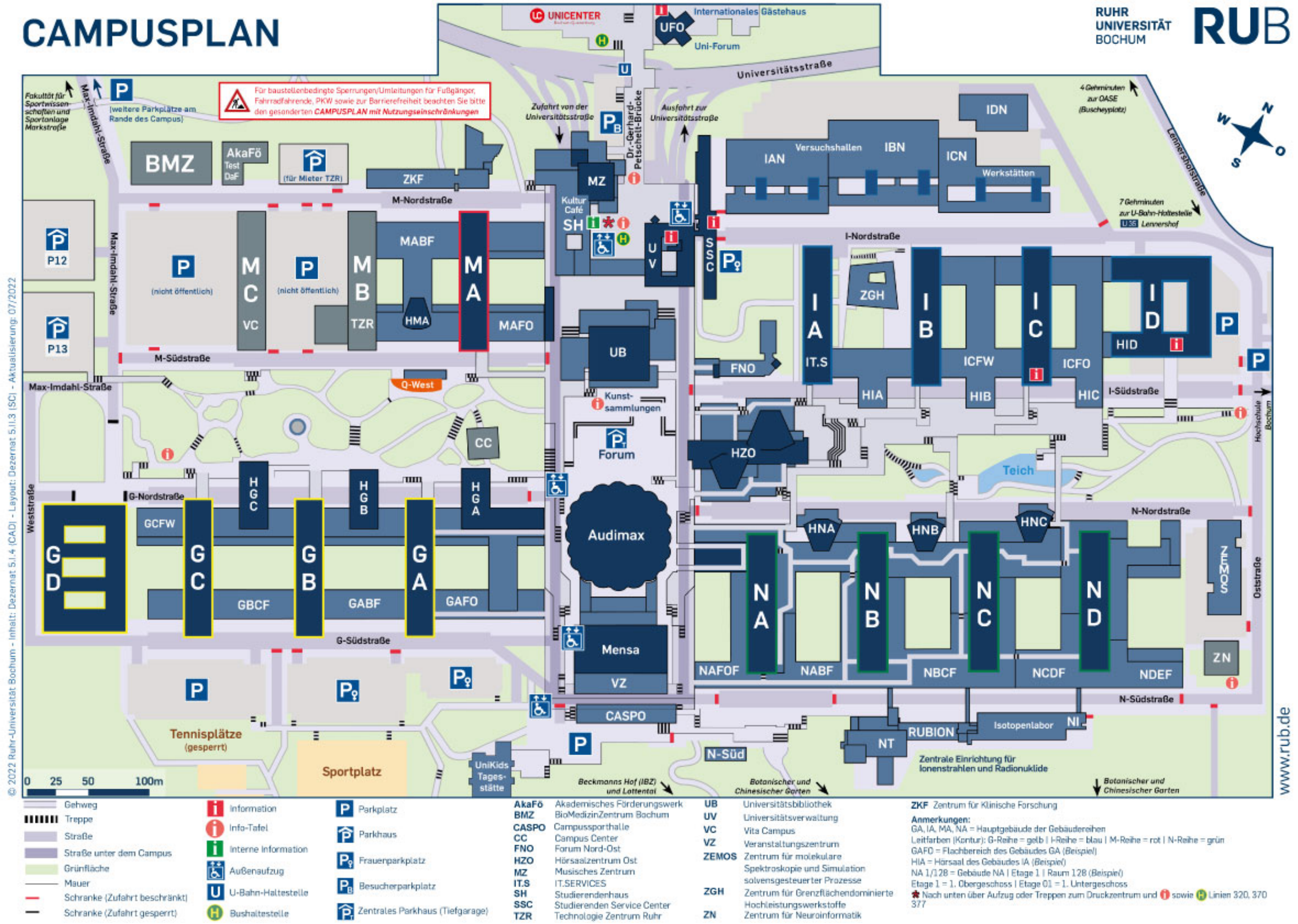
Distance measures for immersed graphs

Representations of geometric networks:

- Embedded graphs: Drawings without crossings
- Immersed graphs: Drawings that may contain crossings
- Plane graphs: Graphs embedded in $\mathbb{R}^2$

We consider non-degenerate straight-line immersions





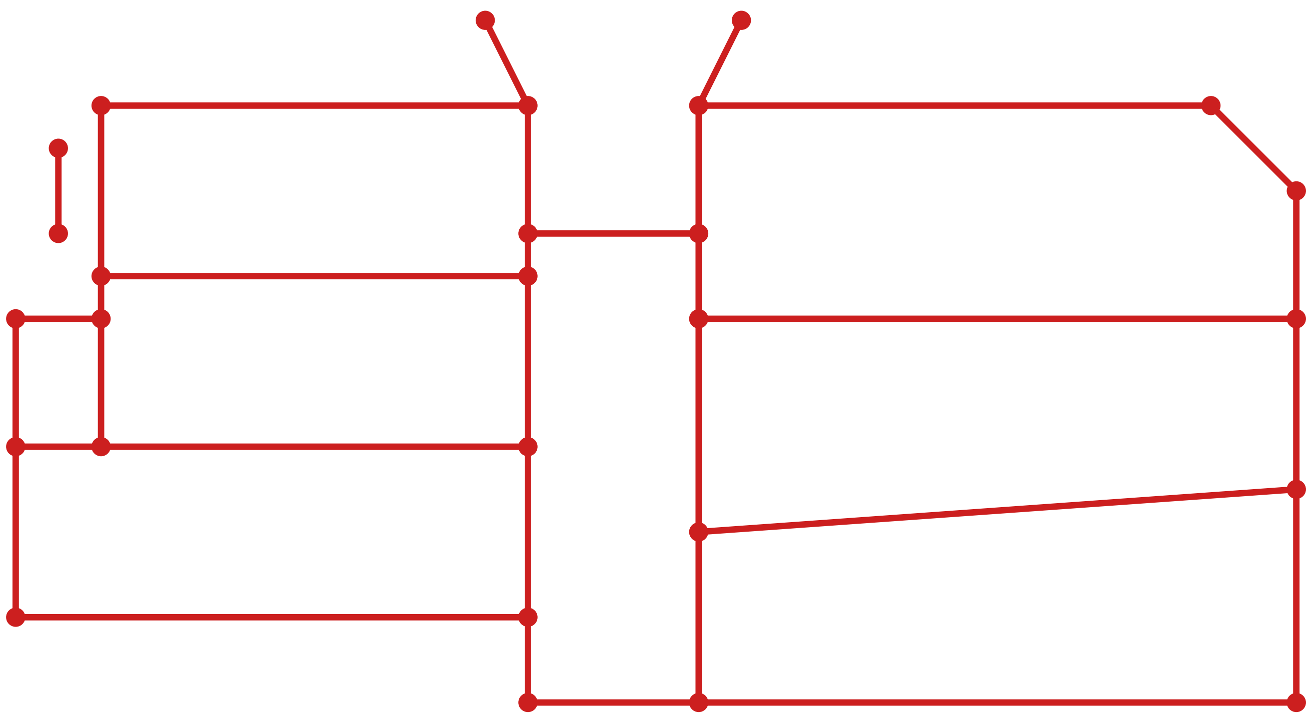
Ruhr-Universität Bochum, Dezernat 5.I.4 & Dezernat 5.II.3

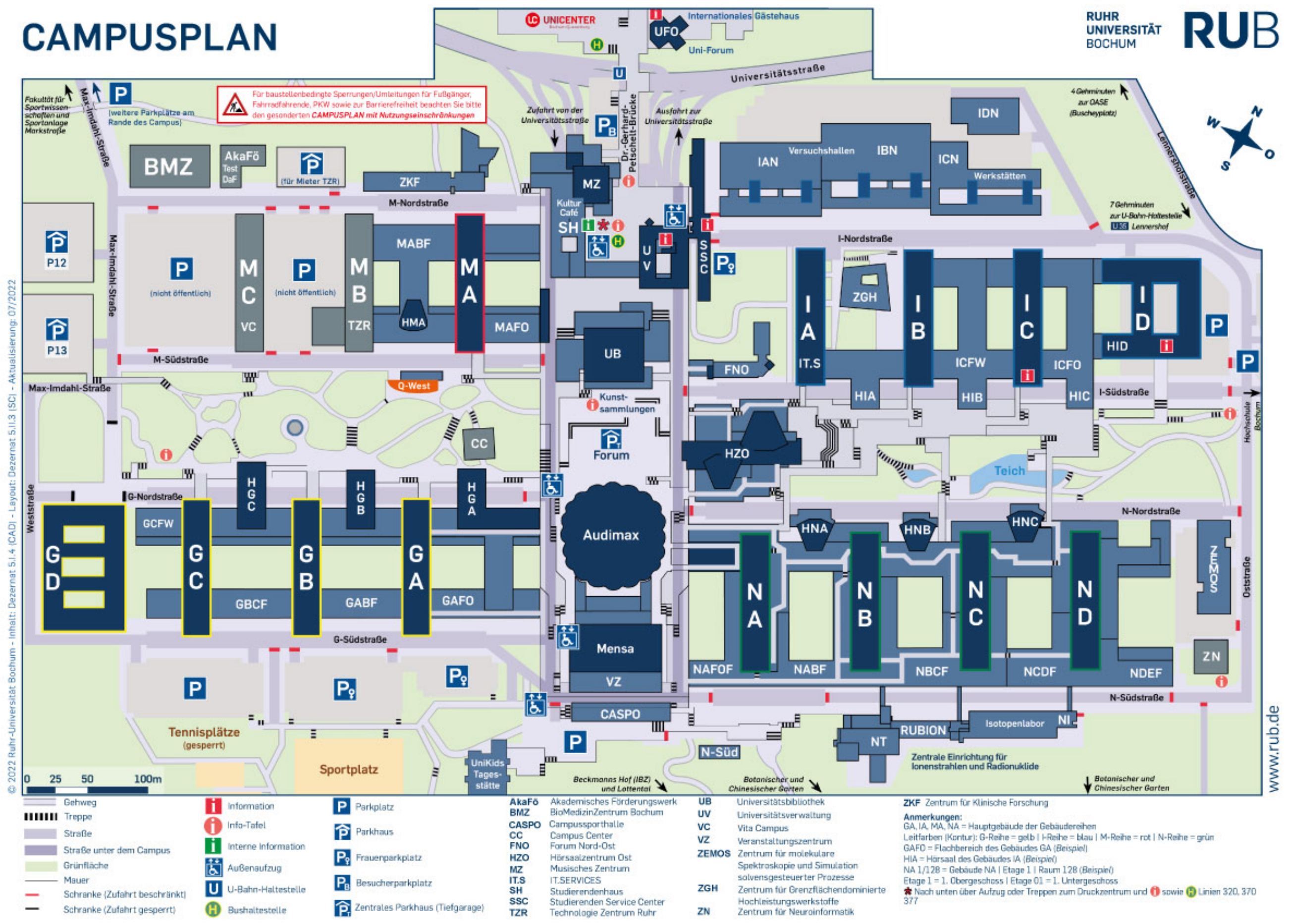
Distance measures for immersed graphs

Representations of geometric networks:

- Embedded graphs: Drawings without crossings
- Immersed graphs: Drawings that may contain crossings
- Plane graphs: Graphs embedded in $\mathbb{R}^2$

We consider non-degenerate straight-line immersions





Ruhr-Universität Bochum, Dezernat 5.I.4 & Dezernat 5.II.3

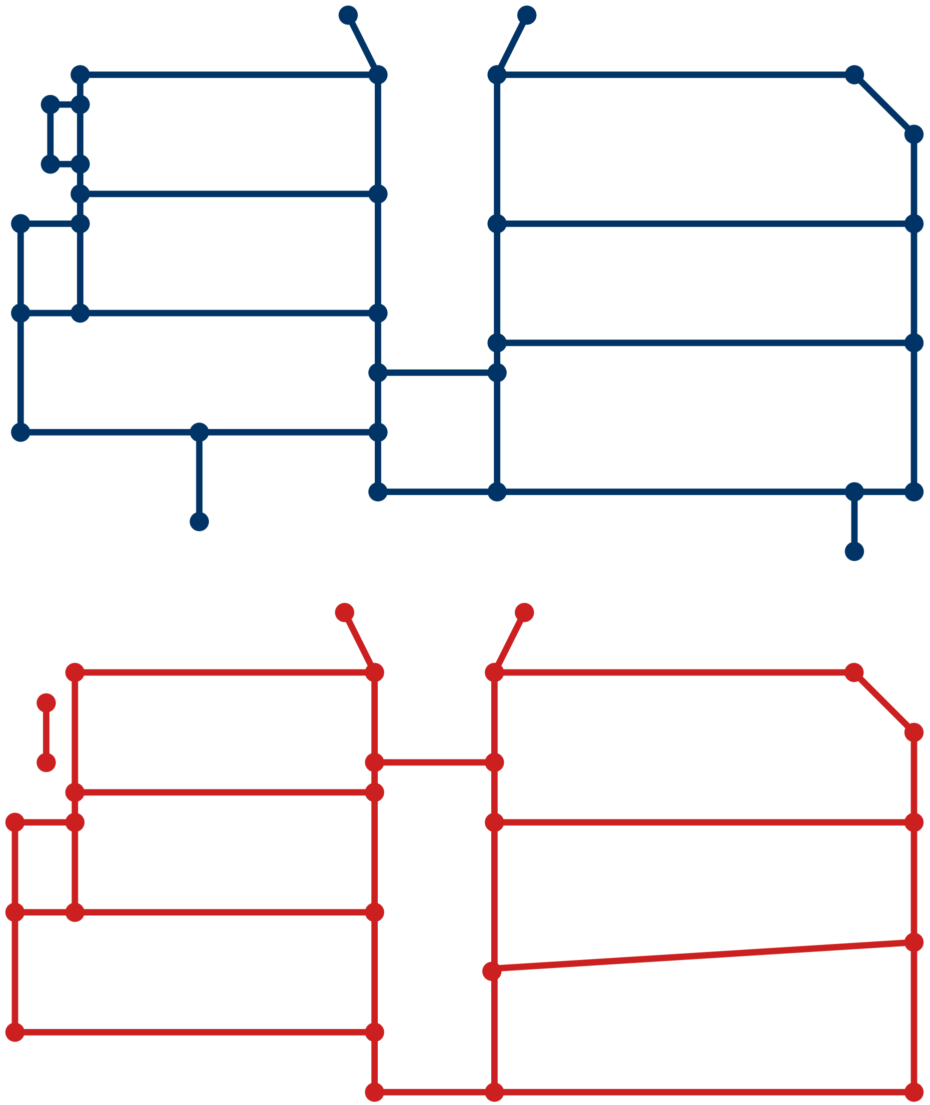
Distance measures for immersed graphs

Representations of geometric networks:

- Embedded graphs: Drawings without crossings
- Immersed graphs: Drawings that may contain crossings
- Plane graphs: Graphs embedded in $\mathbb{R}^2$

We consider non-degenerate straight-line immersions

Given two representations of networks: how to compare them?

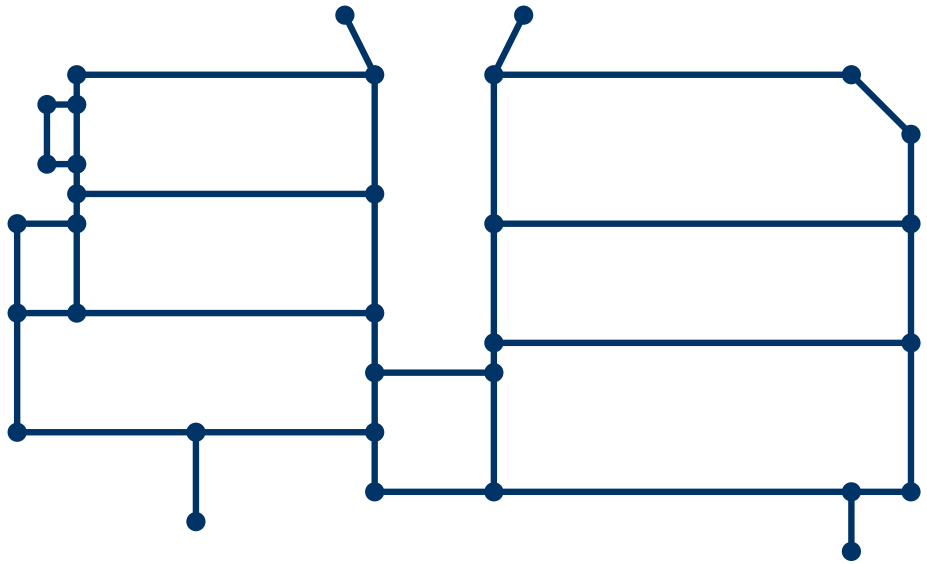


Distance measures for immersed graphs

Representations of geometric networks:

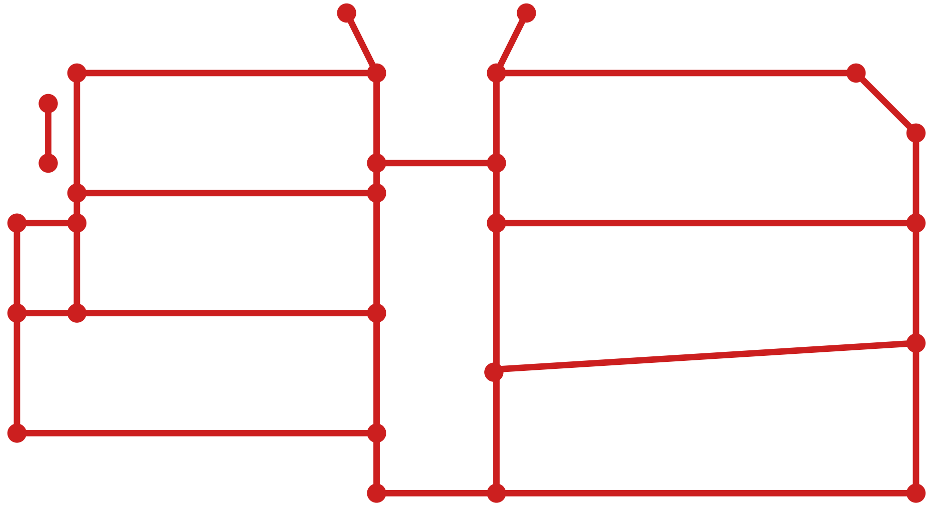
- Embedded graphs: Drawings without crossings
- Immersed graphs: Drawings that may contain crossings
- Plane graphs: Graphs embedded in $\mathbb{R}^2$

We consider non-degenerate straight-line immersions



Given two representations of networks: how to compare them?

- Many approaches: edit distances, Fréchet distance, traversal based distances, LPH based distances,...
- Here: Weak Graph Distance due to Akitaya et al.



→ Akitaya et al.: Distance measures for embedded graphs, CGTA 95, 2021.

Table of Contents

Recap: Weak Fréchet distance

RUB

Let $s_1, s_2\colon [0,1] \to \mathbb{R}^d$ be curves. Their *weak Fréchet distance* is defined by

$$\delta_{wF}(s_1, s_2) = \inf_{\alpha,\beta:[0,1]\to[0,1]} \max_{t\in[0,1]} d(s_1(\alpha(t)), s_2(\beta(t)))$$

continuous surjection

Recap: Weak Fréchet distance

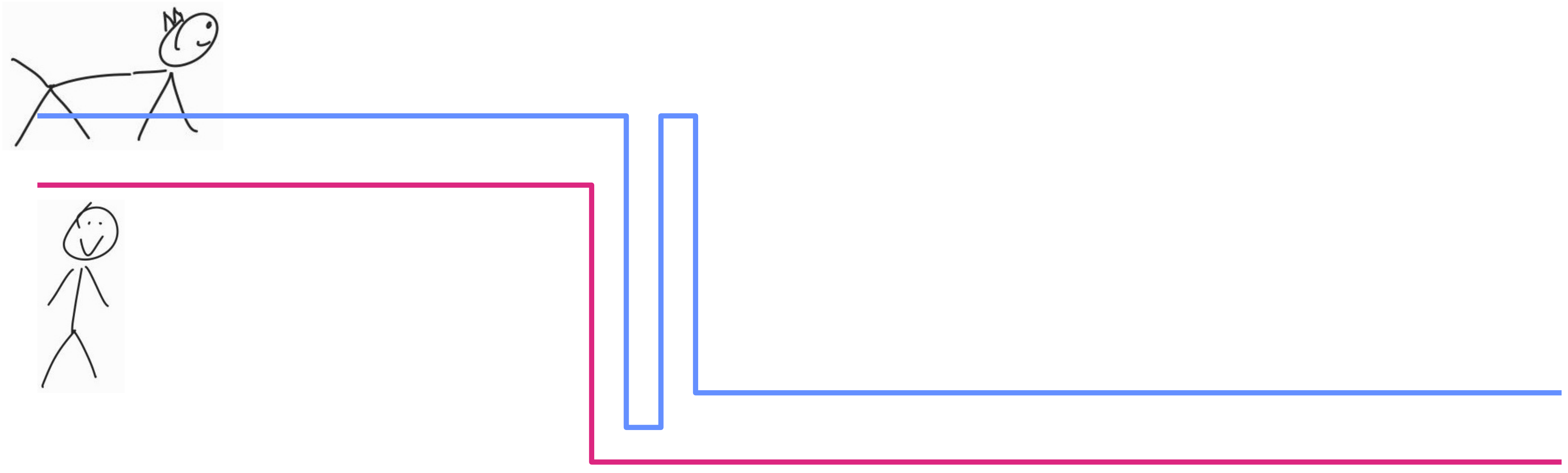
Let $s_1, s_2\colon [0,1] \to \mathbb{R}^d$ be curves. Their *weak Fréchet distance* is defined by

$$\delta_{wF}(s_1, s_2) = \inf_{\alpha,\beta:[0,1]\to[0,1]} \max_{t\in[0,1]} d(s_1(\alpha(t)), s_2(\beta(t)))$$

continuous surjection

Intuition: Person and dog:

Weak Fréchet distance equals shortest leash length *when both are allowed to backtrack along their trajectory*



Recap: Weak Fréchet distance

RUB

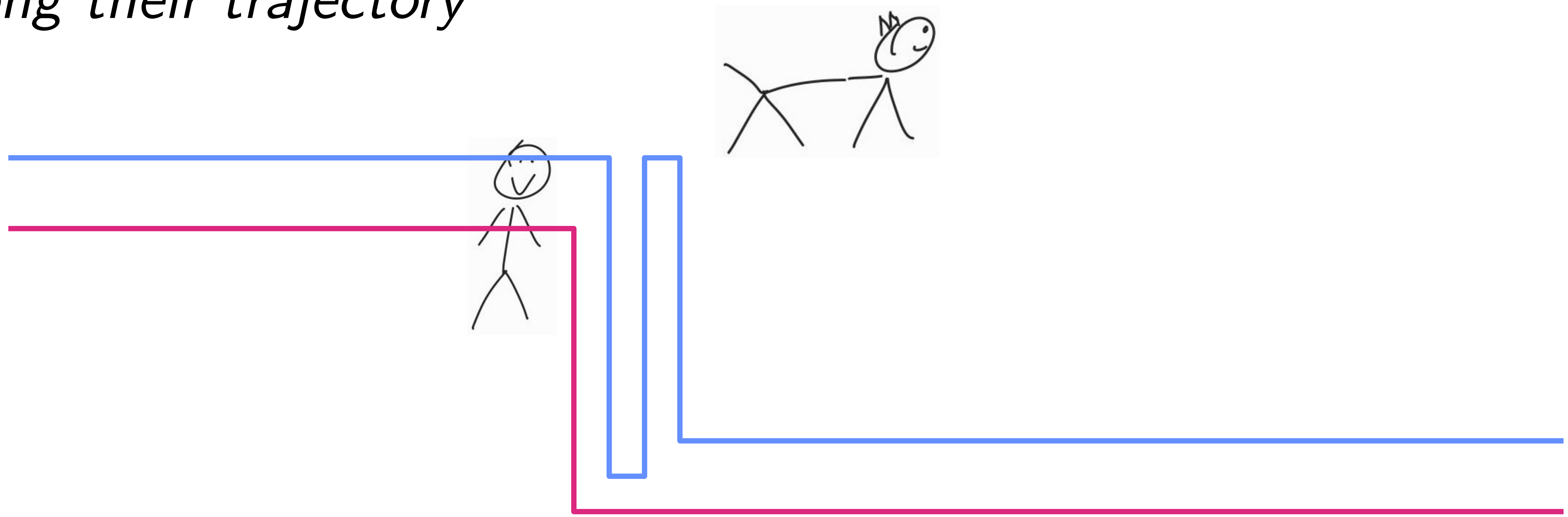
Let $s_1, s_2 \colon [0,1] \to \mathbb{R}^d$ be curves. Their *weak Fréchet distance* is defined by

$$\delta_{wF}(s_1, s_2) = \inf_{\alpha,\beta:[0,1]\to[0,1]} \max_{t\in[0,1]} d(s_1(\alpha(t)), s_2(\beta(t)))$$

continuous surjection

Intuition: Person and dog:

Weak Fréchet distance equals shortest leash length *when both are allowed to backtrack along their trajectory*



Recap: Weak Fréchet distance

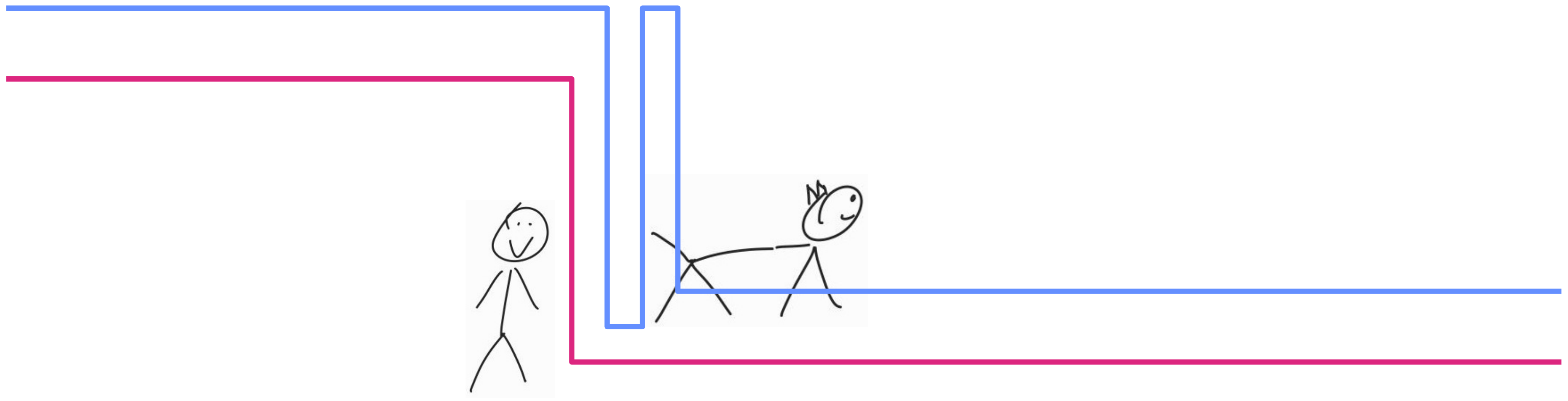
Let $s_1, s_2\colon [0,1] \to \mathbb{R}^d$ be curves. Their *weak Fréchet distance* is defined by

$$\delta_{wF}(s_1, s_2) = \inf_{\alpha,\beta:[0,1]\to[0,1]} \max_{t\in[0,1]} d(s_1(\alpha(t)), s_2(\beta(t)))$$

continuous surjection

Intuition: Person and dog:

Weak Fréchet distance equals shortest leash length *when both are allowed to backtrack along their trajectory*



Recap: Weak Fréchet distance

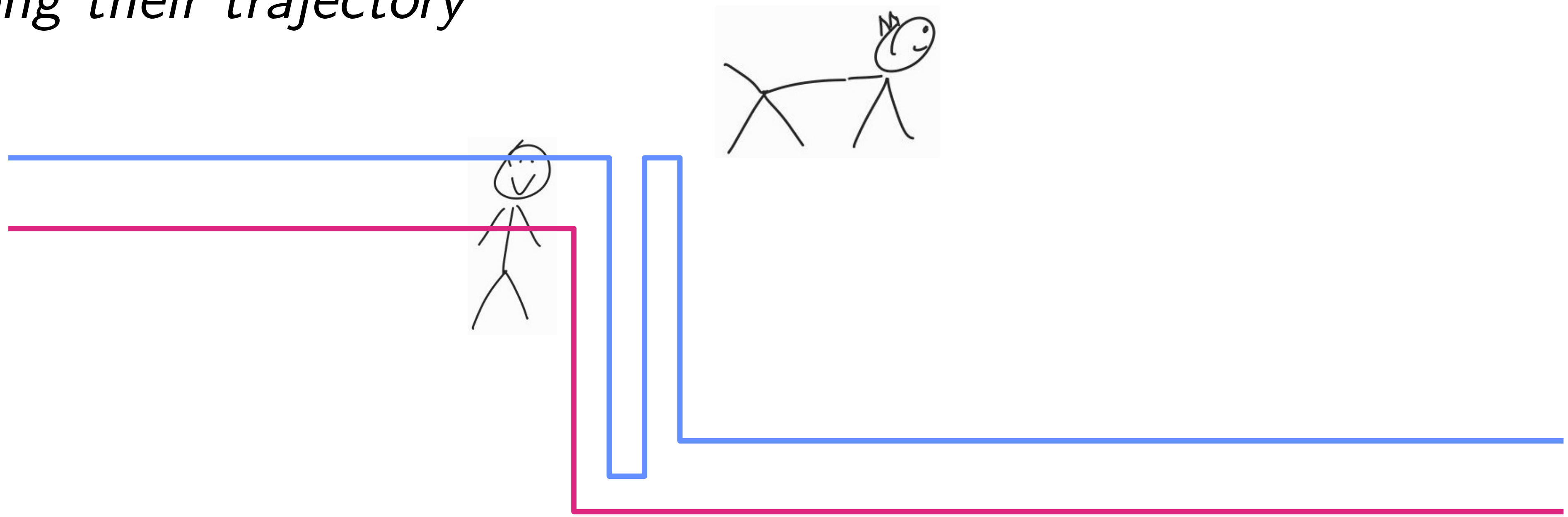
Let $s_1, s_2\colon [0,1] \to \mathbb{R}^d$ be curves. Their *weak Fréchet distance* is defined by

$$\delta_{wF}(s_1, s_2) = \inf_{\alpha,\beta:[0,1]\to[0,1]} \max_{t\in[0,1]} d(s_1(\alpha(t)), s_2(\beta(t)))$$

continuous surjection

Intuition: Person and dog:

Weak Fréchet distance equals shortest leash length *when both are allowed to backtrack along their trajectory*



Recap: Weak Fréchet distance

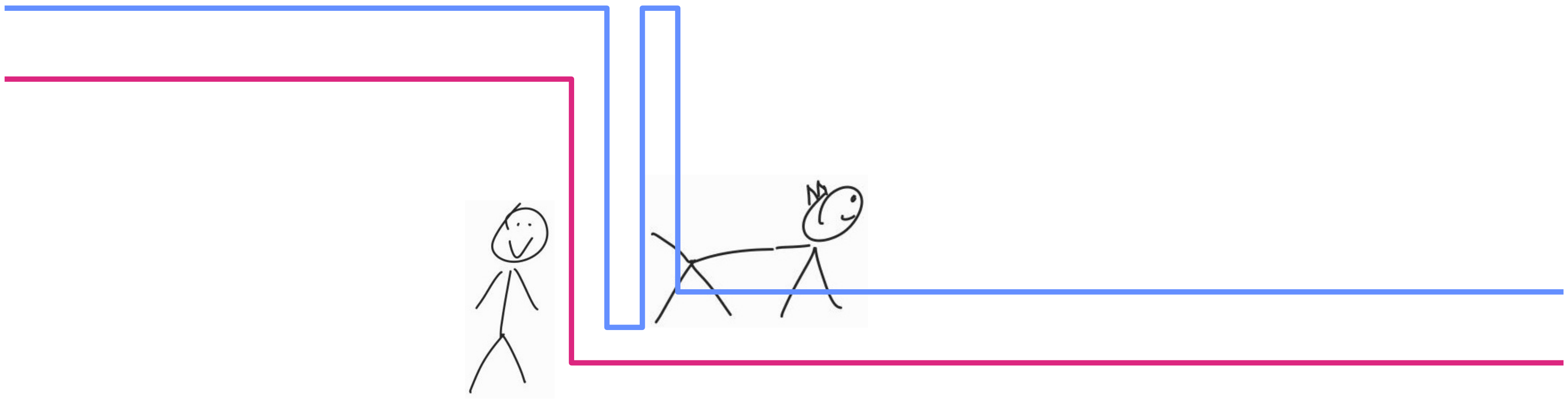
Let $s_1, s_2 \colon [0,1] \to \mathbb{R}^d$ be curves. Their *weak Fréchet distance* is defined by

$$\delta_{wF}(s_1, s_2) = \inf_{\alpha,\beta:[0,1]\to[0,1]} \max_{t\in[0,1]} d(s_1(\alpha(t)), s_2(\beta(t)))$$

continuous surjection

Intuition: Person and dog:

Weak Fréchet distance equals shortest leash length *when both are allowed to backtrack along their trajectory*



Recap: Weak Fréchet distance

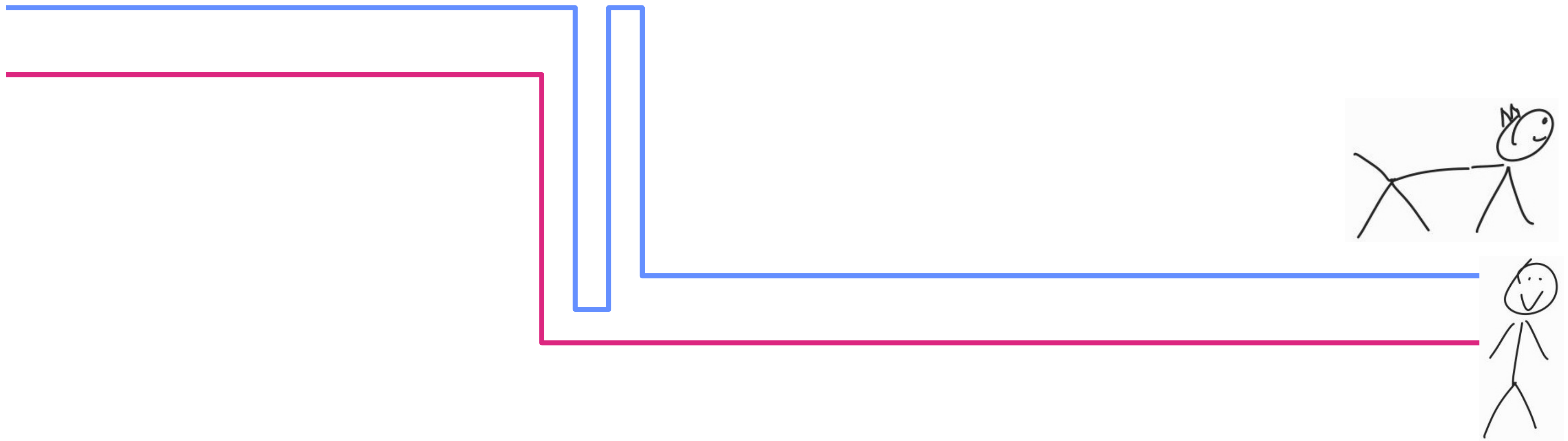
Let $s_1, s_2\colon [0,1] \to \mathbb{R}^d$ be curves. Their *weak Fréchet distance* is defined by

$$\delta_{wF}(s_1, s_2) = \inf_{\alpha,\beta:[0,1]\to[0,1]} \max_{t\in[0,1]} d(s_1(\alpha(t)), s_2(\beta(t)))$$

continuous surjection

Intuition: Person and dog:

Weak Fréchet distance equals shortest leash length *when both are allowed to backtrack along their trajectory*



Recap: Weak Fréchet distance

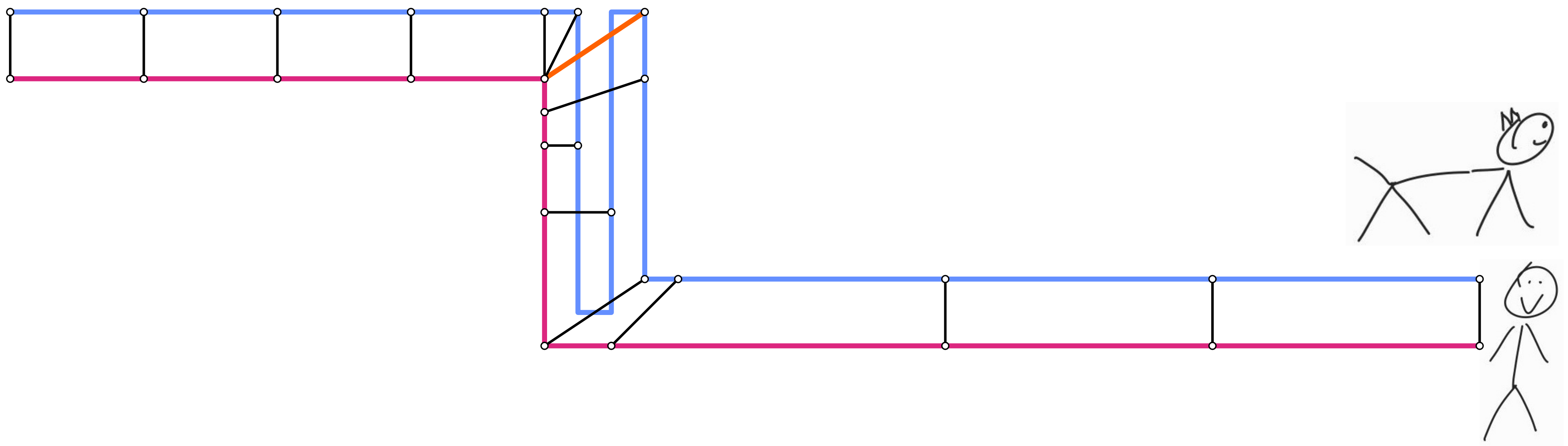
Let $s_1, s_2\colon [0,1] \to \mathbb{R}^d$ be curves. Their *weak Fréchet distance* is defined by

$$\delta_{wF}(s_1, s_2) = \inf_{\alpha,\beta:[0,1]\to[0,1]} \max_{t\in[0,1]} d(s_1(\alpha(t)), s_2(\beta(t)))$$

continuous surjection

Intuition: Person and dog:

Weak Fréchet distance equals shortest leash length *when both are allowed to backtrack along their trajectory*



Weak graph distance

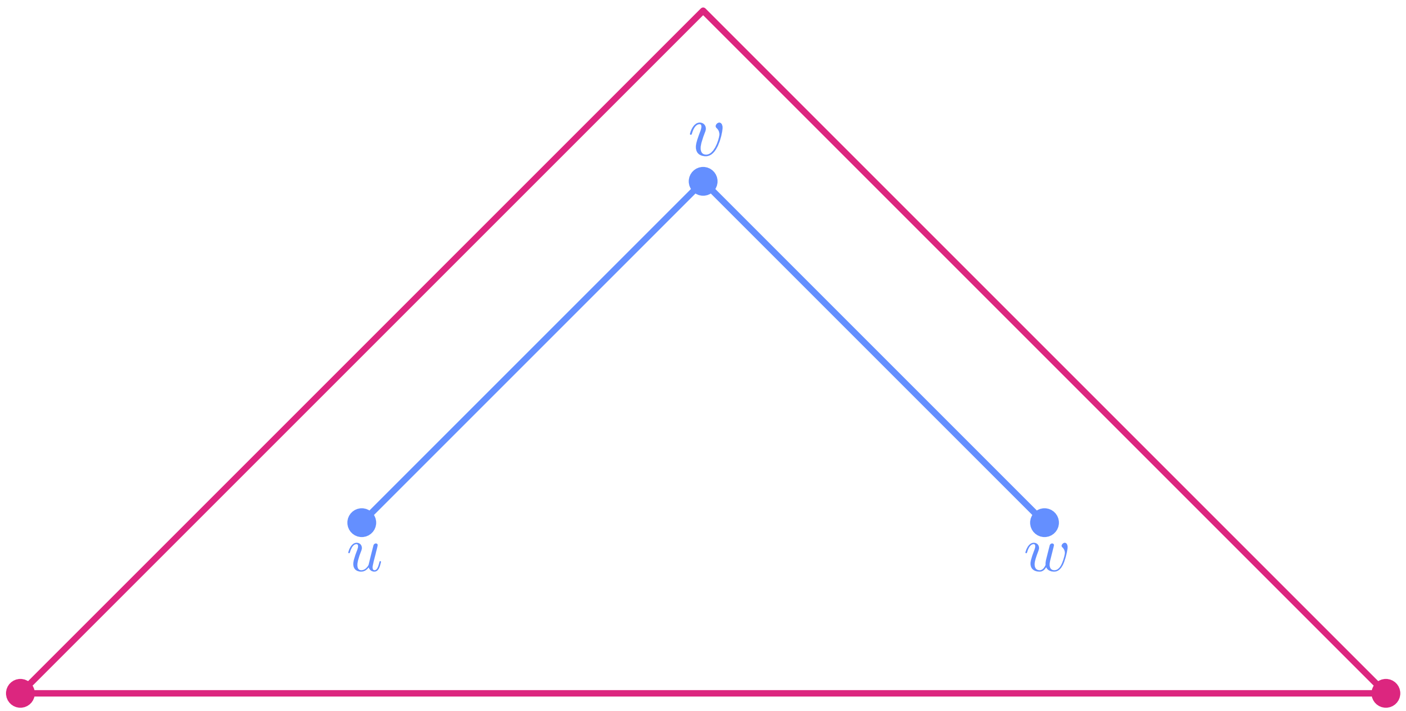
Let G_1, G_2 be immersed graphs. A *graph mapping* $s\colon G_1 \to G_2$ maps

- each vertex v of G_1 to a point $s(v)$ on an edge of G_2
- each edge $\{u, v\}$ of G_1 to a simple path from $s(u)$ to $s(v)$ in G_2

Weak graph distance

Let G_1, G_2 be immersed graphs. A *graph mapping* $s\colon G_1 \to G_2$ maps

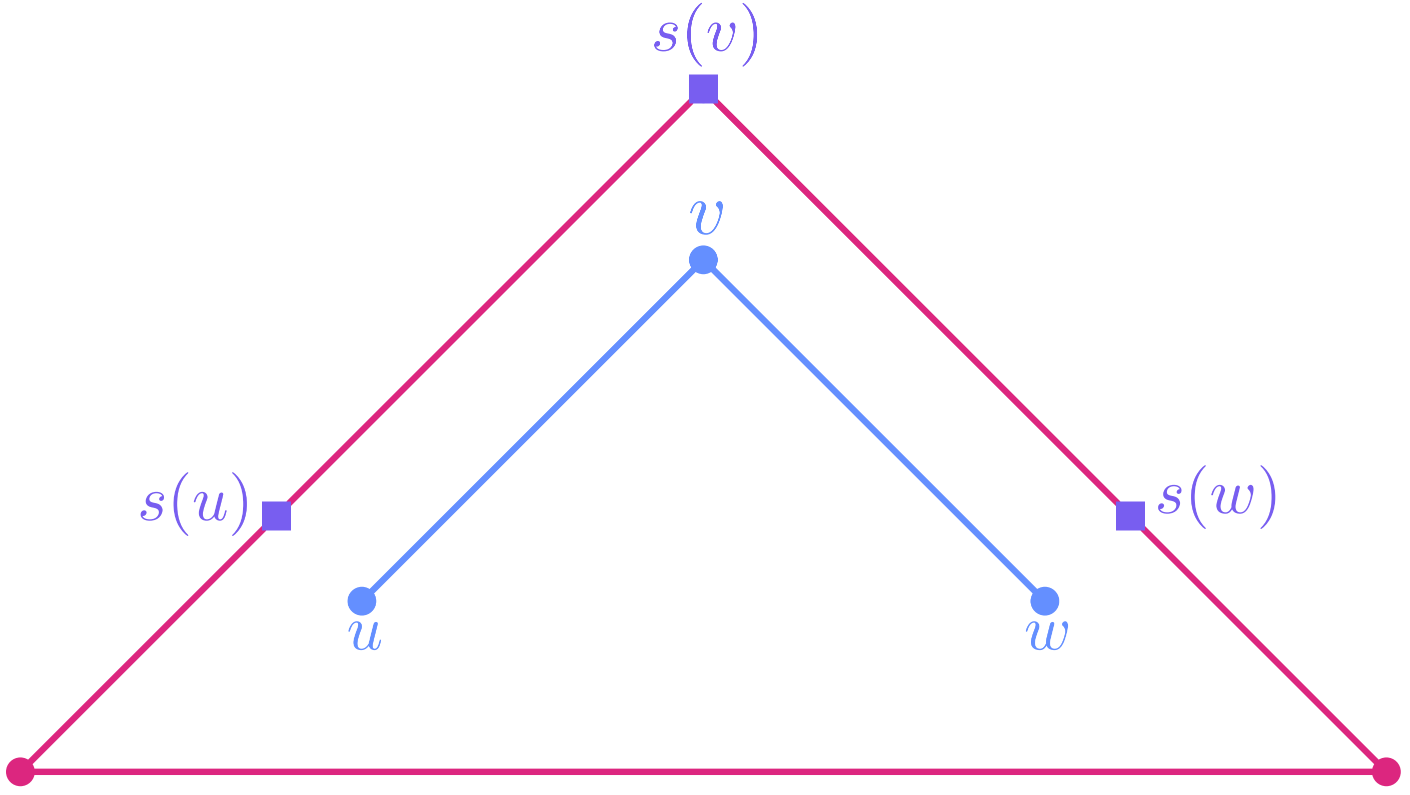
- each vertex v of G_1 to a point $s(v)$ on an edge of G_2
- each edge $\{u, v\}$ of G_1 to a simple path from $s(u)$ to $s(v)$ in G_2



Weak graph distance

Let G_1, G_2 be immersed graphs. A *graph mapping* $s \colon G_1 \to G_2$ maps

- each vertex v of G_1 to a point $s(v)$ on an edge of G_2
- each edge $\{u, v\}$ of G_1 to a simple path from $s(u)$ to $s(v)$ in G_2

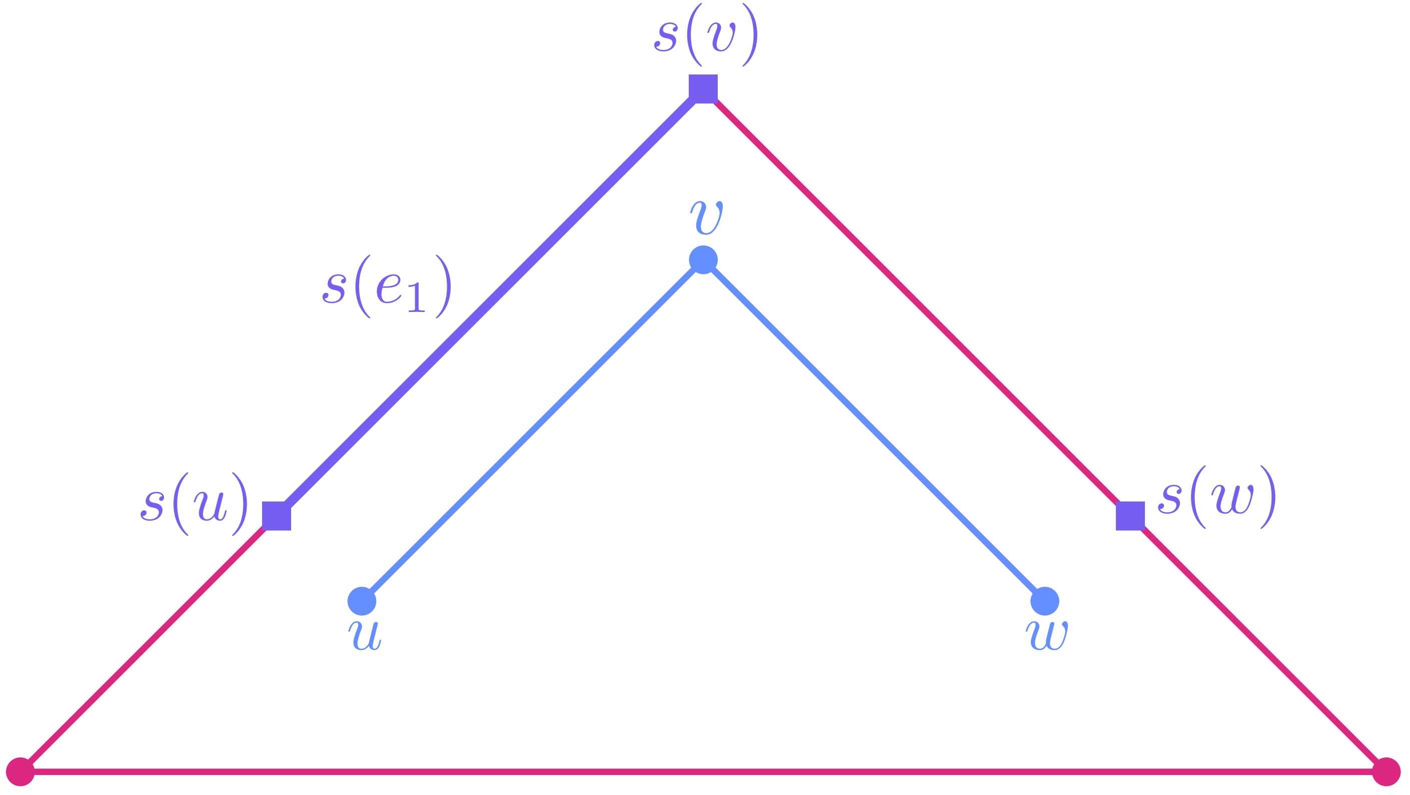


Weak graph distance

RUB

Let G_1, G_2 be immersed graphs. A *graph mapping* $s\colon G_1 \to G_2$ maps

- each vertex v of G_1 to a point $s(v)$ on an edge of G_2
- each edge $\{u, v\}$ of G_1 to a simple path from $s(u)$ to $s(v)$ in G_2

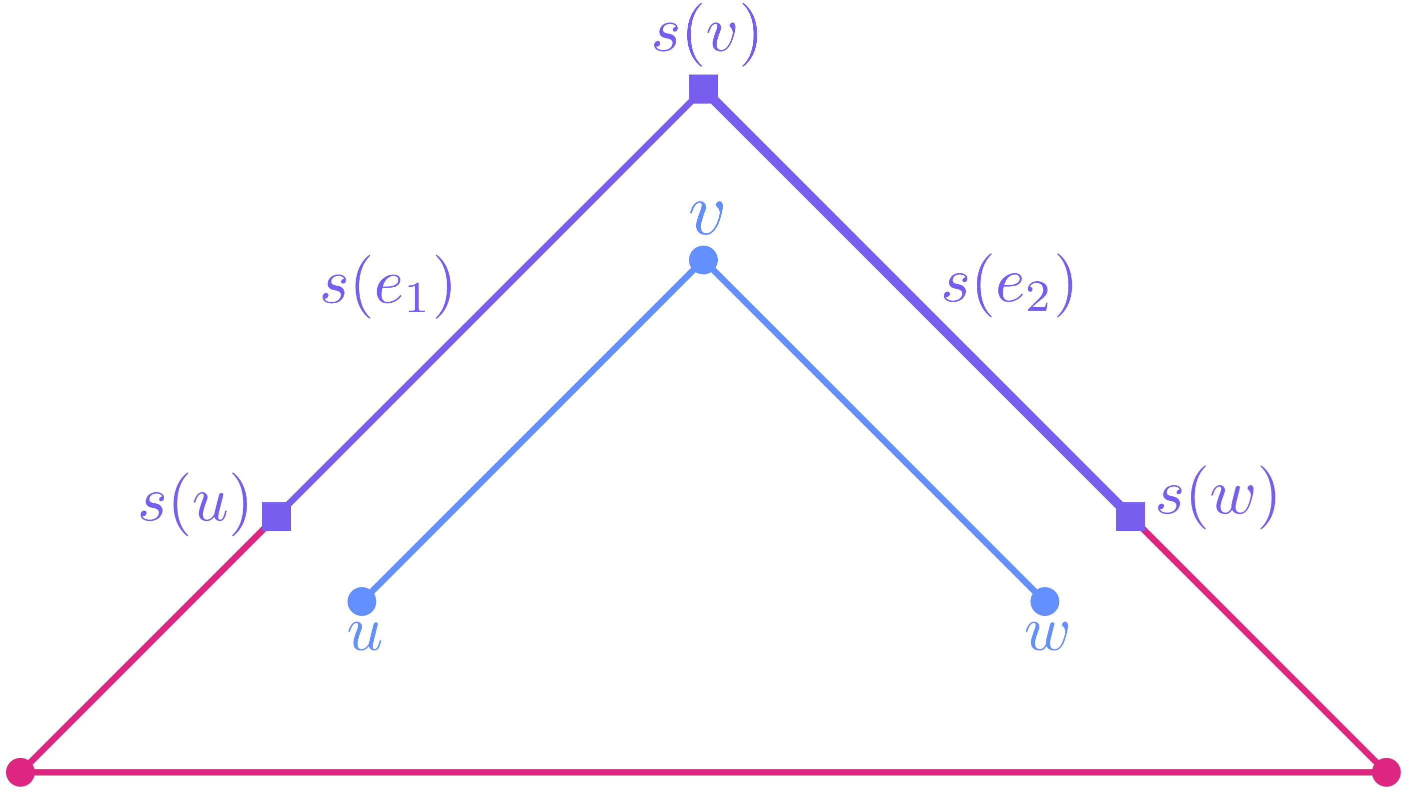


Weak graph distance

RUB

Let G_1, G_2 be immersed graphs. A *graph mapping* $s\colon G_1 \to G_2$ maps

- each vertex v of G_1 to a point $s(v)$ on an edge of G_2
- each edge $\{u, v\}$ of G_1 to a simple path from $s(u)$ to $s(v)$ in G_2



Weak graph distance

RUB

Let G_1, G_2 be immersed graphs. A *graph mapping* $s\colon G_1 \to G_2$ maps

- each vertex v of G_1 to a point $s(v)$ on an edge of G_2
- each edge $\{u, v\}$ of G_1 to a simple path from $s(u)$ to $s(v)$ in G_2

The *directed weak graph distance* from G_1 to G_2 is defined as

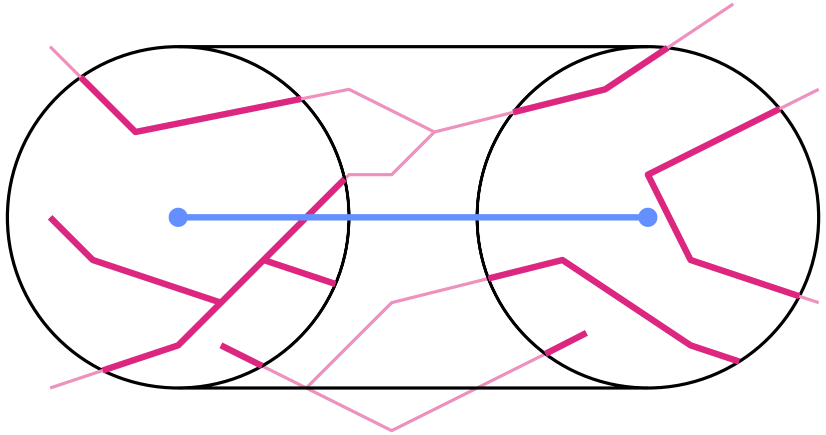
$$\vec{\delta}_{wG} = \min_{s:G_1 \to G_2} \max_{e \in E(G_1)} \delta_{wF}(e, s(e))$$

graph mapping   interpreted as curves

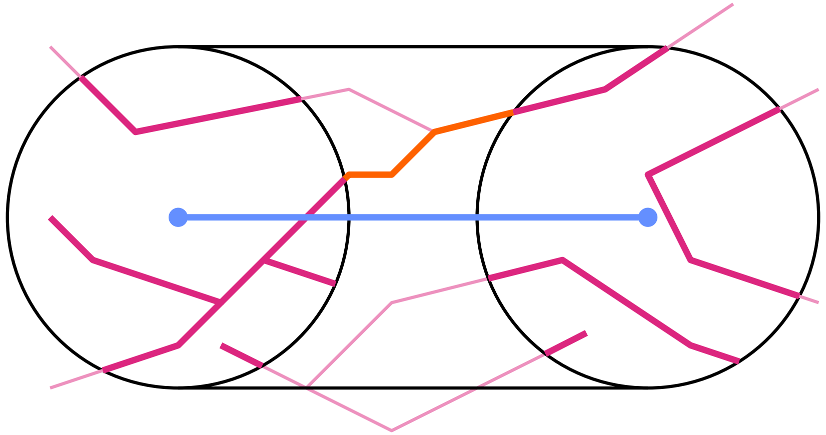
Undirected version: $\delta_{wG}(G_1, G_2) = \max\{\vec{\delta}_{wG}(G_1, G_2), \vec{\delta}_{wG}(G_2, G_1)\}$

General decision algorithm due to Akitaya et al.

Vertex placement of $v \in V(G_1)$: connected component of $G_2 \cap B_\varepsilon(v)$



General decision algorithm due to Akitaya et al.



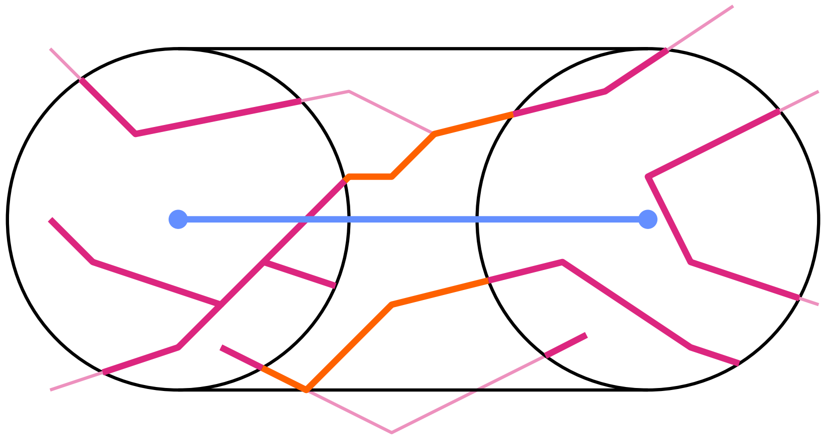
Vertex placement of $v \in V(G_1)$: connected component of $G_2 \cap B_\varepsilon(v)$

(Weak) edge placement of $e = \{u, v\} \in E(G_1)$: path P in G_2 connecting placements of u and v s.t. $\delta_{wF}(e, P) \leq \varepsilon$

General decision algorithm due to Akitaya et al.

Vertex placement of $v \in V(G_1)$: connected component of $G_2 \cap B_\varepsilon(v)$

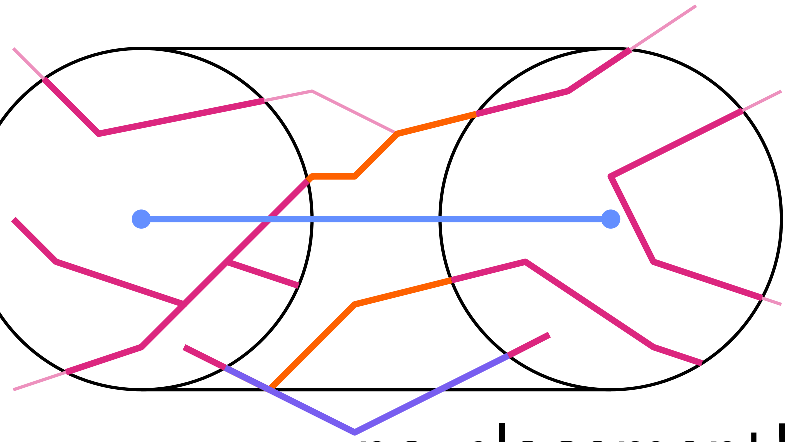
(Weak) edge placement of $e = \{u, v\} \in E(G_1)$: path P in G_2 connecting placements of u and v s.t. $\delta_{wF}(e, P) \leq \varepsilon$



General decision algorithm due to Akitaya et al.

Vertex placement of $v \in V(G_1)$: connected component of $G_2 \cap B_\varepsilon(v)$

(Weak) edge placement of $e = \{u, v\} \in E(G_1)$: path P in G_2 connecting placements of u and v s.t. $\delta_{wF}(e, P) \leq \varepsilon$

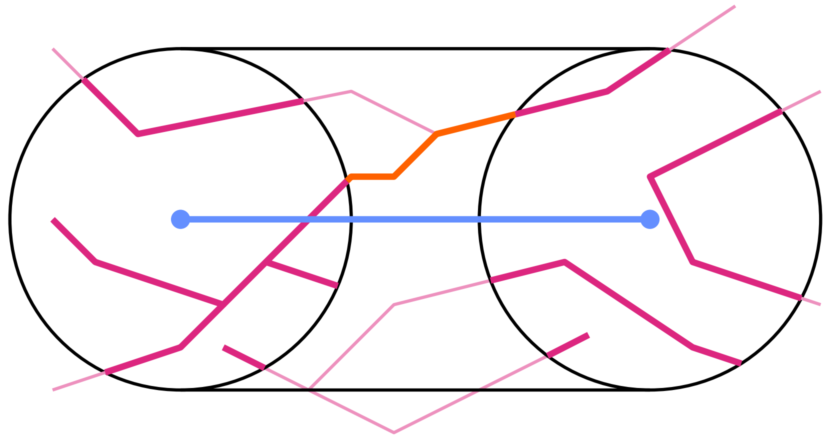


General decision algorithm due to Akitaya et al.

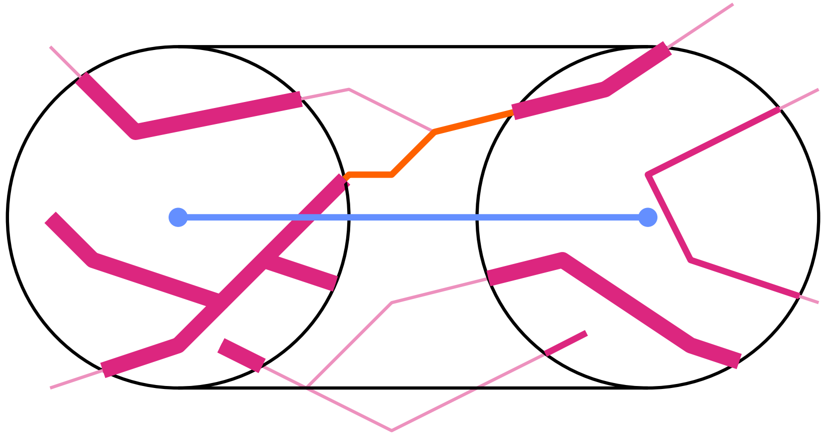
Vertex placement of $v \in V(G_1)$: connected component of $G_2 \cap B_\varepsilon(v)$

(Weak) edge placement of $e = \{u, v\} \in E(G_1)$: path P in G_2 connecting placements of u and v s.t. $\delta_{wF}(e, P) \leq \varepsilon$

(Weak) placement of G_1: graph mapping that maps each edge to a (weak) edge placement



General decision algorithm due to Akitaya et al.



Vertex placement of $v \in V(G_1)$: connected component of $G_2 \cap B_\varepsilon(v)$

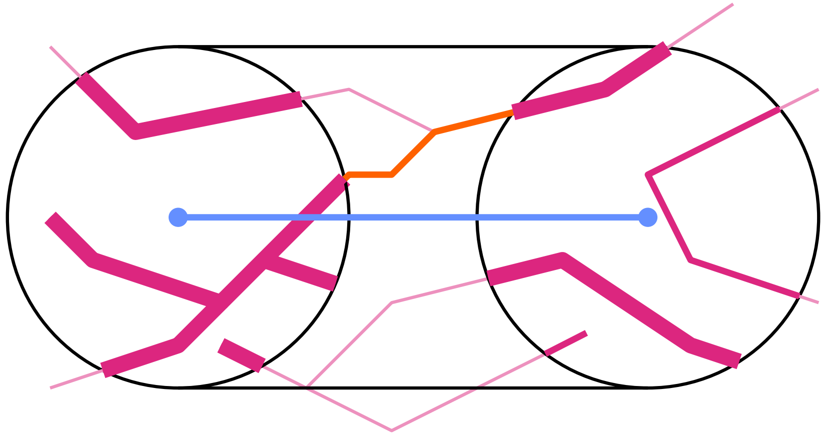
(Weak) edge placement of $e = \{u, v\} \in E(G_1)$: path P in G_2 connecting placements of u and v s.t. $\delta_{wF}(e, P) \leq \varepsilon$

(Weak) placement of G_1: graph mapping that maps each edge to a (weak) edge placement

Valid vertex placement: Is connected to a placement of each adjacent vertex through an edge placement

General decision algorithm due to Akitaya et al.

RUB



Vertex placement of $v \in V(G_1)$: connected component of $G_2 \cap B_\varepsilon(v)$

(Weak) edge placement of $e = \{u, v\} \in E(G_1)$: path P in G_2 connecting placements of u and v s.t. $\delta_{wF}(e, P) \leq \varepsilon$

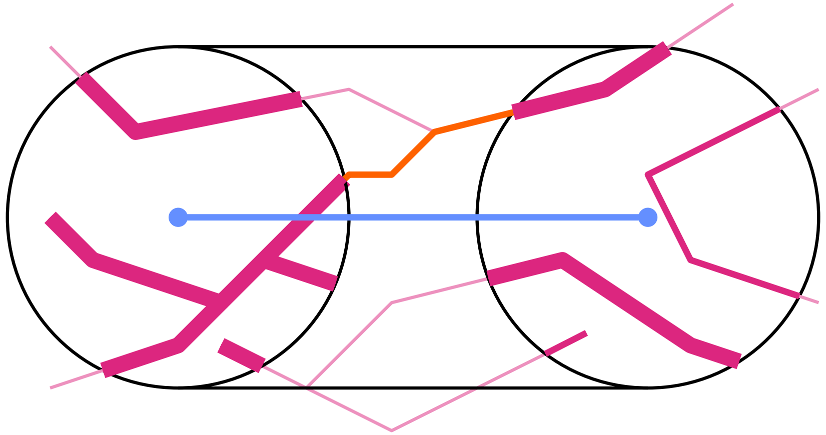
(Weak) placement of G_1: graph mapping that maps each edge to a (weak) edge placement

Valid vertex placement: Is connected to a placement of each adjacent vertex through an edge placement

General decision algorithm:

1. Compute vertex placements.
2. Compute reachability information.
3. Delete invalid placements.
4. Decide whether a placement of G_1 exists.

General decision algorithm due to Akitaya et al.



Vertex placement of $v \in V(G_1)$: connected component of $G_2 \cap B_\varepsilon(v)$

(Weak) edge placement of $e = \{u, v\} \in E(G_1)$: path P in G_2 connecting placements of u and v s.t. $\delta_{wF}(e, P) \leq \varepsilon$

(Weak) placement of G_1: graph mapping that maps each edge to a (weak) edge placement

Valid vertex placement: Is connected to a placement of each adjacent vertex through an edge placement

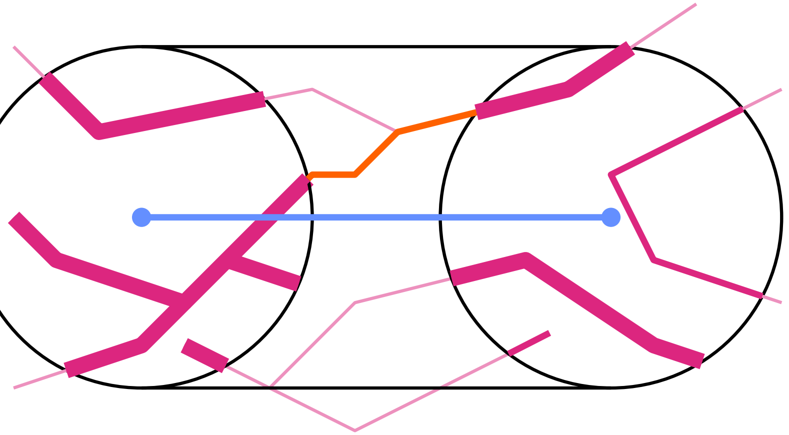
General decision algorithm:

1. Compute vertex placements.	
2. Compute reachability information.	quadratic time
3. Delete invalid placements.	
4. Decide whether a placement of G_1 exists.	NP-complete in general

General decision algorithm due to Akitaya et al.

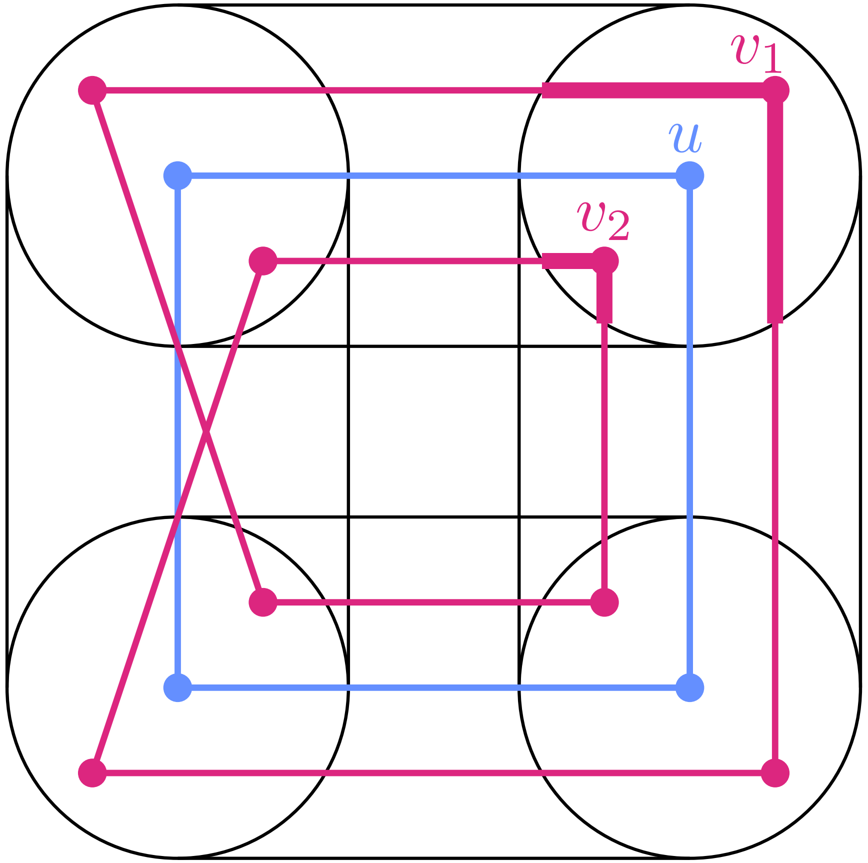
Vertex placement of $v \in V(G_1)$: connected component of $G_2 \cap B_\varepsilon(v)$

(Weak) edge placement of $e = \{u, v\} \in E(G_1)$: path P in G_2 connecting placements of u and v s.t. $\delta_{wF}(e, P) \leq \varepsilon$



(Weak) placement of G_1: graph mapping that maps each edge to a (weak) edge placement

Valid vertex placement: Is connected to a placement of each adjacent vertex through an edge placement



General decision algorithm:

1. Compute vertex placements.
2. Compute reachability information.
3. Delete invalid placements.
4. Decide whether a placement of G_1 exists.

Hardness of deciding $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ if G_1 is plane

RUB

Theorem: Deciding whether $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ is NP-complete even if G_1 is plane and G_2 is immersed in $\mathbb{R}^2$.

Hardness of deciding $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ if G_1 is plane

RUB

Theorem: Deciding whether $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ is NP-complete even if G_1 is plane and G_2 is immersed in $\mathbb{R}^2$.

Sketch of the proof by reduction from $\textsc{Planar 3Col}$:

Let $G = (V, E)$ be the (planar) input graph

Hardness of deciding $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ if G_1 is plane

RUB

Theorem: Deciding whether $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ is NP-complete even if G_1 is plane and G_2 is immersed in $\mathbb{R}^2$.

Sketch of the proof by reduction from $\textsc{Planar 3Col}$:

Let $G = (V, E)$ be the (planar) input graph

1. Construct a crossing-free embedding of G

Hardness of deciding $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ if G_1 is plane

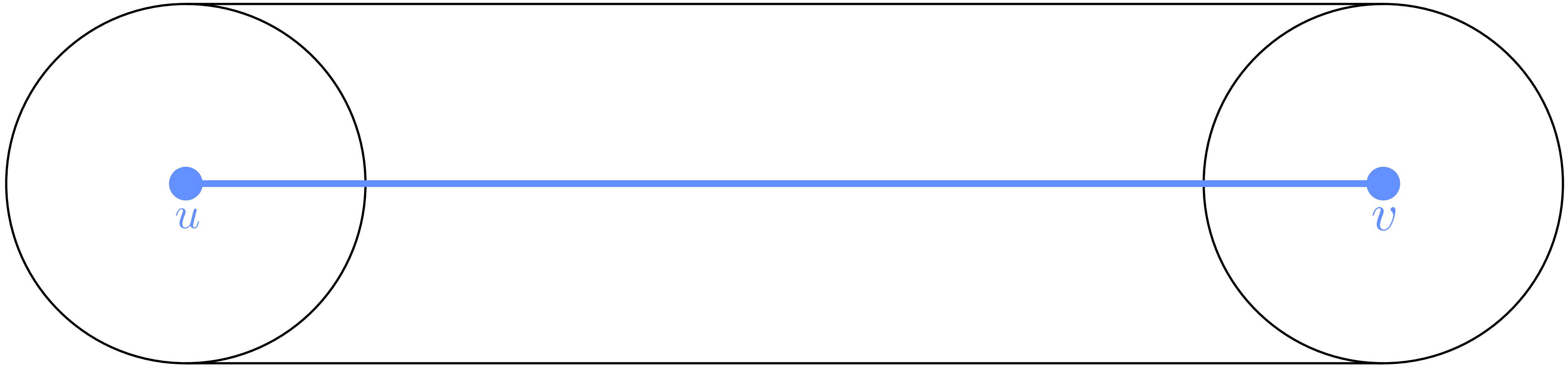
RUB

Theorem: Deciding whether $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ is NP-complete even if G_1 is plane and G_2 is immersed in $\mathbb{R}^2$.

Sketch of the proof by reduction from Planar 3Col:

Let $G = (V, E)$ be the (planar) input graph

2. Choose ε s.t. all ε-balls and tubes are separated



Hardness of deciding $\vec{\delta}_{wG}(G_1, G_2) \le \varepsilon$ if G_1 is plane

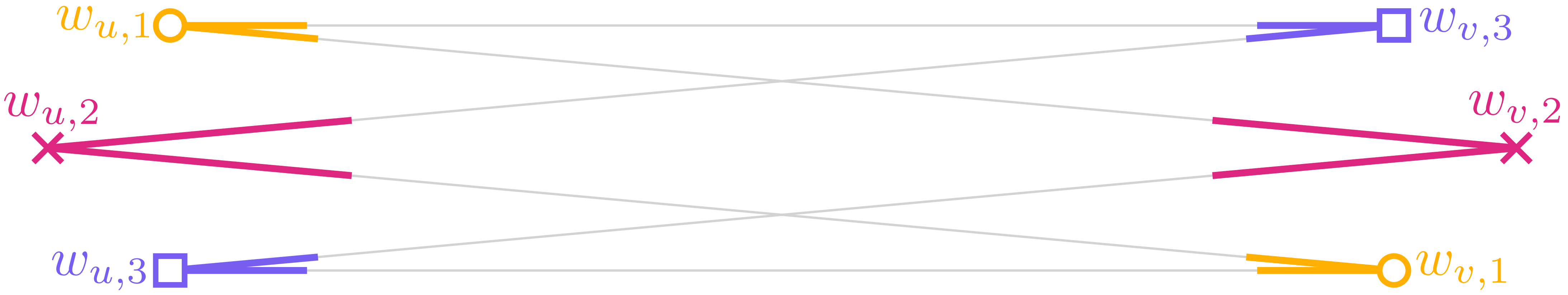
RUB

Theorem: Deciding whether $\vec{\delta}_{wG}(G_1, G_2) \le \varepsilon$ is NP-complete even if G_1 is plane and G_2 is immersed in $\mathbb{R}^2$.

Sketch of the proof by reduction from PLANAR 3COL:

Let $G = (V, E)$ be the (planar) input graph

3. Construct G_c with
   - vertices $w_{u,i}$ for $u \in V$, $i \in [3]$
   - edges $\{w_{u,i}, w_{v,j}\}$ for $\{u, v\} \in E$, $i \neq j$



Hardness of deciding $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ if G_1 is plane

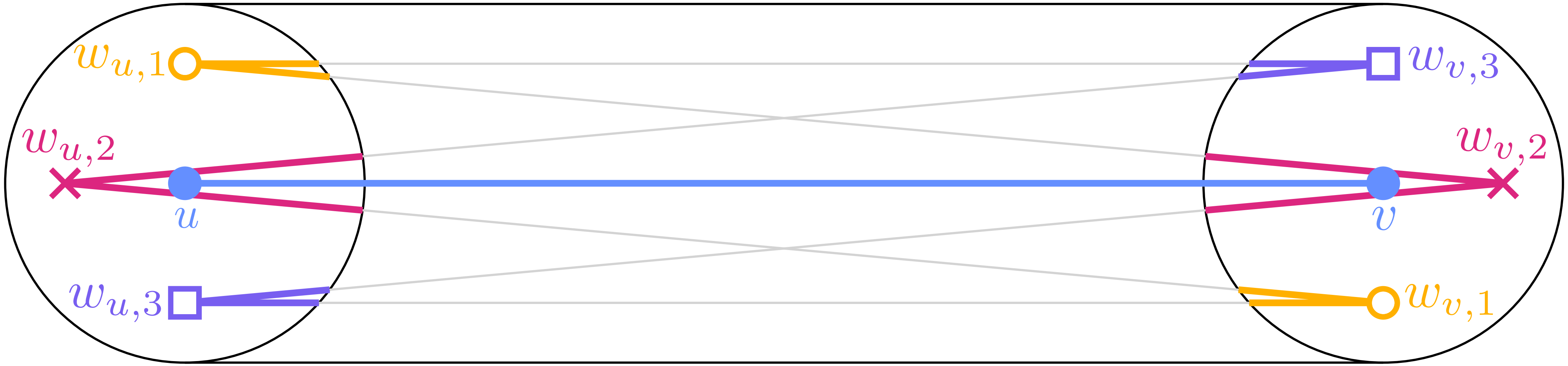
RUB

Theorem: Deciding whether $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ is NP-complete even if G_1 is plane and G_2 is immersed in $\mathbb{R}^2$.

Sketch of the proof by reduction from Planar 3Col:

Let $G = (V, E)$ be the (planar) input graph

4. Immerse G_c s.t. $w_{u,i}$ lies in $B_\varepsilon(u)$



Hardness of deciding $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ if G_1 is plane

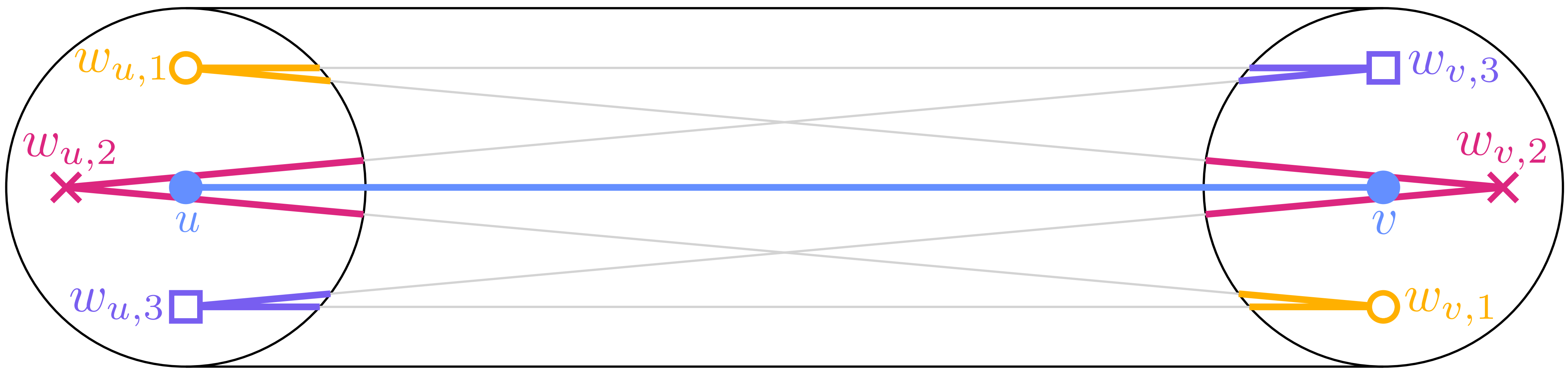
RUB

Theorem: Deciding whether $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ is NP-complete even if G_1 is plane and G_2 is immersed in $\mathbb{R}^2$.

Sketch of the proof by reduction from PLANAR 3COL:

Let $G = (V, E)$ be the (planar) input graph

Idea: Vertex placements $\leftrightarrow$ colors



Hardness of deciding $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ if G_1 is plane

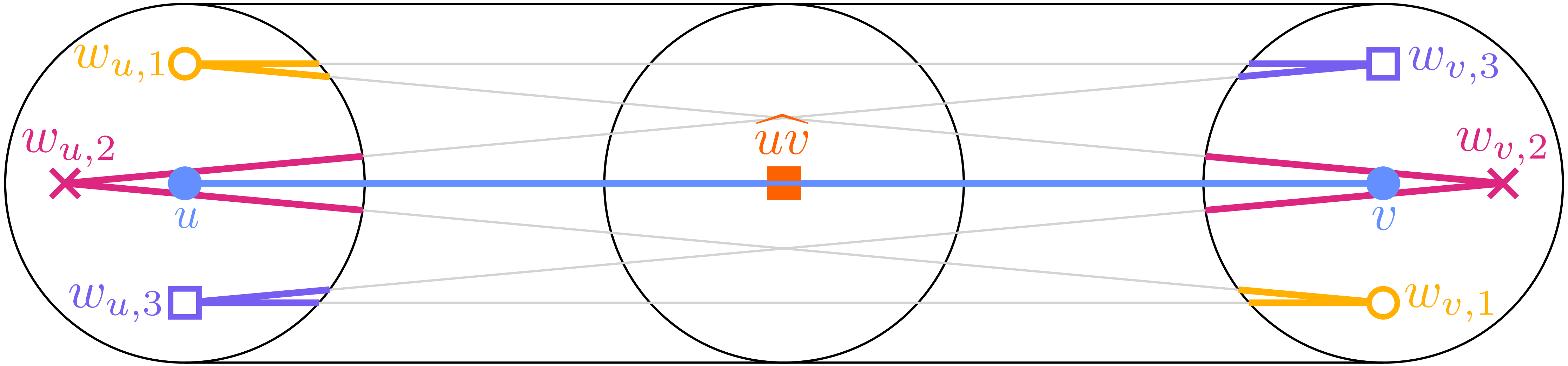
RUB

Theorem: Deciding whether $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ is NP-complete even if G_1 is plane and G_2 is immersed in $\mathbb{R}^2$.

Sketch of the proof by reduction from Planar 3Col:

Let $G = (V, E)$ be the (planar) input graph

$\tilde{1}$. Construct a crossing-free embedding of G and insert a vertex $\widehat{uv}$ in the middle of each edge $\{u, v\}$ $\quad \rightarrow G_p$



Hardness of deciding $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ if G_1 is plane

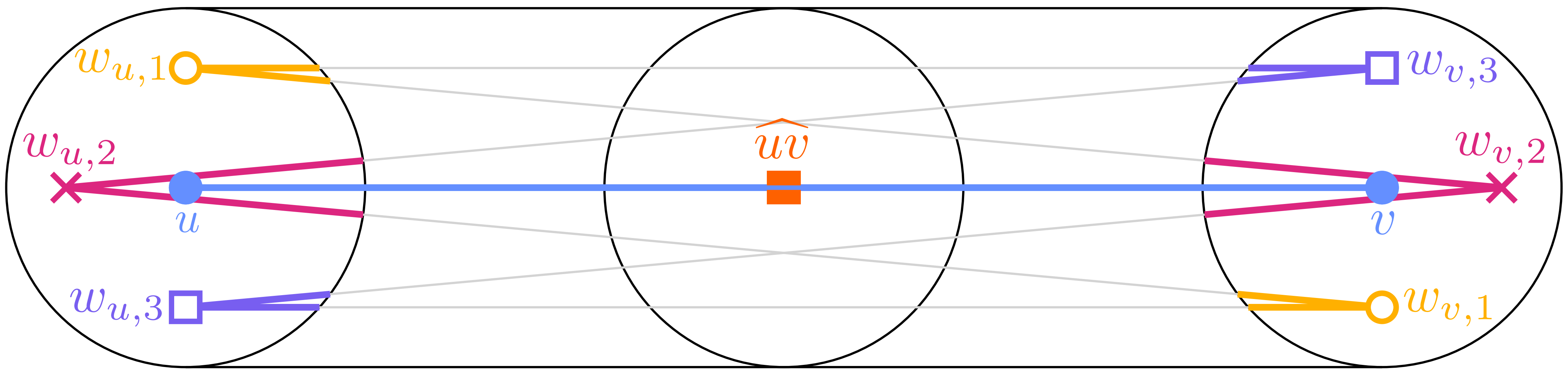
RUB

Theorem: Deciding whether $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ is NP-complete even if G_1 is plane and G_2 is immersed in $\mathbb{R}^2$.

Sketch of the proof by reduction from $\textsc{Planar 3Col}$:

Let $G = (V, E)$ be the (planar) input graph

$\rightarrow$ Each $\{u, v\}$ must be placed through some $\{w_{u,i}, w_{v,j}\}$



Hardness of deciding $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ if G_1 is plane

RUB

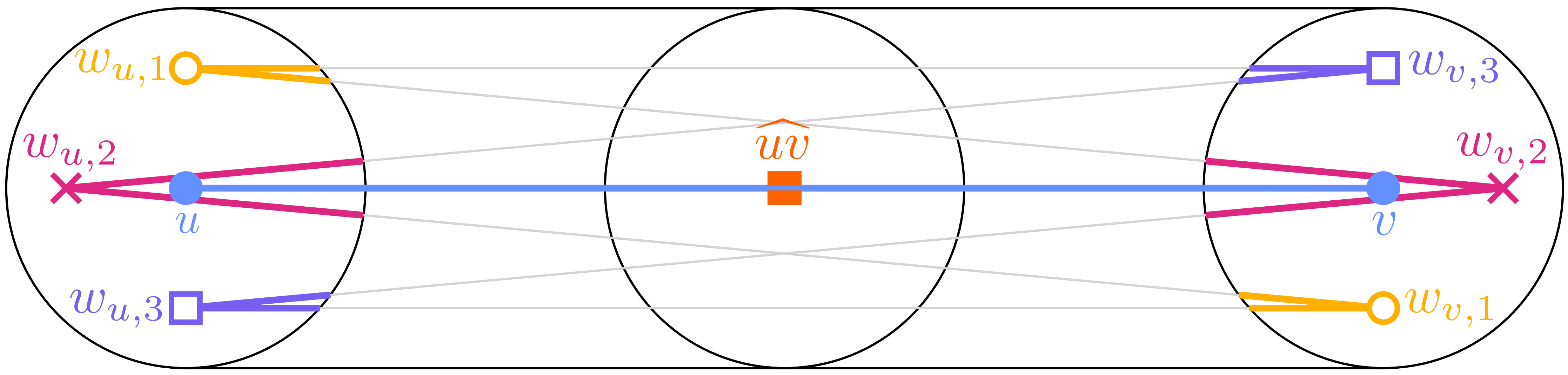
Theorem: Deciding whether $\vec{\delta}_{wG}(G_1, G_2) \leq \varepsilon$ is NP-complete even if G_1 is plane and G_2 is immersed in $\mathbb{R}^2$.

Sketch of the proof by reduction from PLANAR 3COL:

Let $G = (V, E)$ be the (planar) input graph

$\rightarrow$ Each $\{u, v\}$ must be placed through some $\{w_{u,i}, w_{v,j}\}$

$\rightarrow$ Consistent graph mapping $\leftrightarrow$ consistent 3-coloring



Hardness of approximating $\vec{\delta}_{wG}(G_1, G_2)$ if G_1 is plane

RUB

Corollary: The weak graph distance is NP-hard to approximate within any constant ratio $c \geq 1$ even if G_1 is plane, G_2 is immersed in $\mathbb{R}^2$.

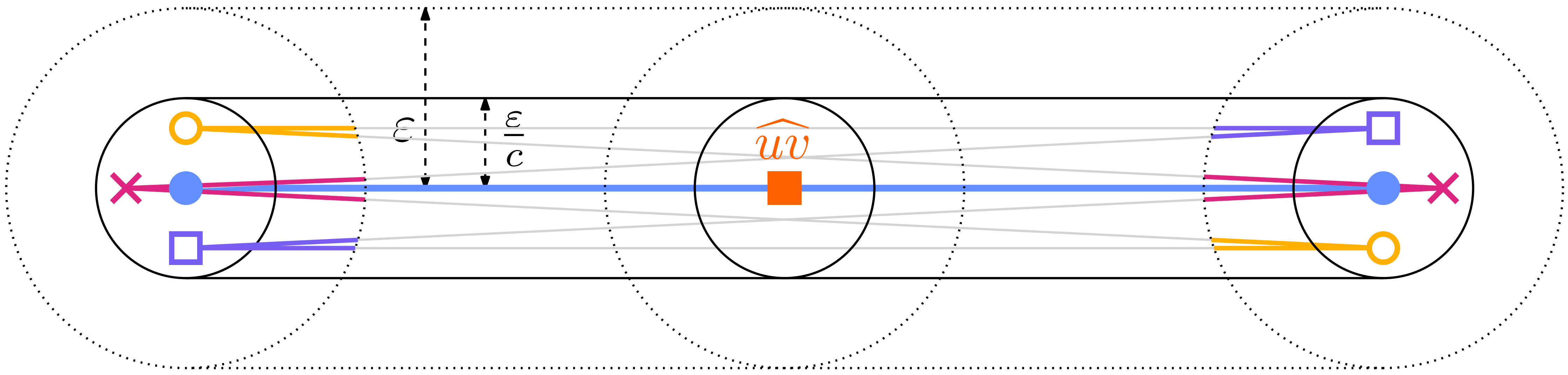
Hardness of approximating $\vec{\delta}_{wG}(G_1, G_2)$ if G_1 is plane

RUB

Corollary: The weak graph distance is NP-hard to approximate within any constant ratio $c \geq 1$ even if G_1 is plane, G_2 is immersed in $\mathbb{R}^2$.

Proof idea:

Immerse G_c within $\frac{\varepsilon}{c}$-balls instead



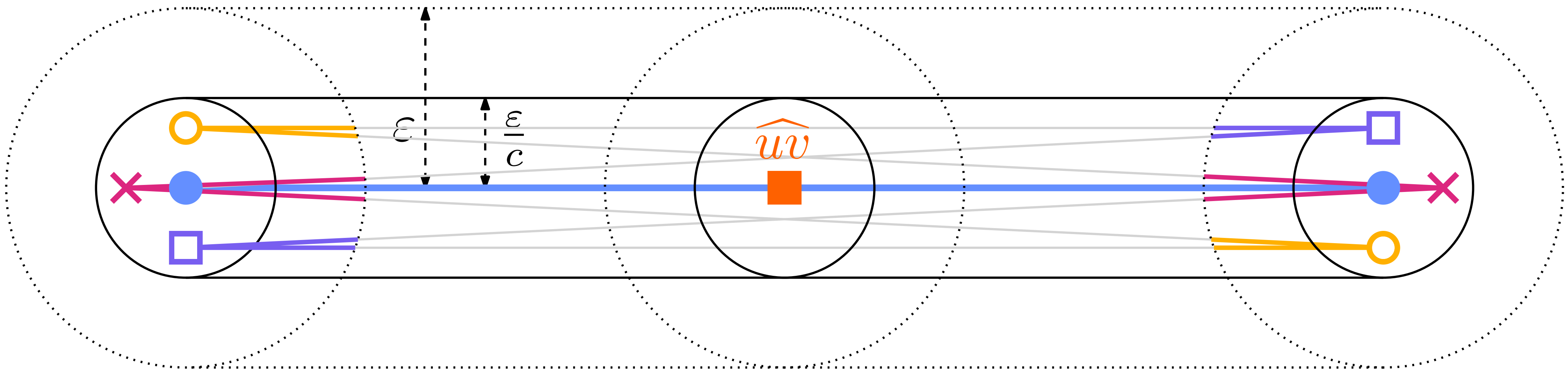
Hardness of approximating $\vec{\delta}_{wG}(G_1, G_2)$ if G_1 is plane

Corollary: The weak graph distance is NP-hard to approximate within any constant ratio $c \geq 1$ even if G_1 is plane, G_2 is immersed in $\mathbb{R}^2$.

Proof idea:

Immerse G_c within $\frac{\varepsilon}{c}$-balls instead

Positive case: upper bound becomes $\vec{\delta}_{wG}(G_p, G_c) \leq \frac{\varepsilon}{c}$



Hardness of approximating $\vec{\delta}_{wG}(G_1, G_2)$ if G_1 is plane

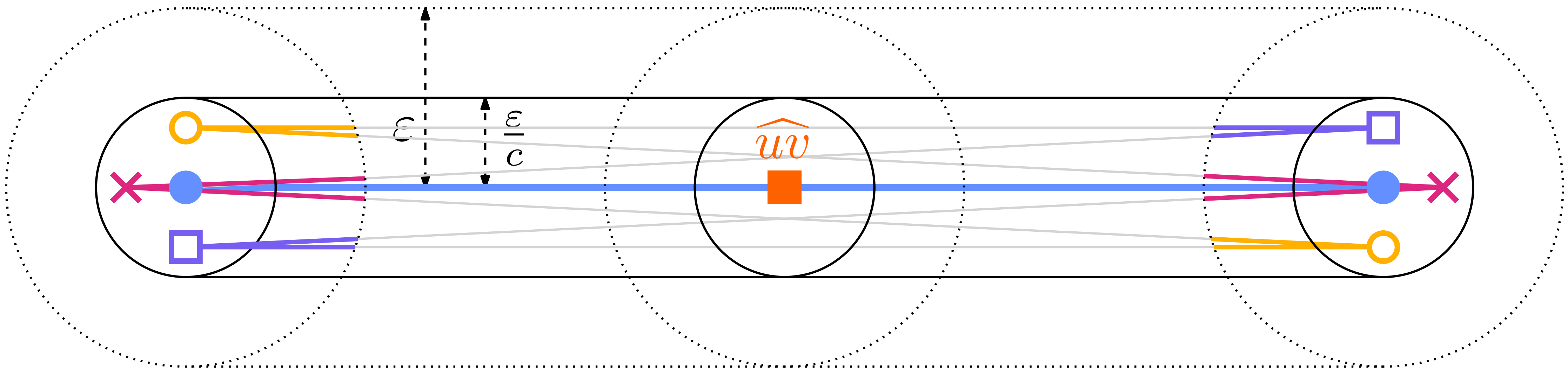
Corollary: The weak graph distance is NP-hard to approximate within any constant ratio $c \geq 1$ even if G_1 is plane, G_2 is immersed in $\mathbb{R}^2$.

Proof idea:

Immerse G_c within $\frac{\varepsilon}{c}$-balls instead

Positive case: upper bound becomes $\vec{\delta}_{wG}(G_p, G_c) \leq \frac{\varepsilon}{c}$

Negative case: Lower bound of ε remains intact



Hardness of the embedded case in $\mathbb{R}^d$, $d \geq 3$

RUB

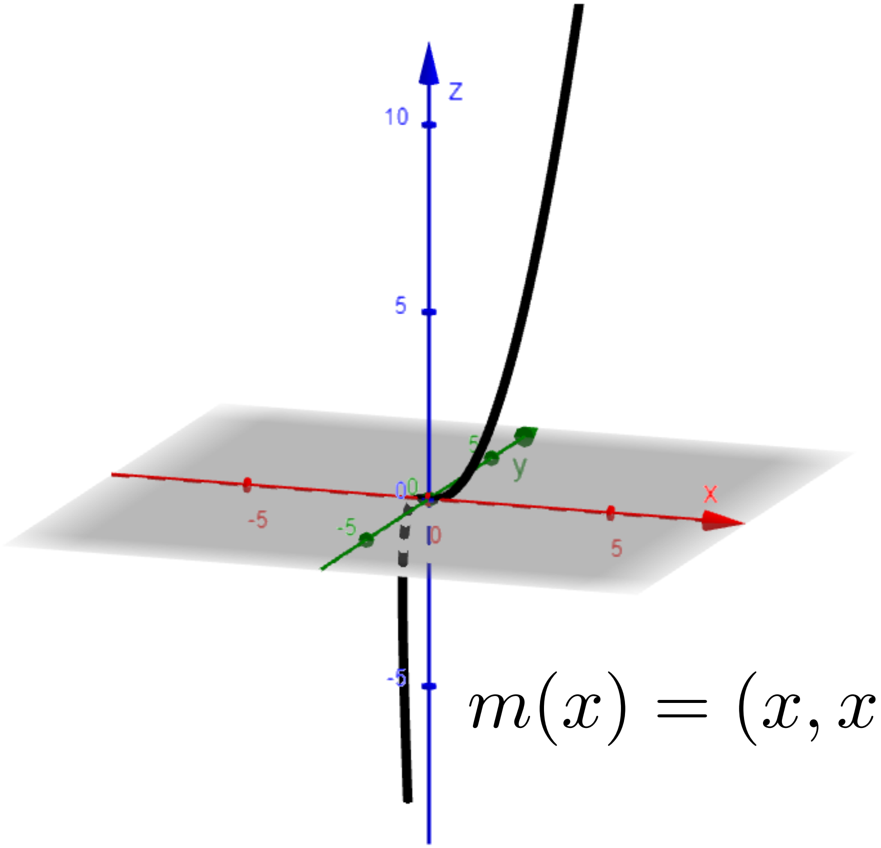
Theorem: The weak graph distance is NP-hard to approximate within any constant ratio $c \geq 1$ if G_1, G_2 are embedded in $\mathbb{R}^d$ for any $d \geq 3$.

Hardness of the embedded case in $\mathbb{R}^d$, $d \geq 3$

Theorem: The weak graph distance is NP-hard to approximate within any constant ratio $c \geq 1$ if G_1, G_2 are embedded in $\mathbb{R}^d$ for any $d \geq 3$.

Proof idea:

(Up to details,) embed on the 3-dim moment curve instead



$$m(x) = (x, x^2, x^3)$$

Crossing-rigid weak graph distances

RUB

WGD has nice properties, but is NP-complete even in the 2-dimensional case

Crossing-rigid weak graph distances

WGD has nice properties, but is NP-complete even in the 2-dimensional case

$\rightarrow$ currently unknown whether WGD is FPT when parameterized in $\#$crossings

Crossing-rigid weak graph distances

WGD has nice properties, but is NP-complete even in the 2-dimensional case

$\rightarrow$ currently unknown whether WGD is FPT when parameterized in #crossings

Now: crossing-rigid weak graph distances

Crossing-rigid weak graph distances

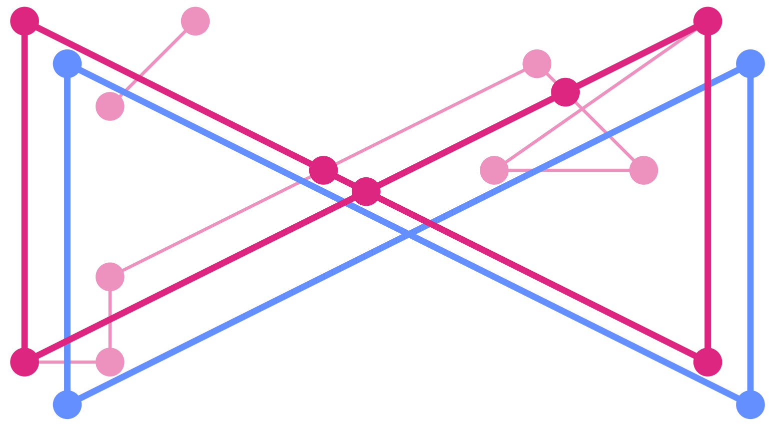
Intuition: map crossings onto crossings

Crossing-rigid weak graph distances

Intuition: map crossings onto crossings

(Informal) Definition: A graph mapping that maps each edge with n crossings to a path containing

- up to n crossings is *loosely crossing-rigid*
- at least one* and at most n crossings is *crossing-rigid*
- exactly n crossings is *strictly crossing-rigid*

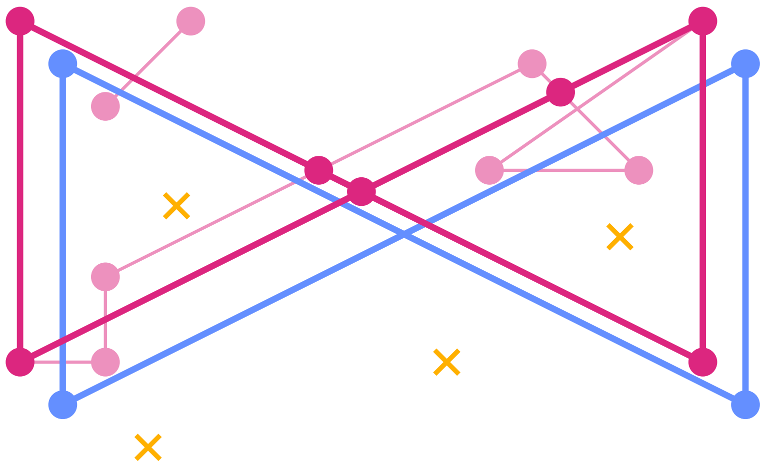


Crossing-rigid weak graph distances

Intuition: map crossings onto crossings

(Informal) Definition: A graph mapping that maps each edge with n crossings to a path containing

- up to n crossings is *loosely crossing-rigid*
- at least one* and at most n crossings is *crossing-rigid*
- exactly n crossings is *strictly crossing-rigid*

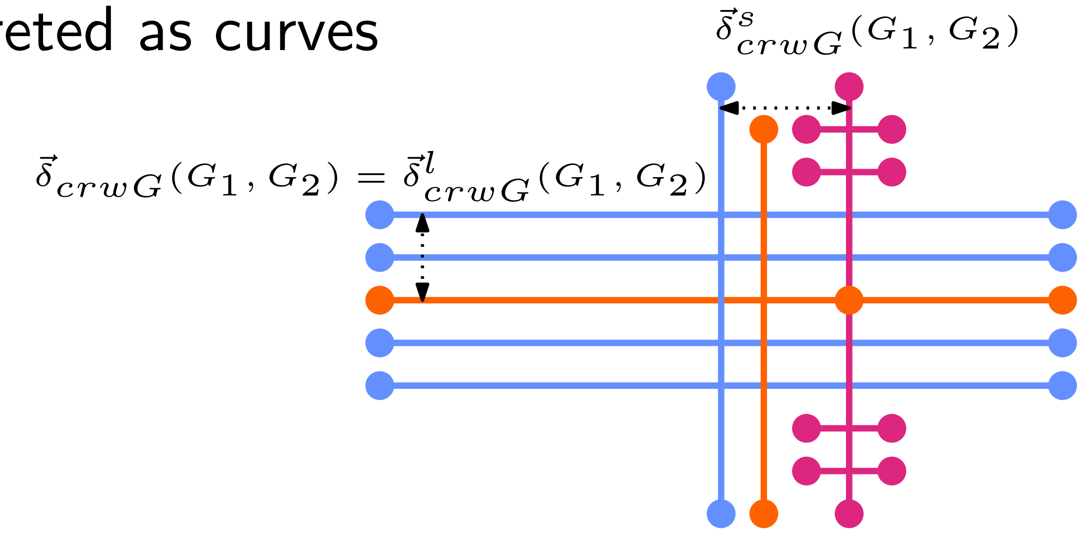


The *directed crossing-rigid weak graph distance* is defined as

$$\vec{\delta}_{crwG}(G_1, G_2) = \inf_{s: G_1 \to G_2} \max_{e \in E} \delta_{wF}(e, s(e))$$

crossing-rigid interpreted as curves

Analogous: $\vec{\delta}^{l}_{crwG}$, $\vec{\delta}^{s}_{crwG}$



Decision algorithm for the existence of crossing-rigid placements

RUB

Intuition: assign crossings and check whether the fixed assignment allows WGD $\leq \varepsilon$

Decision algorithm for the existence of crossing-rigid placements

Intuition: assign crossings and check whether the fixed assignment allows WGD $\leq \varepsilon$

1. Compute crossings.
2. Compute vertex placements.
3. **for each** valid* assignment of crossings of G_2 to crossings of G_1:
4. Compute reachability information (under assignment).
5. Delete invalid placements.
6. **if** there exists a placement for G_1, **return** true.
7. **return** false.

Decision algorithm for the existence of crossing-rigid placements

RUB

Intuition: assign crossings and check whether the fixed assignment allows WGD $\leq \varepsilon$

1. Compute crossings. 2. Compute vertex placements.	polynomial time
3. **for each** valid* assignment of crossings of G_2 to crossings of G_1:	FPT-like bound
4. Compute reachability information (under assignment). 5. Delete invalid placements.	polynomial time
6. **if** there exists a placement for G_1, **return** true.	Conj.: linear*
7. **return** false.	polynomial time

Decision algorithm for the existence of crossing-rigid placements

Intuition: assign crossings and check whether the fixed assignment allows WGD $\leq \varepsilon$

1.	Compute crossings.	polynomial time
2.	Compute vertex placements.	
3.	**for each** valid* assignment of crossings of G_2 to crossings of G_1:	FPT-like bound
4.	Compute reachability information (under assignment).	polynomial time
5.	Delete invalid placements.	
6.	**if** there exists a placement for G_1, **return** true.	Conj.: linear*
7.	**return** false.	polynomial time

Caveat: since placements are no longer compact, there might be no ε-placement but still $\delta_{crwG}^{(l)/(r)}(G_1, G_2) = \varepsilon$

Outlook

RUB

We have seen:

- $\vec{\delta}_{wG}(G_1, G_2)$ is NP-hard to approximate up to c even if G_1 is plane
- $\vec{\delta}_{wG}(G_1, G_2)$ is NP-hard to approximate for G_1, G_2 embedded in $\mathbb{R}^d$, $d \geq 3$
- The CRWGDs could be good alternatives for graphs immersed in $\mathbb{R}^2$

Outlook

RUB

We have seen:

- $\vec{\delta}_{wG}(G_1, G_2)$ is NP-hard to approximate up to c even if G_1 is plane
- $\vec{\delta}_{wG}(G_1, G_2)$ is NP-hard to approximate for G_1, G_2 embedded in $\mathbb{R}^d$, $d \geq 3$
- The CRWGDs could be good alternatives for graphs immersed in $\mathbb{R}^2$

Natural next steps:

- verify conjecture on the tractability of crossing-rigid weak graph distances
- (metric) properties of crossing-rigid distances
- research the parameterized complexity of the unmodified WGD and the complexity of our measures without regularity conditions
- experimental evaluations of the different distance measures

Outlook

RUB

We have seen:

- $\vec{\delta}_{wG}(G_1, G_2)$ is NP-hard to approximate up to c even if G_1 is plane
- $\vec{\delta}_{wG}(G_1, G_2)$ is NP-hard to approximate for G_1, G_2 embedded in $\mathbb{R}^d$, $d \geq 3$
- The CRWGDs could be good alternatives for graphs immersed in $\mathbb{R}^2$

Natural next steps:

- verify conjecture on the tractability of crossing-rigid weak graph distances
- (metric) properties of crossing-rigid distances
- research the parameterized complexity of the unmodified WGD and the complexity of our measures without regularity conditions
- experimental evaluations of the different distance measures

Thank you