Fixed-Parameter Algorithms for Computing RAC Drawings of Graphs

Cornelius Brand · Robert Ganian · **Sebastian Röder** · **Florian Schager**

22.09.2023 · GD '23

TU WIEN
ac ALGORITHMS AND COMPLEXITY GROUP

Right-angle crossing (RAC) drawings

- **Problem:** Crossings in non-planar graphs impair readability
- Empirical studies show that sharp angles are problematic [Huang et al. 2008]

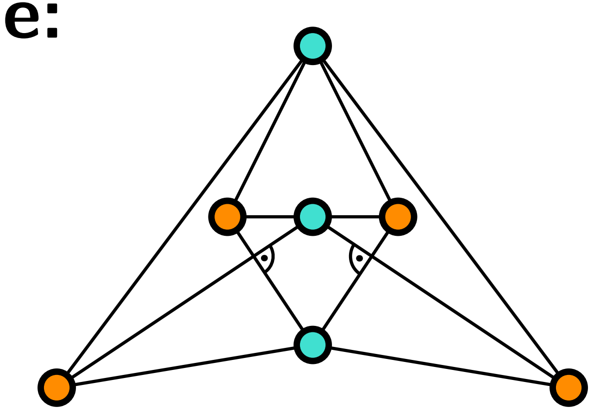
Right-angle crossing (RAC) drawings

- **Problem:** Crossings in non-planar graphs impair readability
- Empirical studies show that sharp angles are problematic [Huang et al. 2008]
- **Solution:** Allow crossings only at right angles

Right-angle crossing (RAC) drawings

- **Problem:** Crossings in non-planar graphs impair readability
- Empirical studies show that sharp angles are problematic [Huang et al. 2008]
- **Solution:** Allow crossings only at right angles

Example:

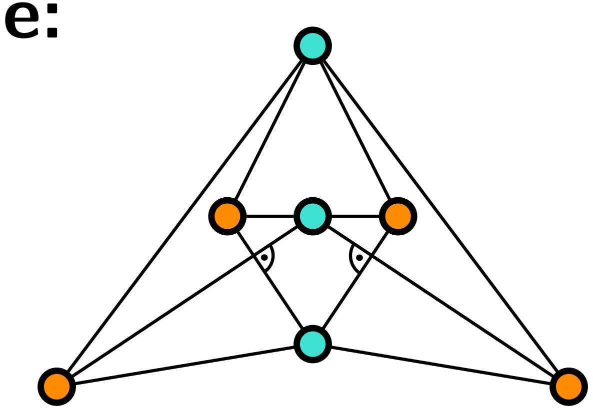


RAC drawing of $K_{3,4}$

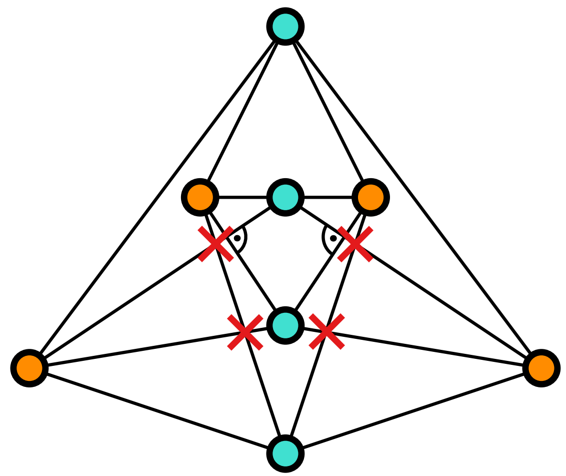
Right-angle crossing (RAC) drawings

- **Problem:** Crossings in non-planar graphs impair readability
- Empirical studies show that sharp angles are problematic [Huang et al. 2008]
- **Solution:** Allow crossings only at right angles

Example:



RAC drawing of $K_{3,4}$



$K_{4,4}$ does not admit a RAC drawing

Generalized RAC drawings

RAC DRAWING

Instance: A graph G

Question: Does G admit a straight-line RAC drawing?

Generalized RAC drawings

RAC DRAWING

Instance: A graph G

Question: Does G admit a straight-line RAC drawing?

Theorem: Deciding whether a graph G admits a RAC drawing is NP-hard

[Argyriou et al. 2010]

Generalized RAC drawings

β-Bend RAC DRAWING

Instance: A graph G, **an integer** $\beta \in \{0, 1, 2, 3\}$

Question: Does G admit a RAC drawing **with** β **bends per edge**?

Generalized RAC drawings

β**-Bend** RAC DRAWING

Instance: A graph G, **an integer** $\beta \in \{0, 1, 2, 3\}$

Question: Does G admit a RAC drawing **with** β **bends per edge**?

0-bends: $m \leq 4n - 10$

[Didimo, Eades and Liotta 2011]

NP-hard

Generalized RAC drawings

β-**Bend** RAC DRAWING

Instance: A graph G, **an integer** $\beta \in \{0, 1, 2, 3\}$

Question: Does G admit a RAC drawing **with** β **bends per edge**?

0-bends: $m \leq 4n - 10$

[Didimo, Eades and Liotta 2011]

NP-hard

1-bends: $m \leq 5.5n - \mathcal{O}(1)$

[Angelini, Bekos, Förster and Kaufmann 2018]

2-bends: $m \leq 74.2n$

[Arikushi, Fulek, Keszegh, Morić and Tóth 2012]

Complexity unknown

Generalized RAC drawings

β-Bend RAC DRAWING

Instance: A graph G, **an integer** $\beta \in \{0, 1, 2, 3\}$

Question: Does G admit a RAC drawing **with** β **bends per edge**?

0-bends: $m \leq 4n - 10$

[Didimo, Eades and Liotta 2011]

NP-hard

1-bends: $m \leq 5.5n - \mathcal{O}(1)$

[Angelini, Bekos, Förster and Kaufmann 2018]

2-bends: $m \leq$ ~~$74.2n$~~ $24n - 26$

[Arikushi, Fulek, Keszegh, Morić and Tóth 2012]

Complexity unknown

Generalized RAC drawings

β-**Bend** RAC DRAWING

Instance: A graph G, **an integer** $\beta \in \{0, 1, 2, 3\}$

Question: Does G admit a RAC drawing **with** β **bends per edge**?

0-bends: $m \leq 4n - 10$

[Didimo, Eades and Liotta 2011]

NP-hard

1-bends: $m \leq 5.5n - \mathcal{O}(1)$

[Angelini, Bekos, Förster and Kaufmann 2018]

2-bends: $m \leq$ ~~$74.2n$~~ ~~$24n - 26$~~ $20n - 22$

[Arikushi, Fulek, Keszegh, Morić and Tóth 2012]

Complexity unknown

Generalized RAC drawings

β-Bend RAC DRAWING

Instance: A graph G, **an integer** $\beta \in \{0, 1, 2, 3\}$

Question: Does G admit a RAC drawing **with** β **bends per edge**?

0-bends: $m \leq 4n - 10$
[Didimo, Eades and Liotta 2011]
NP-hard

1-bends: $m \leq 5.5n - \mathcal{O}(1)$
[Angelini, Bekos, Förster and Kaufmann 2018]
2-bends: $m \leq$ ~~$74.2n$~~ ~~$24n - 26$~~ $20n - 22$
[Arikushi, Fulek, Keszegh, Morić and Tóth 2012]
Complexity unknown

3-bends: always drawable
[Didimo, Eades and Liotta 2011]
linear time

Generalized RAC drawings

β-Bend RAC DRAWING

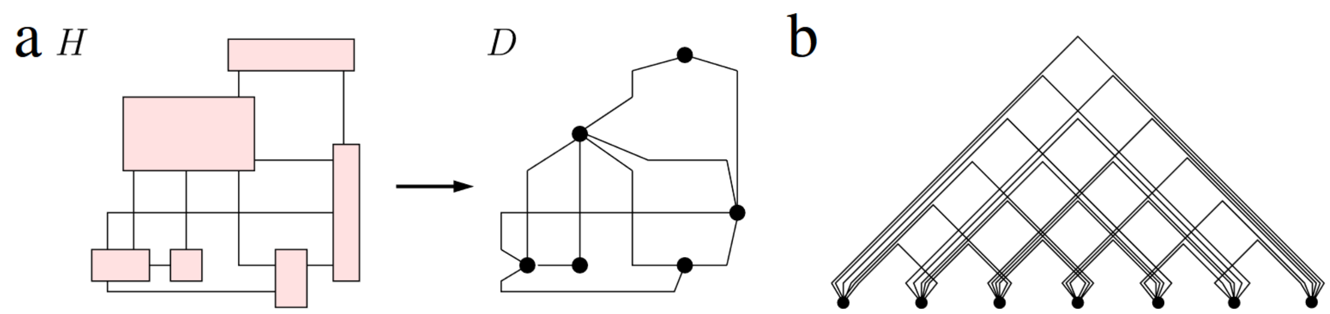
Instance: A graph G, **an integer** $\beta \in \{0, 1, 2, 3\}$

Question: Does G admit a RAC drawing **with β bends per edge**?

0-bends: $m \leq 4n - 10$
[Didimo, Eades and Liotta 2011]
NP-hard

1-bends: $m \leq 5.5n - \mathcal{O}(1)$
[Angelini, Bekos, Förster and Kaufmann 2018]
2-bends: $m \leq$ ~~$74.2n$~~ ~~$24n - 26$~~ $20n - 22$
[Arikushi, Fulek, Keszegh, Morić and Tóth 2012]
Complexity unknown

3-bends: always drawable
[Didimo, Eades and Liotta 2011]
linear time



Generalized RAC drawings

β-Bend RAC DRAWING

Instance: A graph G, **an integer** $\beta \in \{0, 1, 2, 3\}$

Question: Does G admit a RAC drawing **with** β **bends per edge**?

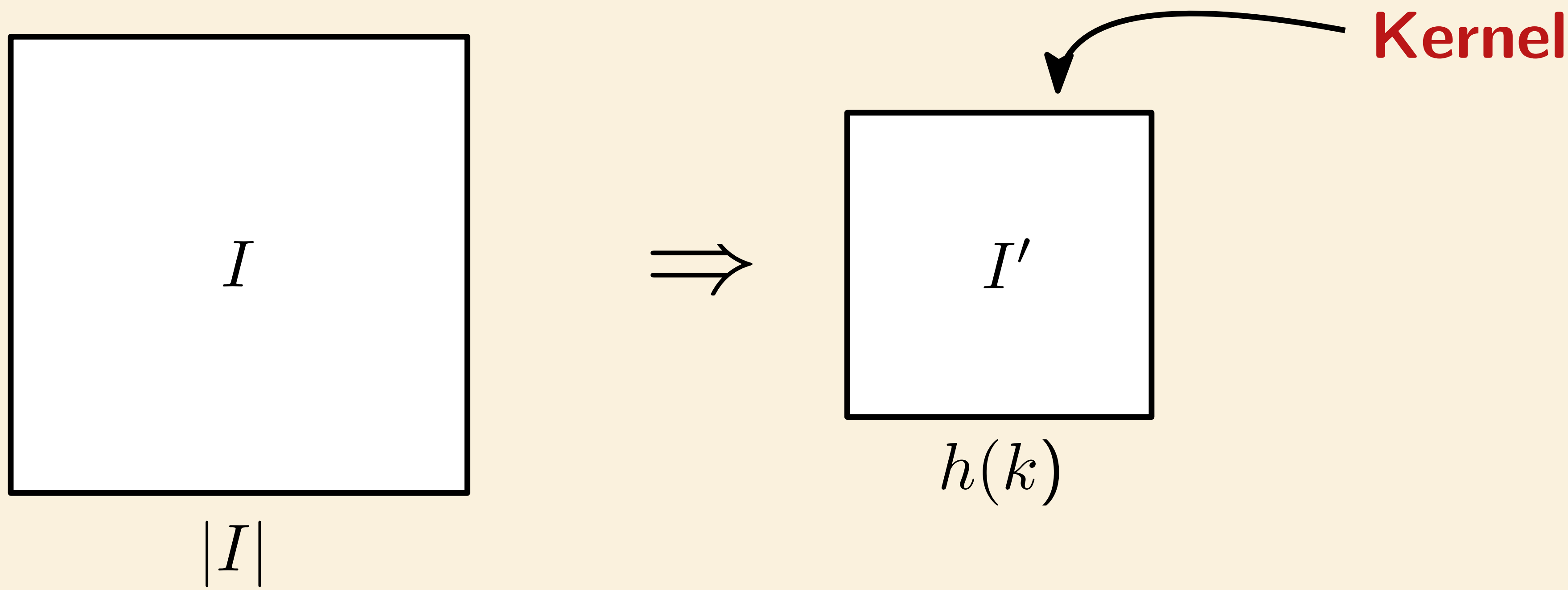
BEND-RESTRICTED RAC DRAWING (BRAC)

Instance: A graph G, an integer $b \geq 0$, edge labelling $\beta : E \mapsto \{0, 1, 2, 3\}$.

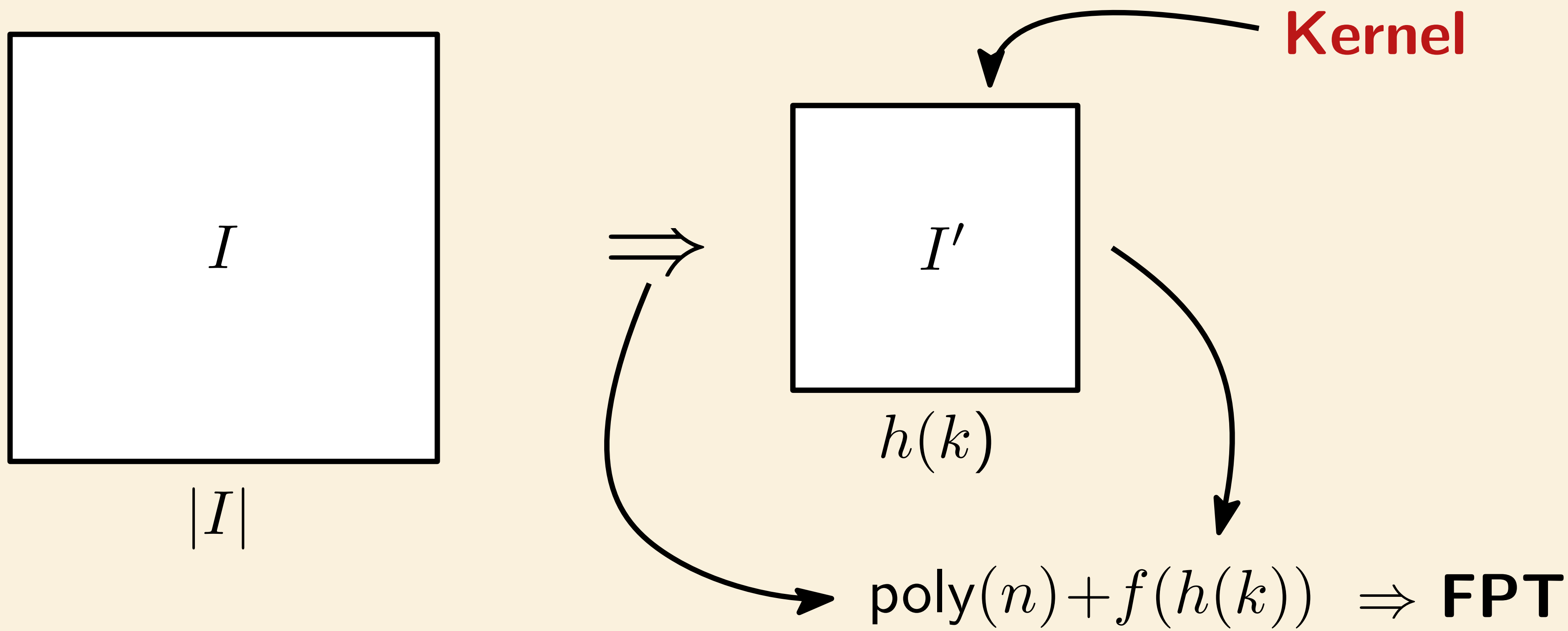
Question: Does G admit a RAC drawing, with

- at most b **total bends** and
- at most $\beta(e)$ bends for each edge e?

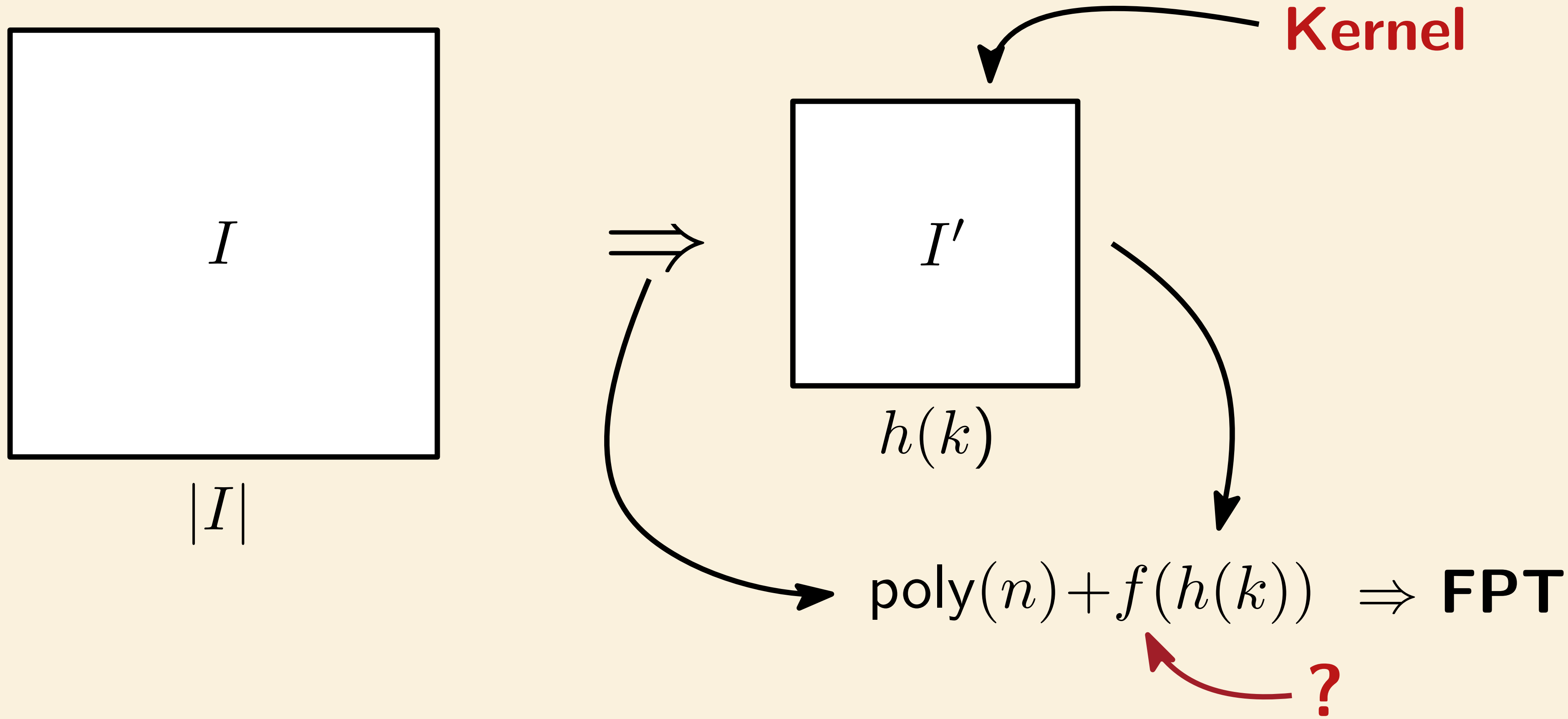
Side note: Kernelization



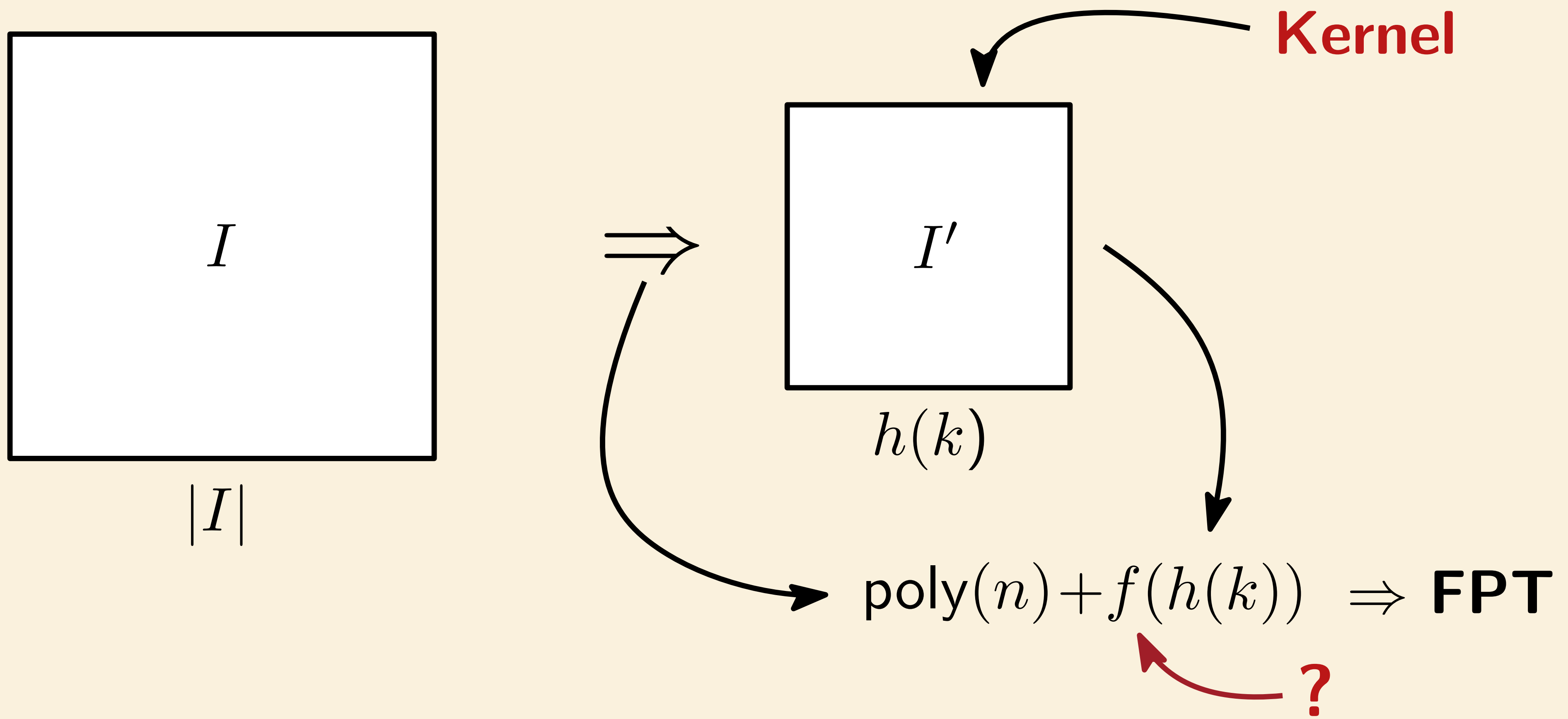
Side note: Kernelization



Side note: Kernelization



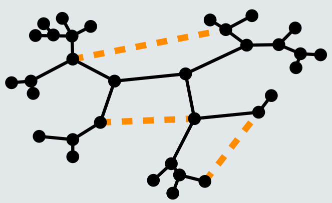
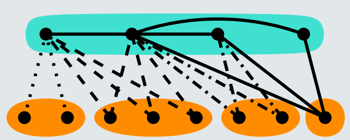
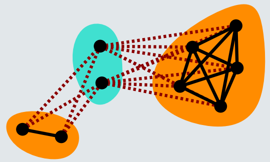
Side note: Kernelization



Theorem. An instance (G, b, β) of BRAC can be solved in time $m^{\mathcal{O}(m^2)}$.

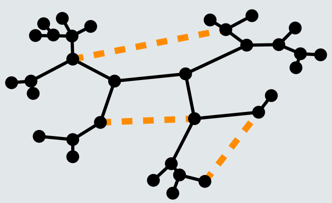
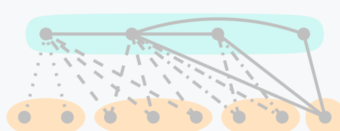
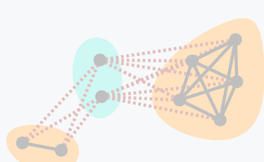
Results

- b-BEND β-RESTRICTED RAC DRAWING (BRAC) is *fixed-parameter tractable* when parameterized by

	Feedback edge number $\mathbf{fen}(G)$	$2^{\mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}} + \mathcal{O}(\lvert E(G)\rvert)$
	Vertex cover number $\mathbf{vcn}(G) + b$	$2^{2^{\mathcal{O}(\mathbf{vcn}(G)+\log b)}} + \mathcal{O}(\lvert E(G)\rvert)$
	Neighborhood diversity $\mathbf{nd}(G) + b$	$2^{b^{\mathcal{O}(\mathbf{nd}(G))}} + \mathcal{O}(\lvert E(G)\rvert)$

Results

- b-BEND β-RESTRICTED RAC DRAWING (BRAC) is *fixed-parameter tractable* when parameterized by

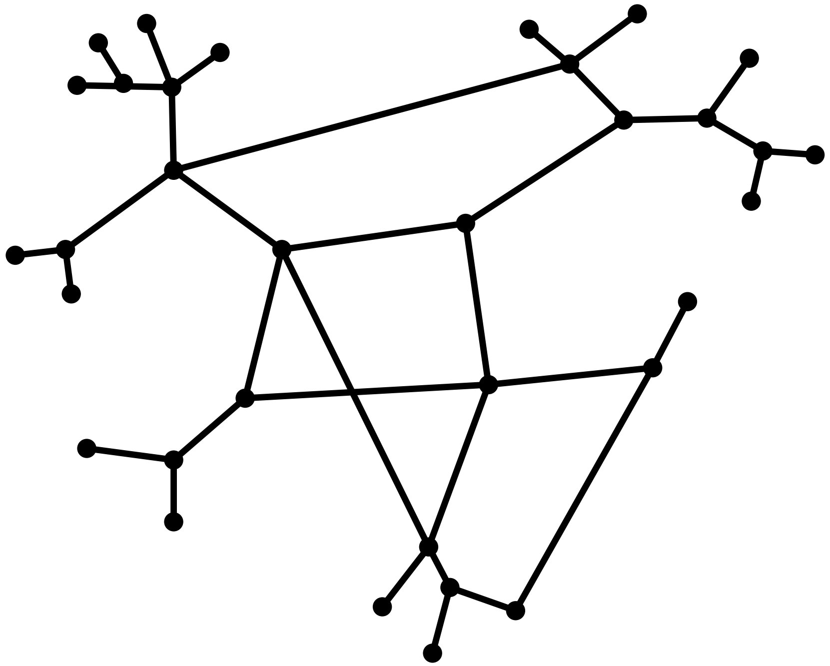
	Feedback edge number $\mathbf{fen}(G)$	$2^{\mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}} + \mathcal{O}(\lvert E(G)\rvert)$
	Vertex cover number $\mathbf{vcn}(G) + b$	$2^{2^{\mathcal{O}(\mathbf{vcn}(G)+\log b)}} + \mathcal{O}(\lvert E(G)\rvert)$
	Neighborhood diversity $\mathbf{nd}(G) + b$	$2^{b^{\mathcal{O}(\mathbf{nd}(G))}} + \mathcal{O}(\lvert E(G)\rvert)$

Feedback Edge Number (FEN)

Definition (Feedback Edge Number)**.** *Let* $G = (V, E)$ *be a graph. The feedback edge number* $\mathbf{fen}(G)$ *is the minimal number of edges, whose removal yields an acyclic graph.*

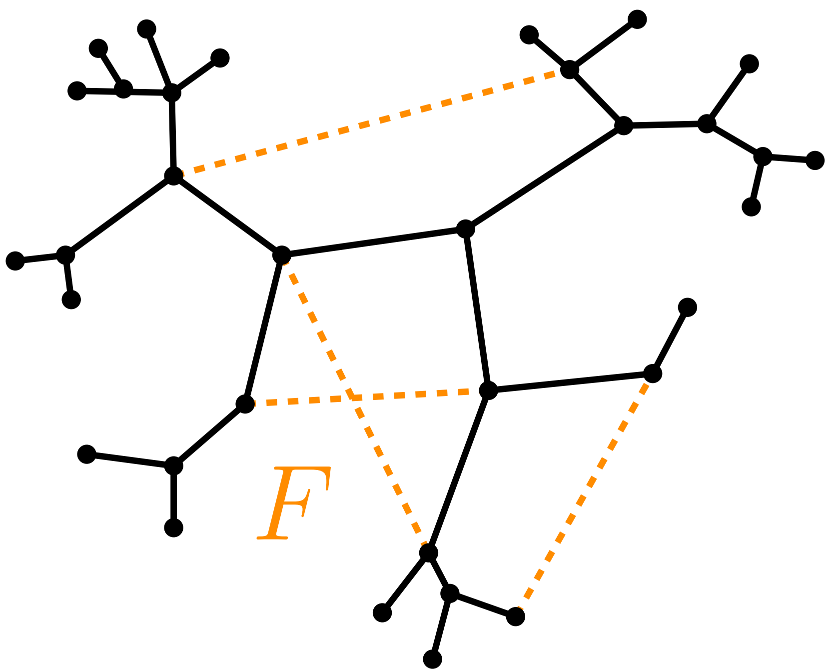
Feedback Edge Number (FEN)

Definition (Feedback Edge Number). *Let $G = (V, E)$ be a graph. The feedback edge number $\mathbf{fen}(G)$ is the minimal number of edges, whose removal yields an acyclic graph.*



Feedback Edge Number (FEN)

Definition (Feedback Edge Number)**.** *Let* $G = (V, E)$ *be a graph. The feedback edge number* $\mathbf{fen}(G)$ *is the minimal number of edges, whose removal yields an acyclic graph.*



FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

FEN - Kernel construction

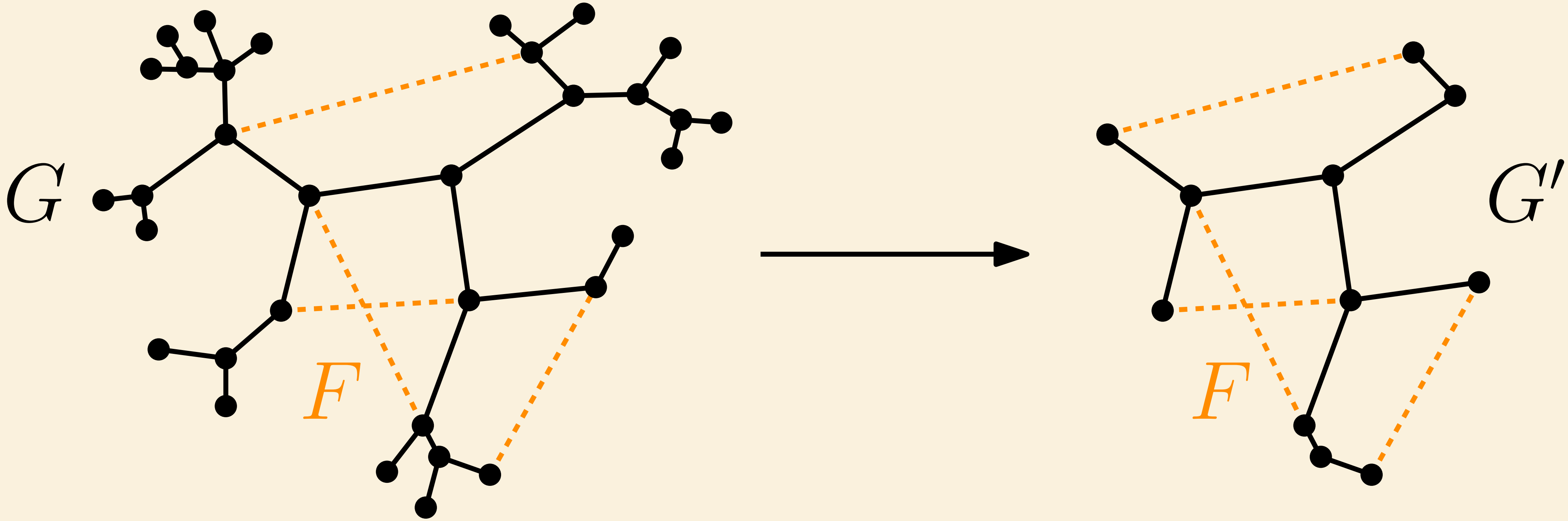
Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

- **Step 1**: Iteratively remove vertices of degree one

FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

Step 1: Iteratively remove vertices of degree one



Now $G' - F$ is a tree with at most $2 \cdot \mathbf{fen}(G)$ leaves

FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

- **Step 1**: Iteratively remove vertices of degree one

- **Step 2**: Partition $G' - F$ into $\ell \leq 4 \cdot \mathbf{fen}(G)$ disjoint subpaths

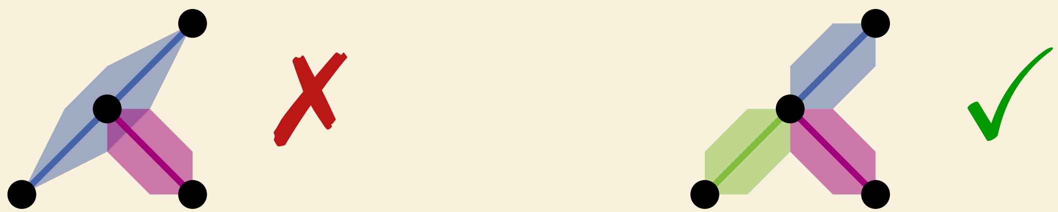
FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

▶ **Step 1**: Iteratively remove vertices of degree one

▼ **Step 2**: Partition $G' - F$ into $\ell \leq 4 \cdot \mathbf{fen}(G)$ disjoint subpaths

We allow the subpaths only to intersect at their respective endpoints



FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

- **Step 1**: Iteratively remove vertices of degree one
- **Step 2**: Partition $G' - F$ into $\ell \leq 4 \cdot \mathbf{fen}(G)$ disjoint subpaths
- **Step 3**: Order the paths $P_1, \ldots, P_\ell$ in increasing order of their length

FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

- **Step 1**: Iteratively remove vertices of degree one
- **Step 2**: Partition $G' - F$ into $\ell \leq 4 \cdot \mathbf{fen}(G)$ disjoint subpaths
- **Step 3**: Order the paths $P_1, \ldots, P_\ell$ in increasing order of their length

  Define $P_0 := F, p_i := |P_i|$

FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

- **Step 1**: Iteratively remove vertices of degree one

- **Step 2**: Partition $G' - F$ into $\ell \leq 4 \cdot \mathbf{fen}(G)$ disjoint subpaths

- **Step 3**: Order the paths $P_1, \ldots, P_\ell$ in increasing order of their length

  Define $P_0 := F, p_i := |P_i|$ $\quad \exists i : p_i > 12\ell \cdot p_{i-1}$?

FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

- **Step 1**: Iteratively remove vertices of degree one

- **Step 2**: Partition $G' - F$ into $\ell \leq 4 \cdot \mathbf{fen}(G)$ disjoint subpaths

- **Step 3**: Order the paths $P_1, \ldots, P_\ell$ in increasing order of their length

  Define $P_0 := F, p_i := |P_i|$ $\exists i : p_i > 12\ell \cdot p_{i-1}$?

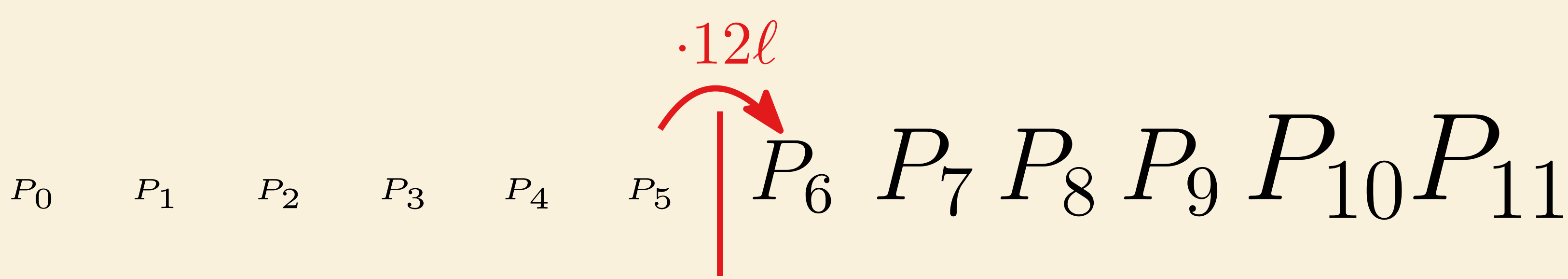
  P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11}

FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

- **Step 1**: Iteratively remove vertices of degree one
- **Step 2**: Partition $G' - F$ into $\ell \leq 4 \cdot \mathbf{fen}(G)$ disjoint subpaths
- **Step 3**: Order the paths $P_1, \ldots, P_\ell$ in increasing order of their length

  Define $P_0 := F, p_i := |P_i|$ $\exists i : p_i > 12\ell \cdot p_{i-1}$?

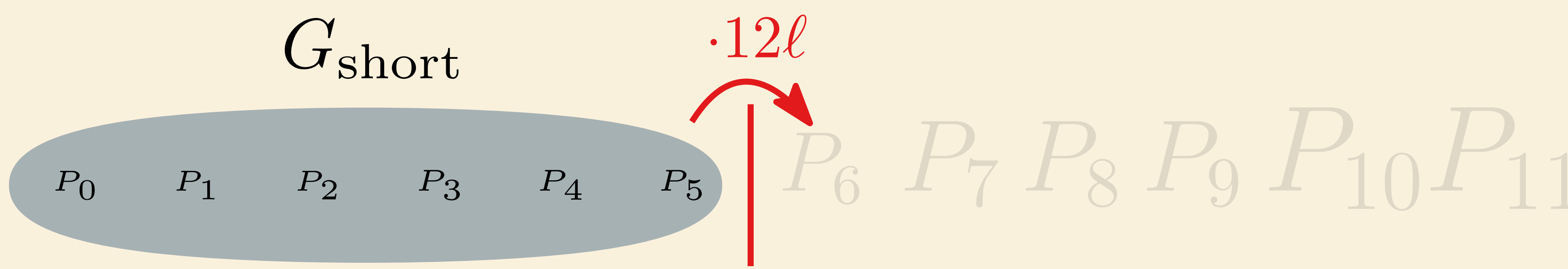


FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

- **Step 1**: Iteratively remove vertices of degree one
- **Step 2**: Partition $G' - F$ into $\ell \leq 4 \cdot \mathbf{fen}(G)$ disjoint subpaths
- **Step 3**: Order the paths $P_1, \ldots, P_\ell$ in increasing order of their length

Define $P_0 := F, p_i := |P_i|$ $\exists i : p_i > 12\ell \cdot p_{i-1}$?

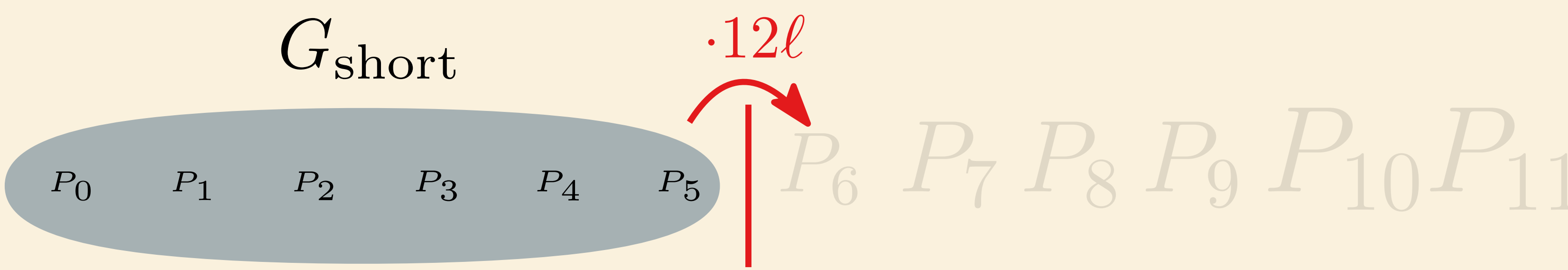


FEN - Kernel construction

Let $G = (V, E)$ be a graph and $F \subset E$ its feedback edge set.

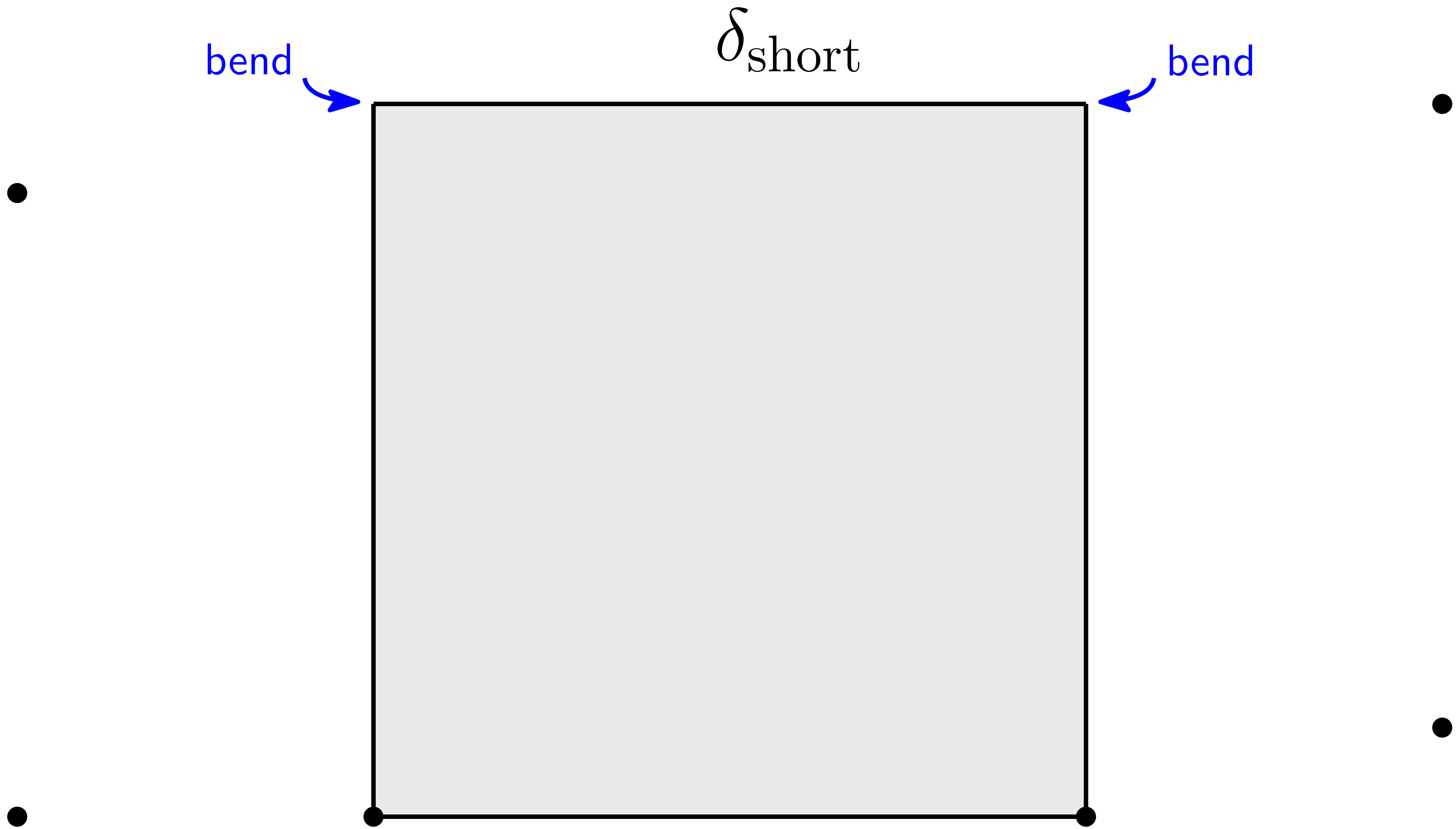
- **Step 1**: Iteratively remove vertices of degree one
- **Step 2**: Partition $G' - F$ into $\ell \leq 4 \cdot \mathbf{fen}(G)$ disjoint subpaths
- **Step 3**: Order the paths $P_1, \ldots, P_\ell$ in increasing order of their length

Define $P_0 := F, p_i := |P_i|$ $\exists i : p_i > 12\ell \cdot p_{i-1}$?

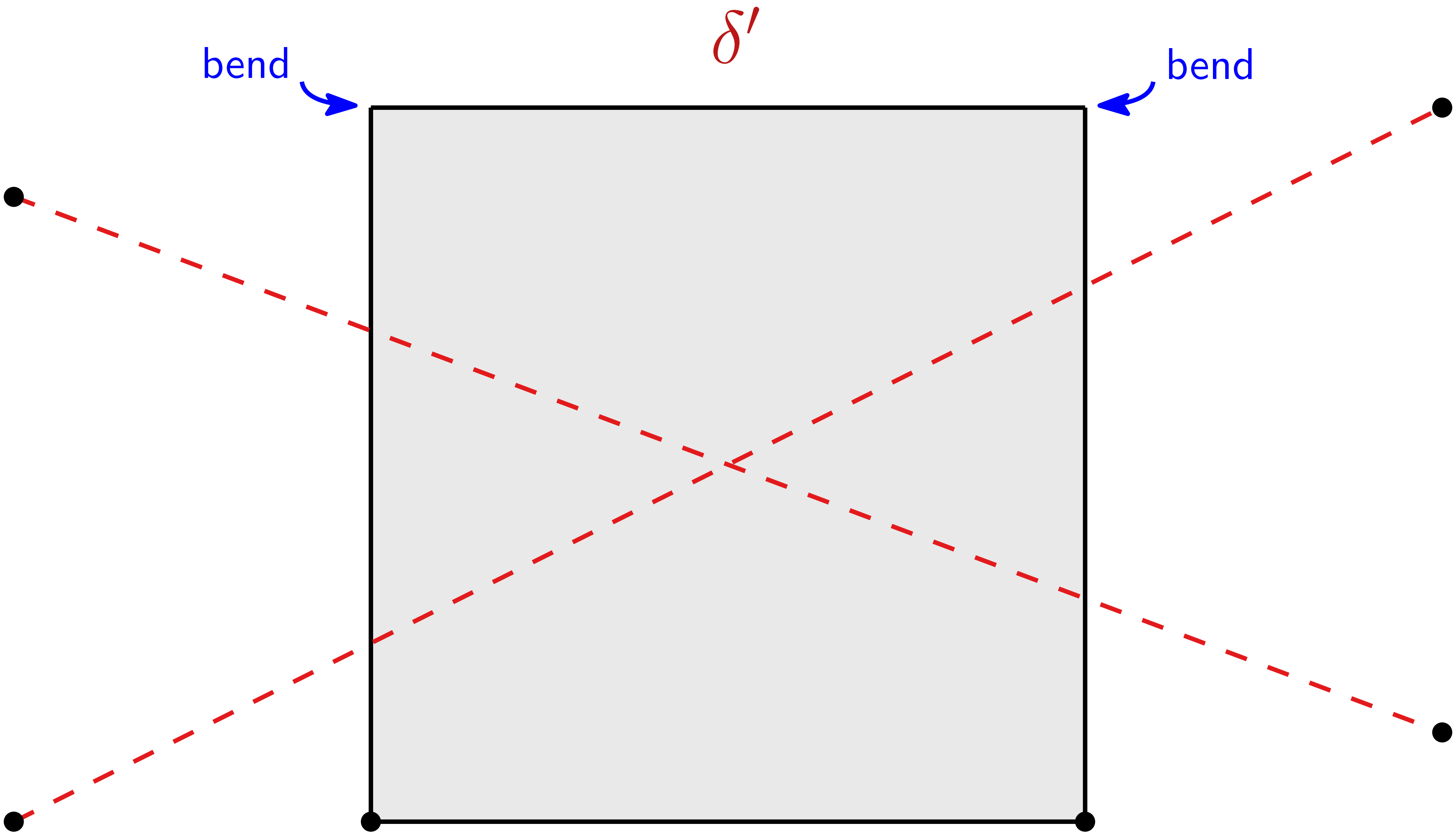


kernel of size $\mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}$

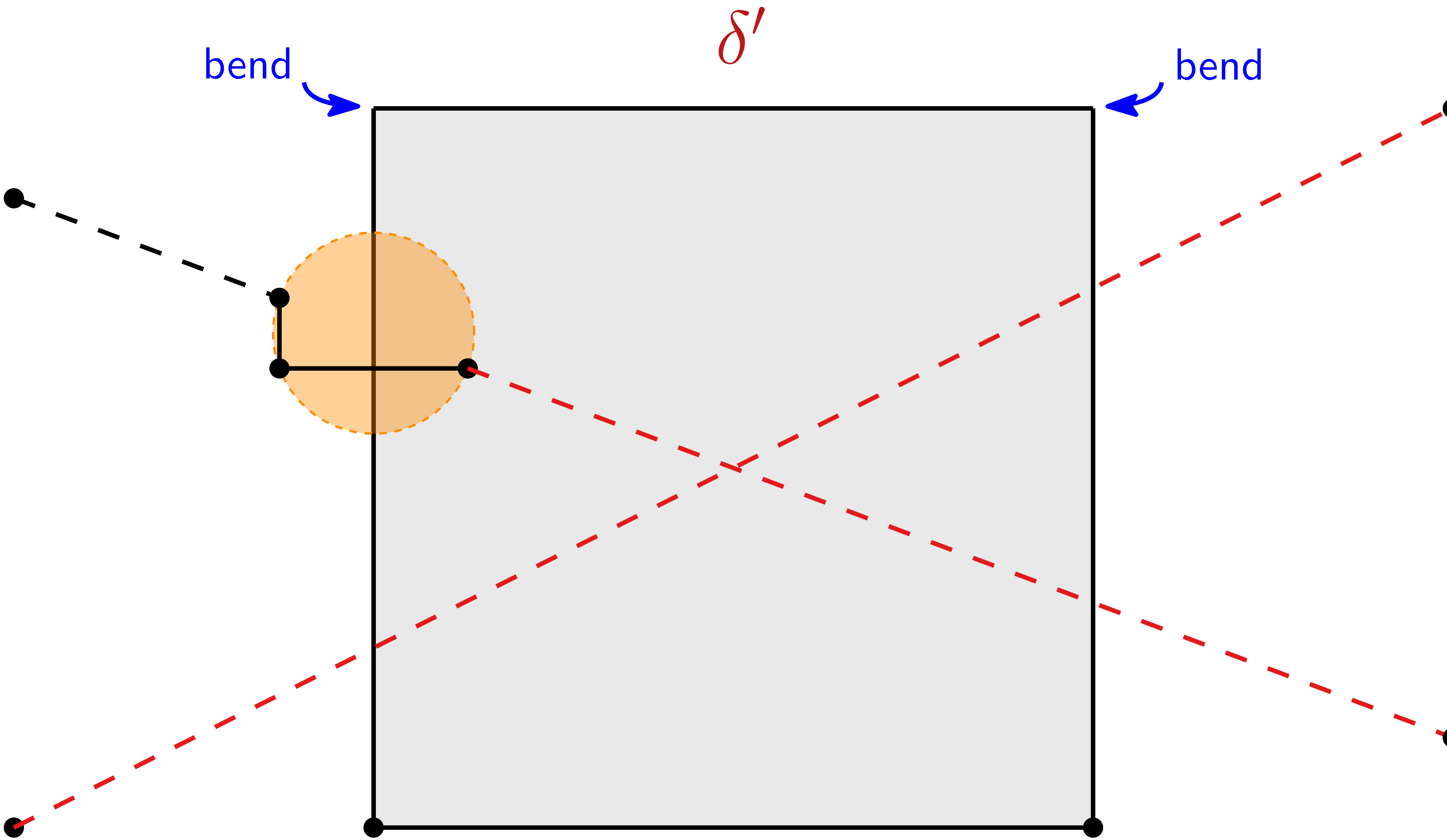
FEN - Kernel correctness



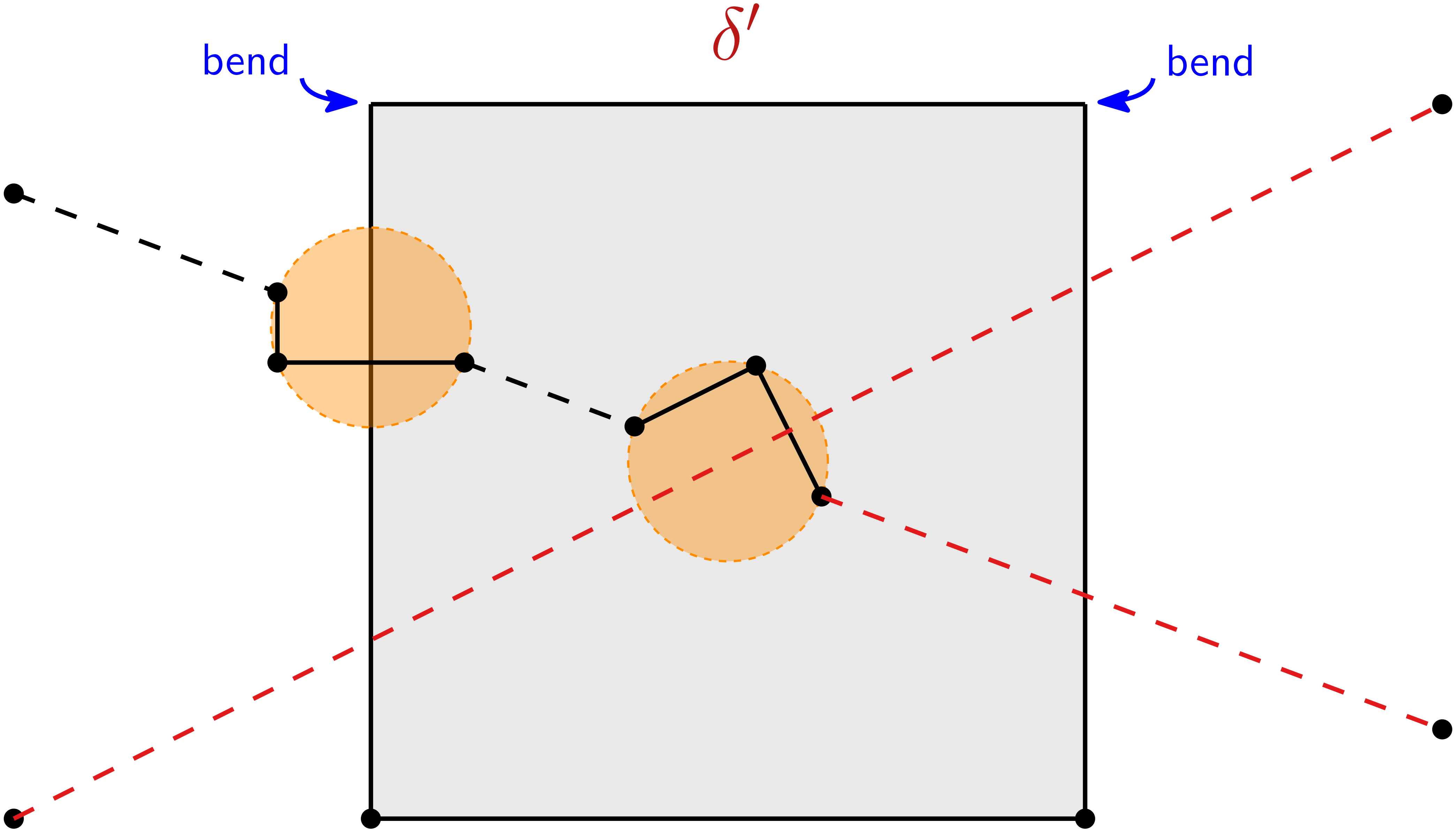
FEN - Kernel correctness



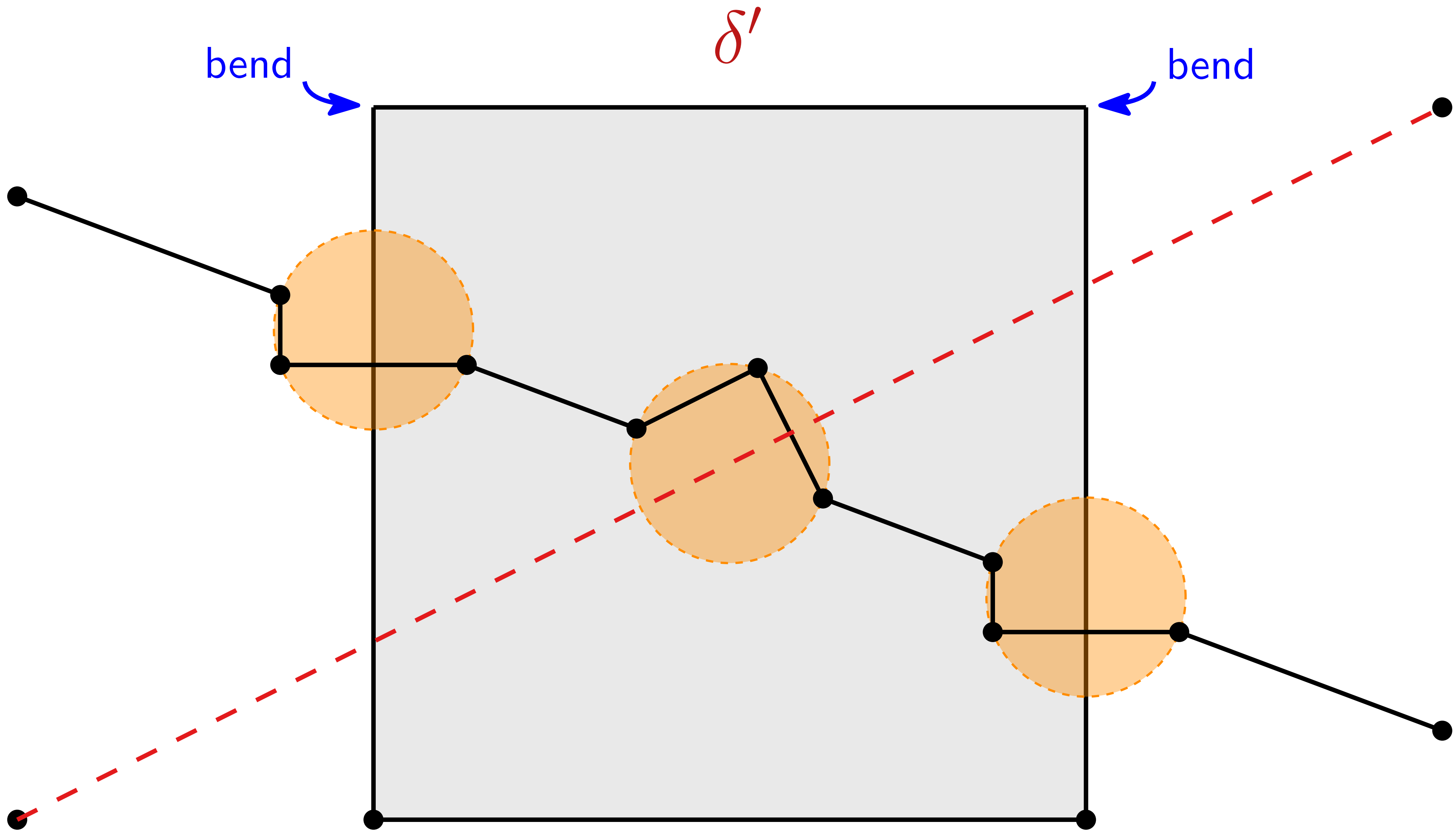
FEN - Kernel correctness



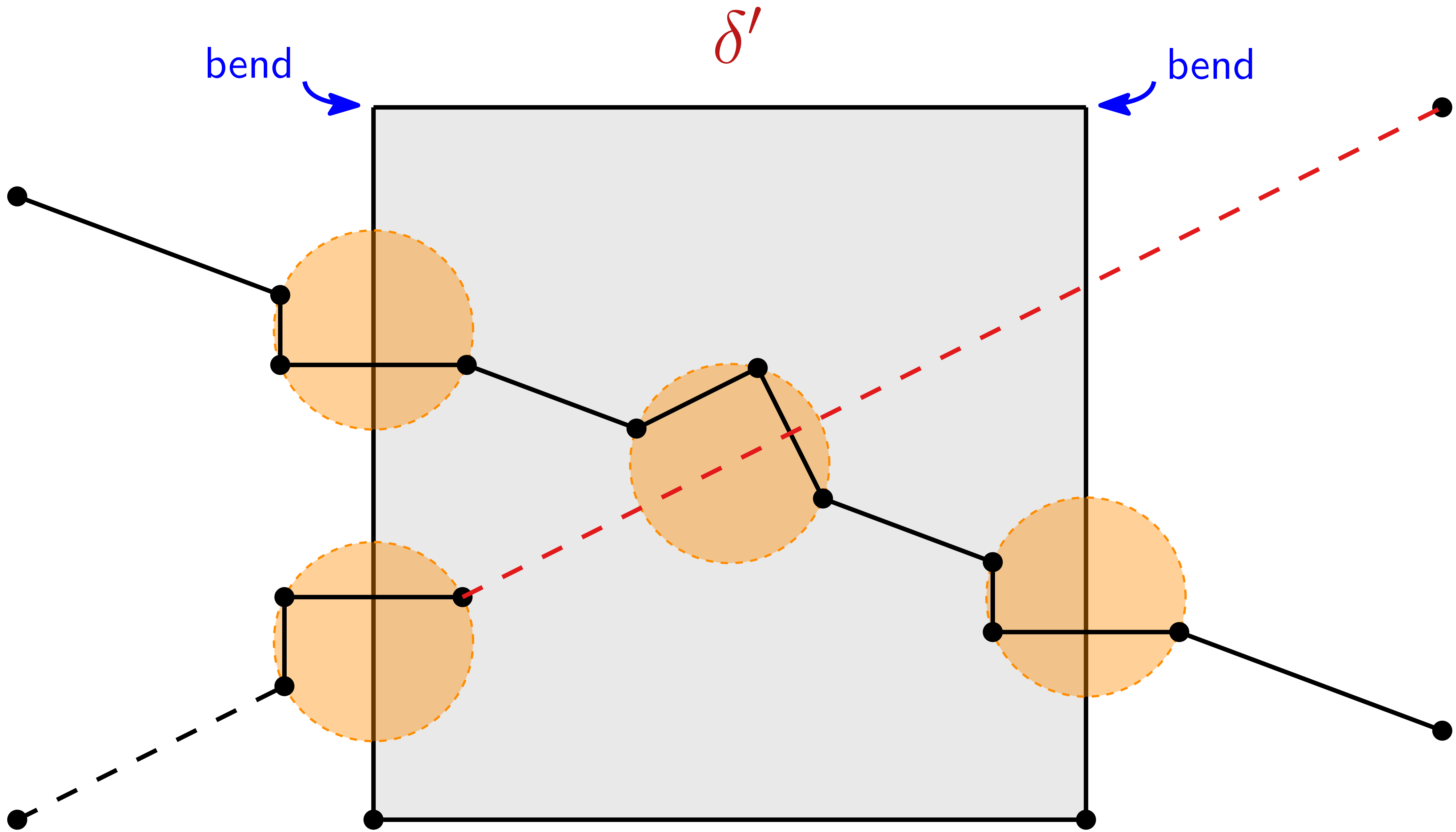
FEN - Kernel correctness



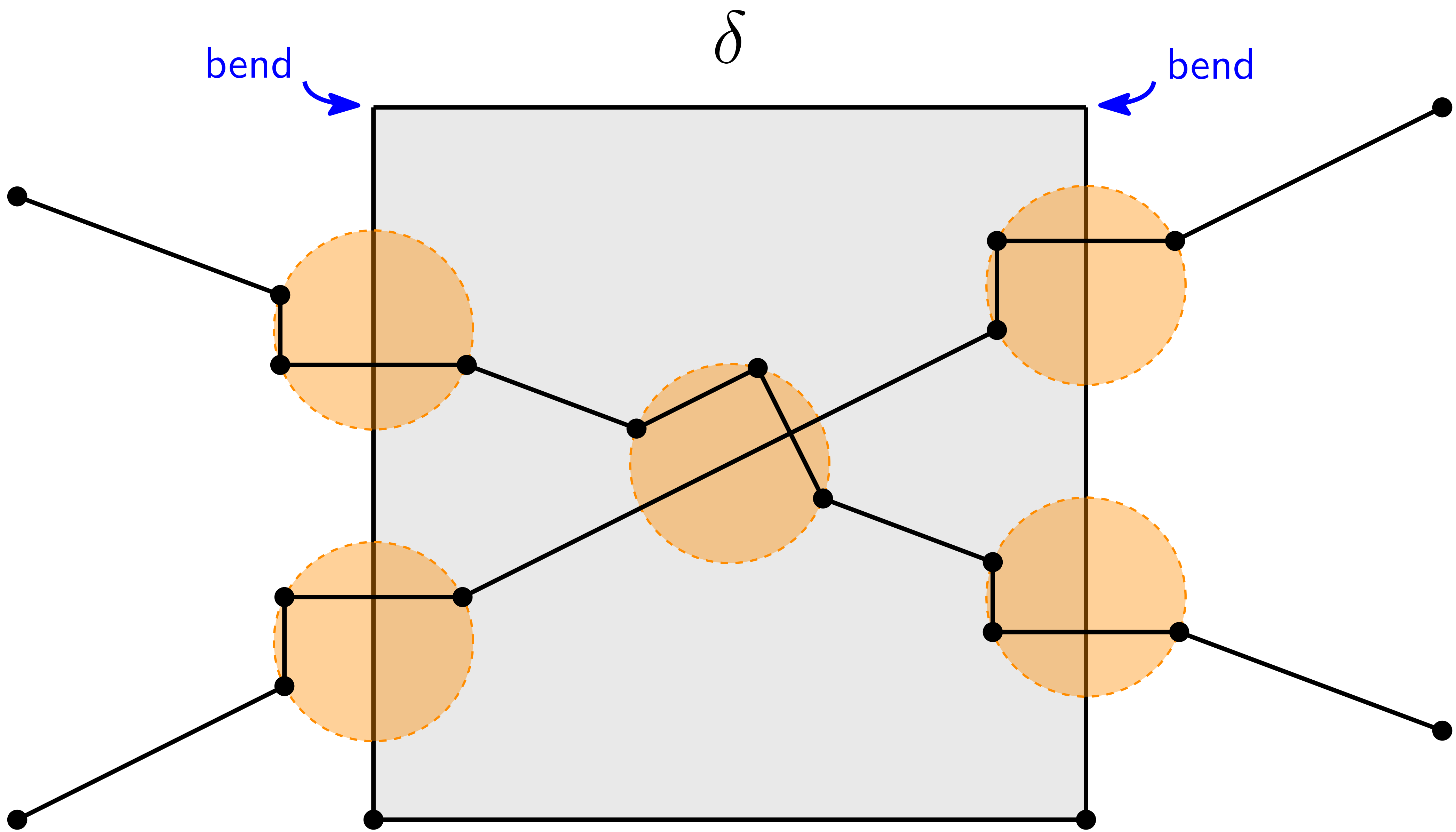
FEN - Kernel correctness



FEN - Kernel correctness



FEN - Kernel correctness



Kernel size

Let i_0 be the smallest i such that $p_{i_0+1} > 12\ell \cdot p_{i_0}$.

- Every long path $P_j \in \mathcal{P}_{\text{long}}$ is crossed at most: $4\ell \cdot p_{i_0}$ times

Kernel size

Let i_0 be the smallest i such that $p_{i_0+1} > 12\ell \cdot p_{i_0}$.

- Every long path $P_j \in \mathcal{P}_{\text{long}}$ is crossed at most: $4\ell \cdot p_{i_0}$ times
  - Each edge in G_{short} has at most 3 bends
  - At most $4\sum_{i=0}^{i_0} p_i$ crossings involving edges from G_{short}

  Total: $4\sum_{i=0}^{i_0} p_i + (\ell - i_0) \leq 4\ell \cdot p_{i_0}$ crossings

Kernel size

Let i_0 be the smallest i such that $p_{i_0+1} > 12\ell \cdot p_{i_0}$.

- Every long path $P_j \in \mathcal{P}_{\text{long}}$ is crossed at most: $4\ell \cdot p_{i_0}$ times

- Each crossing requires three vertices
- Since $|P_j| > 12\ell \cdot p_{i_0}$, each long path has enough vertices

Kernel size

Let i_0 be the smallest i such that $p_{i_0+1} > 12\ell \cdot p_{i_0}$.

- Every long path $P_j \in \mathcal{P}_{\text{long}}$ is crossed at most: $4\ell \cdot p_{i_0}$ times

- Each crossing requires three vertices
- Since $|P_j| > 12\ell \cdot p_{i_0}$, each long path has enough vertices

- $|G_{\text{short}}| = \sum_{i=0}^{i_0} p_i \leq \sum_{i=0}^{i_0} p_0 (12\ell)^i = \mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}$

Kernel size

Let i_0 be the smallest i such that $p_{i_0+1} > 12\ell \cdot p_{i_0}$.

- Every long path $P_j \in \mathcal{P}_{\text{long}}$ is crossed at most: $4\ell \cdot p_{i_0}$ times

- Each crossing requires three vertices
- Since $|P_j| > 12\ell \cdot p_{i_0}$, each long path has enough vertices

- $|G_{\text{short}}| = \sum_{i=0}^{i_0} p_i \le \sum_{i=0}^{i_0} p_0 (12\ell)^i = \mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}$

Solve time $m^{\mathcal{O}(m^2)}$

Kernel size $\mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}$

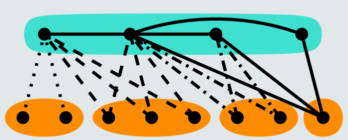
Kernel time $\mathcal{O}(|E(G)|)$

Total $2^{\mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}} + \mathcal{O}(|E(G)|)$

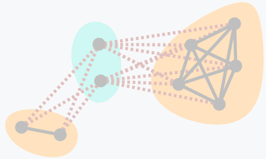
Results

- b-BEND β-RESTRICTED RAC DRAWING (BRAC) is *fixed-parameter tractable* when parameterized by

Feedback edge number $\mathbf{fen}(G)$ — $2^{\mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}} + \mathcal{O}(|E(G)|)$



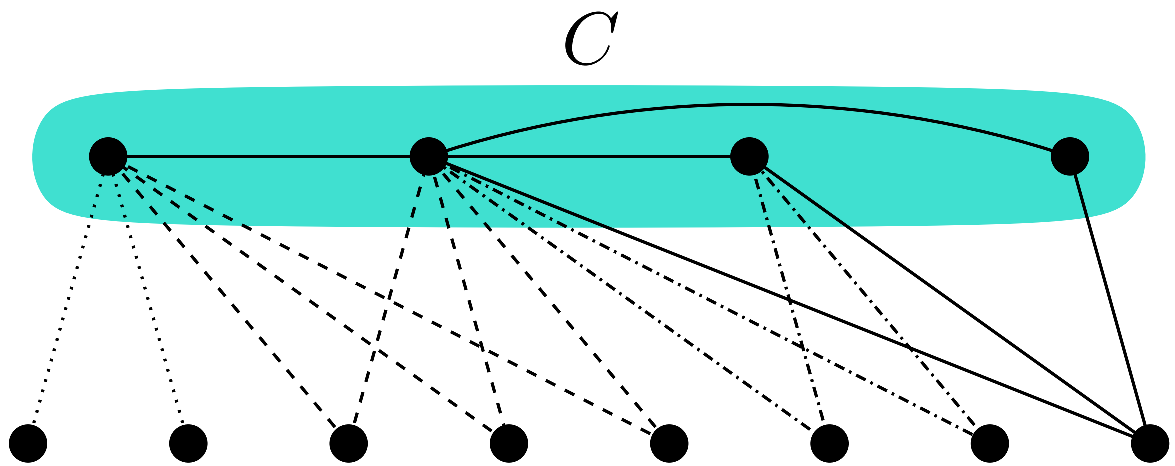
Vertex cover number $\mathbf{vcn}(G) + b$ — $2^{2^{\mathcal{O}(\mathbf{vcn}(G)+\log b)}} + \mathcal{O}(|E(G)|)$



Neighborhood diversity $\mathbf{nd}(G) + b$ — $2^{b^{\mathcal{O}(\mathbf{nd}(G))}} + \mathcal{O}(|E(G)|)$

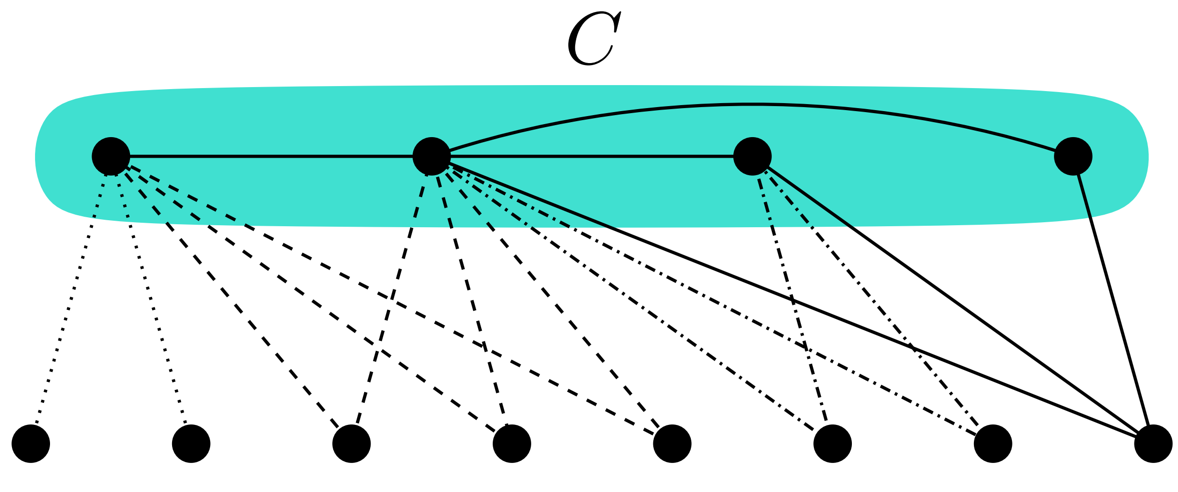
Vertex Cover

Definition. **Vertex cover**: *min. set* $C \subseteq V$ *s.t.* $\forall e \in E$: e *incident to* $v \in C$



Vertex Cover

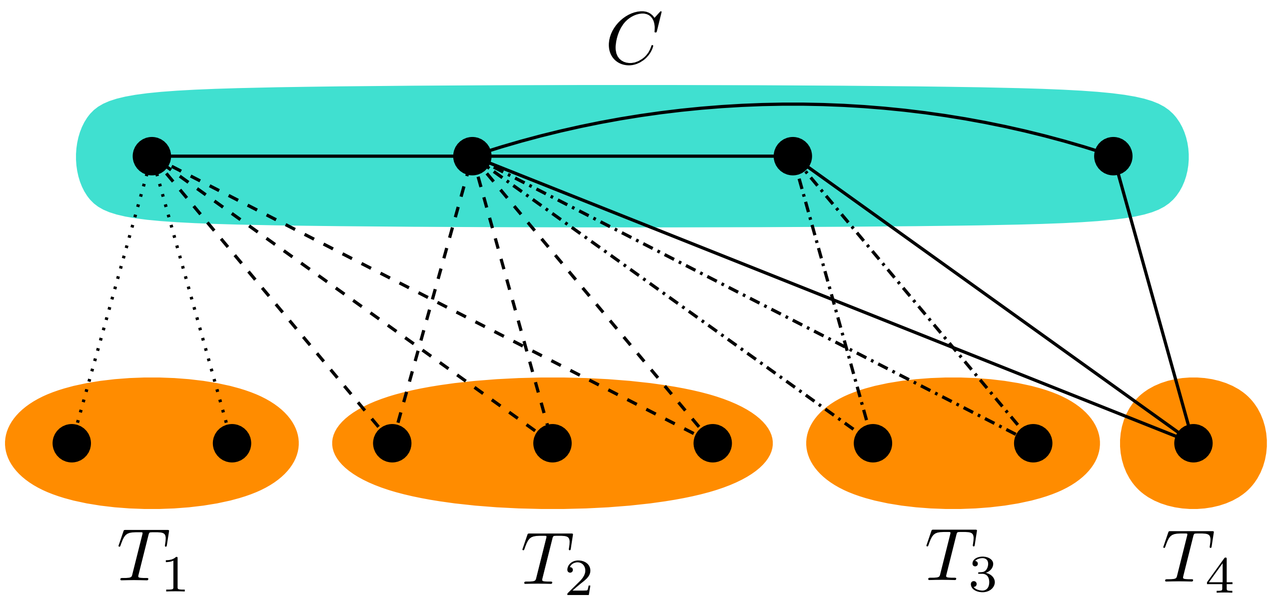
Definition. **Vertex cover**: *min. set* $C \subseteq V$ *s.t.* $\forall e \in E$: e *incident to* $v \in C$



- $k = \mathbf{vcn}(G) = |C|$

Vertex Cover

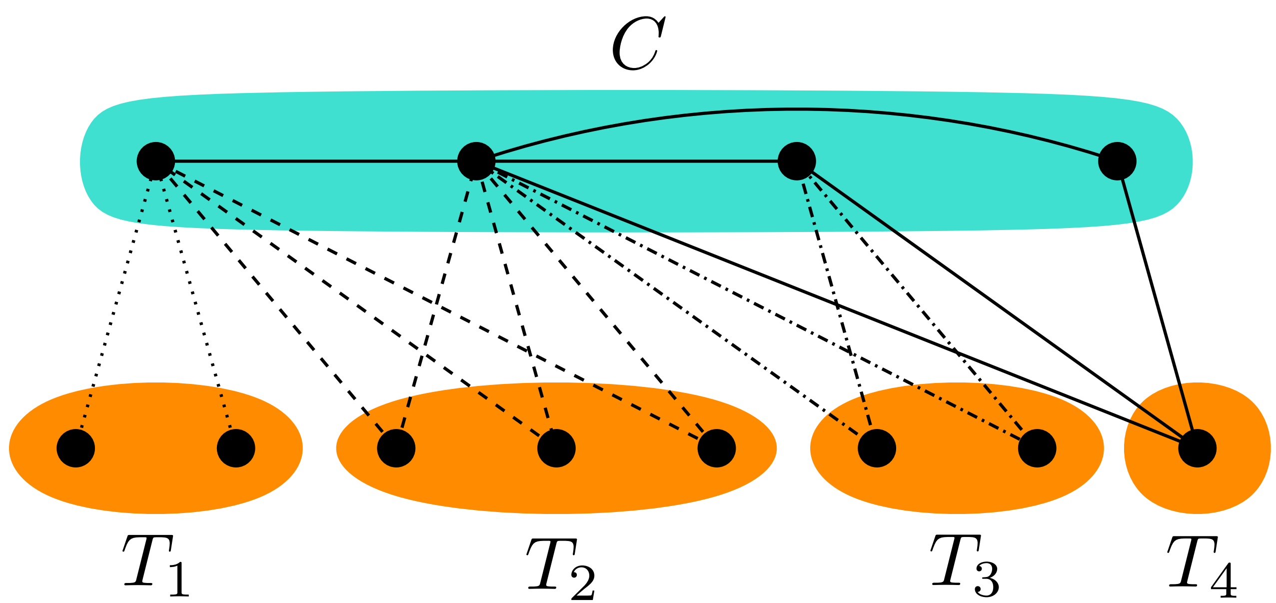
Definition. **Vertex cover**: *min. set* $C \subseteq V$ *s.t.* $\forall e \in E$: e *incident to* $v \in C$



- $k = \mathbf{vcn}(G) = |C|$
- **Type**: Set of vertices in $G - C$ with the same neighbourhood

Vertex Cover

Definition. **Vertex cover**: *min. set* $C \subseteq V$ *s.t.* $\forall e \in E$: e *incident to* $v \in C$

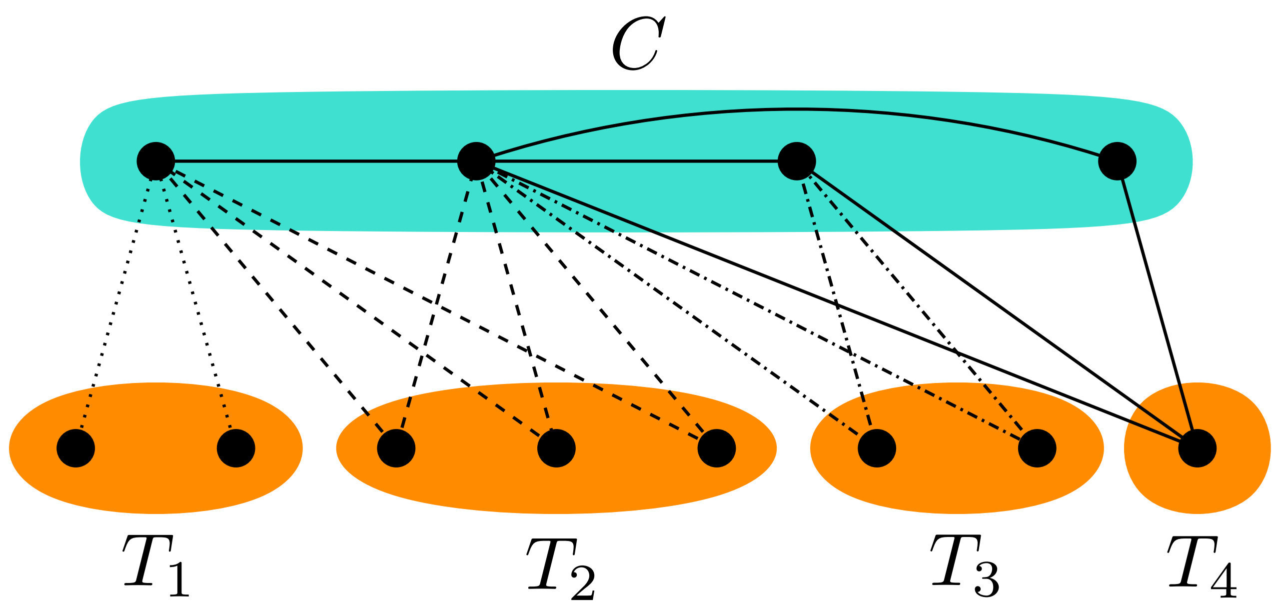


- $k = \mathbf{vcn}(G) = |C|$

- **Type**: Set of vertices in $G - C$ with the same neighbourhood
- At most 2^k types

Vertex Cover

Definition. **Vertex cover**: *min. set* $C \subseteq V$ *s.t.* $\forall e \in E$: e *incident to* $v \in C$



- $k = \mathbf{vcn}(G) = |C|$
- **Type**: Set of vertices in $G - C$ with the same neighbourhood
- At most 2^k types
- **Idea:** limit the number of vertices in each type
  by a function of $\mathbf{vcn}(G) + b$

FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

FTP via $\mathbf{vcn}(G) + b$

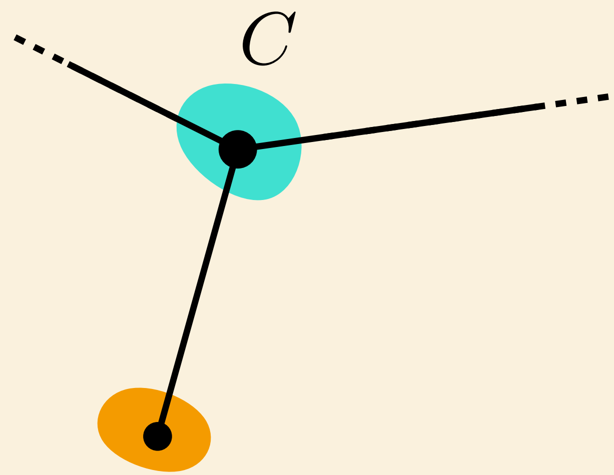
- Distinguish types by number of neighbours $|T_i|$

▶ **Case 1**: $|T_i| = 1$

FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

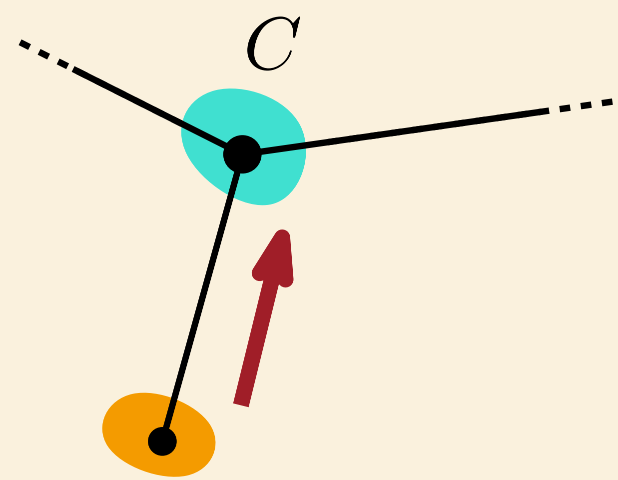
Case 1: $|T_i| = 1$



FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

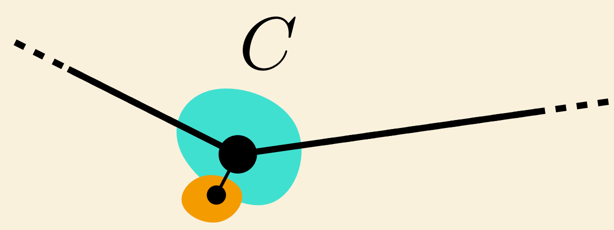
Case 1: $|T_i| = 1$



FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

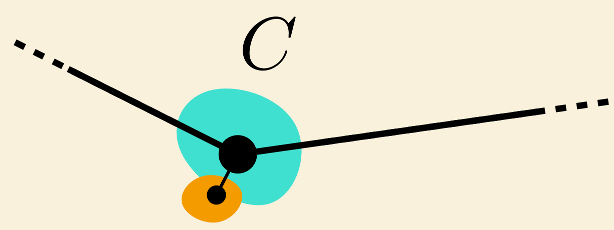
Case 1: $|T_i| = 1$



FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

Case 1: $|T_i| = 1$



$\Rightarrow$ can remove all types with $|T_i| = 1$

FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

▶ **Case 1**: $|T_i| = 1$ $\Rightarrow$ keep 0

FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

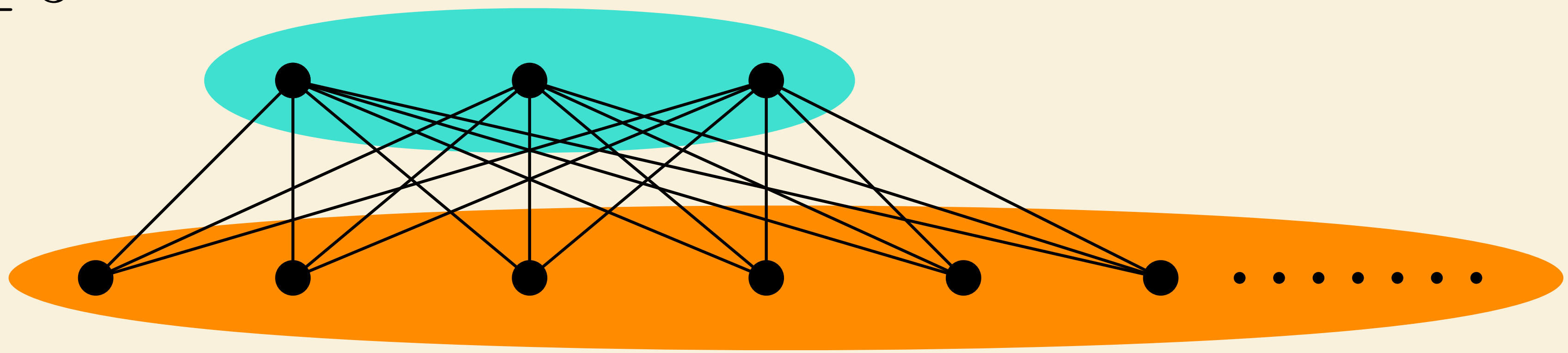
Case 2: $|T_i| \geq 3$

FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$

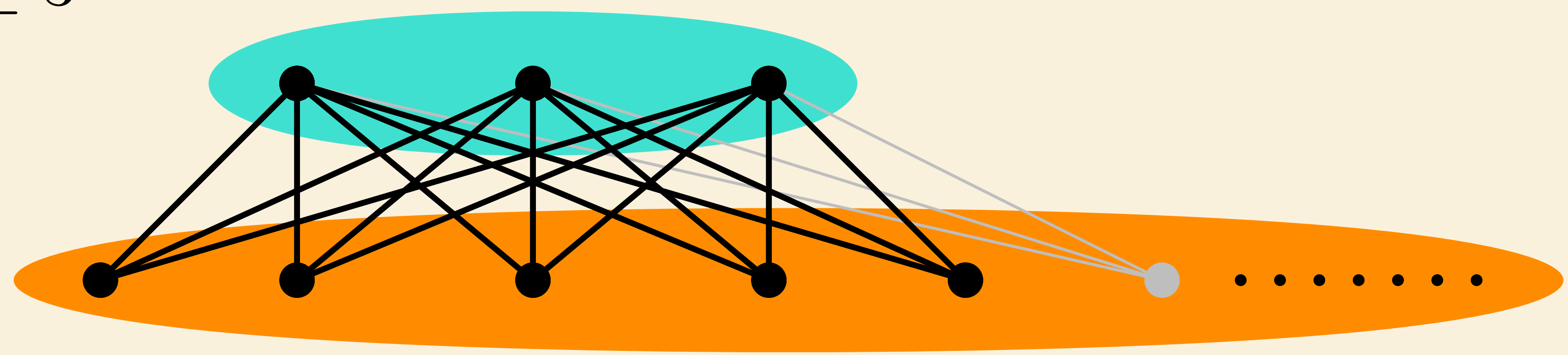


FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$



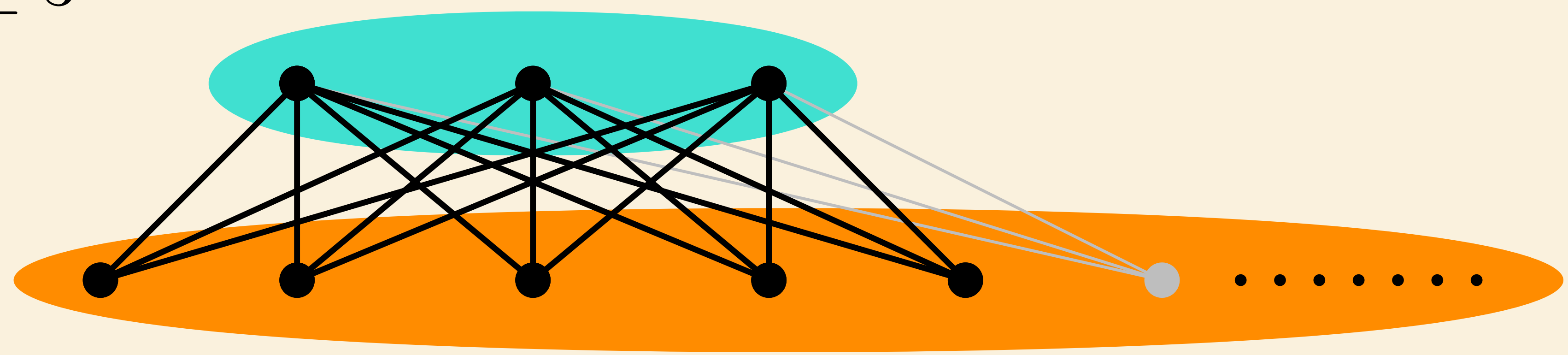
$K_{3,5}$

FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$



$K_{3,5}$

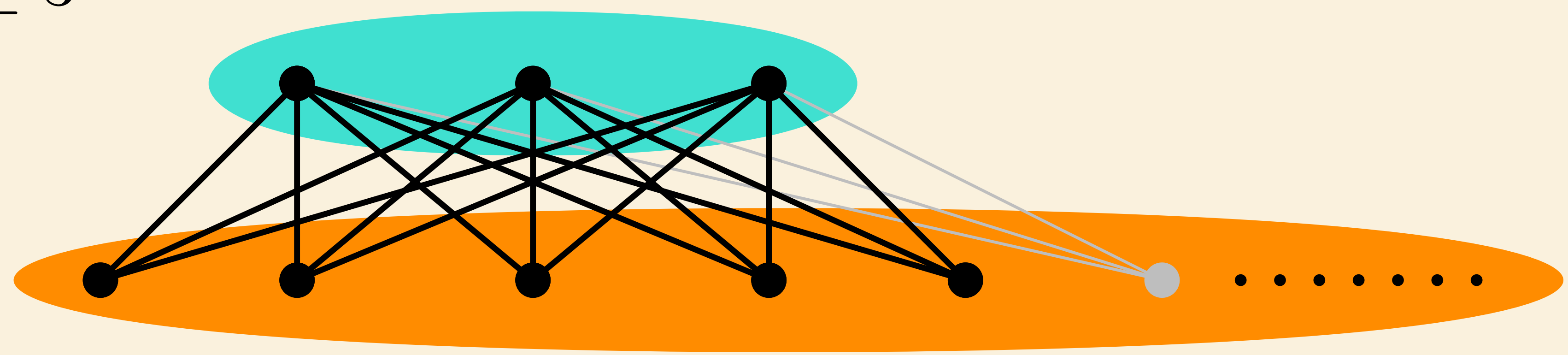
Lemma. $K_{3,5}$ does **not** admit a RAC drawing [Didimo, Eades, Liotta 2010]

FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$



$K_{3,5}$

Lemma. $K_{3,5}$ does **not** admit a RAC drawing [Didimo, Eades, Liotta 2010]

$\Rightarrow$ always no-instance with $\geq 5 + b$ vertices

FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

Case 3: $|T_i| = 2$

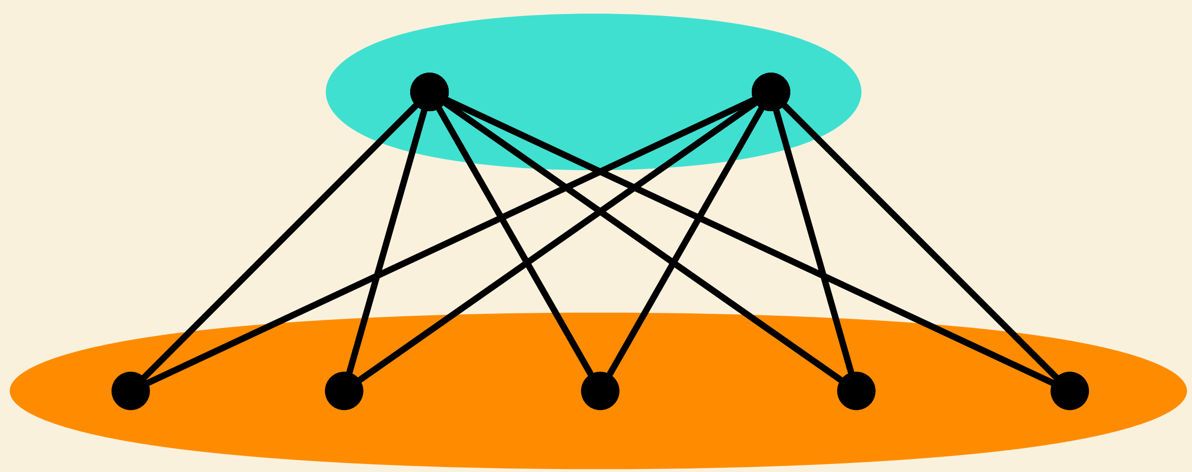
FTP via $\mathbf{vcn}(G) + b$

- Distinguish types by number of neighbours $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

Case 3: $|T_i| = 2$



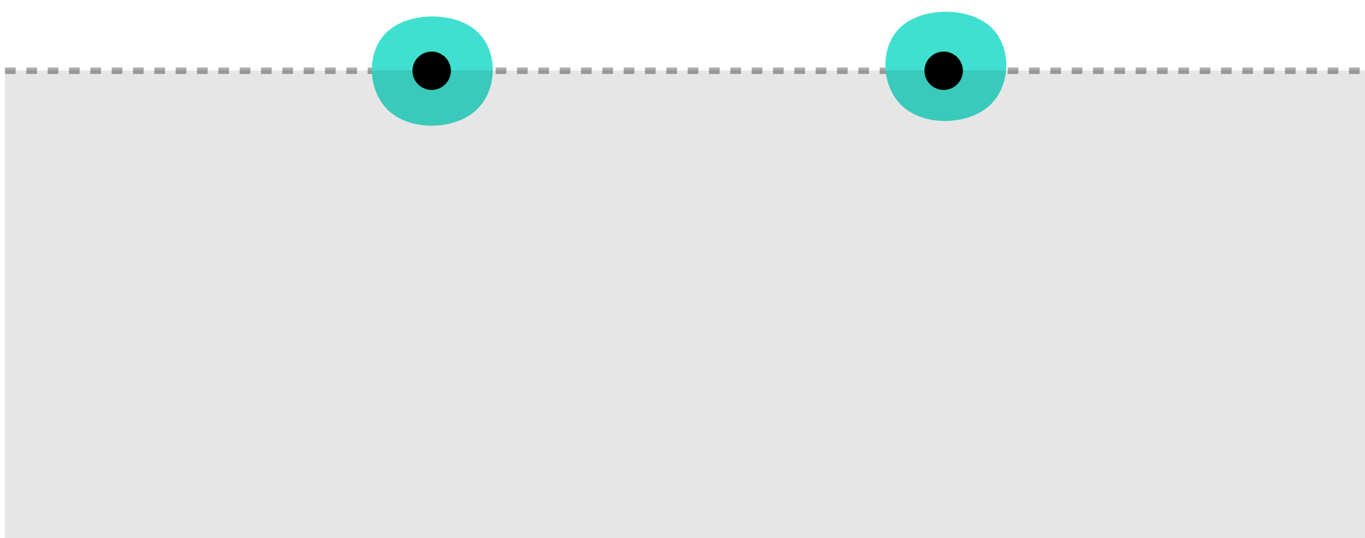
Case 3: $|T_i| = 2$

Case 3: $|T_i| = 2$

Lemma. At most 4 vertices of a type T with $|T| = 2$ can be involved in a crossing within T.

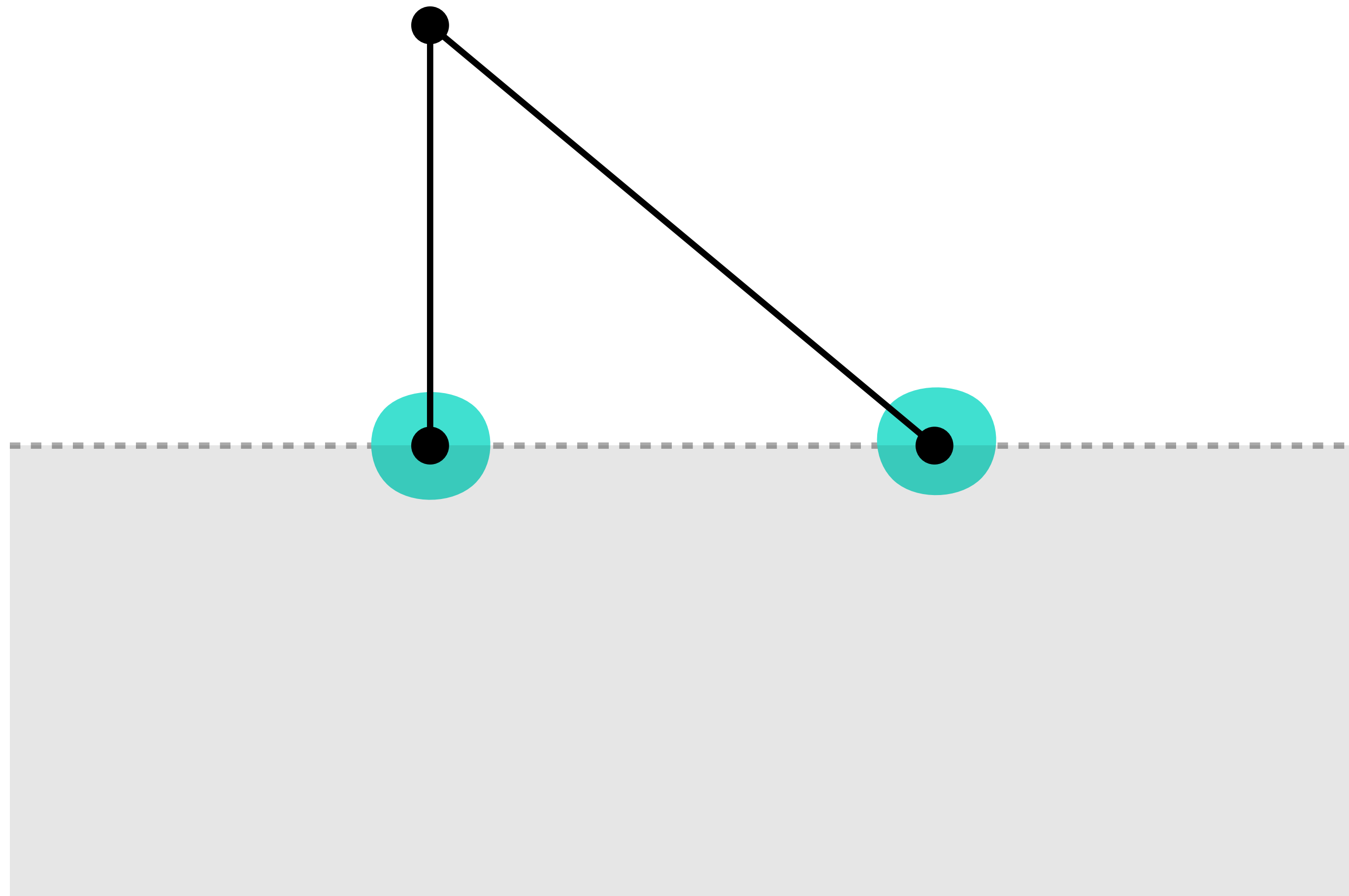
Case 3: $|T_i| = 2$

Lemma. At most 4 vertices of a type T with $|T| = 2$ can be involved in a crossing within T.



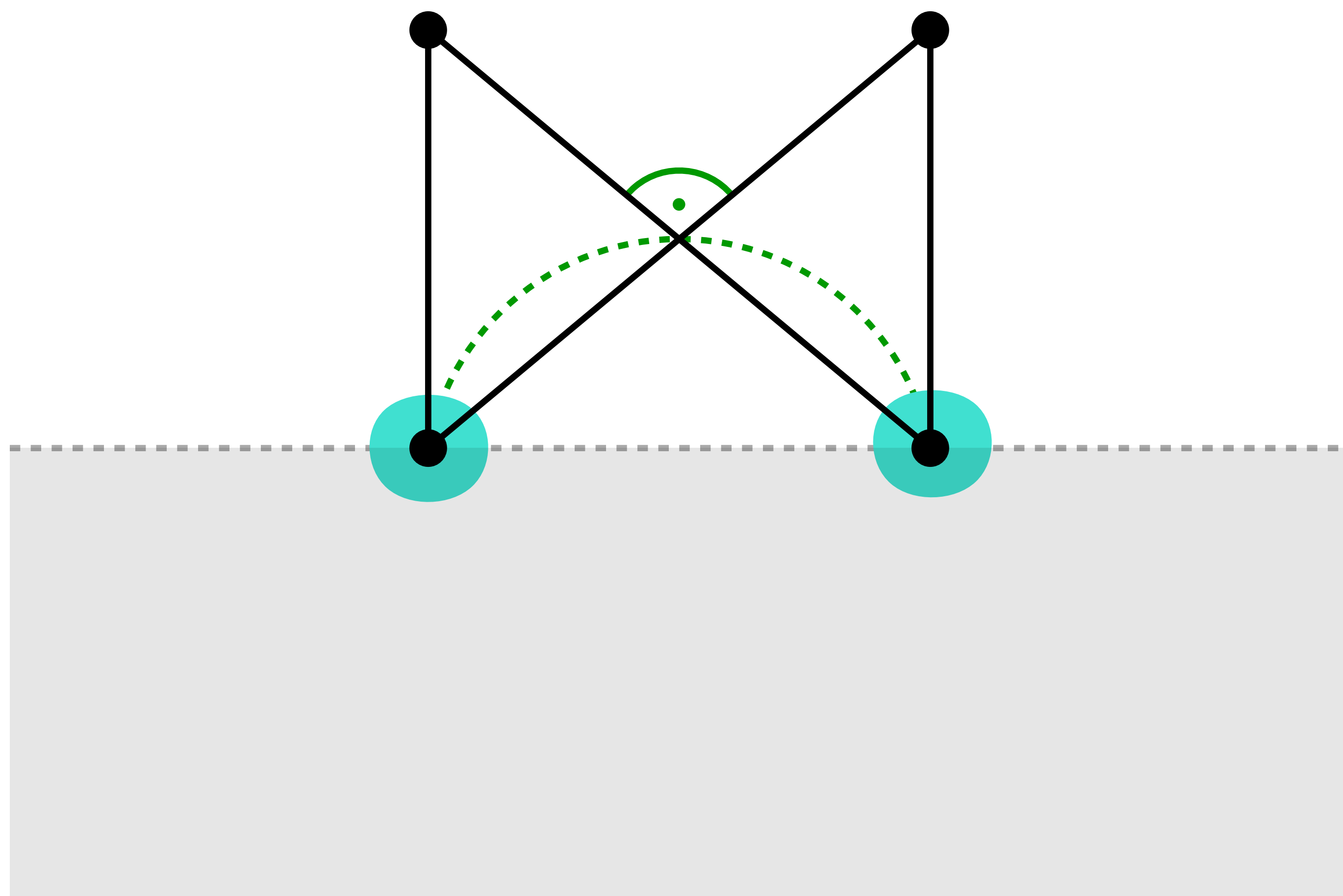
Case 3: $|T_i| = 2$

Lemma. At most 4 vertices of a type T with $|T| = 2$ can be involved in a crossing within T.



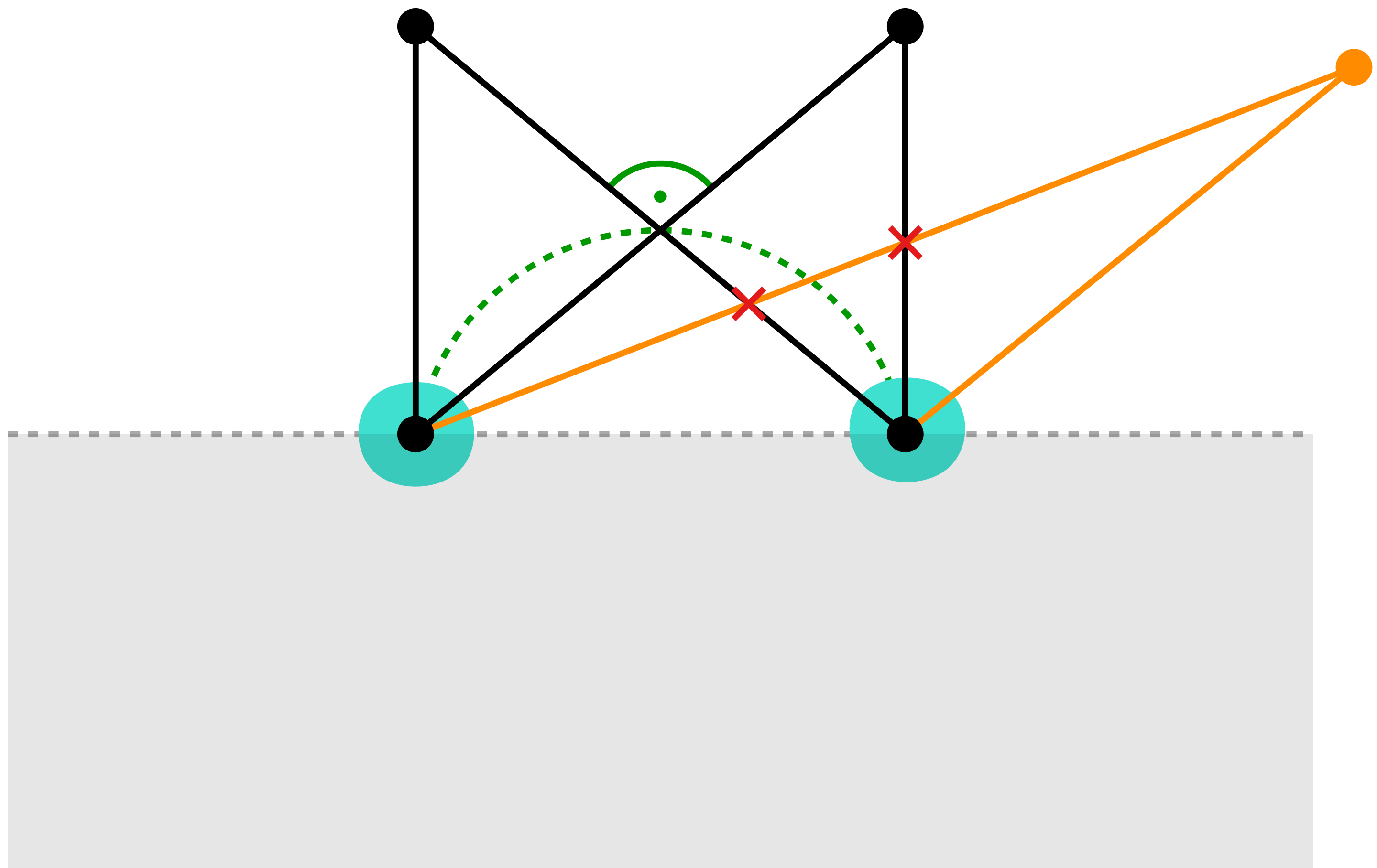
Case 3: $|T_i| = 2$

Lemma. At most 4 vertices of a type T with $|T| = 2$ can be involved in a crossing within T.



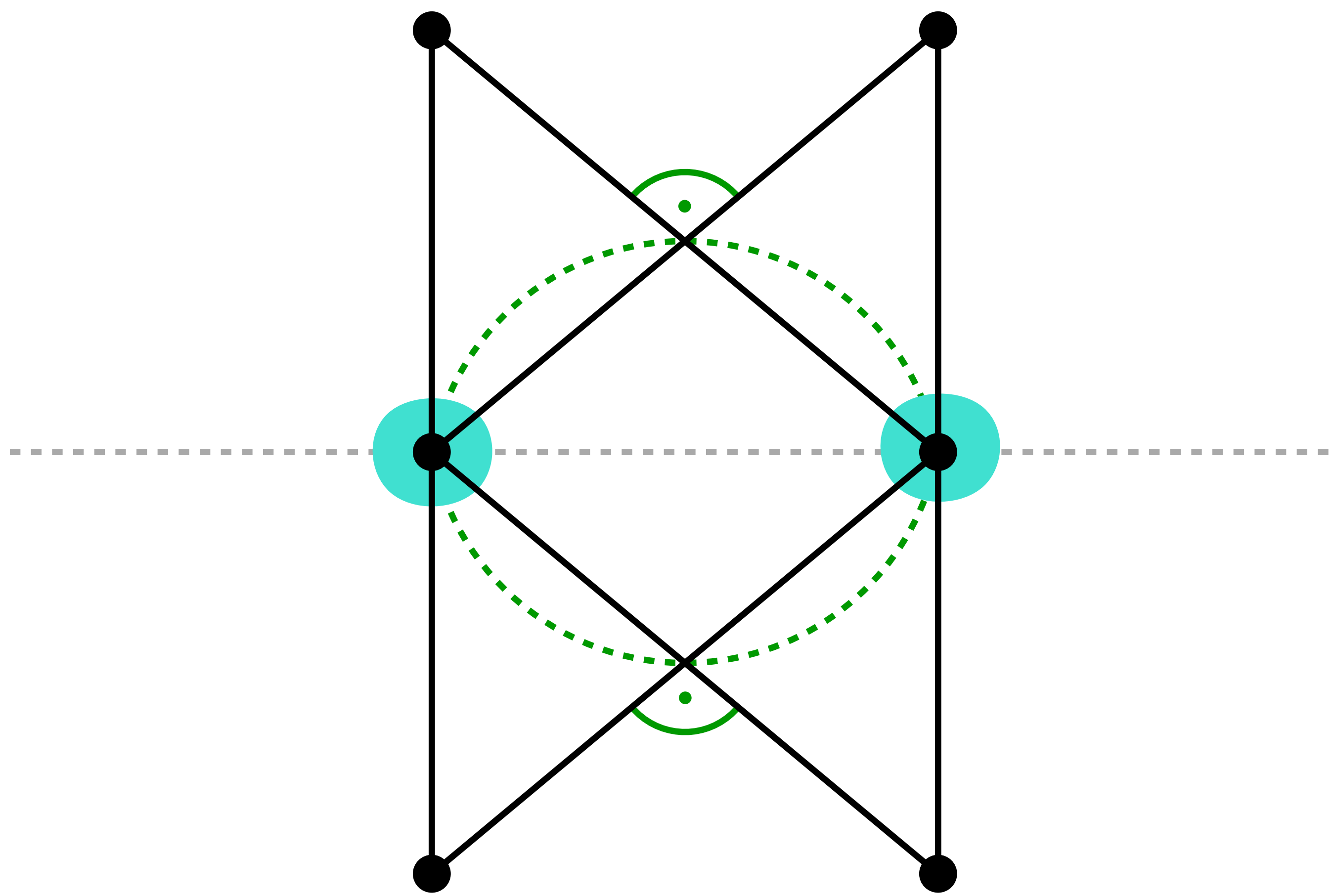
Case 3: $|T_i| = 2$

Lemma. At most 4 vertices of a type T with $|T| = 2$ can be involved in a crossing within T.



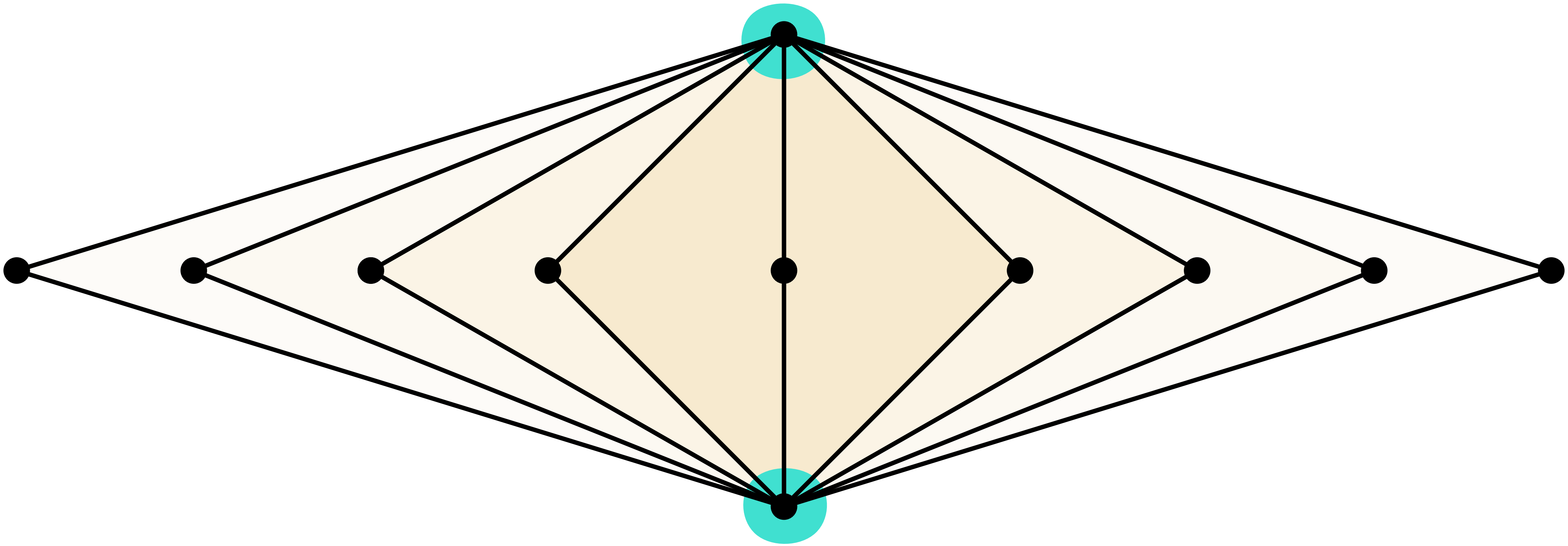
Case 3: $|T_i| = 2$

Lemma. At most 4 vertices of a type T with $|T| = 2$ can be involved in a crossing within T.



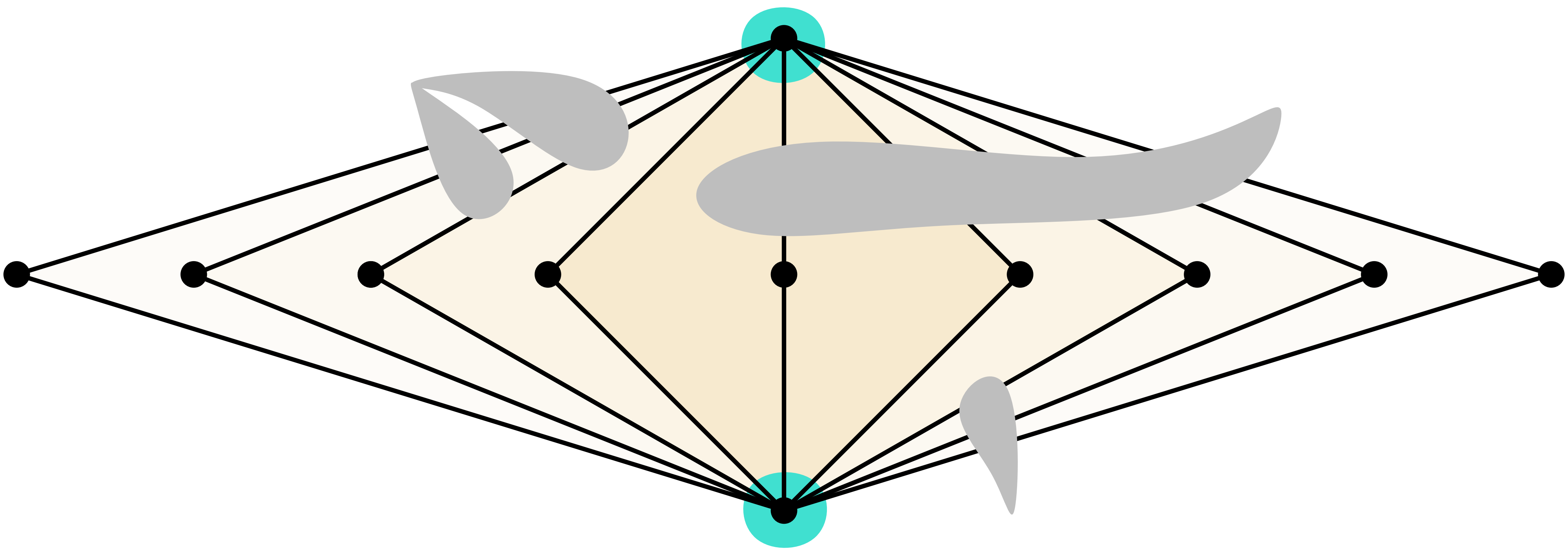
Case 3: $|T_i| = 2$

- All other vertices of T_i form a hierarchy



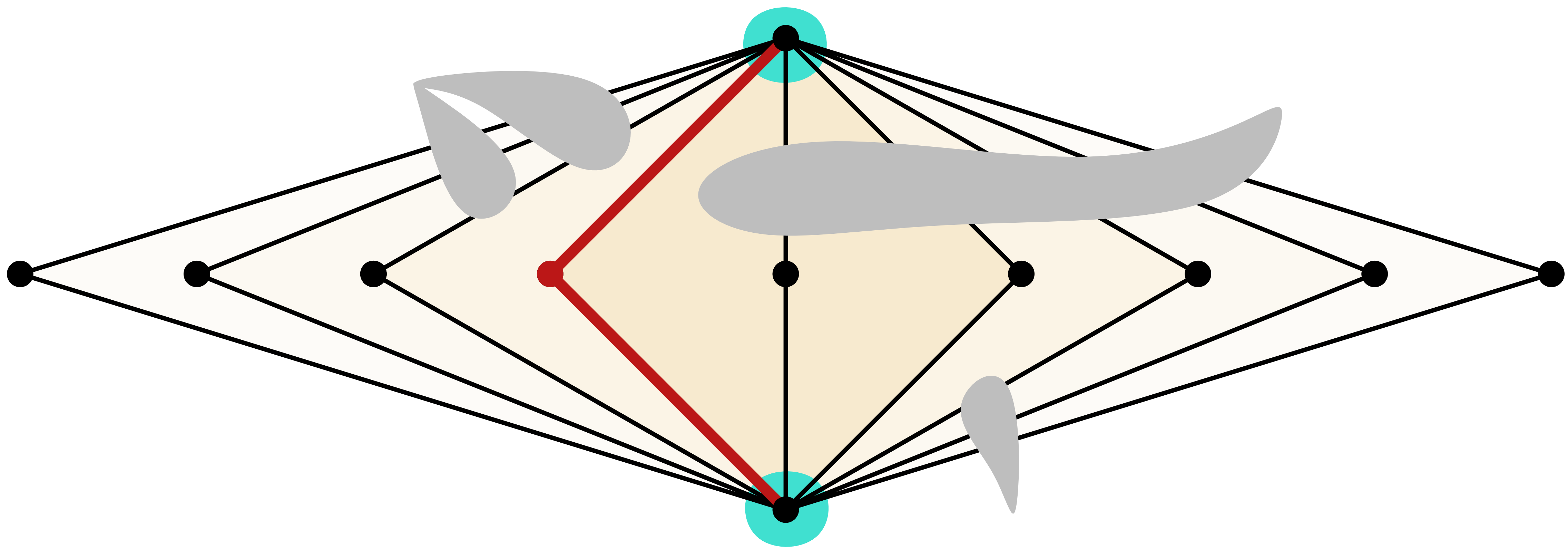
Case 3: $|T_i| = 2$

- All other vertices of T_i form a hierarchy



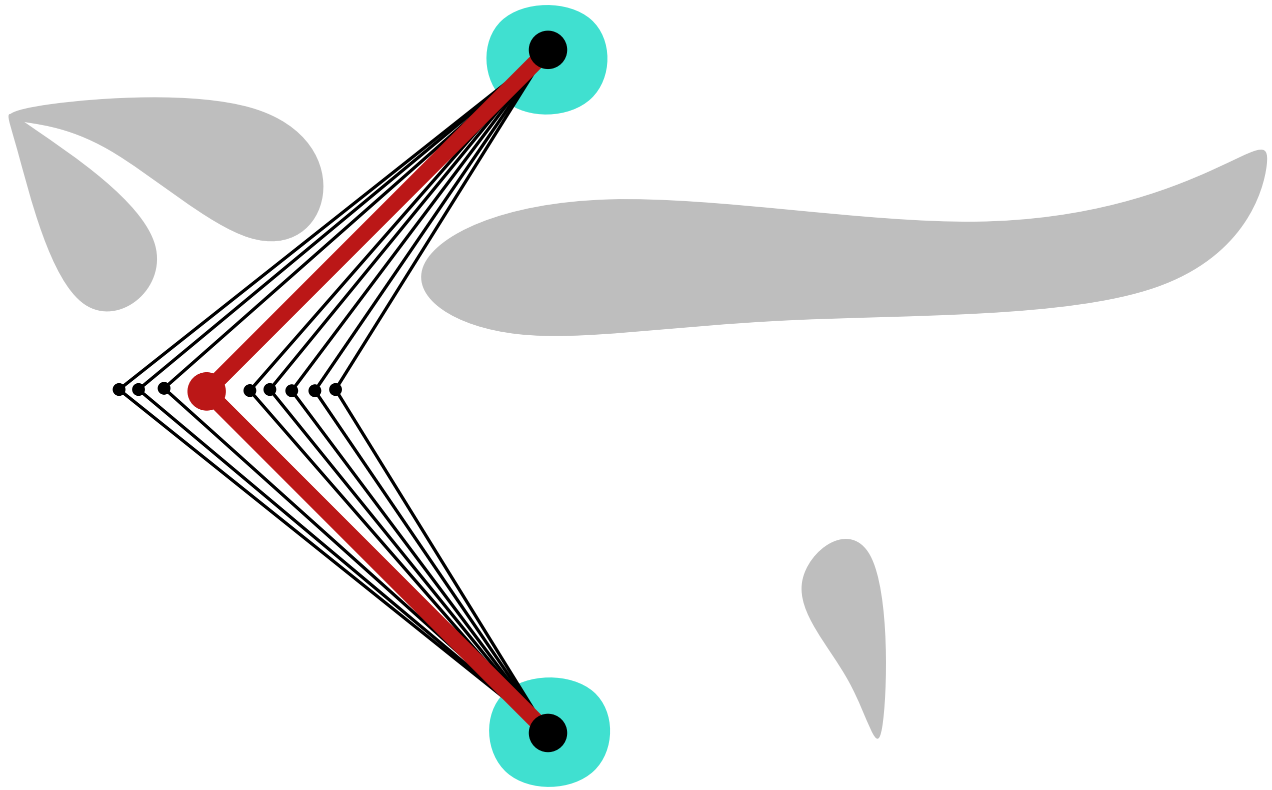
Case 3: $|T_i| = 2$

- All other vertices of T_i form a hierarchy



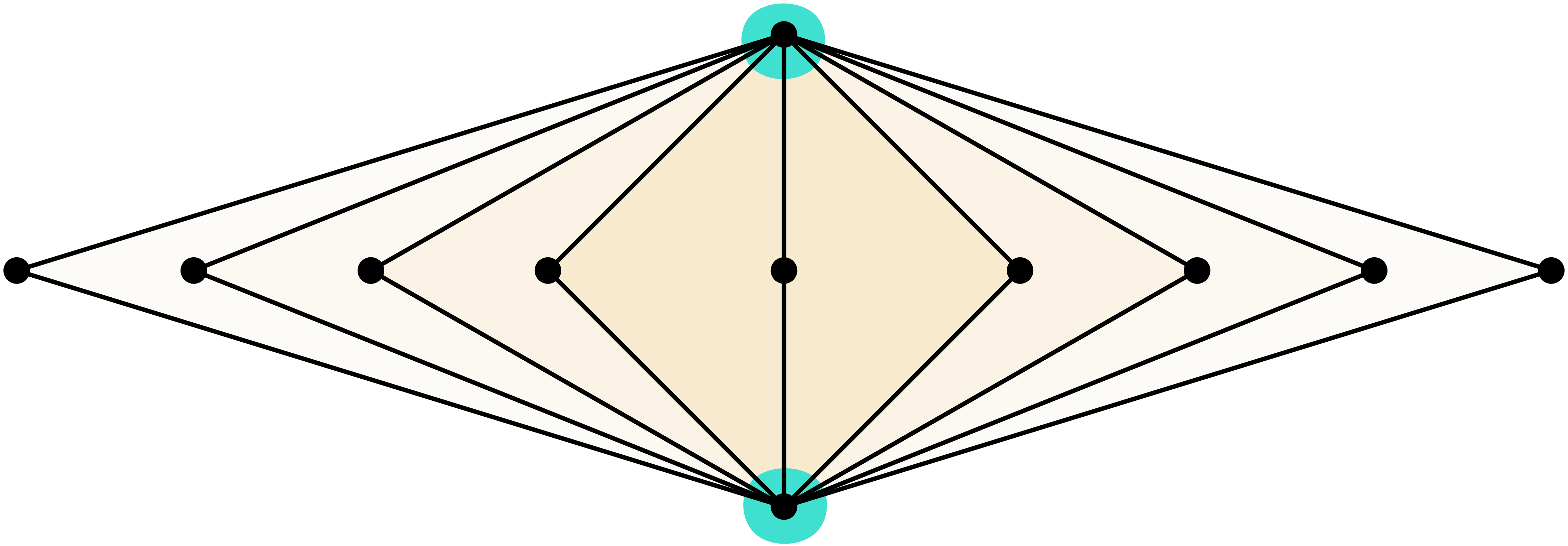
Case 3: $|T_i| = 2$

- All other vertices of T_i form a hierarchy



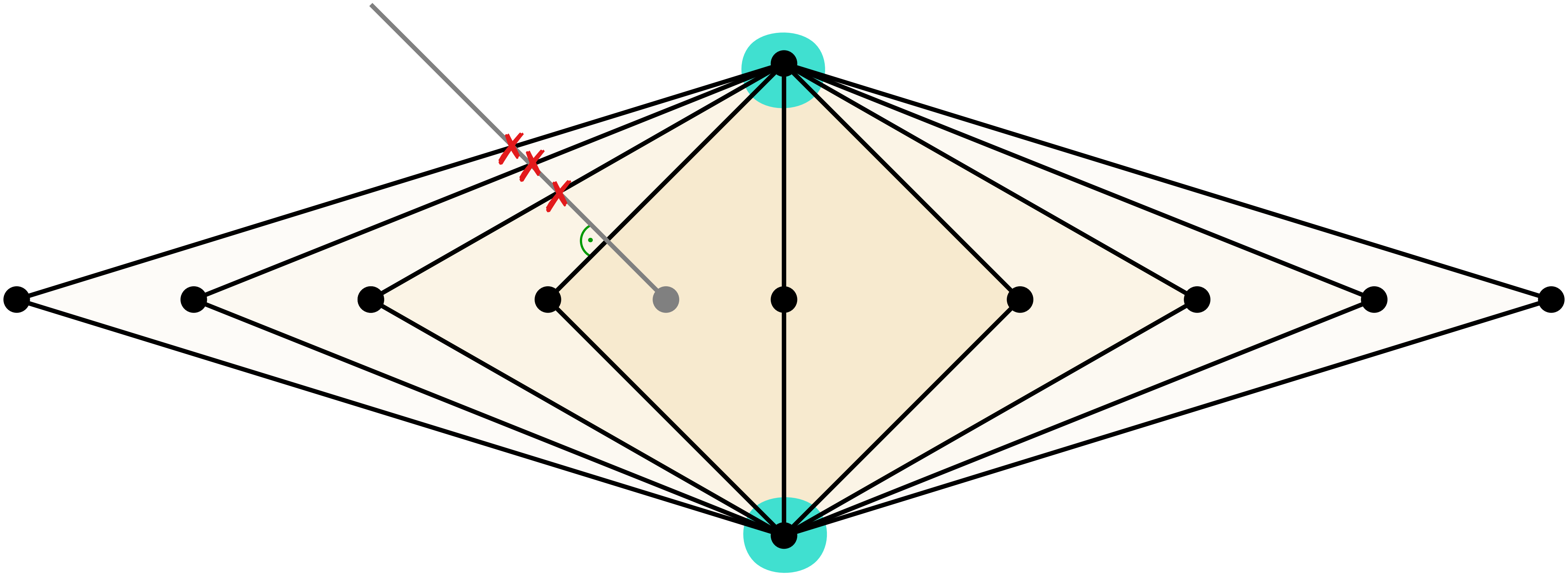
Case 3: $|T_i| = 2$

- All other vertices of T_i form a hierarchy



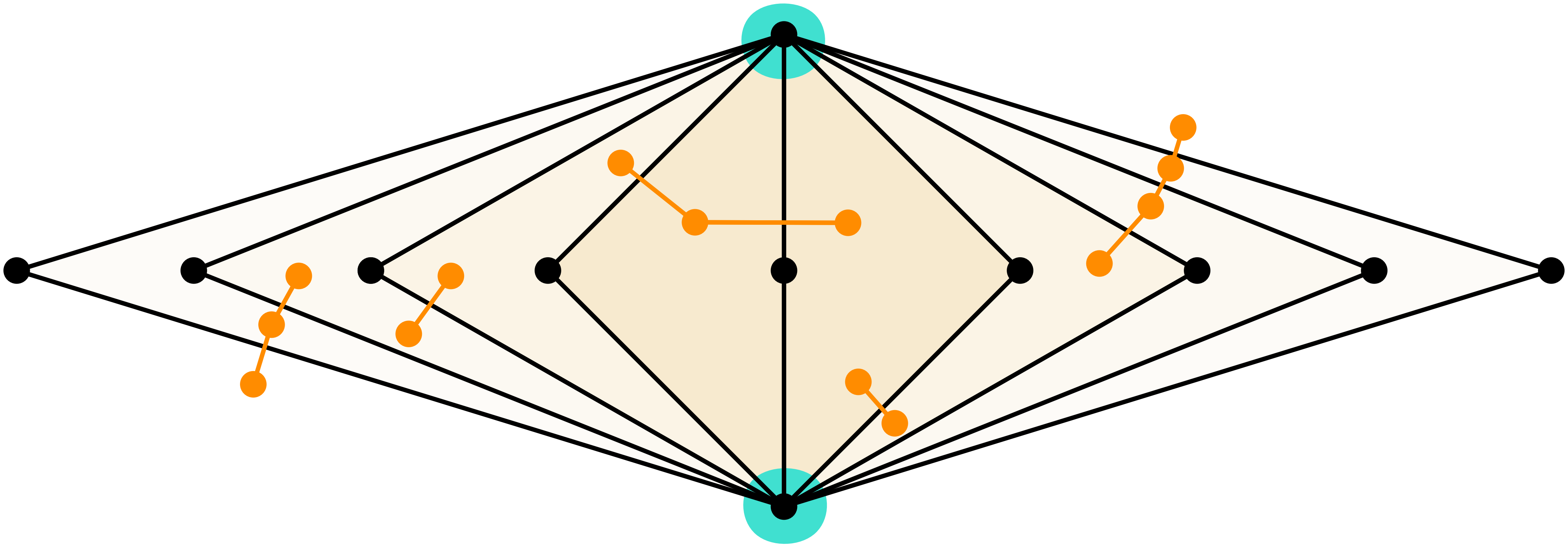
Case 3: $|T_i| = 2$

- All other vertices of T_i form a hierarchy



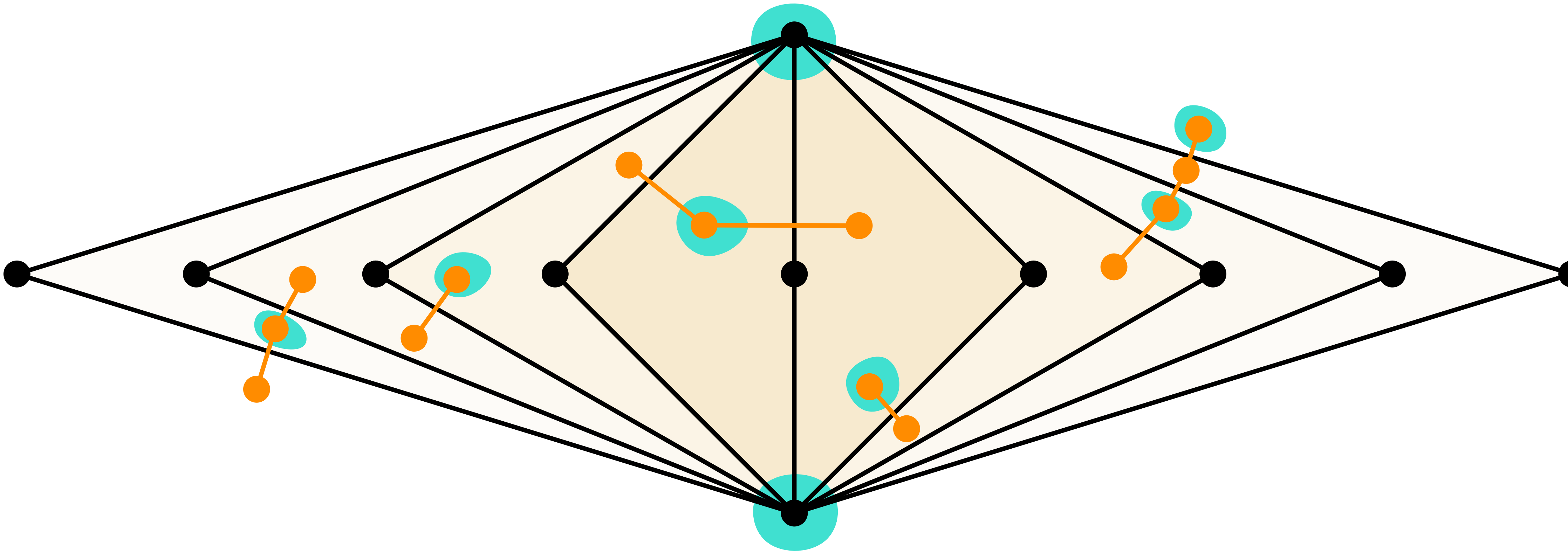
Case 3: $|T_i| = 2$

- All other vertices of T_i form a hierarchy



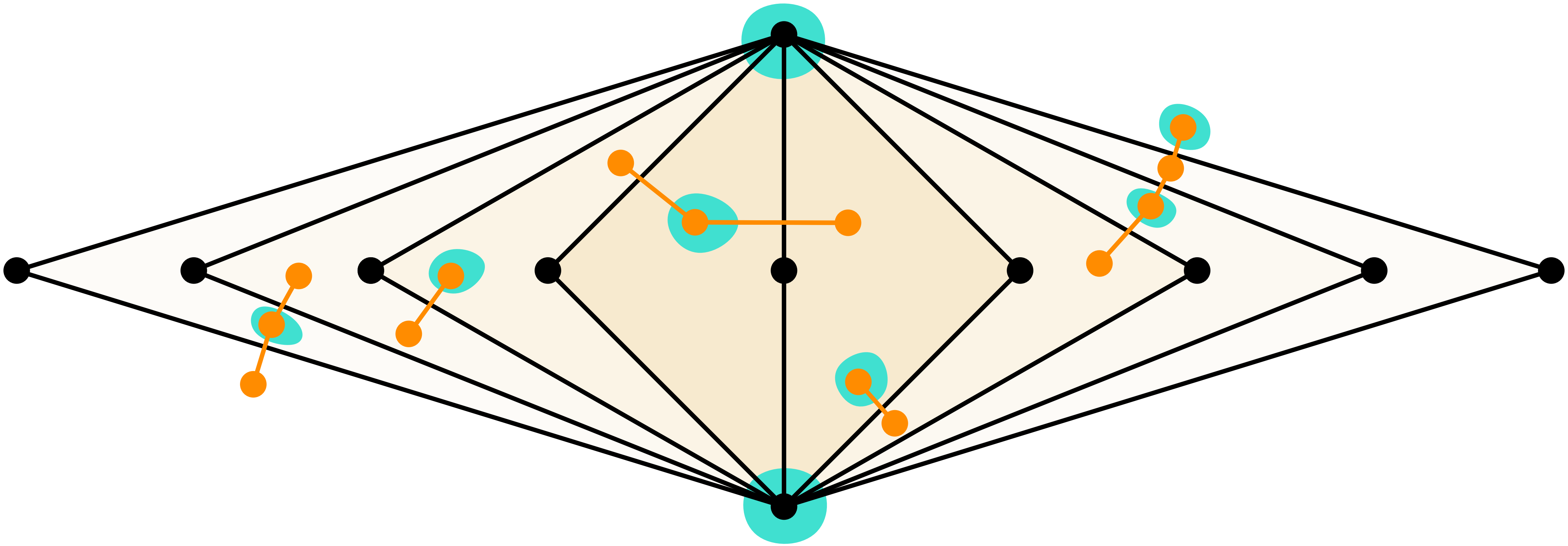
Case 3: $|T_i| = 2$

- All other vertices of T_i form a hierarchy



Case 3: $|T_i| = 2$

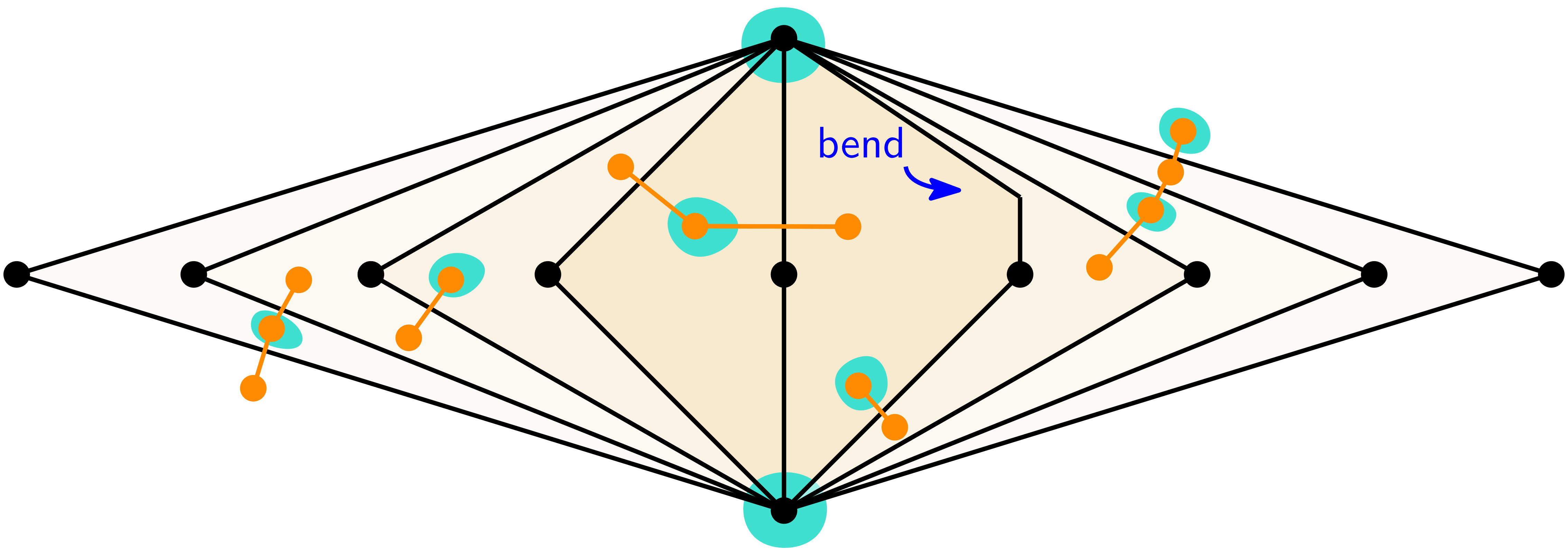
- All other vertices of T_i form a hierarchy



- Crossing-free edge guaranteed with $> 3k$ members

Case 3: $|T_i| = 2$

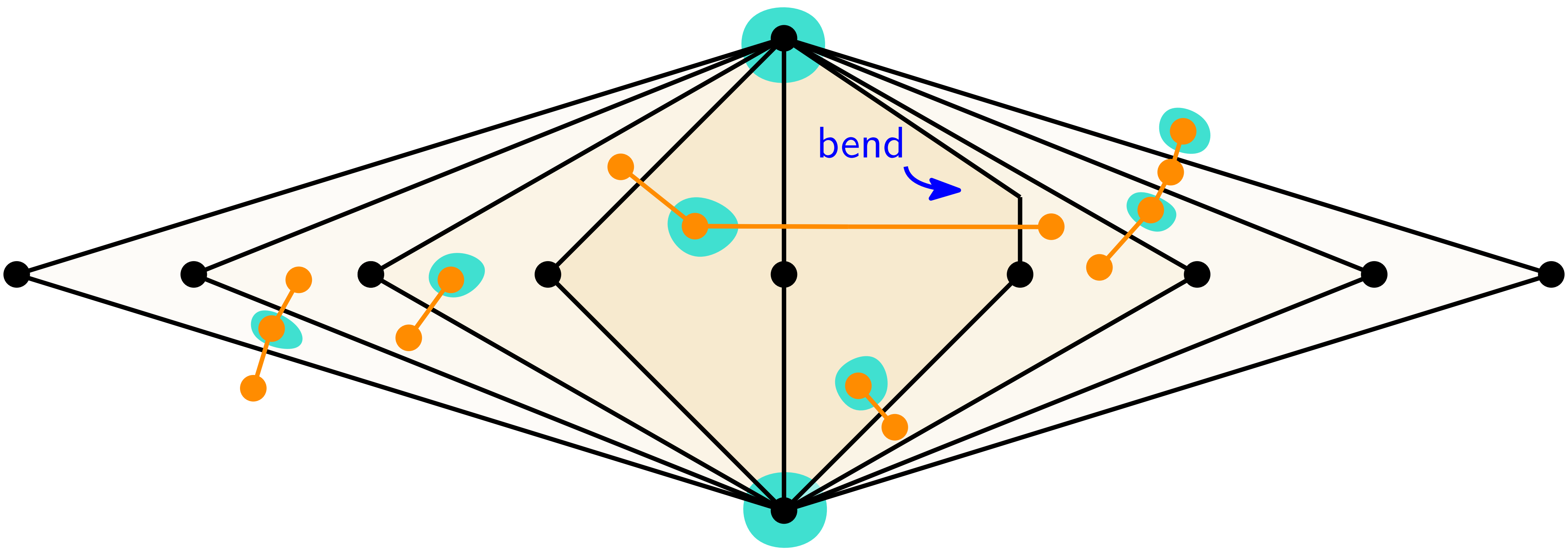
- All other vertices of T_i form a hierarchy



- Crossing-free edge guaranteed with $> 3k$ members

Case 3: $|T_i| = 2$

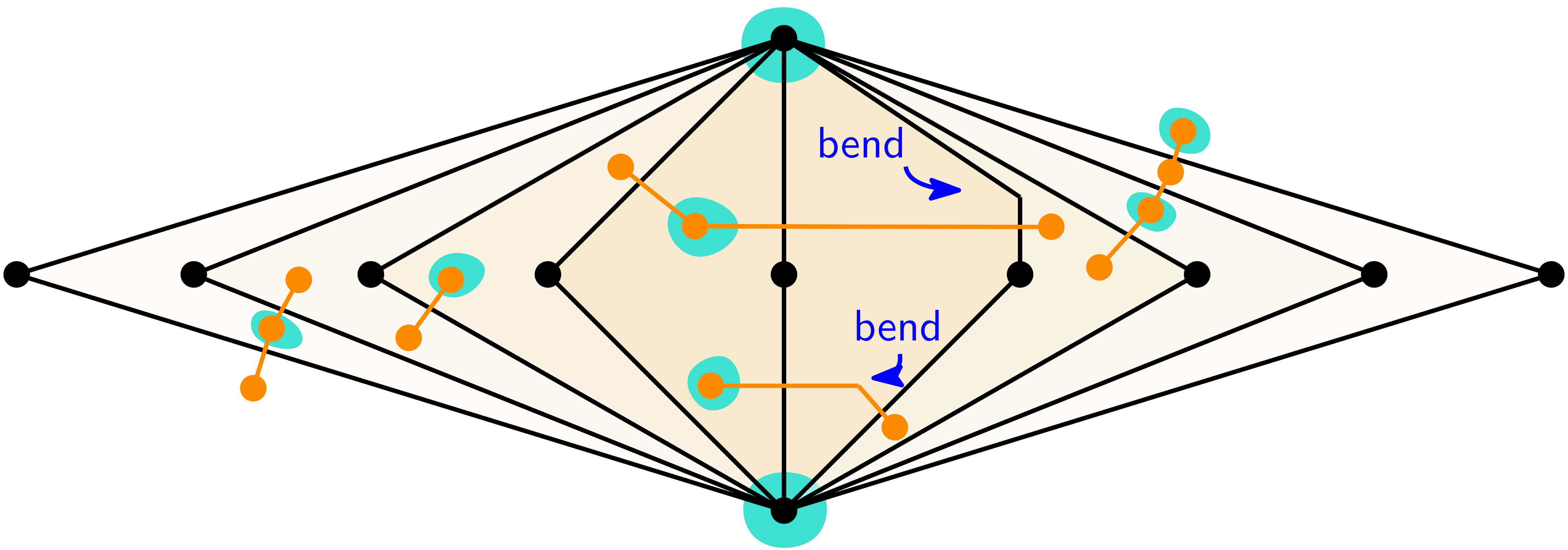
- All other vertices of T_i form a hierarchy



- Crossing-free edge guaranteed with $> 3k$ members

Case 3: $|T_i| = 2$

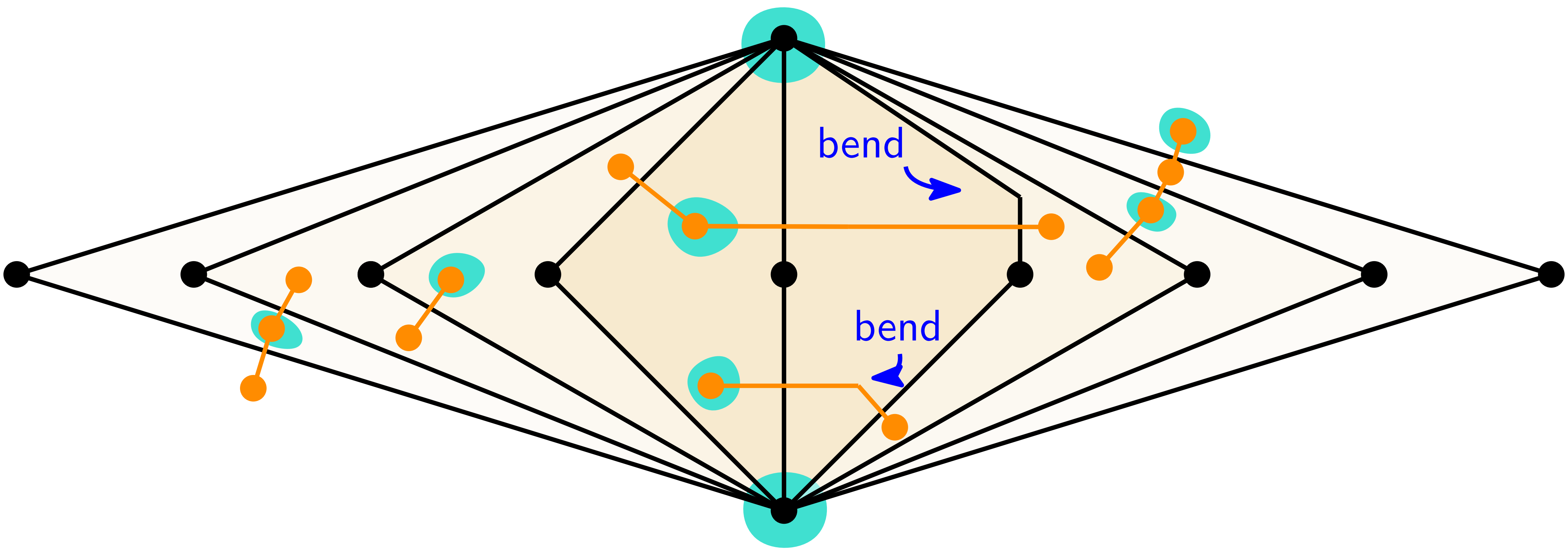
- All other vertices of T_i form a hierarchy



- Crossing-free edge guaranteed with $> 3k$ members

Case 3: $|T_i| = 2$

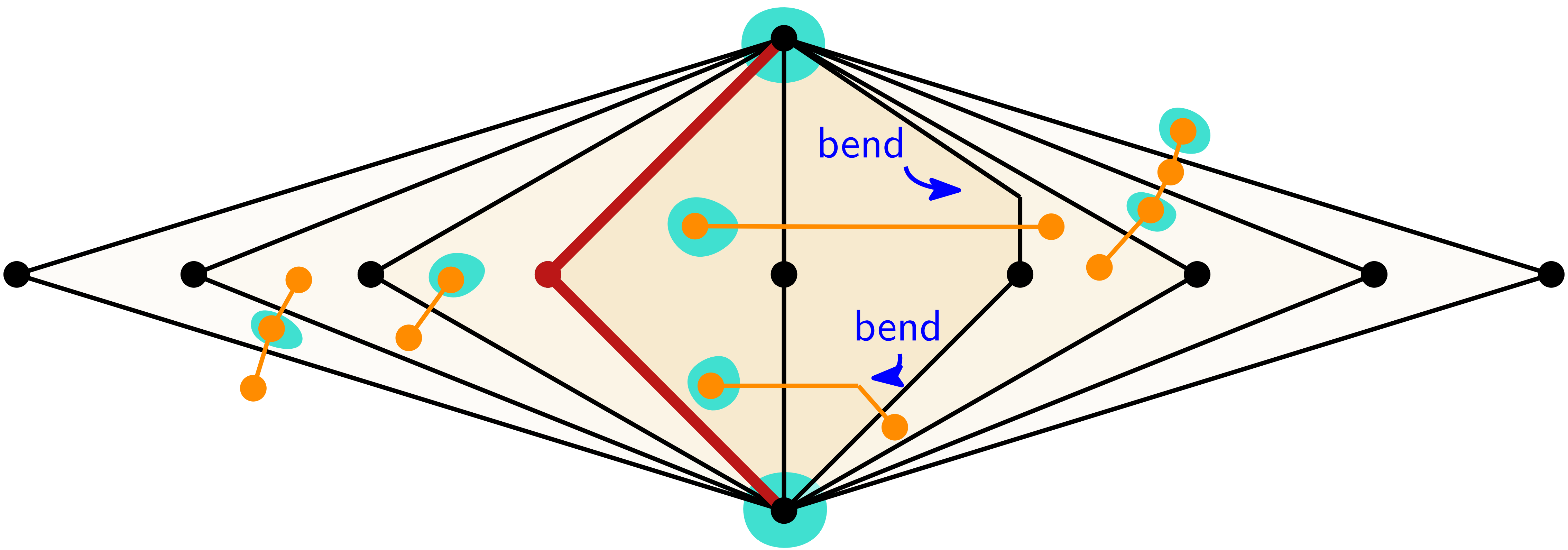
- All other vertices of T_i form a hierarchy



- Crossing-free edge guaranteed with $> 3k$ $+b$ bends members

Case 3: $|T_i| = 2$

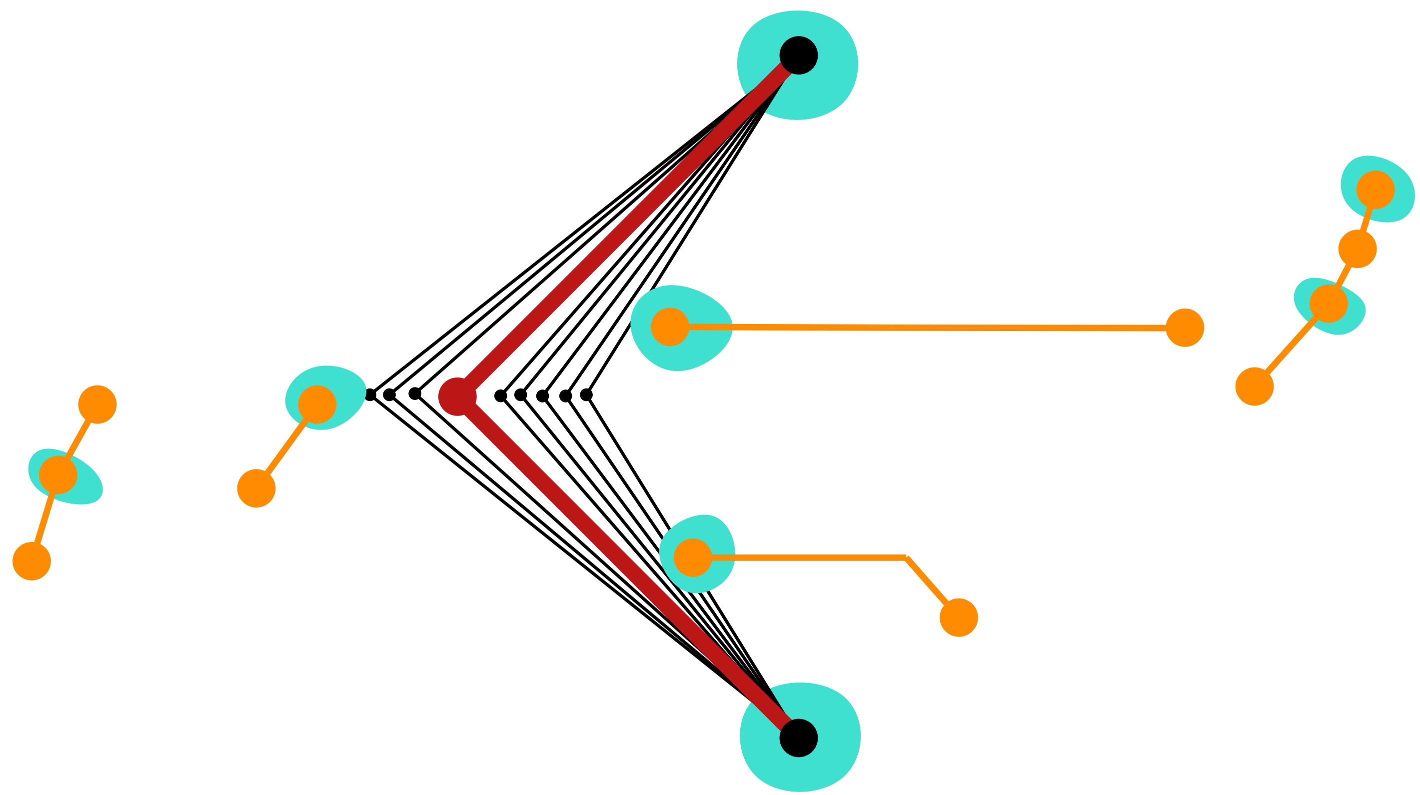
- All other vertices of T_i form a hierarchy



- Crossing-free edge guaranteed with $> 3k$ $+b$ bends members

Case 3: $|T_i| = 2$

- All other vertices of T_i form a hierarchy



- Crossing-free edge guaranteed with $> 3k$ $+b$ bends members

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

Case 3: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

Case 3: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

Total: $|V| =$

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

Case 3: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

Total: $|V| = \underbrace{k}_{\mathbf{vcn}(G)}$

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

Case 3: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

Total: $|V| = \underbrace{k}_{\mathbf{vcn}(G)} + k \cdot \underbrace{0}_{\text{Case 1}}$

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

▶ **Case 1**: $|T_i| = 1$ $\Rightarrow$ keep 0

▶ **Case 2**: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

▶ **Case 3**: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

Total: $|V| = \underbrace{k}_{\mathbf{vcn}(G)} + k \cdot \underbrace{0}_{\text{Case 1}} + 2^k \cdot \underbrace{(5+b)}_{\text{Case 2}}$

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

Case 3: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

Total: $|V| = \underbrace{k}_{\mathbf{vcn}(G)} + k \cdot \underbrace{0}_{\text{Case 1}} + 2^k \cdot \underbrace{(5 + b)}_{\text{Case 2}} + k^2 \cdot \underbrace{(3k + b)}_{\text{Case 3}}$

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

Case 3: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

Total: $|V| = \underbrace{k}_{\mathbf{vcn}(G)} + k \cdot \underbrace{0}_{\text{Case 1}} + 2^k \cdot \underbrace{(5 + b)}_{\text{Case 2}} + k^2 \cdot \underbrace{(3k + b)}_{\text{Case 3}} = \mathcal{O}(b \cdot 2^k)$

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

Case 3: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

Total: $|V| = \underbrace{k}_{\mathbf{vcn}(G)} + k \cdot \underbrace{0}_{\text{Case 1}} + 2^k \cdot \underbrace{(5+b)}_{\text{Case 2}} + k^2 \cdot \underbrace{(3k+b)}_{\text{Case 3}} = \mathcal{O}(b \cdot 2^k)$

Solve time $m^{\mathcal{O}(m^2)}$

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

▶ **Case 1**: $|T_i| = 1$ $\Rightarrow$ keep 0

▶ **Case 2**: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

▶ **Case 3**: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

Total: $|V| = \underbrace{k}_{\mathbf{vcn}(G)} + k \cdot \underbrace{0}_{\text{Case 1}} + 2^k \cdot \underbrace{(5 + b)}_{\text{Case 2}} + k^2 \cdot \underbrace{(3k + b)}_{\text{Case 3}} = \mathcal{O}(b \cdot 2^k)$

Solve time $m^{\mathcal{O}(m^2)}$

Kernel size $\mathcal{O}(b \cdot 2^k)$

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

Case 1: $|T_i| = 1$ $\Rightarrow$ keep 0

Case 2: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$

Case 3: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

Total: $|V| = \underbrace{k}_{\mathbf{vcn}(G)} + k \cdot \underbrace{0}_{\text{Case 1}} + 2^k \cdot \underbrace{(5 + b)}_{\text{Case 2}} + k^2 \cdot \underbrace{(3k + b)}_{\text{Case 3}} = \mathcal{O}(b \cdot 2^k)$

Solve time	Kernel size	Kernel time
$m^{\mathcal{O}(m^2)}$	$\mathcal{O}(b \cdot 2^k)$	$\mathcal{O}(\lvert E(G)\rvert)$

FPT via $\mathbf{vcn}(G) + b$

- Distinguish types by cardinality $|T_i|$

- **Case 1**: $|T_i| = 1$ $\Rightarrow$ keep 0
- **Case 2**: $|T_i| \geq 3$ $\Rightarrow$ keep $5 + b$
- **Case 3**: $|T_i| = 2$ $\Rightarrow$ keep $3k + b$

Total: $|V| = \underbrace{k}_{\mathbf{vcn}(G)} + k \cdot \underbrace{0}_{\text{Case 1}} + 2^k \cdot \underbrace{(5+b)}_{\text{Case 2}} + k^2 \cdot \underbrace{(3k+b)}_{\text{Case 3}} = \mathcal{O}(b \cdot 2^k)$

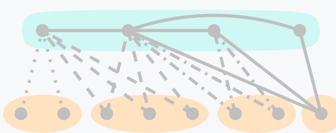
Solve time	Kernel size	Kernel time
$m^{\mathcal{O}(m^2)}$	$\mathcal{O}(b \cdot 2^k)$	$\mathcal{O}(\lvert E(G)\rvert)$

Total $2^{2^{\mathcal{O}(k+\log b)}} + \mathcal{O}(|E(G)|)$

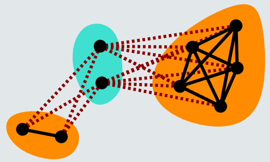
Results

- b-BEND β-RESTRICTED RAC DRAWING (BRAC) is *fixed-parameter tractable* when parameterized by

Feedback edge number $\mathbf{fen}(G)$ | $2^{\mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}} + \mathcal{O}(|E(G)|)$



Vertex cover number $\mathbf{vcn}(G) + b$ | $2^{2^{\mathcal{O}(\mathbf{vcn}(G)+\log b)}} + \mathcal{O}(|E(G)|)$



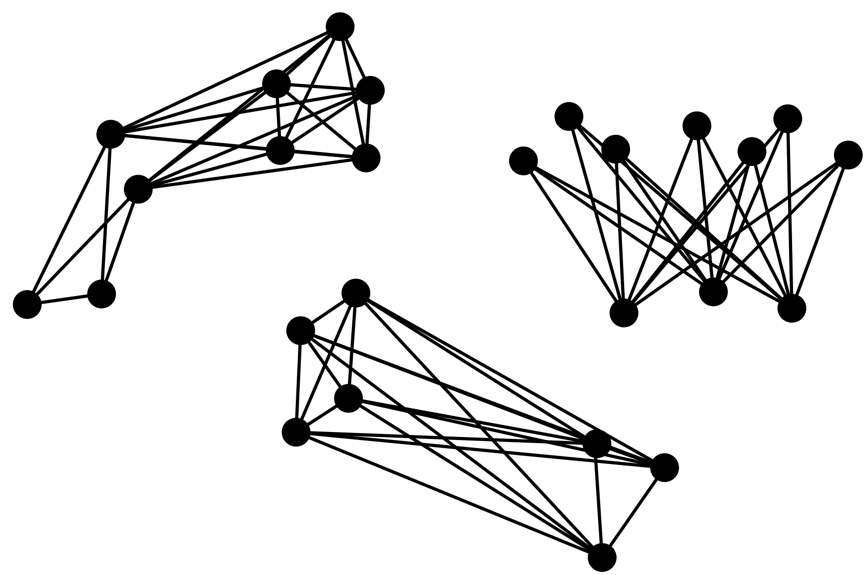
Neighborhood diversity $\mathbf{nd}(G) + b$ | $2^{b^{\mathcal{O}(\mathbf{nd}(G))}} + \mathcal{O}(|E(G)|)$

Neighbourhood Diversity

Definition. **Neighborhood diversity** $\mathbf{nd}(G)$: *min* k *s.t.* $\exists$ k*-partition* neighborhood equivalent for each vertex in same partition

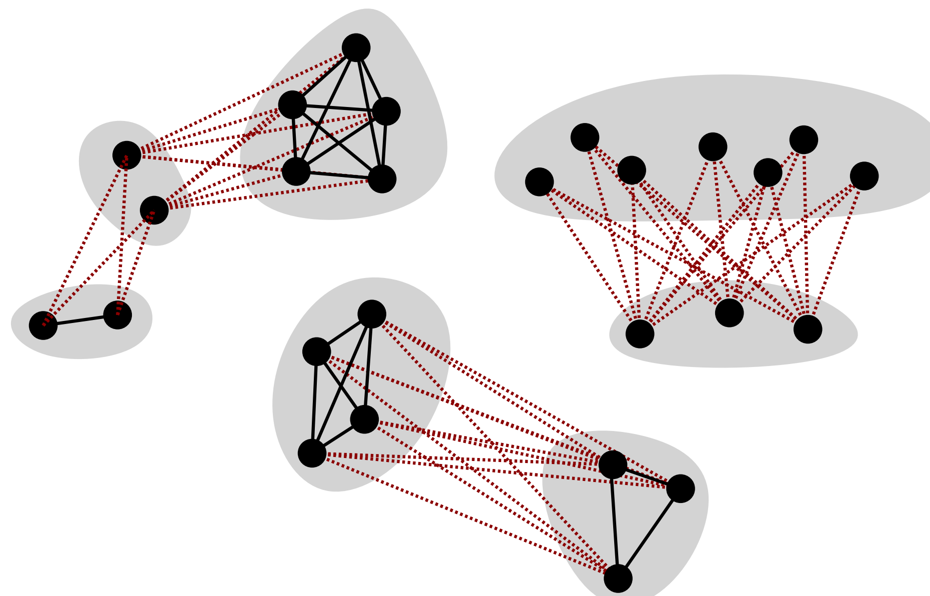
Neighbourhood Diversity

Definition. **Neighborhood diversity** $\mathbf{nd}(G)$: *min* k *s.t.* $\exists$ k*-partition* neighborhood equivalent for each vertex in same partition



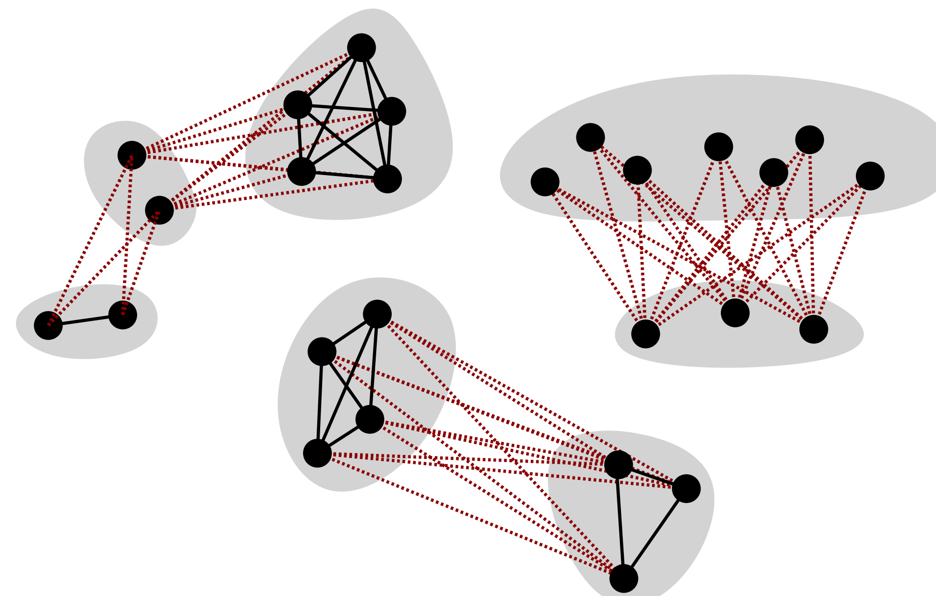
Neighbourhood Diversity

Definition. **Neighborhood diversity** $\mathbf{nd}(G)$: *min* k *s.t.* $\exists$ k*-partition* neighborhood equivalent for each vertex in same partition



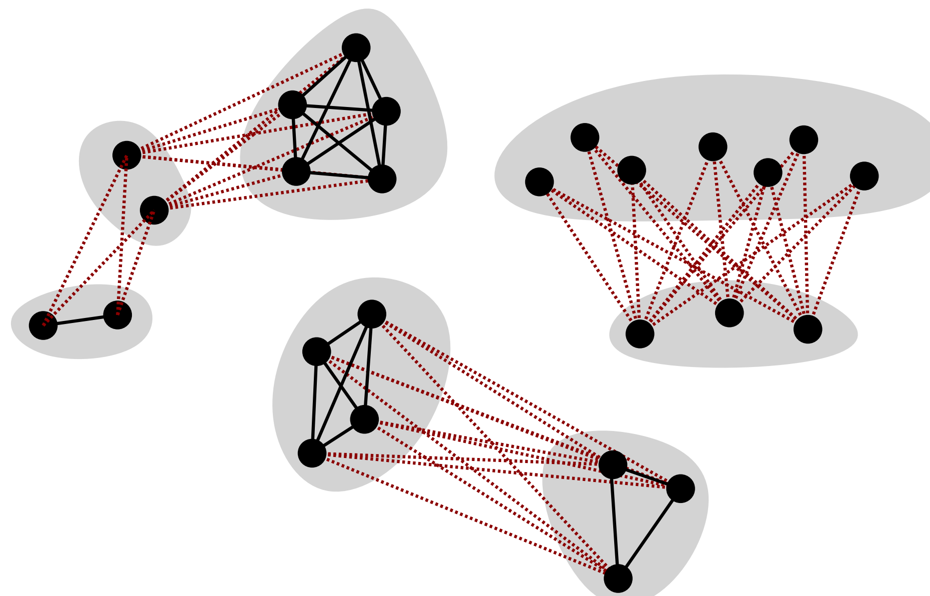
Neighbourhood Diversity

Definition. **Neighborhood diversity** $\mathbf{nd}(G)$: *min* k *s.t.* $\exists$ k*-partition* neighborhood equivalent for each vertex in same partition



$\mathbf{nd}(G) = 7$

Definition. **Neighborhood diversity** $\mathbf{nd}(G)$: *min* k *s.t.* $\exists$ k*-partition* neighborhood equivalent for each vertex in same partition

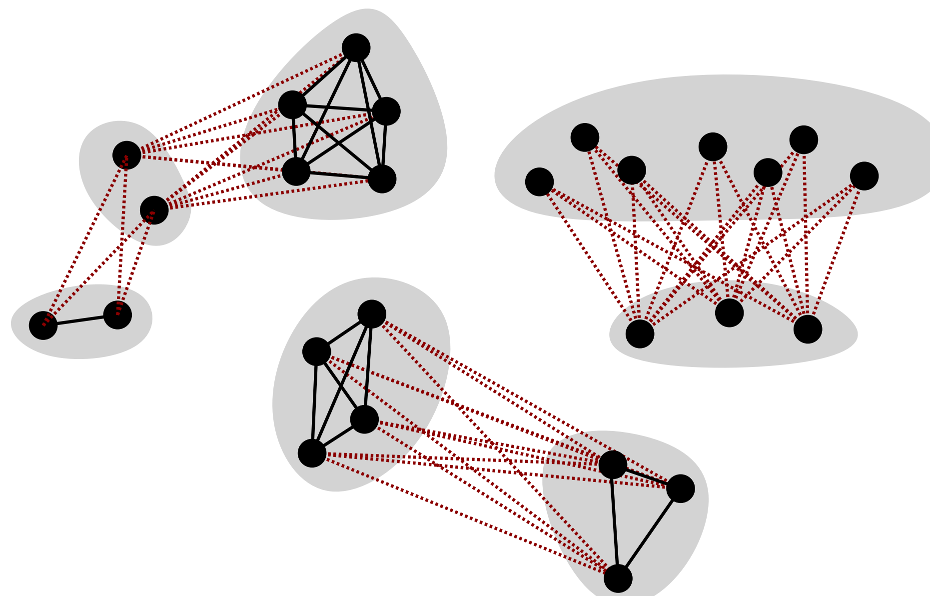


$\mathbf{nd}(G) = 7$

- $\mathbf{nd}(G) \leq f(\mathbf{vcn}(G))$

Neighbourhood Diversity

Definition. **Neighborhood diversity** $\mathbf{nd}(G)$: *min* k *s.t.* $\exists$ k*-partition* neighborhood equivalent for each vertex in same partition



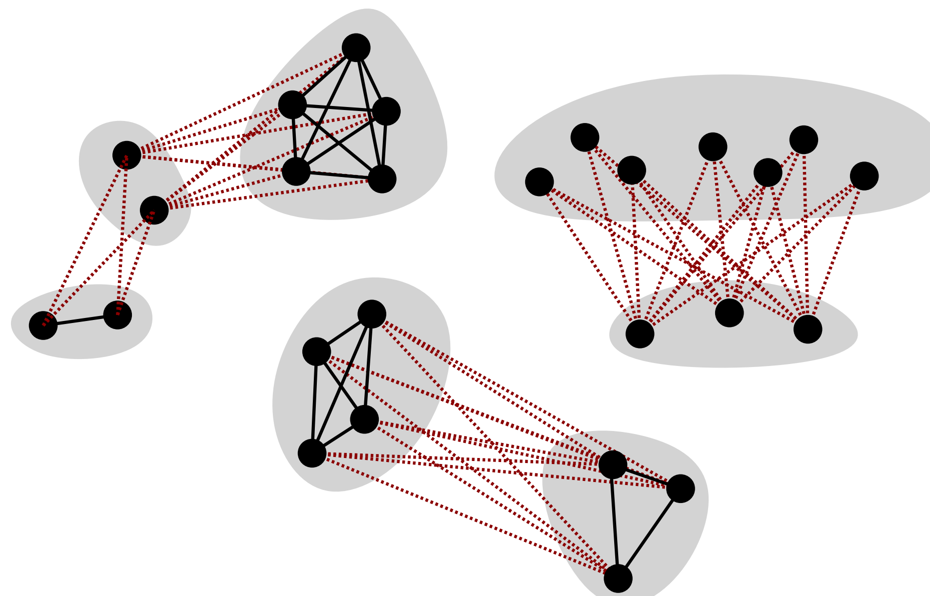
$\mathbf{nd}(G) = 7$

- $\mathbf{nd}(G) \leq f(\mathbf{vcn}(G))$

Lemma. G b-bend RAC drawable $\Rightarrow$ $\mathbf{vcn}(G) \leq 5 \cdot \mathbf{nd}(G) + b$.

Neighbourhood Diversity

Definition. **Neighborhood diversity** $\mathbf{nd}(G)$: *min k s.t. $\exists$ k-partition* neighborhood equivalent for each vertex in same partition



$\mathbf{nd}(G) = 7$

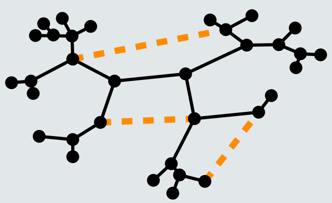
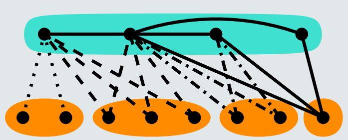
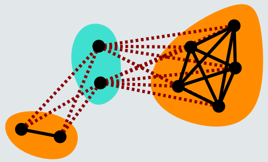
- $\mathbf{nd}(G) \leq f(\mathbf{vcn}(G))$

Lemma. G b-bend RAC drawable $\Rightarrow \mathbf{vcn}(G) \leq 5 \cdot \mathbf{nd}(G) + b$.

- Implies BRAC is FPT by $\mathbf{nd}(G)$

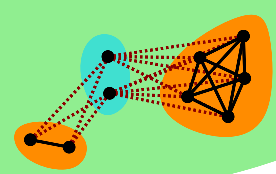
Results

- b-BEND β-RESTRICTED RAC DRAWING (BRAC) is *fixed-parameter tractable* when parameterized by

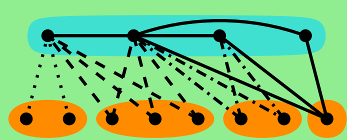
	Feedback edge number $\mathbf{fen}(G)$	$2^{\mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}} + \mathcal{O}(\lvert E(G)\rvert)$
	Vertex cover number $\mathbf{vcn}(G) + b$	$2^{2^{\mathcal{O}(\mathbf{vcn}(G)+\log b)}} + \mathcal{O}(\lvert E(G)\rvert)$
	Neighborhood diversity $\mathbf{nd}(G) + b$	$2^{b^{\mathcal{O}(\mathbf{nd}(G))}} + \mathcal{O}(\lvert E(G)\rvert)$

Open questions

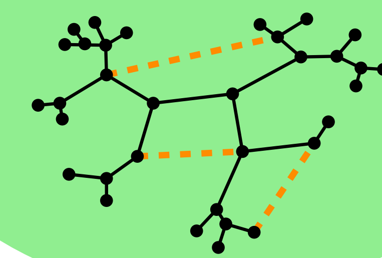
Neighborhood diversity



Vertex cover

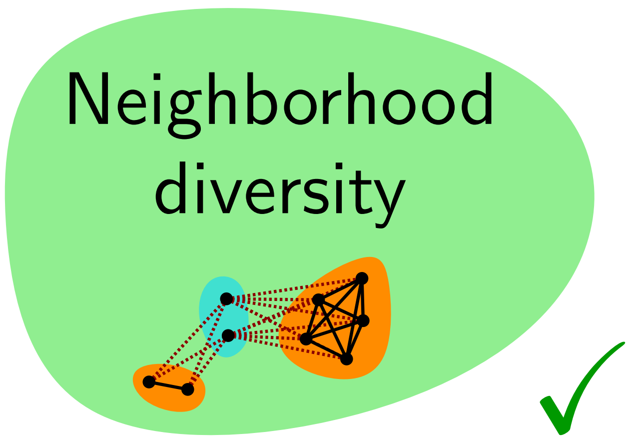


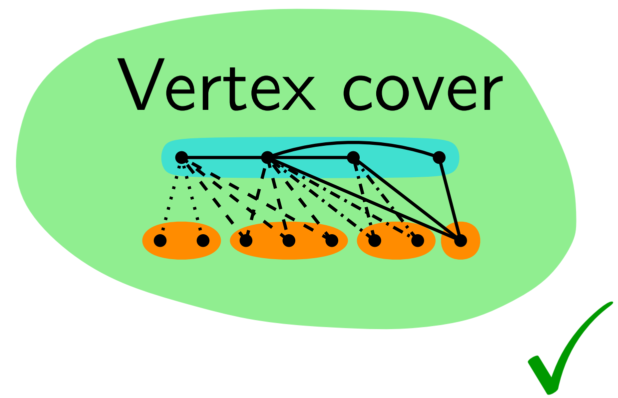
Feedback edge set

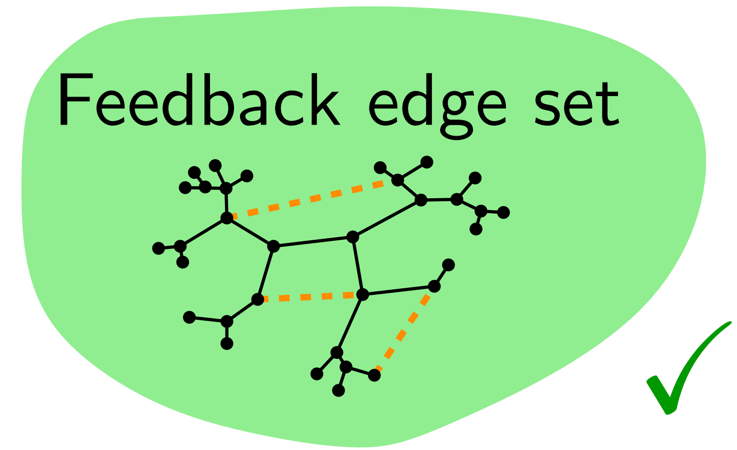


Open questions

- Also FPT by $\mathbf{vcn}(G)$ alone? (instead of $\mathbf{vcn}(G) + b$)

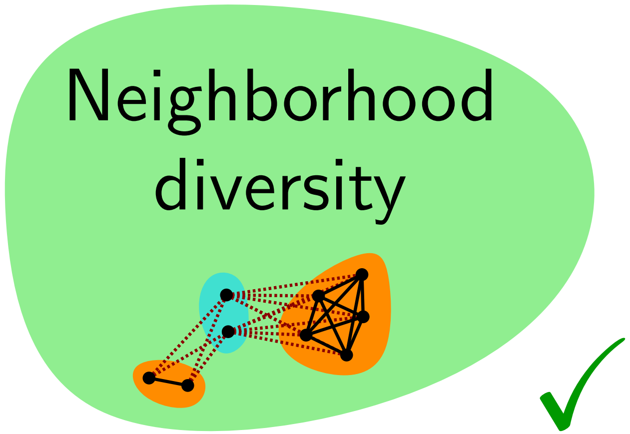


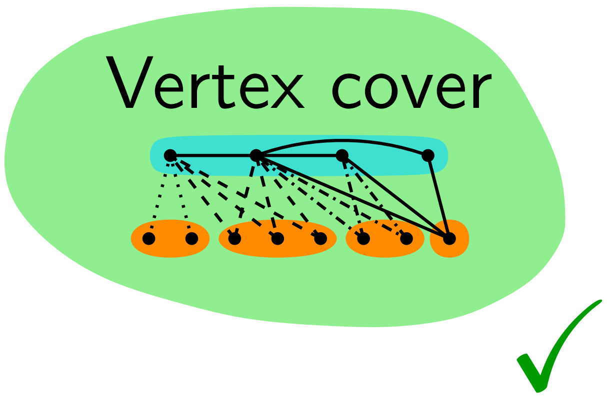


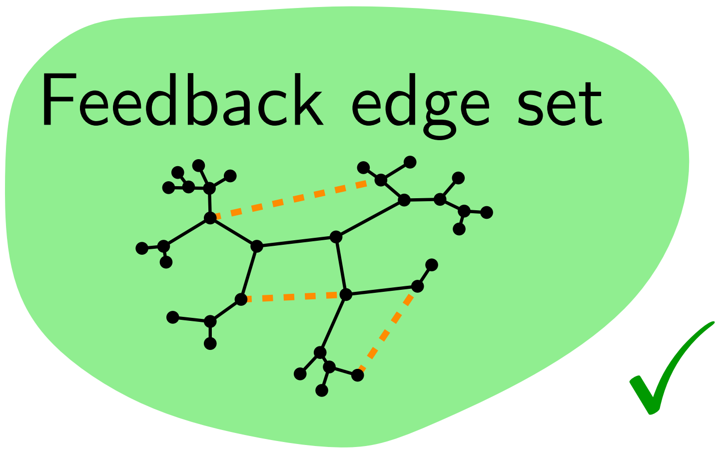


Open questions

- Also FPT by $\mathbf{vcn}(G)$ alone? (instead of $\mathbf{vcn}(G) + b$)
- Obtain smaller (polynomially sized) kernels

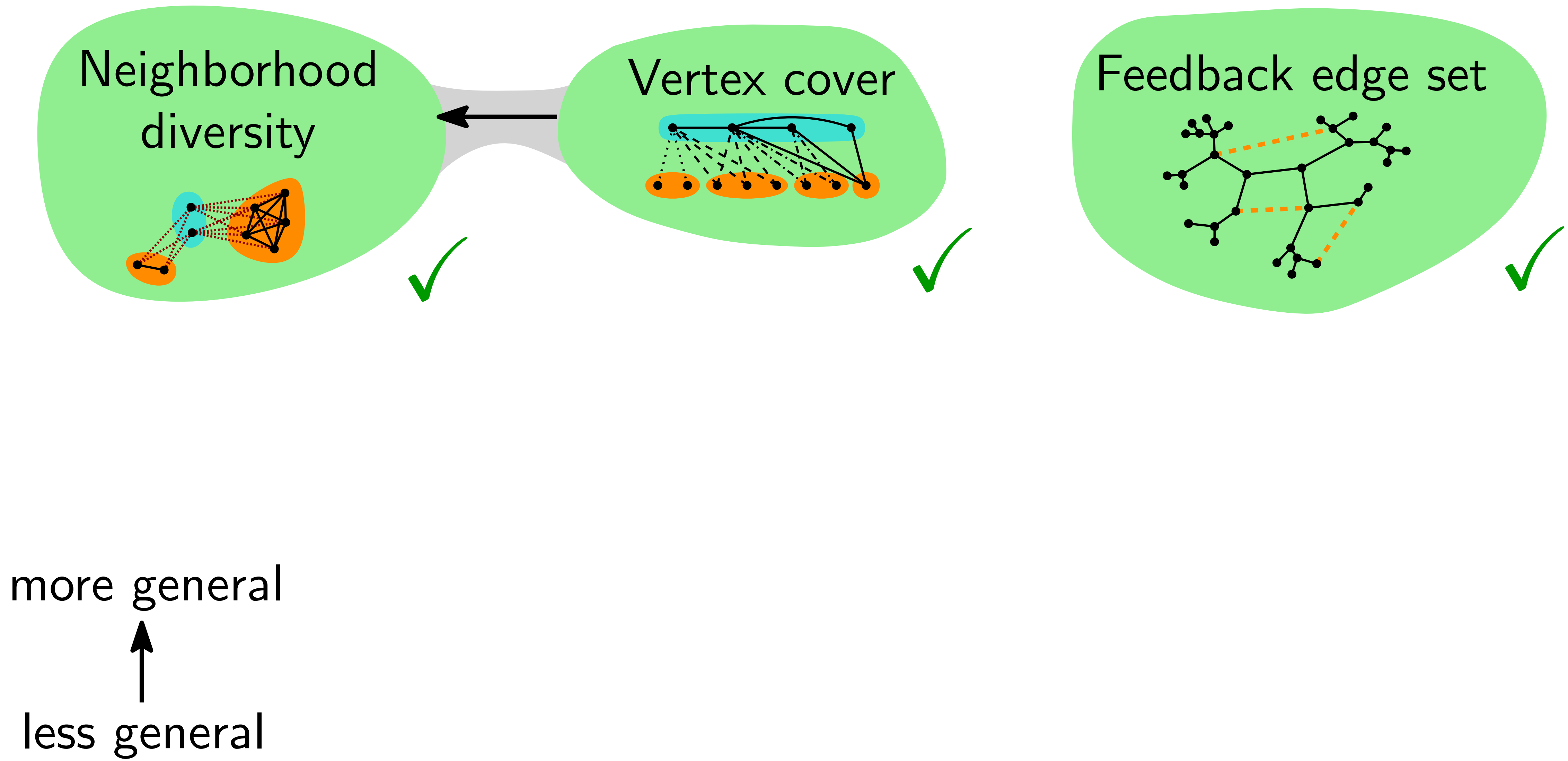






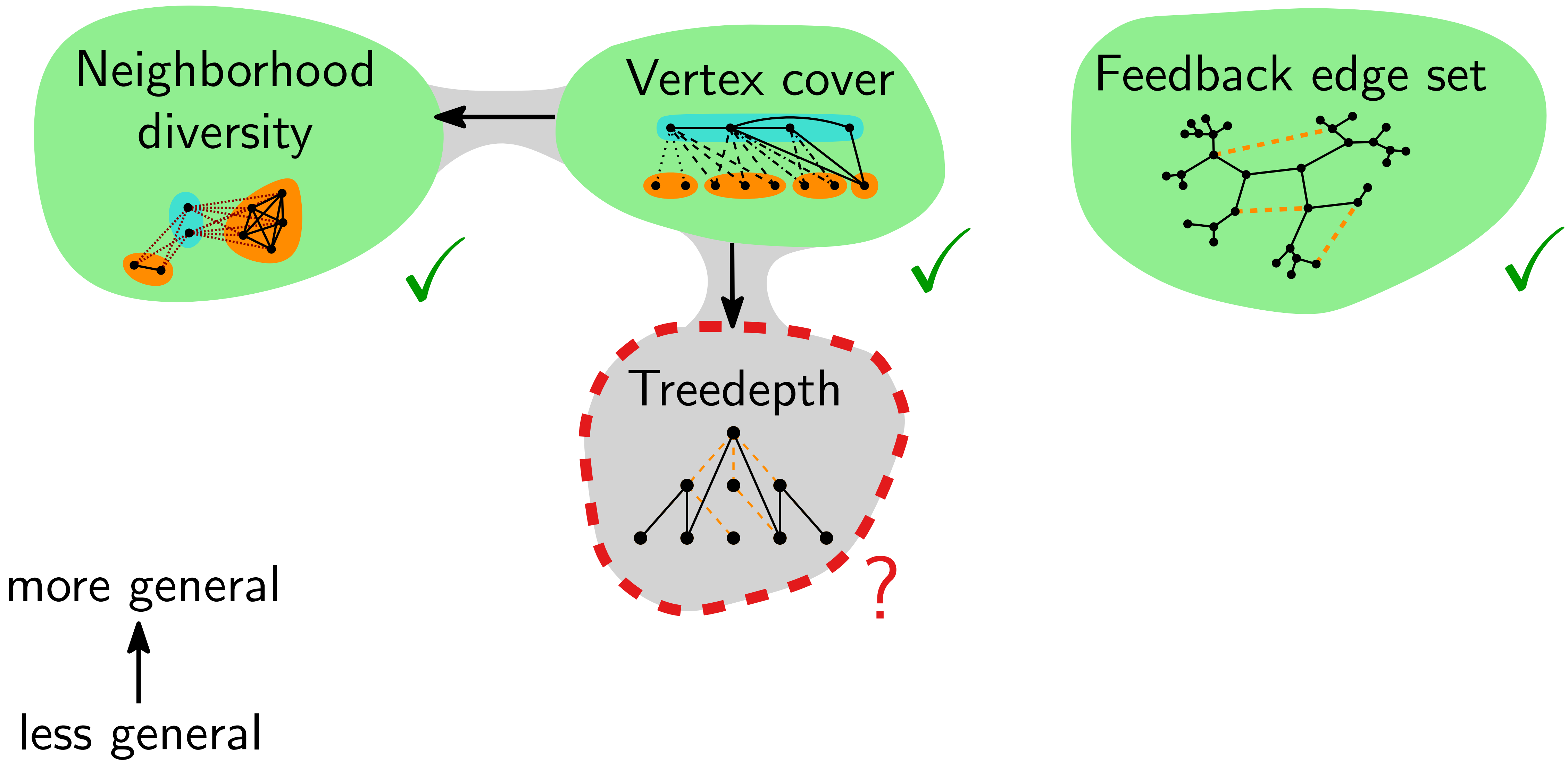
Open questions

- Also FPT by $\mathbf{vcn}(G)$ alone? (instead of $\mathbf{vcn}(G) + b$)
- Obtain smaller (polynomially sized) kernels



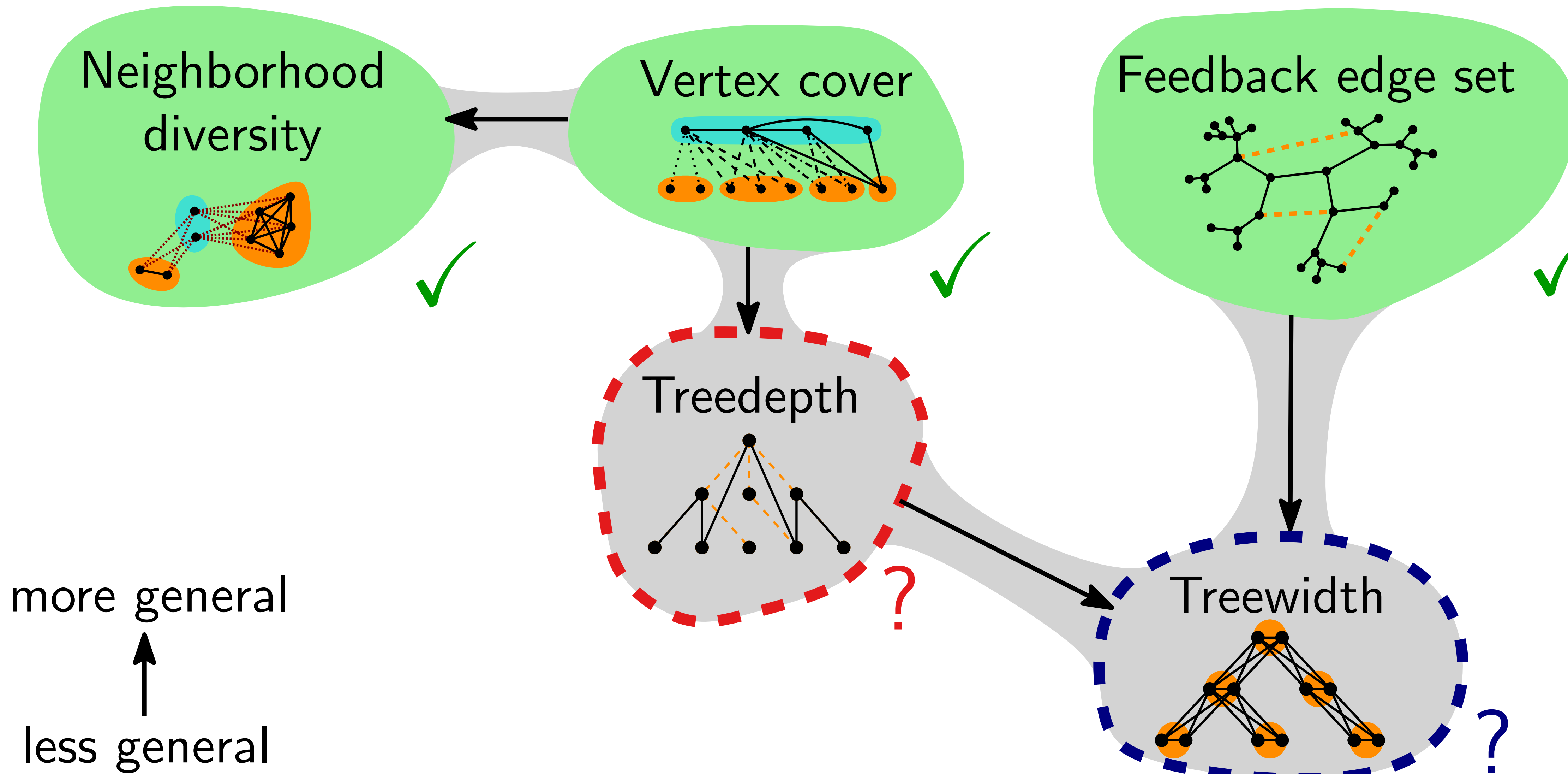
Open questions

- Also FPT by $\mathbf{vcn}(G)$ alone? (instead of $\mathbf{vcn}(G) + b$)
- Obtain smaller (polynomially sized) kernels



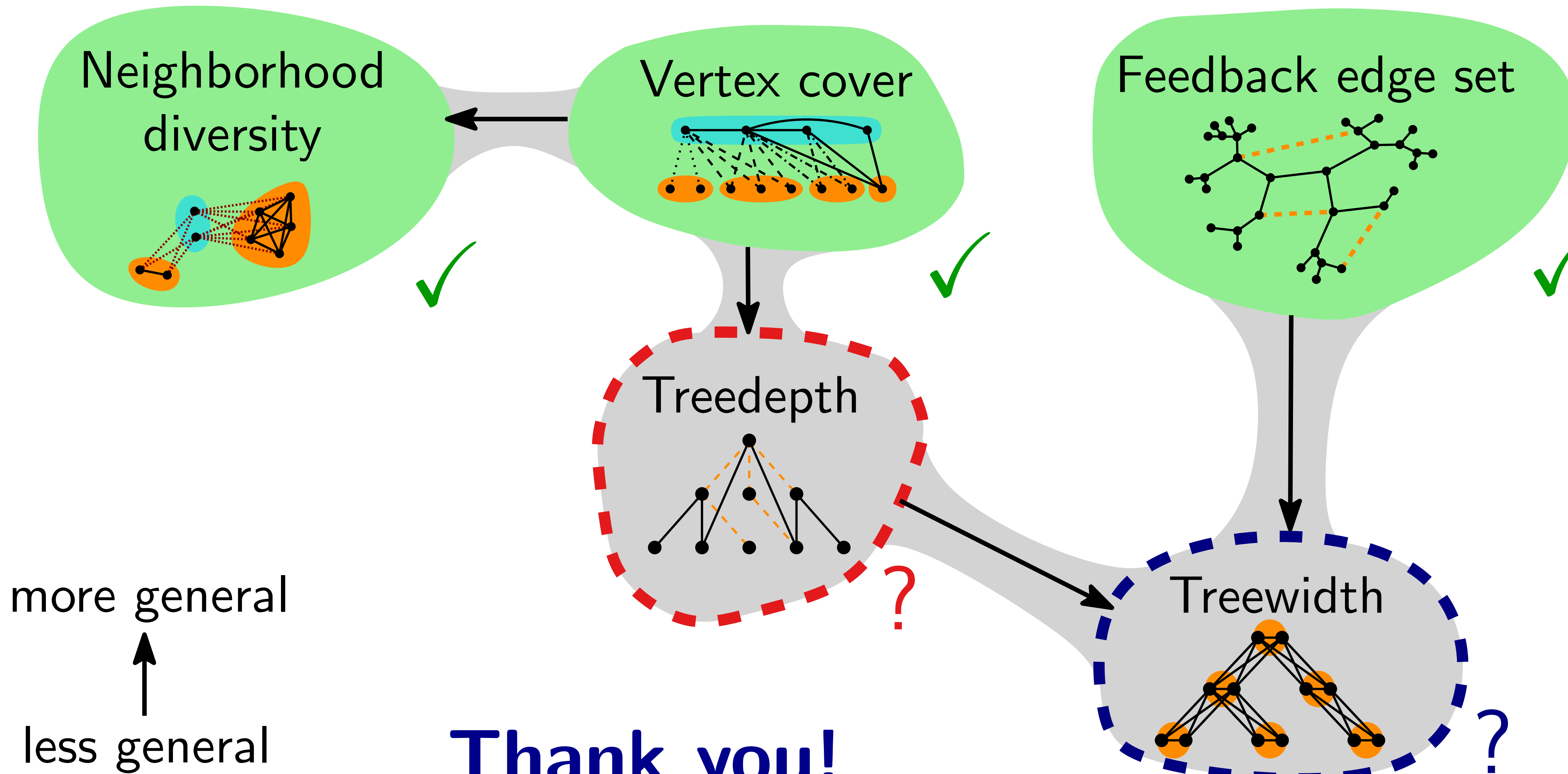
Open questions

- Also FPT by $\mathbf{vcn}(G)$ alone? (instead of $\mathbf{vcn}(G) + b$)
- Obtain smaller (polynomially sized) kernels



Open questions

- Also FPT by $\mathbf{vcn}(G)$ alone? (instead of $\mathbf{vcn}(G) + b$)
- Obtain smaller (polynomially sized) kernels



Thank you!