UNIVERSITÄT TRIER

ALGO

On k-Plane Insertion into Plane Drawings

Julia Katheder

Philipp Kindermann

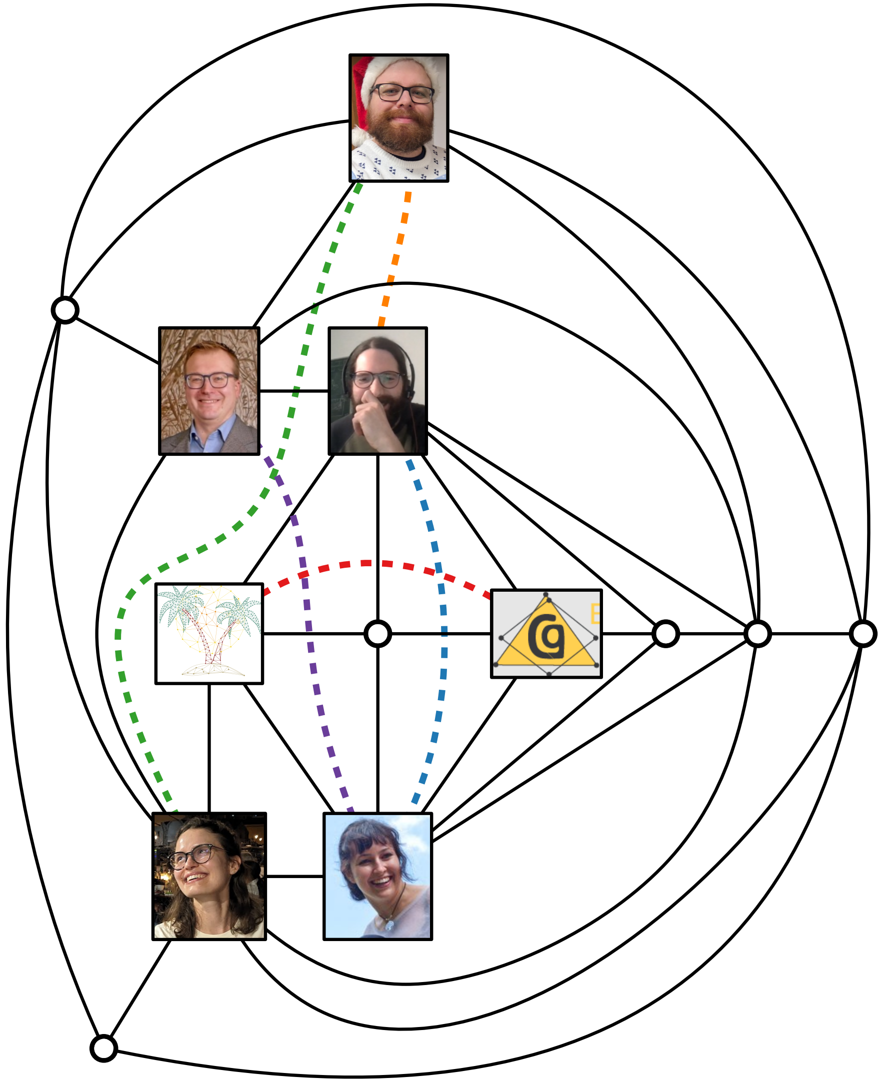
Fabian Klute

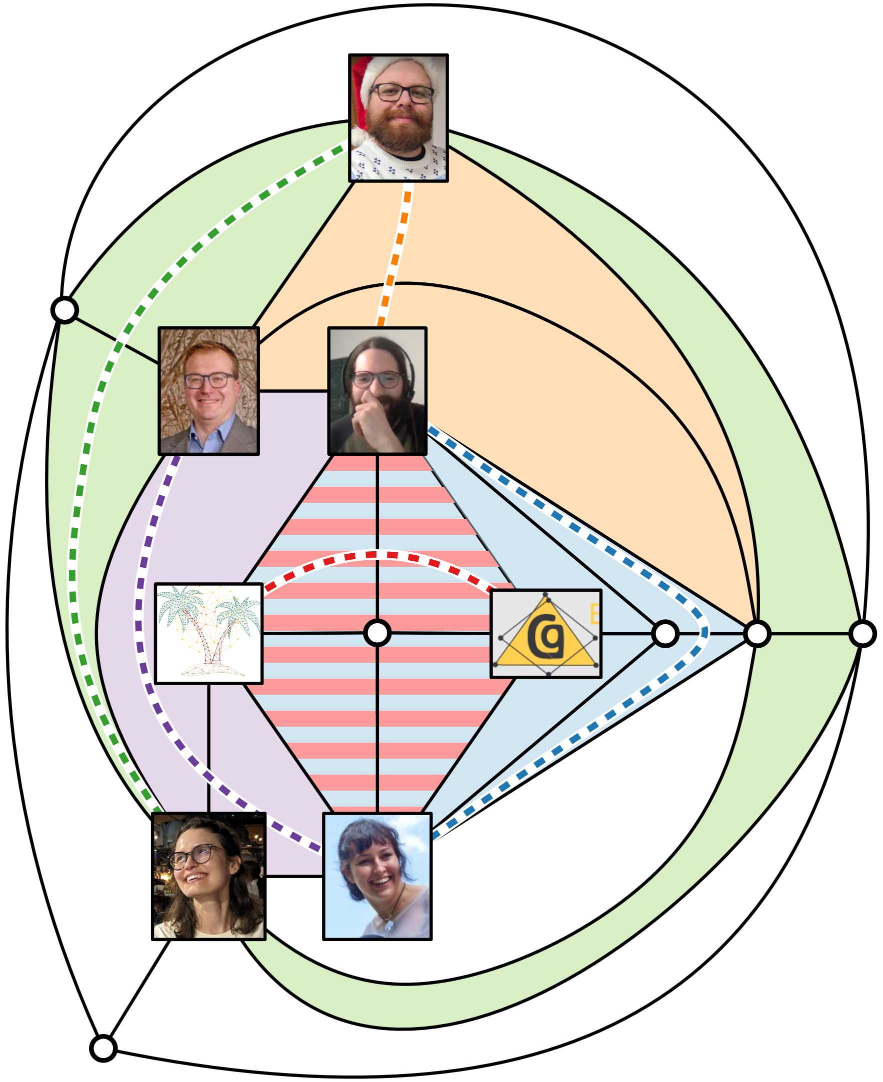
Irene Parada

Ignaz Rutter

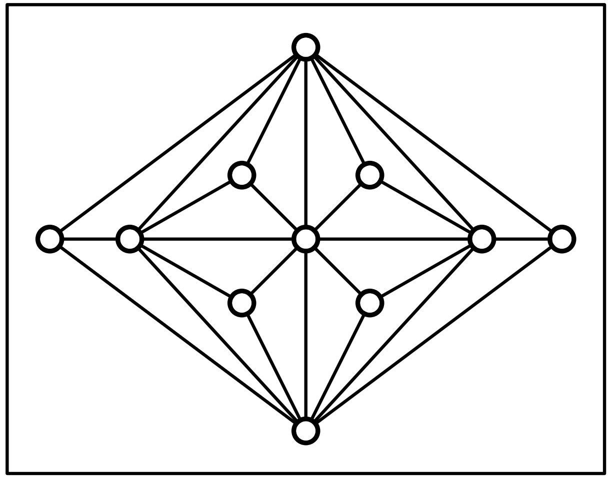
SWGD 2023
Summer Workshop
on Graph Drawing

EuroCG 2024
13-15 March
Ioannina, Greece



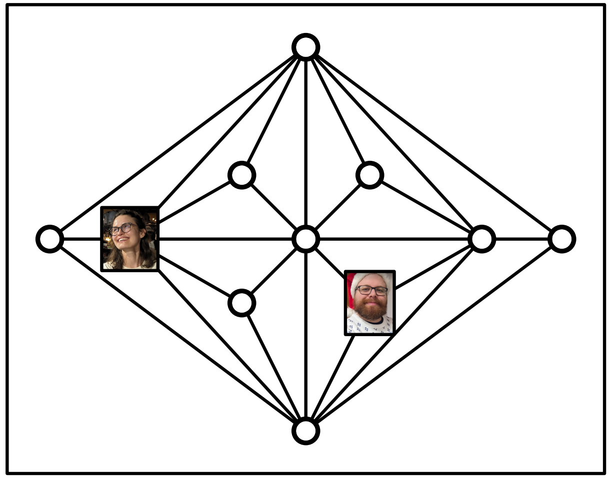


Inserting an Edge Into a Planar Graph



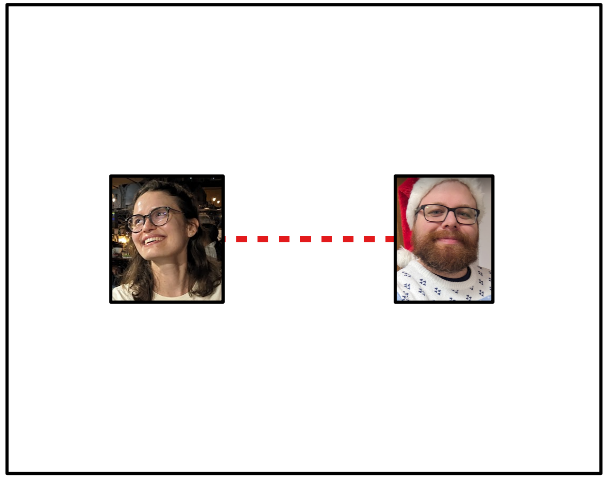
planar graph G

Inserting an Edge Into a Planar Graph



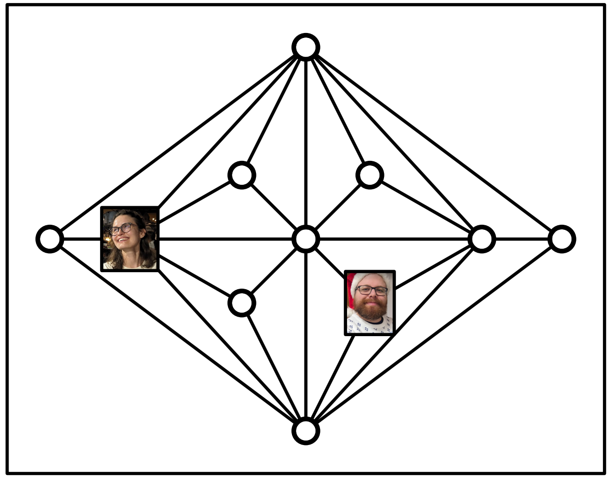
planar graph G

$+$

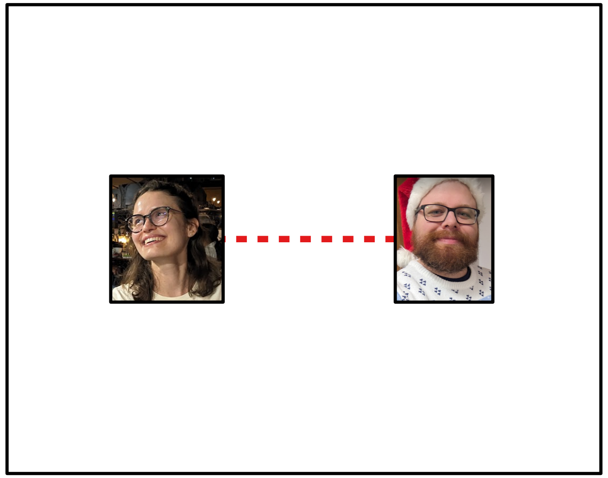


an edge e btw.
2 vertices of G

Inserting an Edge Into a Planar Graph

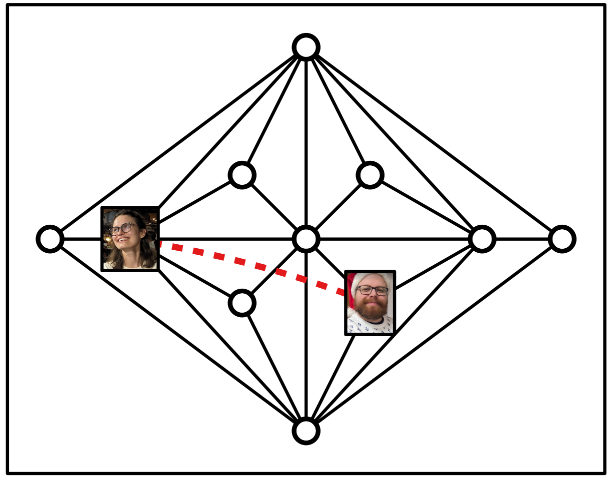


planar graph G



an edge e btw.
2 vertices of G

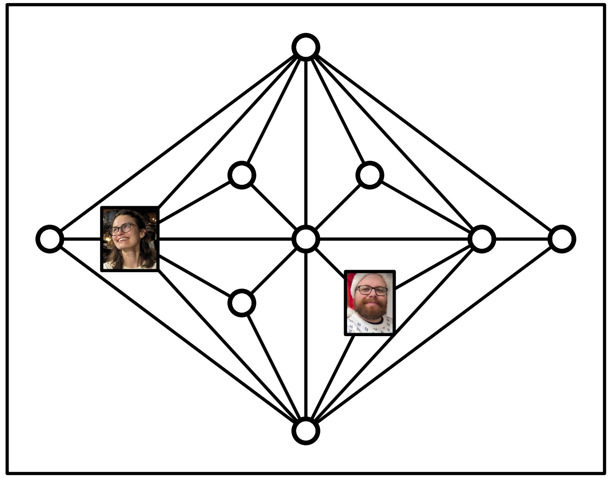
2 crossings



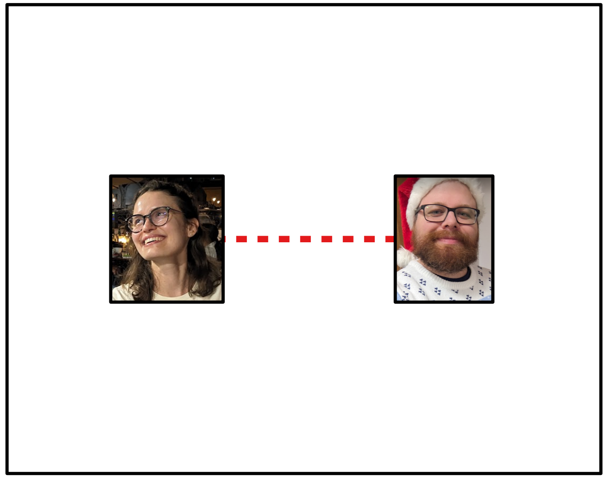
crossing-min. drawing of $G + e$

Inserting an Edge Into a Planar Graph

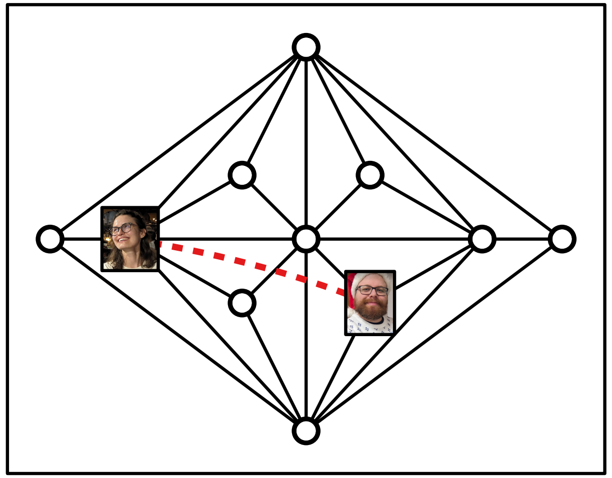
2 crossings



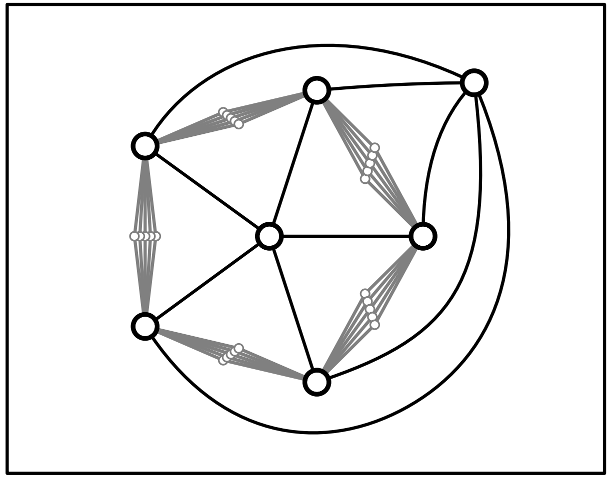
planar graph G



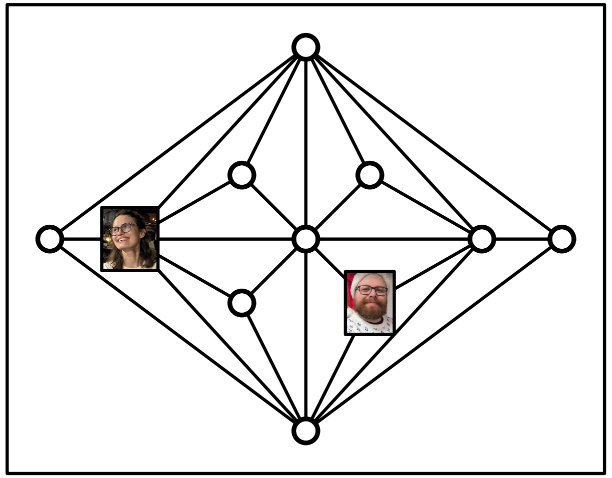
an edge e btw.
2 vertices of G



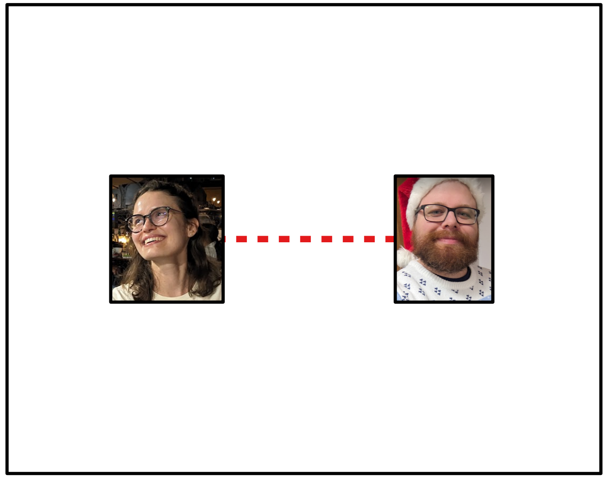
crossing-min. drawing of $G + e$



Inserting an Edge Into a Planar Graph

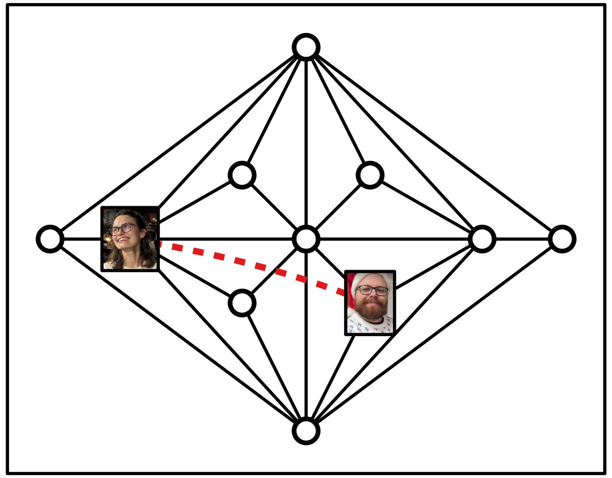


planar graph G

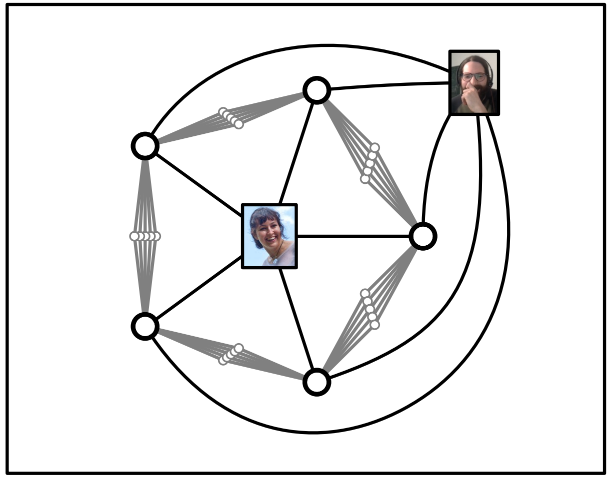


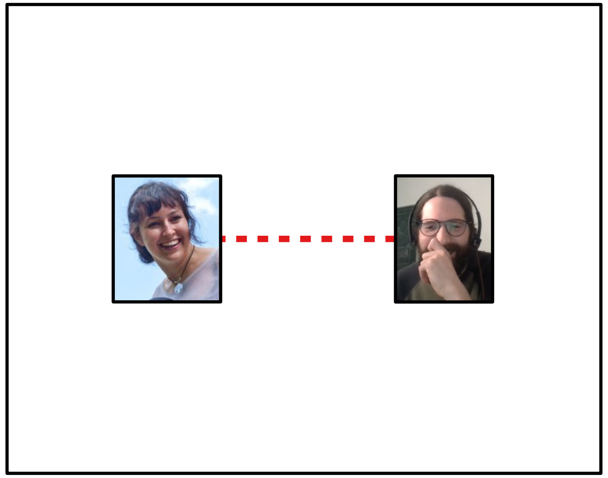
an edge e btw.
2 vertices of G

2 crossings

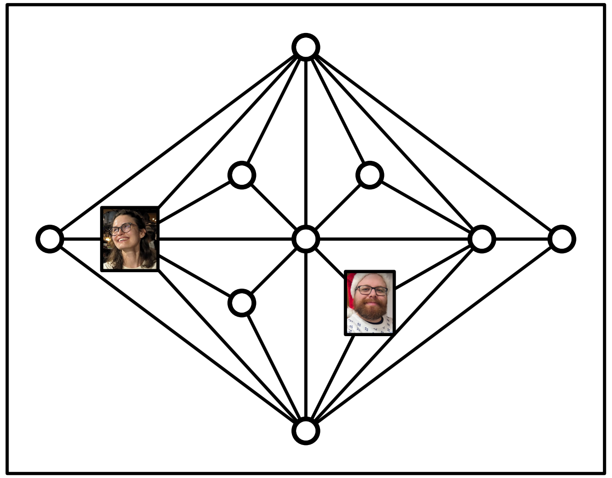


crossing-min. drawing of $G + e$



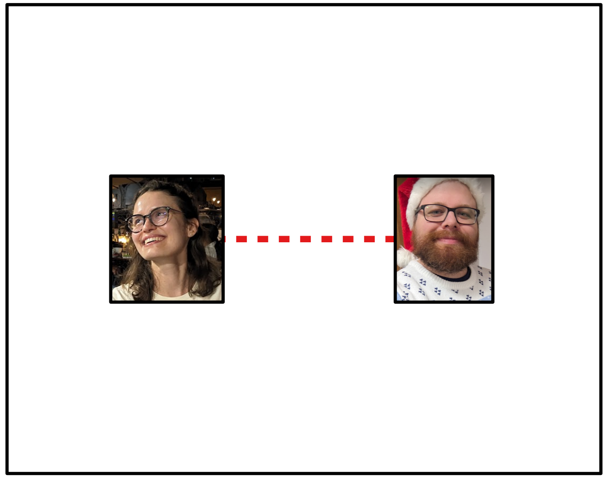


Inserting an Edge Into a Planar Graph



planar graph G

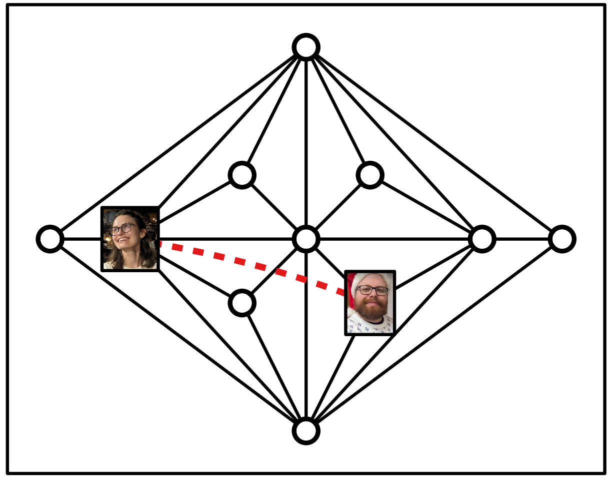
+



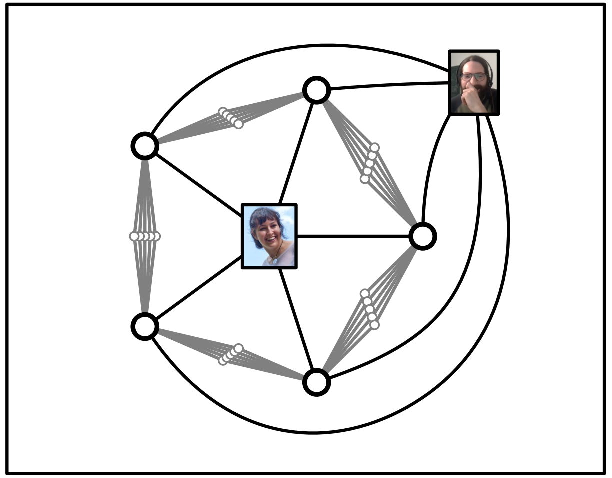
an edge e btw.
2 vertices of G

→

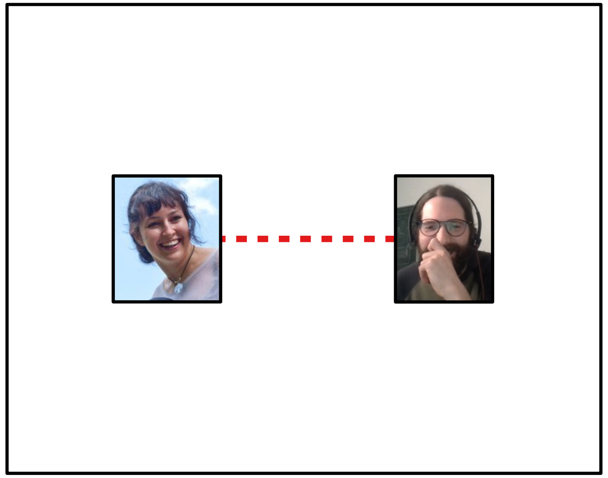
2 crossings



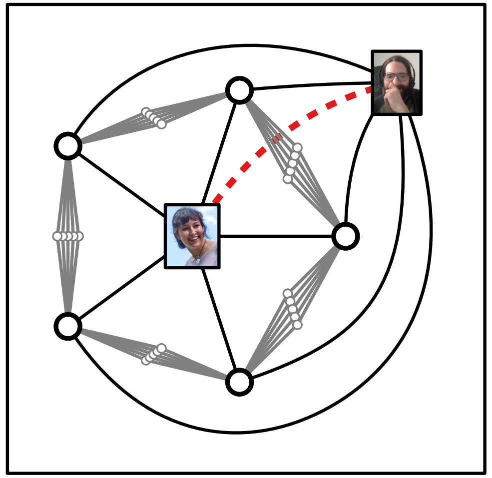
crossing-min. drawing of $G + e$



+

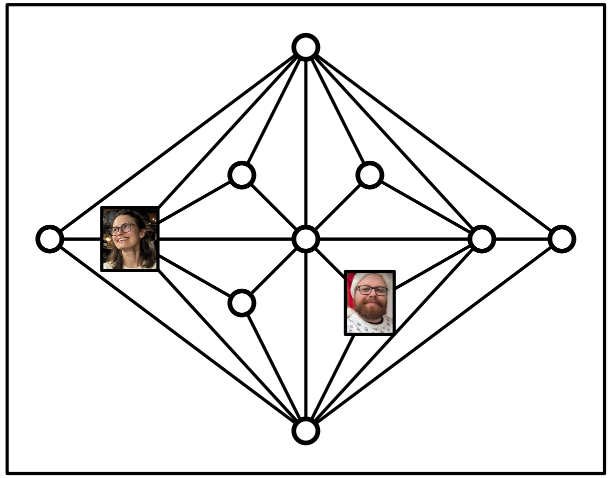


→

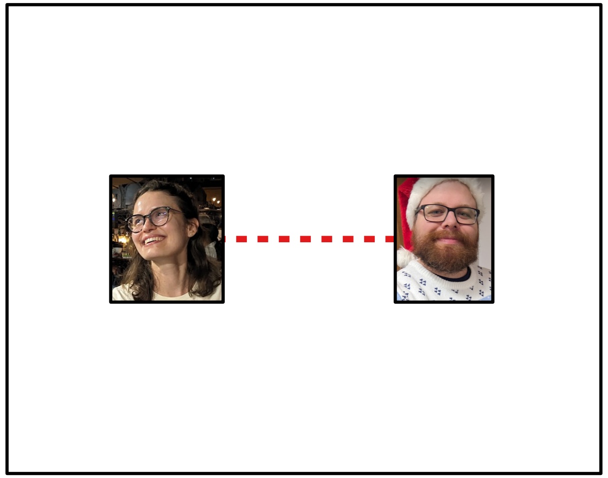


k crossings

Inserting an Edge Into a Planar Graph

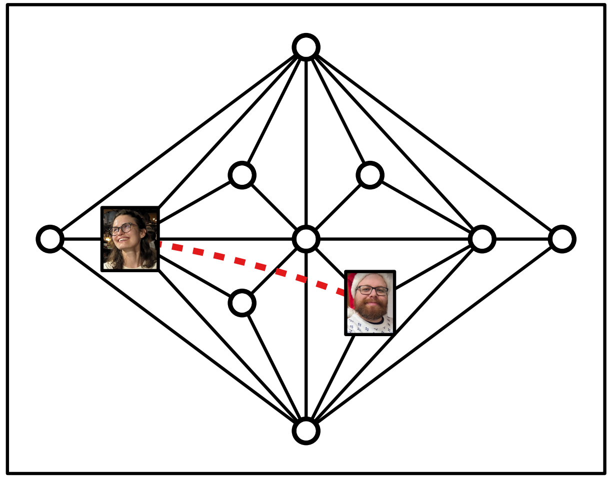


planar graph G

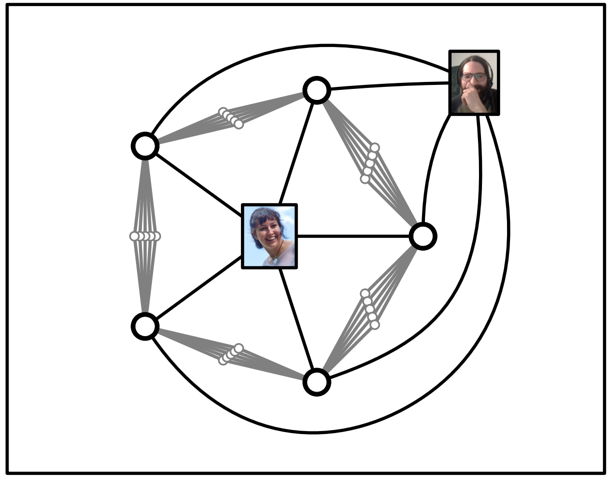


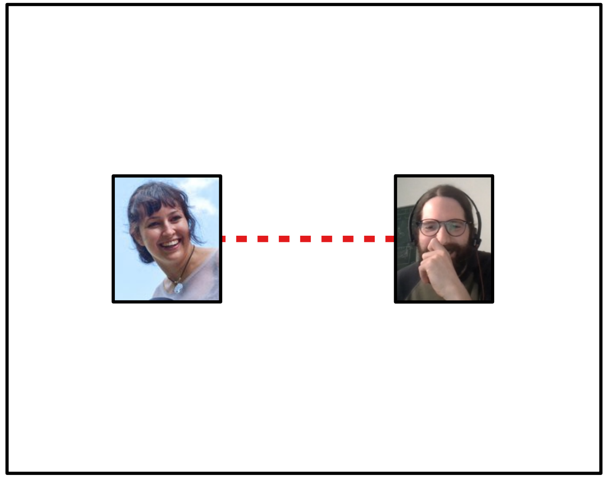
an edge e btw.
2 vertices of G

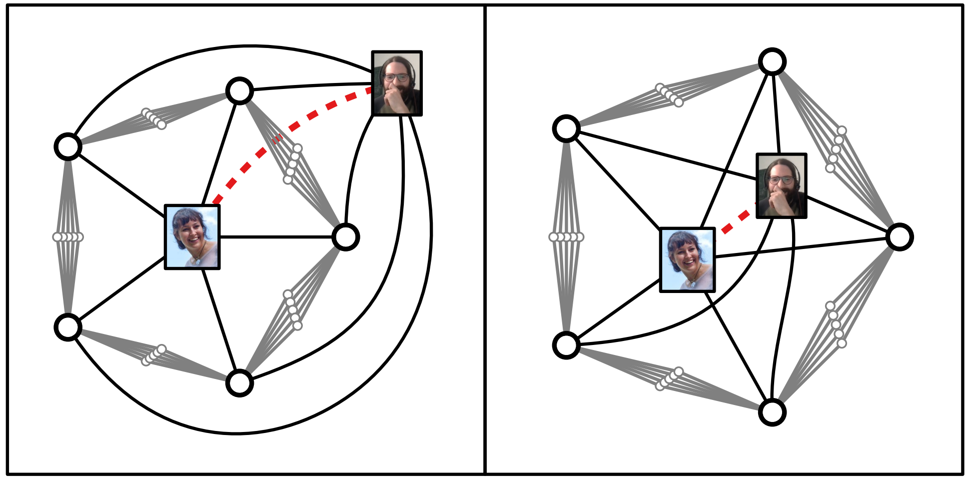
2 crossings



crossing-min. drawing of $G + e$



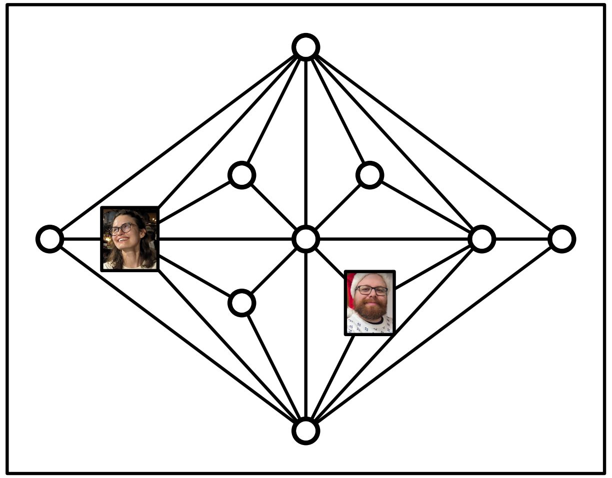




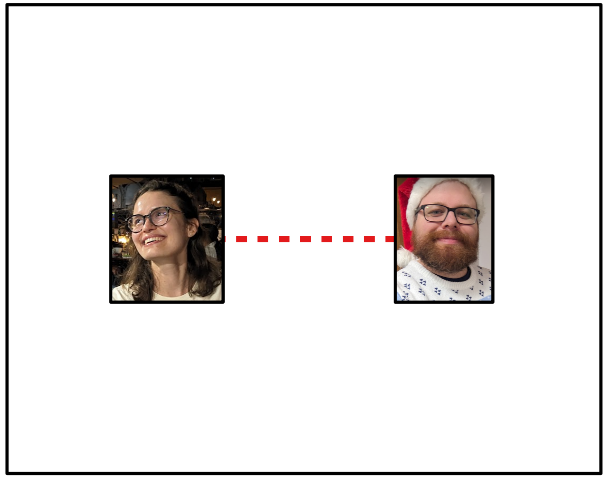
k crossings

4 crossings

Inserting an Edge Into a Planar Graph

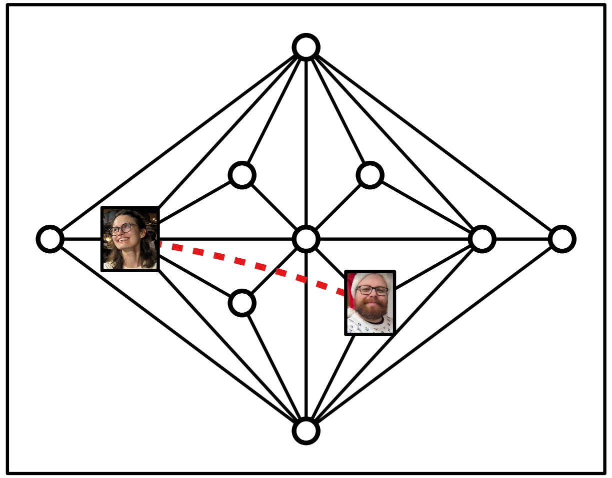


planar graph G

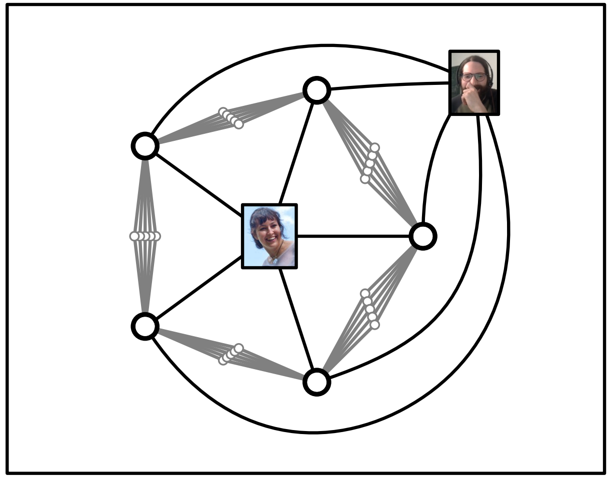


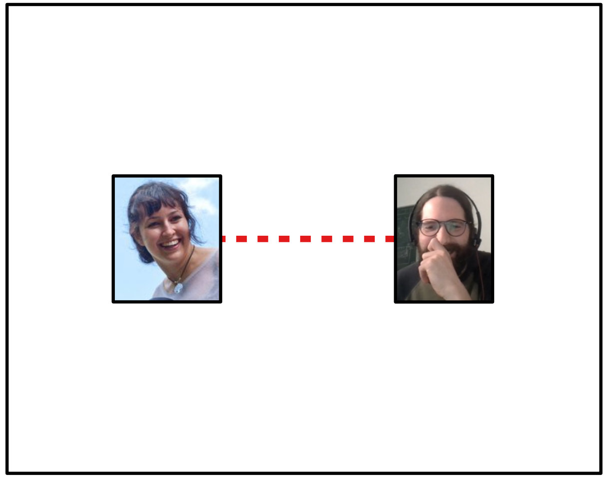
an edge e btw.
2 vertices of G

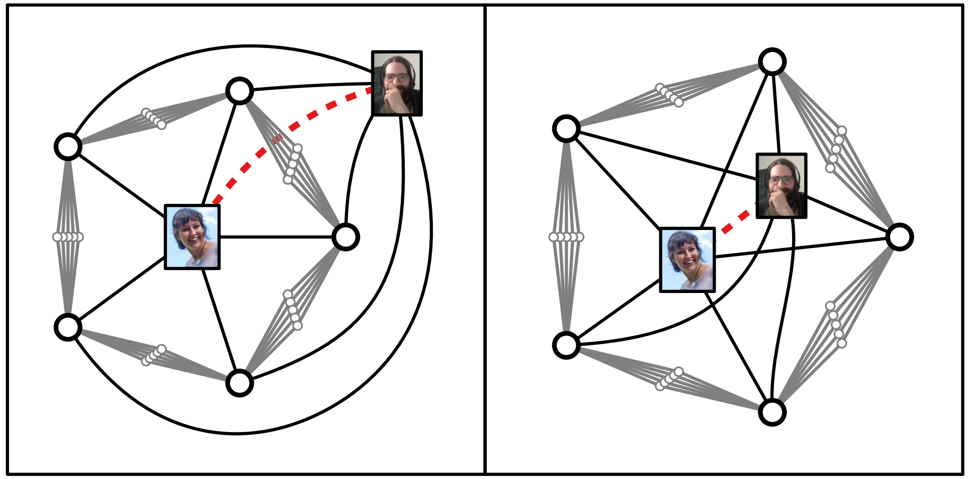
2 crossings



crossing-min. drawing of $G + e$



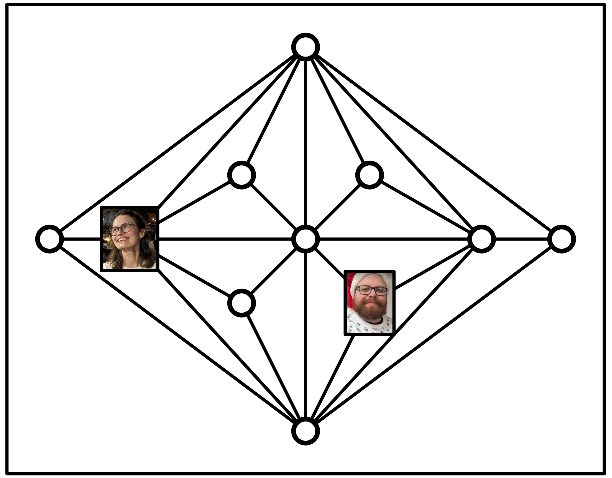




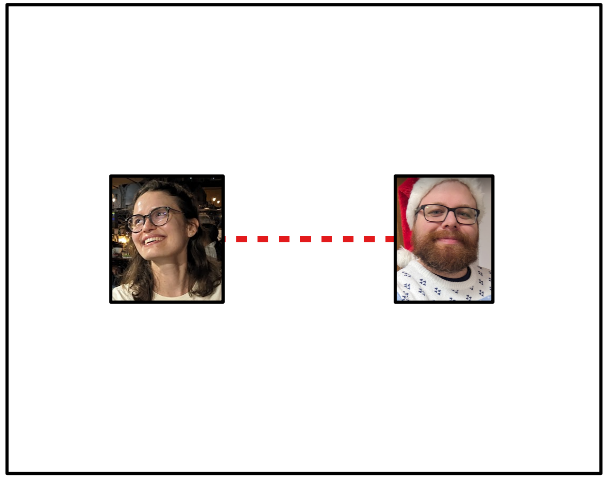
k crossings 4 crossings

This problem is NP-hard. [Cabello & Mohar '08]

Inserting an Edge Into a Planar Graph II

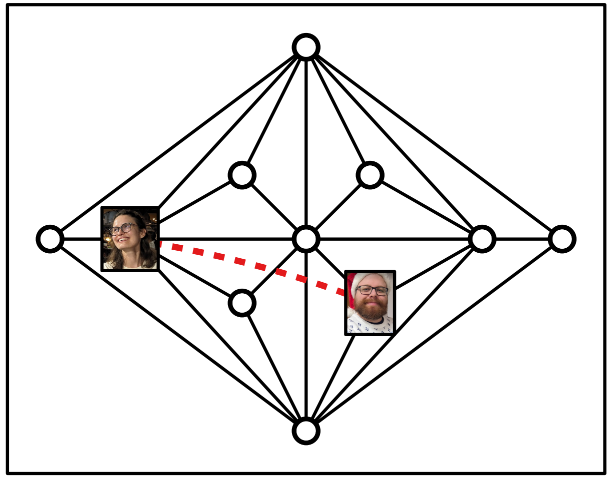


planar graph G

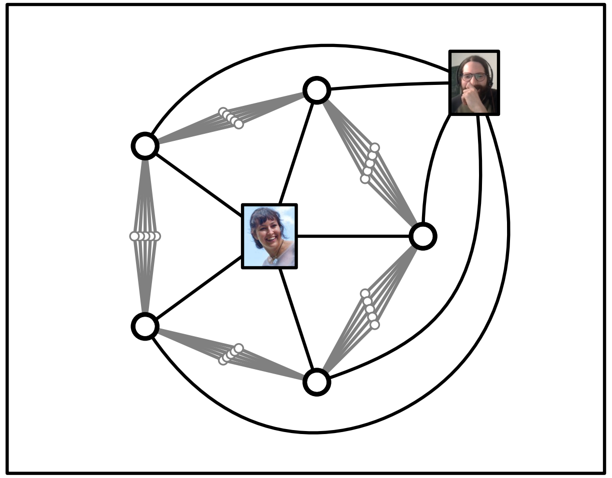


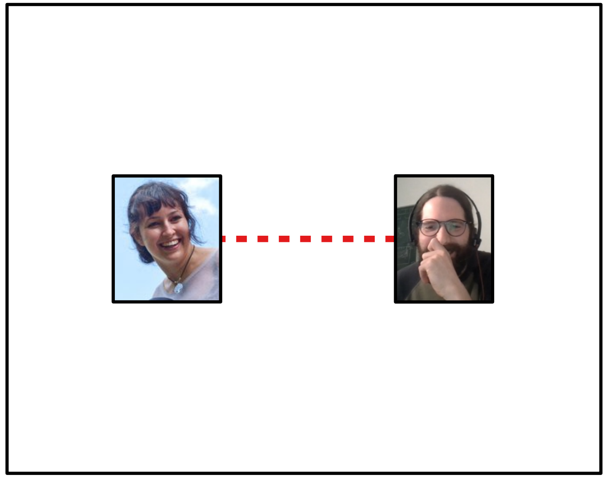
an edge e btw.
2 vertices of G

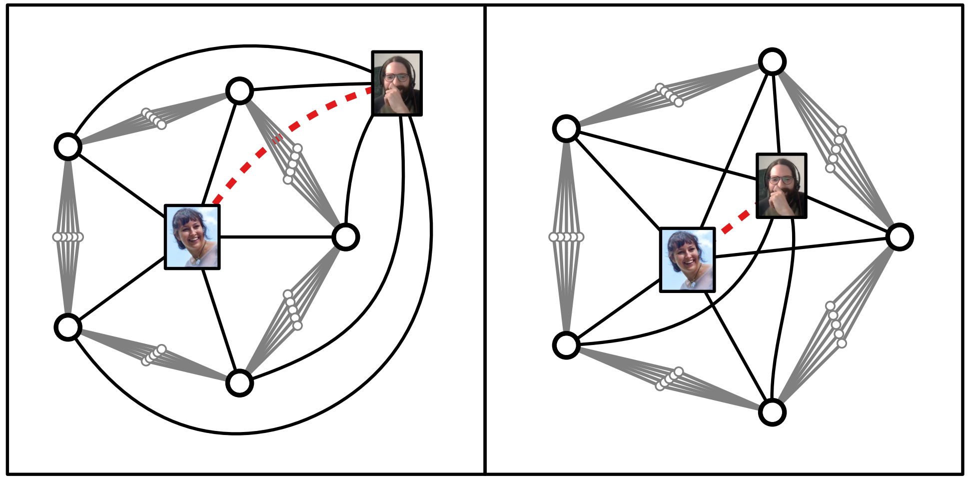
2 crossings



crossing-min. drawing of $G + e$
s.t. G is drawn planar





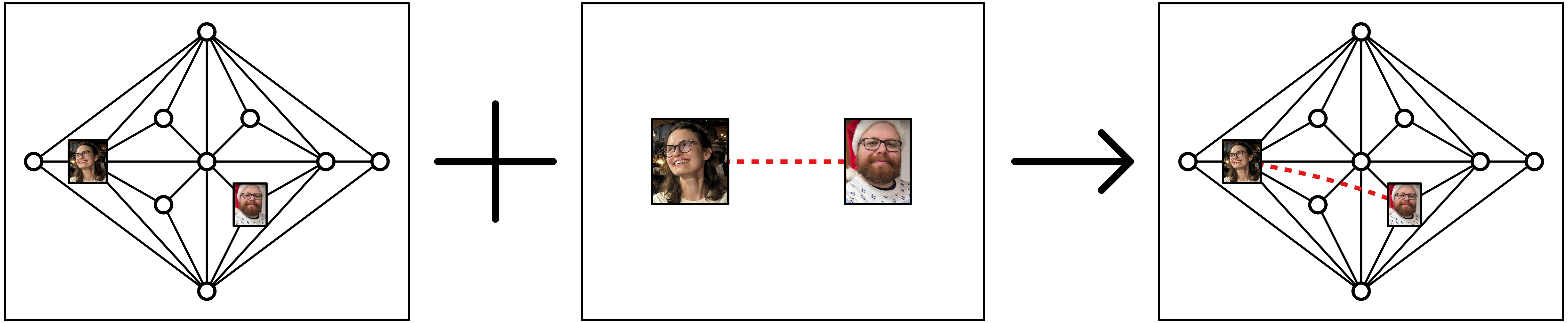


k crossings

4 crossings

Inserting an Edge Into a Planar Graph II

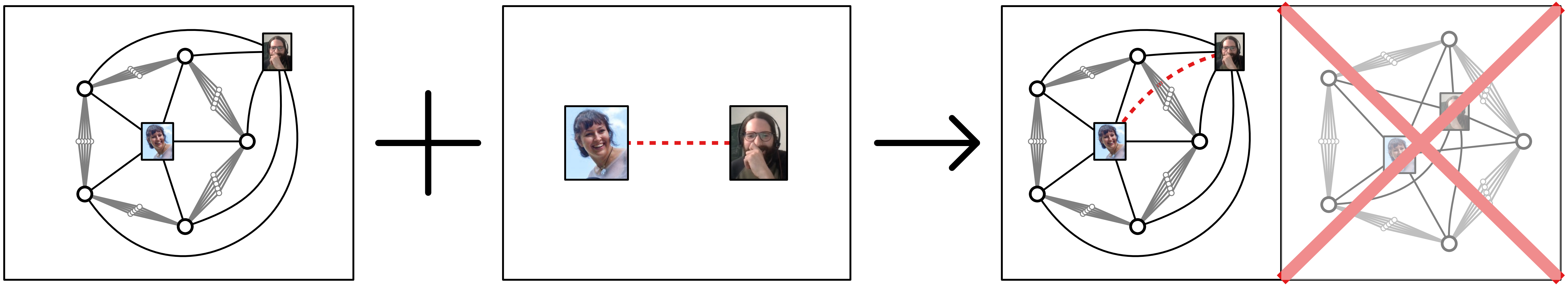
2 crossings



planar graph G

an edge e btw.
2 vertices of G

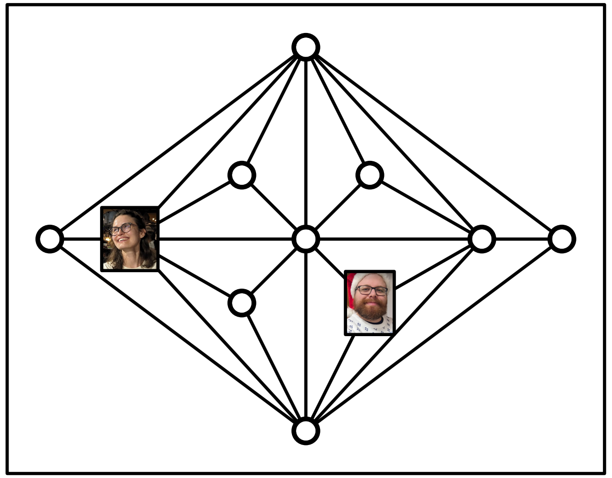
crossing-min. drawing of $G + e$
s.t. G is drawn planar



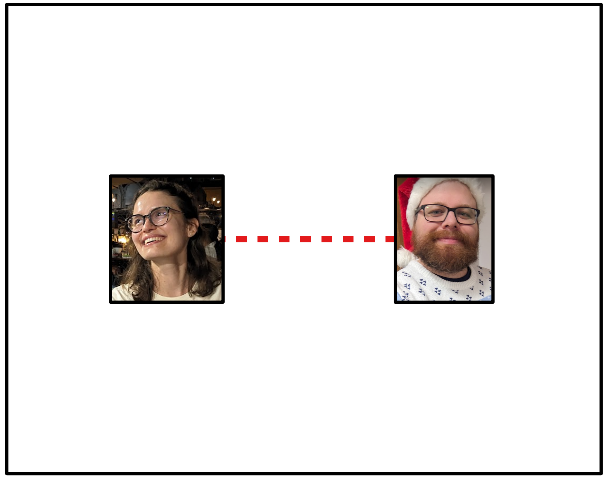
k crossings

4 crossings

Inserting an Edge Into a Planar Graph II

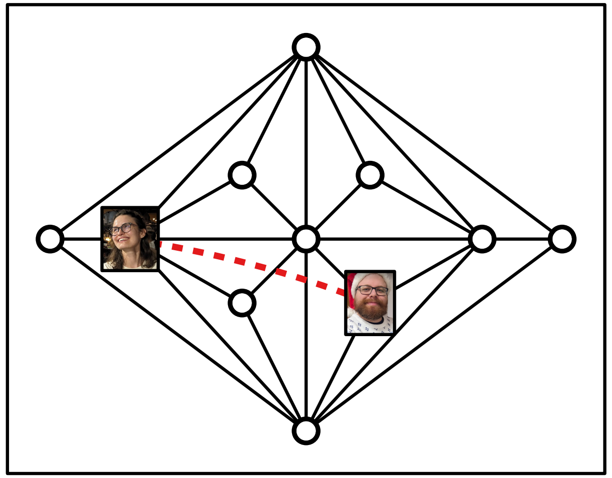


planar graph G

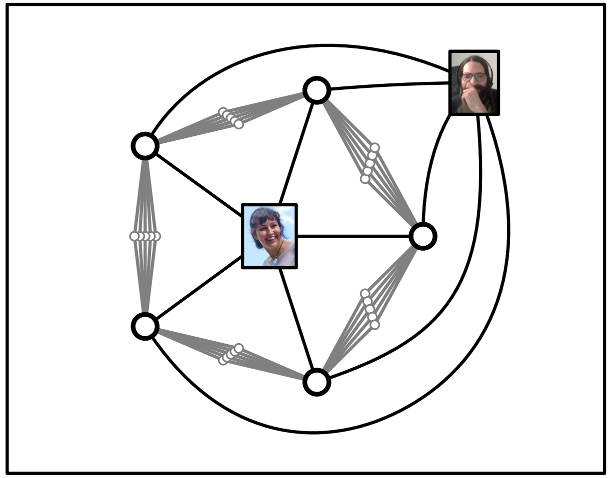


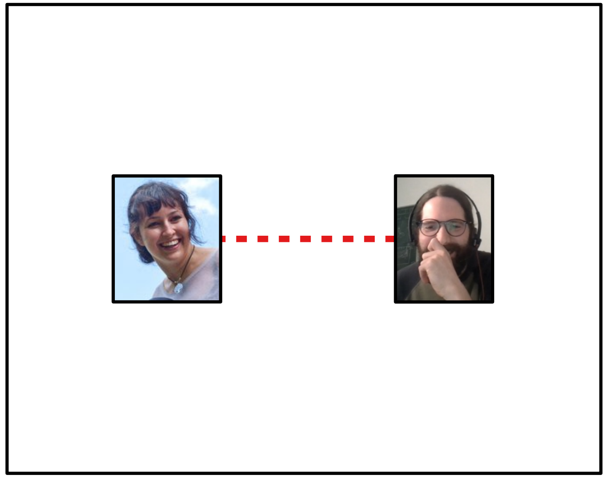
an edge e btw.
2 vertices of G

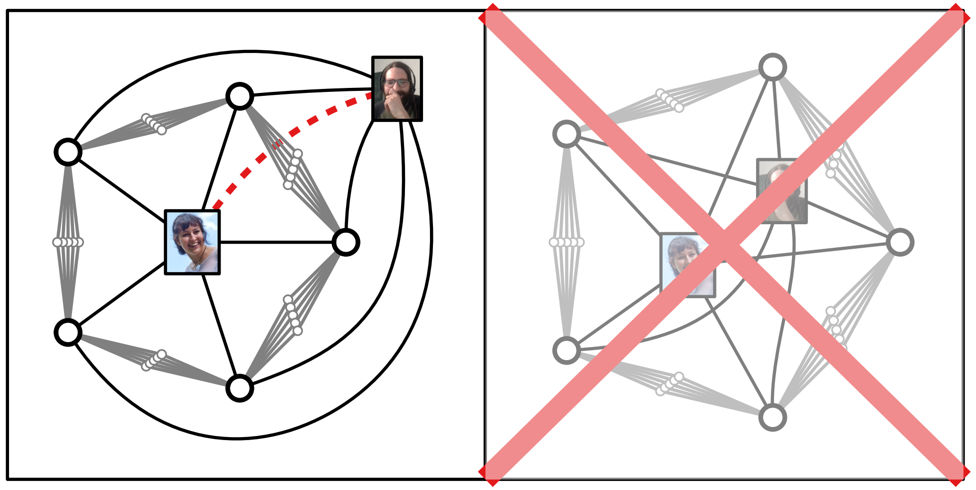
2 crossings



crossing-min. drawing of $G + e$
s.t. G is drawn planar





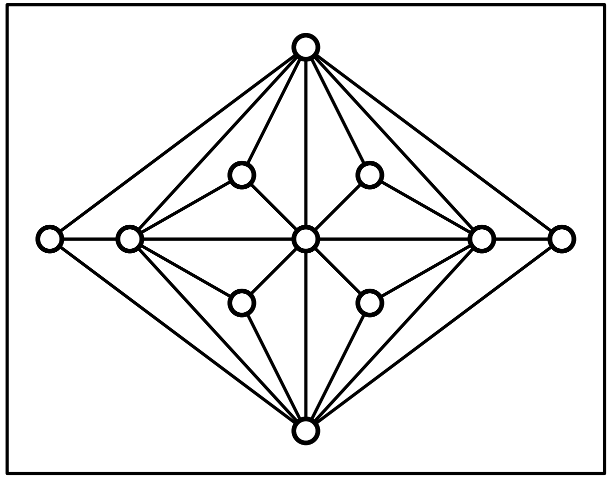


k crossings

4 crossings

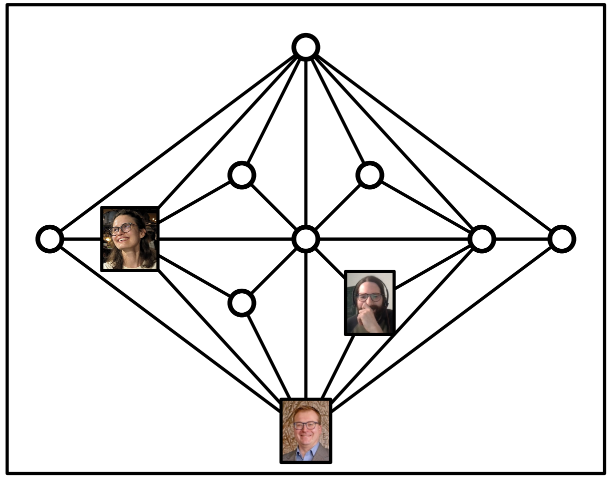
This problem can be solved in $\mathcal{O}(n)$ time. [Gutwenger, Mutzel & Weiskircher '05]

Inserting a **Vertex** Into a Planar Graph

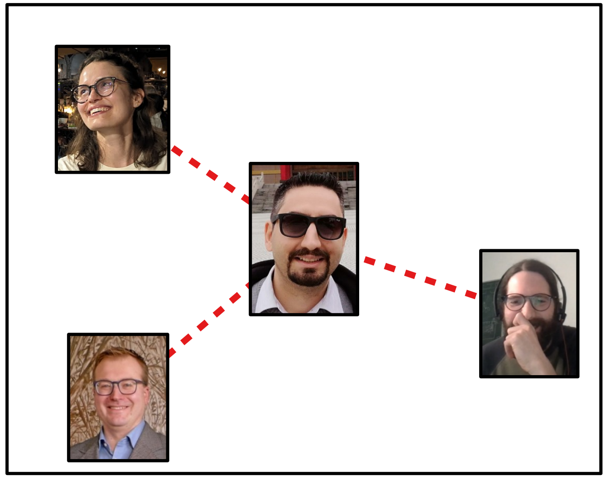


planar graph G

Inserting a **Vertex** Into a Planar Graph

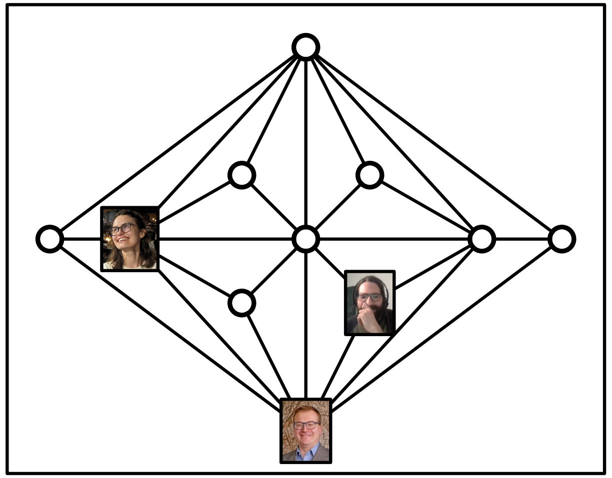


planar graph G

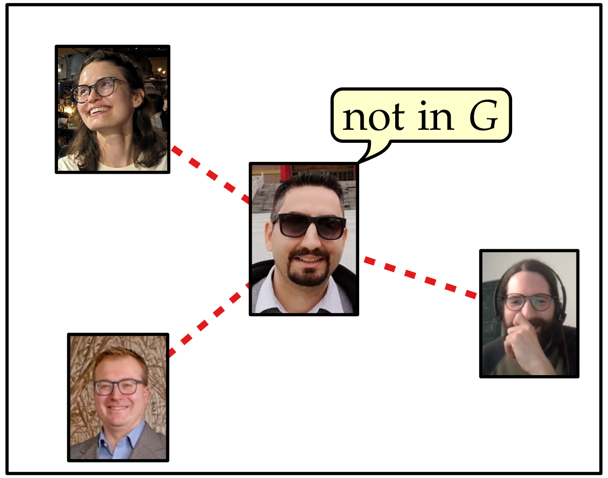


a star S with
its leaves in G

Inserting a **Vertex** Into a Planar Graph

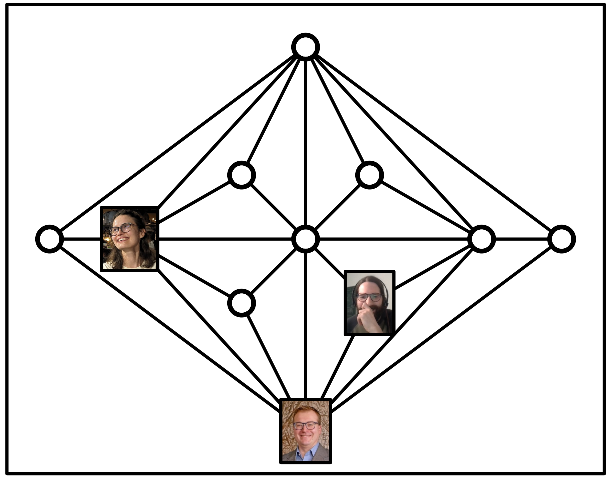


planar graph G

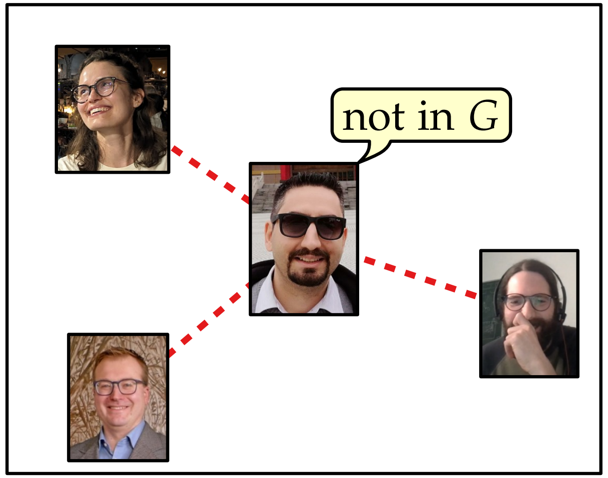


a star S with
its leaves in G

Inserting a **Vertex** Into a Planar Graph

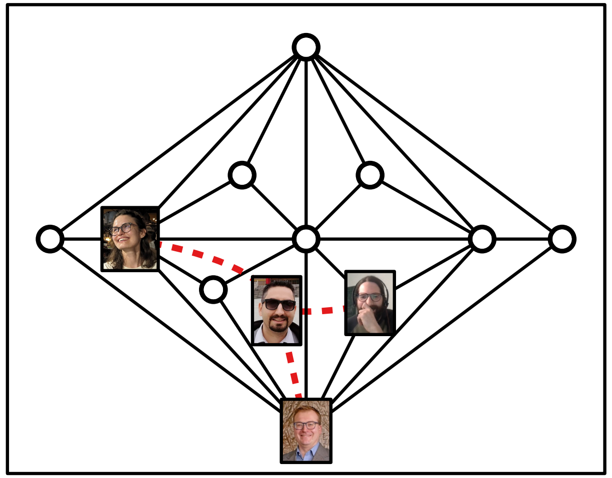


planar graph G



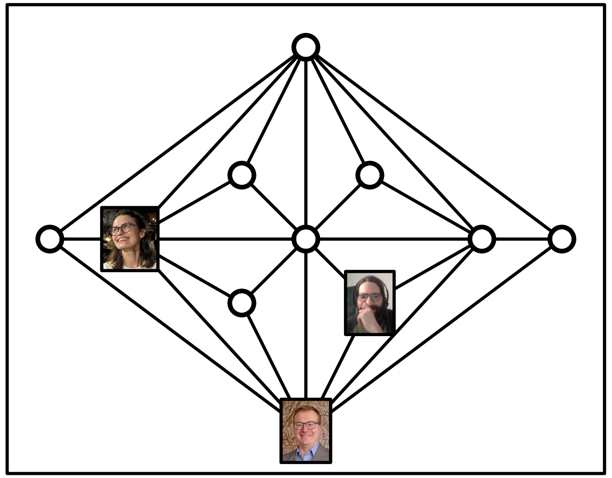
a star S with
its leaves in G

2 crossings

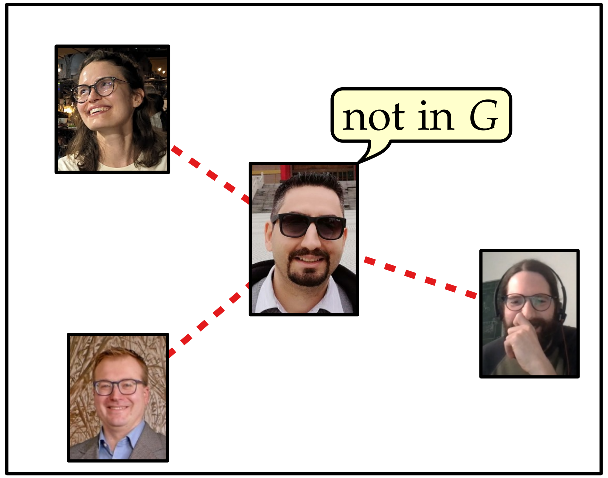


crossing-min. drawing of $G + S$
s.t. G is drawn planar

Inserting a **Vertex** Into a Planar Graph

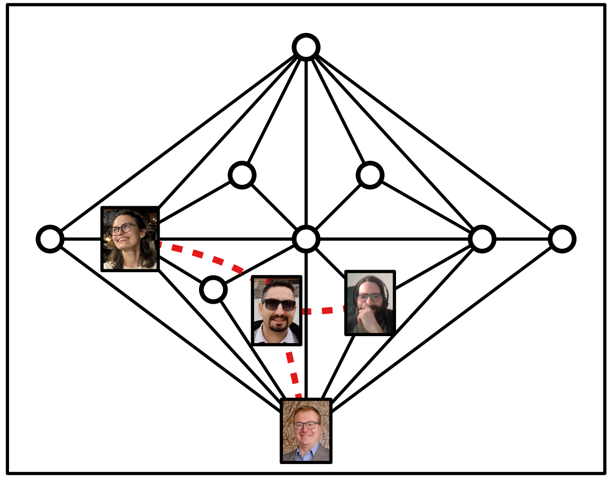


planar graph G



a star S with
its leaves in G

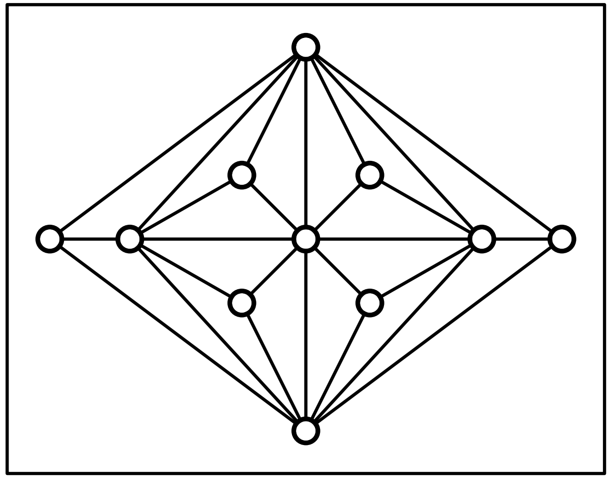
2 crossings



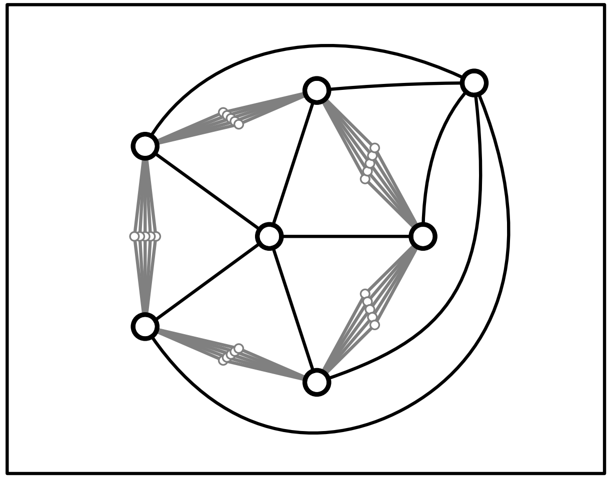
crossing-min. drawing of $G + S$
s.t. G is drawn planar

This problem can be solved in $\mathcal{O}(n^7)$ time. [Chimani, Gutwenger, Mutzel & Wolf '05]

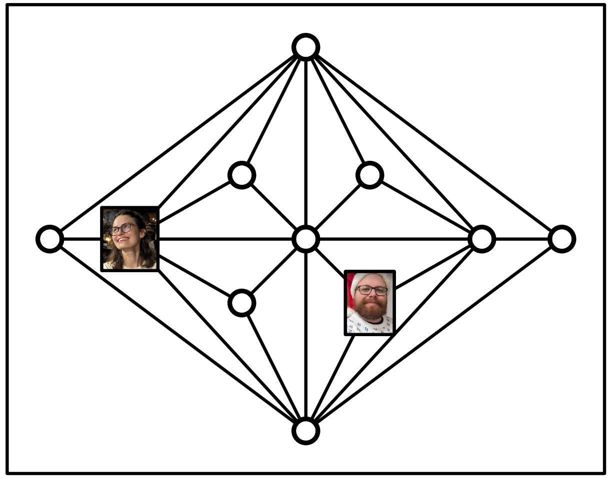
Inserting an Edge Into a **Plane** Graph

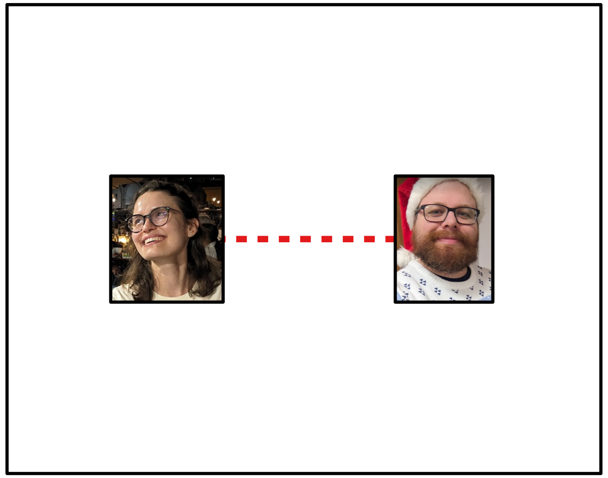


plane graph G
(planar graph
+ planar embedding)



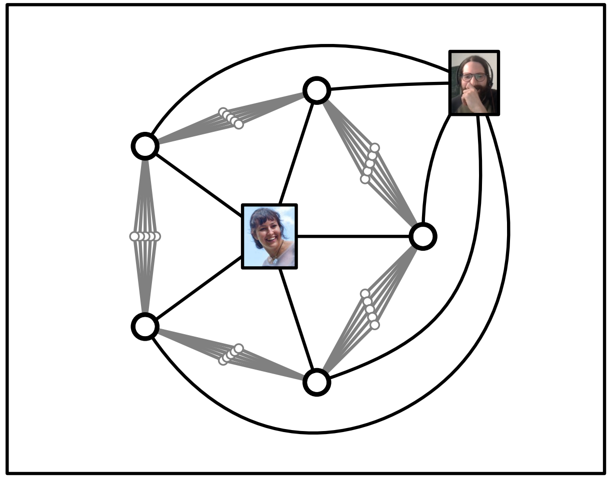
Inserting an Edge Into a **Plane** Graph

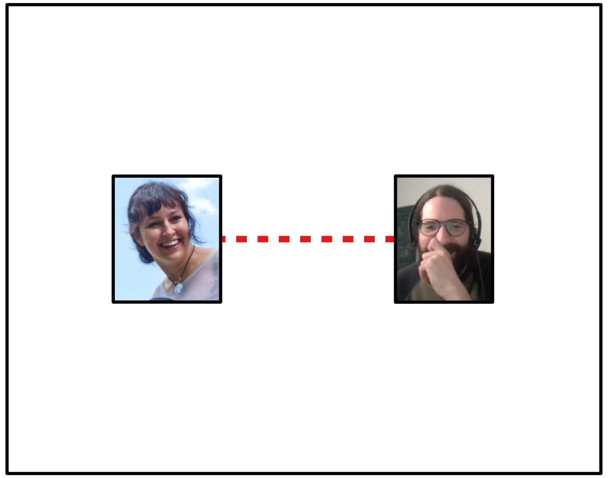




plane graph G
(planar graph
+ planar embedding)

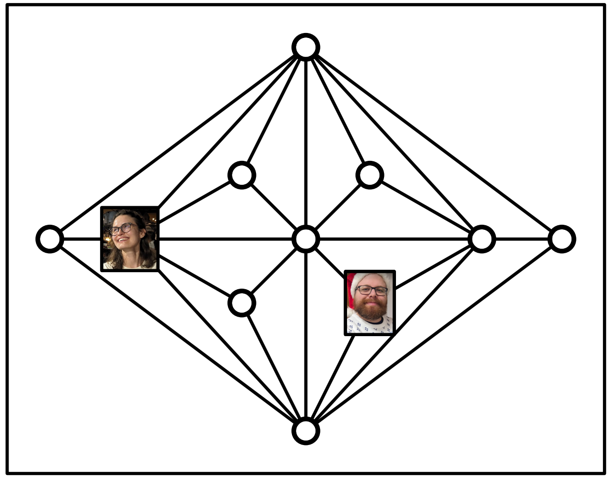
an edge e btw.
2 vertices of G

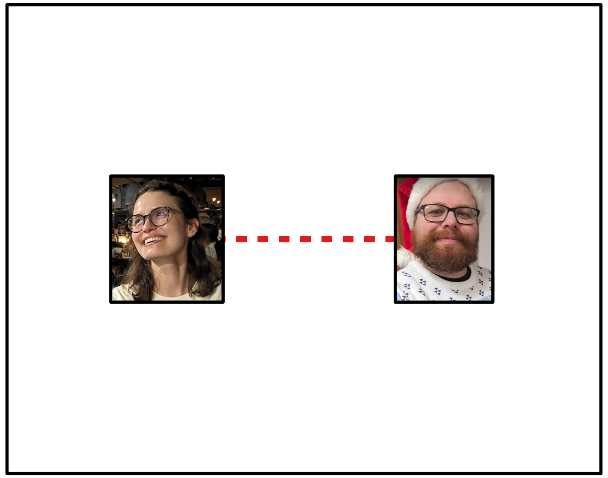


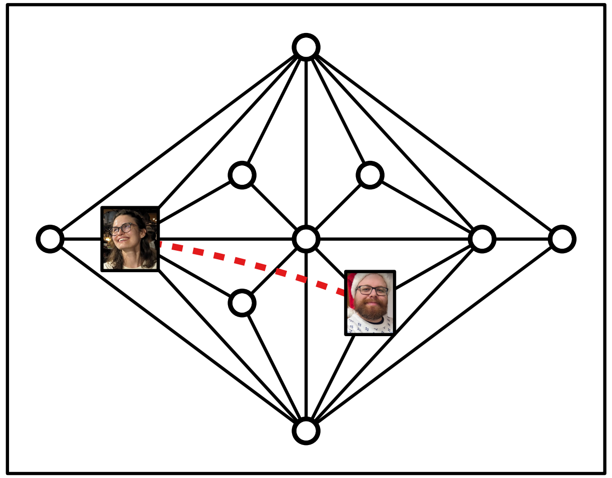


Inserting an Edge Into a **Plane** Graph

2 crossings





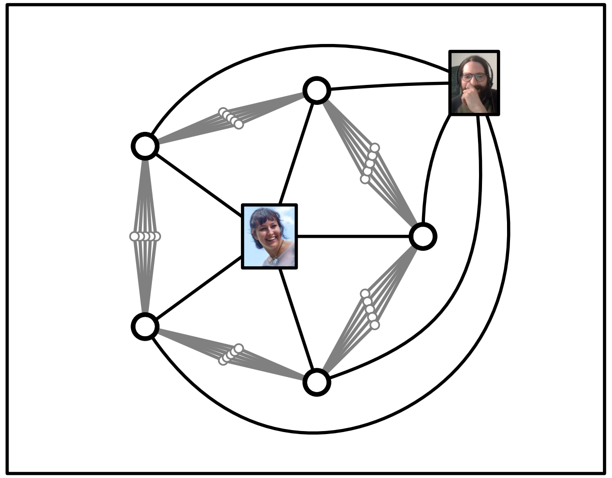


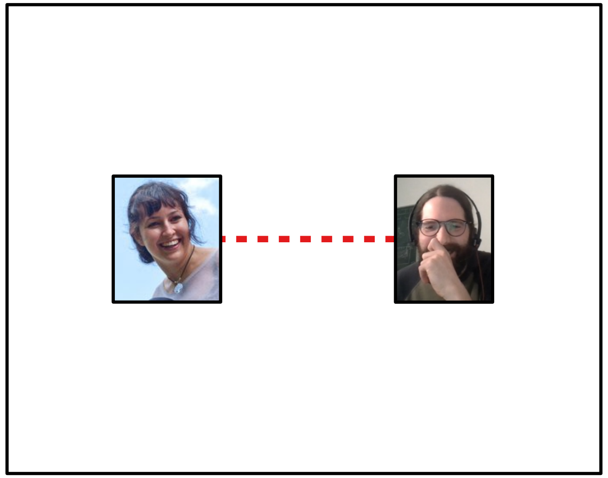
plane graph G
(planar graph
+ planar embedding)

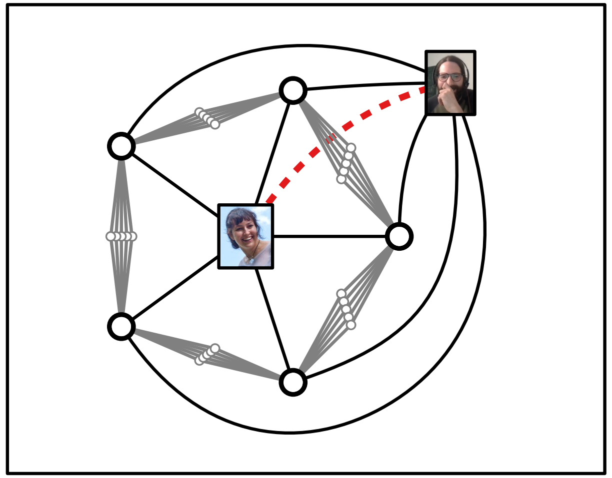
an edge e btw.
2 vertices of G

crossing-min. drawing of $G + e$
that keeps the embedding of G

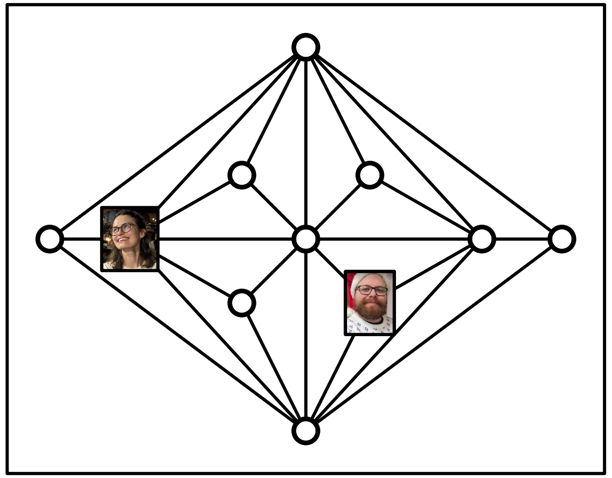
k crossings



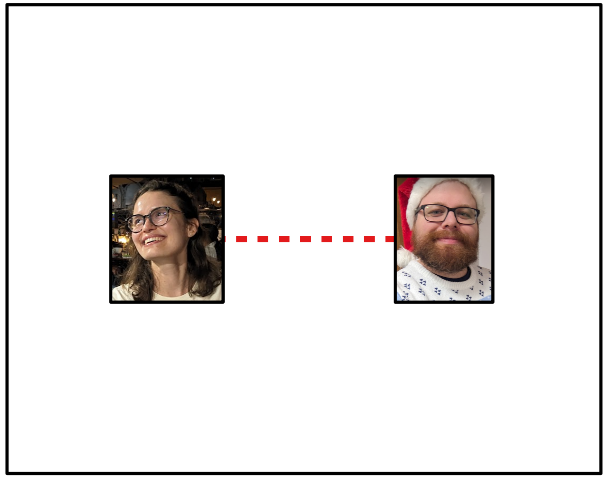




Inserting an Edge Into a **Plane** Graph

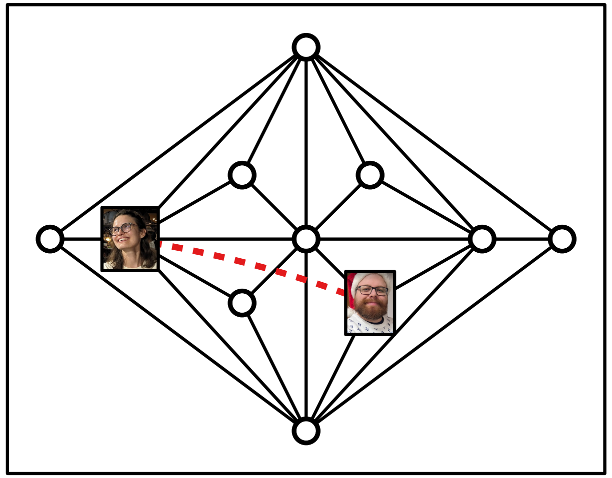


plane graph G
(planar graph
+ planar embedding)

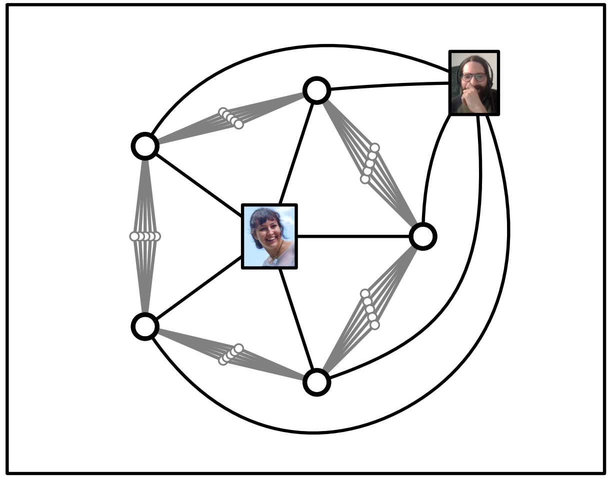


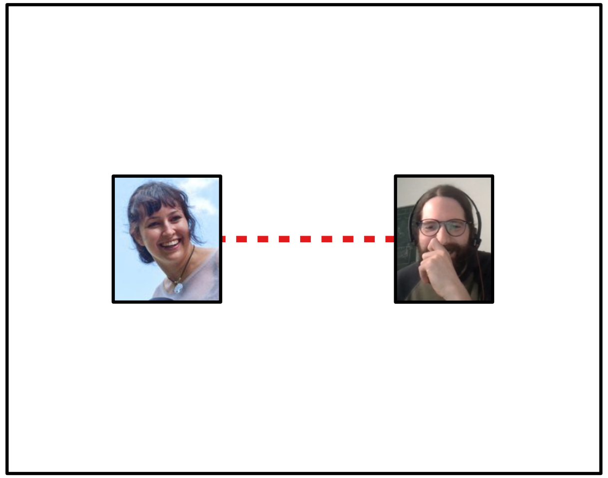
an edge e btw.
2 vertices of G

2 crossings

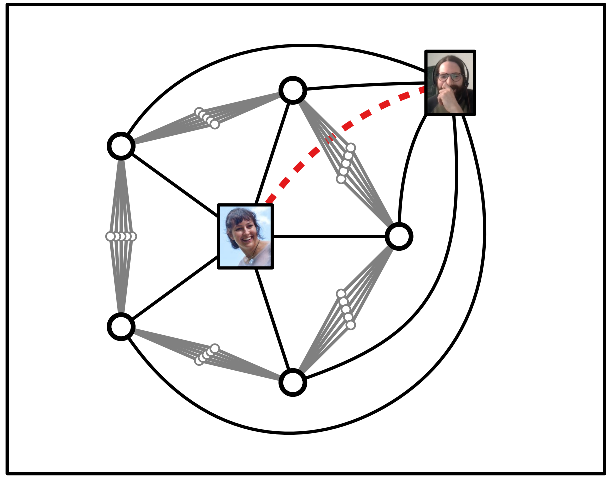


crossing-min. drawing of $G + e$
that keeps the embedding of G



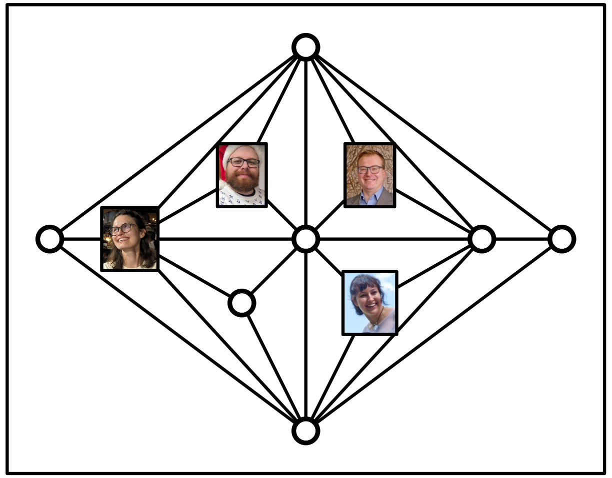


k crossings



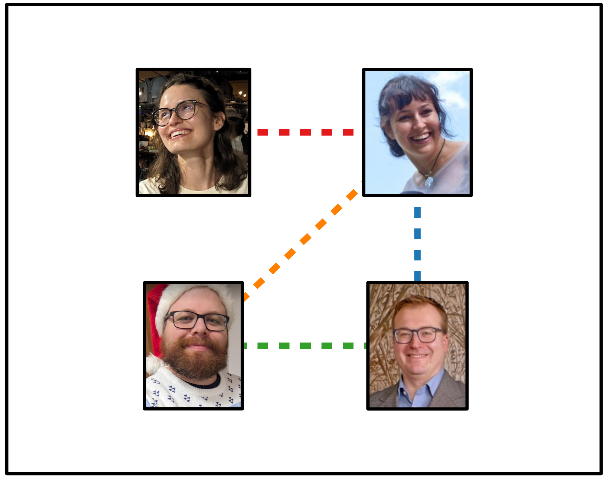
This problem can be solved in $\mathcal{O}(n)$ time. [BFS]

Inserting **Edges** Into a Plane Graph



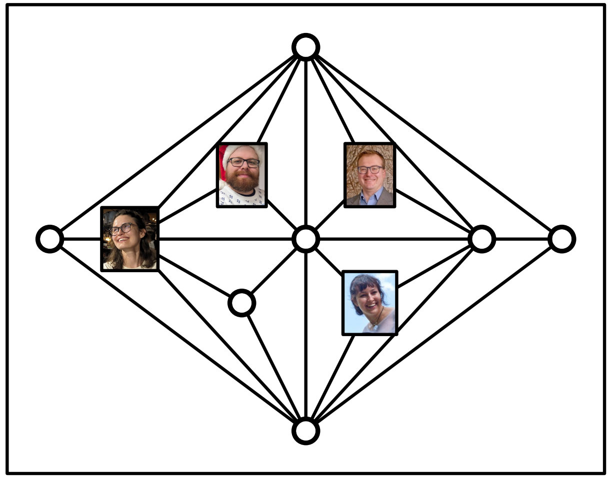
plane graph G

+

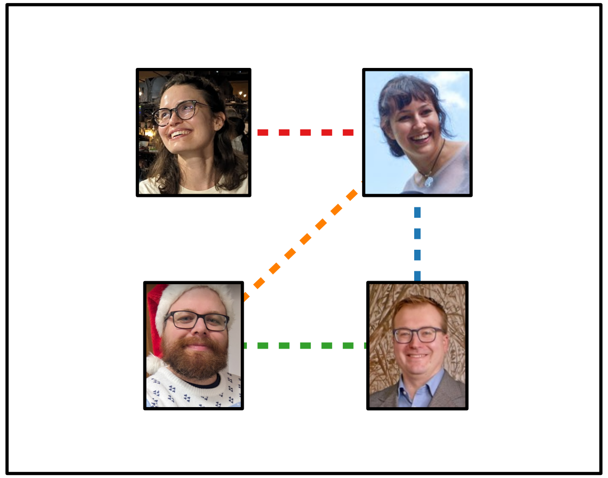


edges E' btw. vtcs in G

Inserting Edges Into a Plane Graph

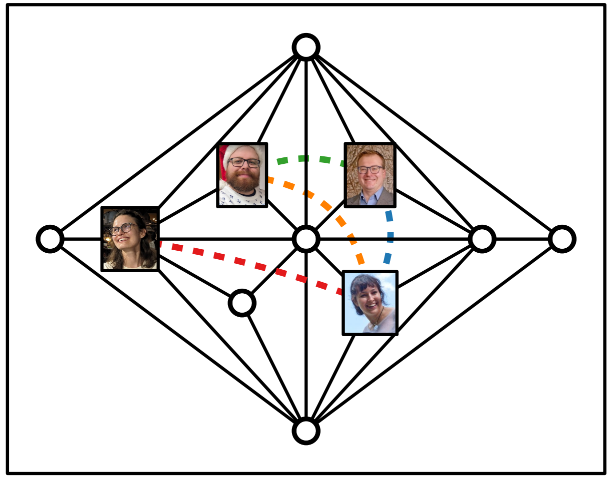


plane graph G



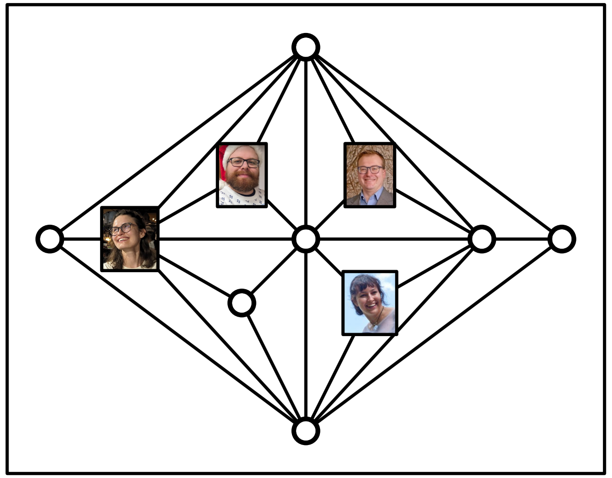
edges E' btw. vtcs in G

7 crossings

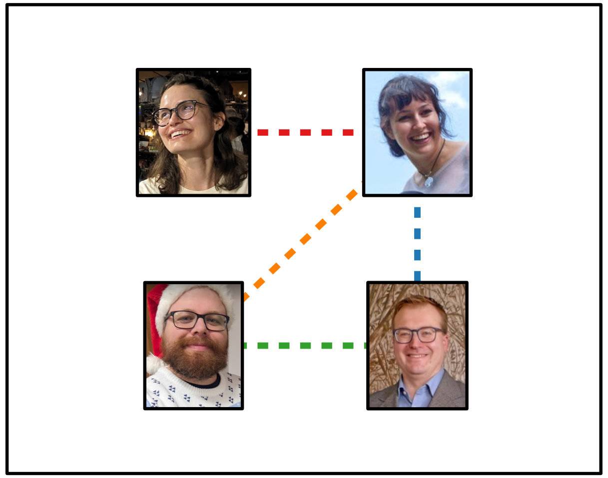


crossing-min. drawing of $G + E'$ that keeps the embedding of G

Inserting Edges Into a Plane Graph

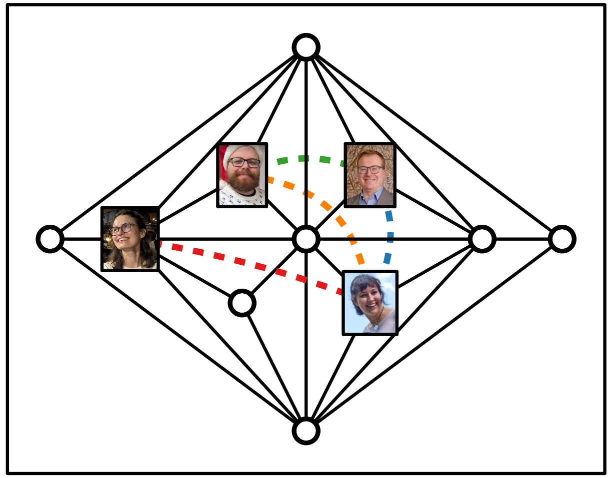


plane graph G



edges E' btw. vtcs in G

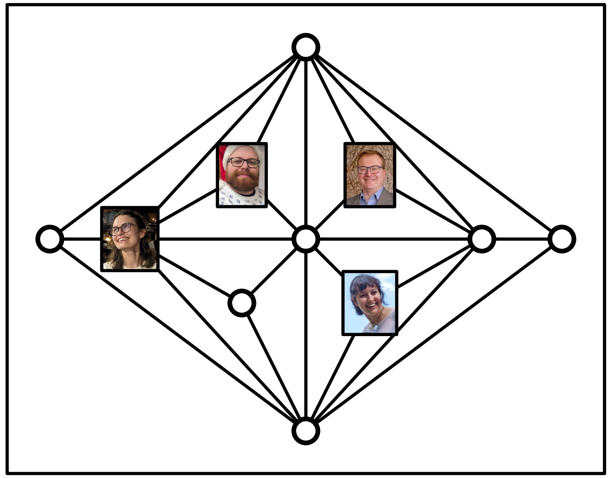
7 crossings



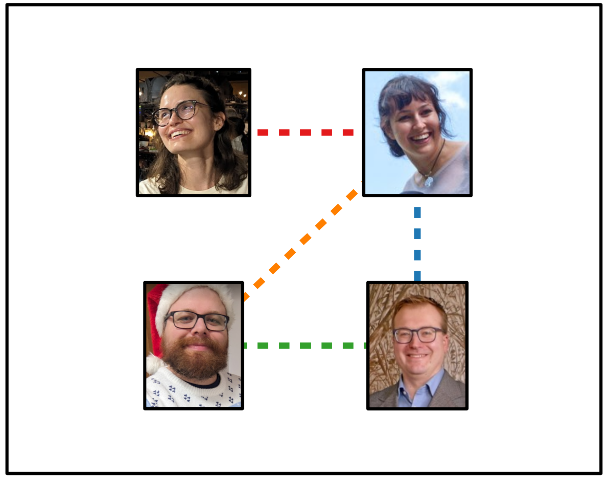
crossing-min. drawing of $G + E'$ that keeps the embedding of G

This problem is NP-hard. [Ziegler '01]

Inserting **Edges** Into a Plane Graph

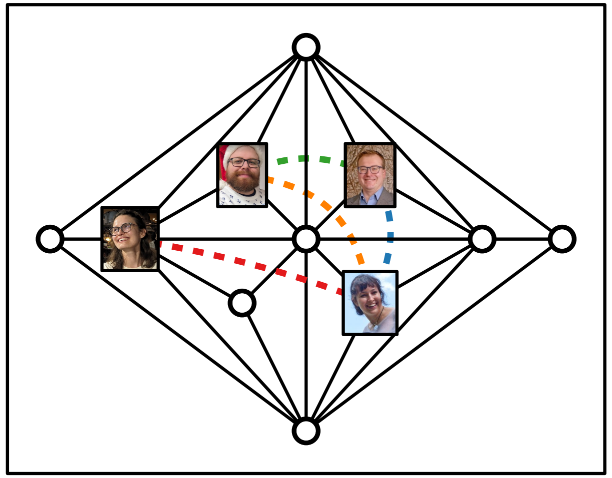


plane graph G



edges E' btw. vtcs in G

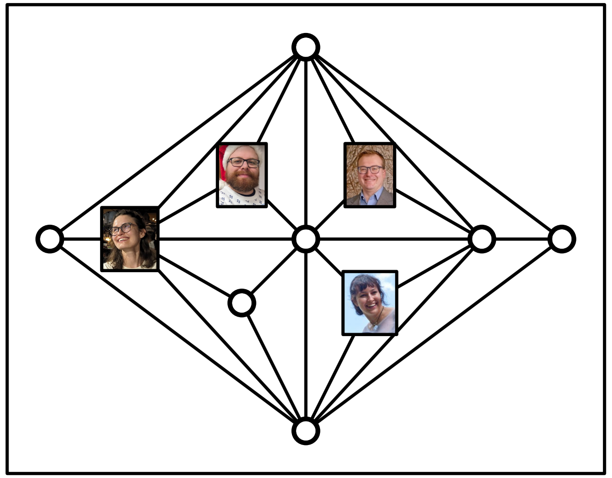
7 crossings



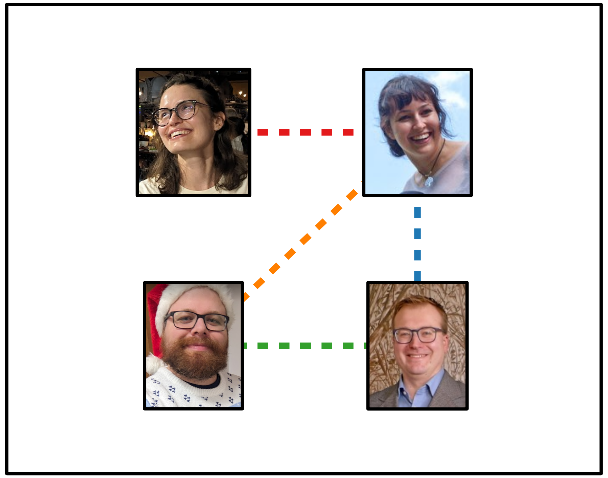
crossing-min. drawing of $G + E'$ that keeps the embedding of G

This problem is NP-hard. [Ziegler '01]
… even if G is biconnected.

Inserting Edges Into a Plane Graph

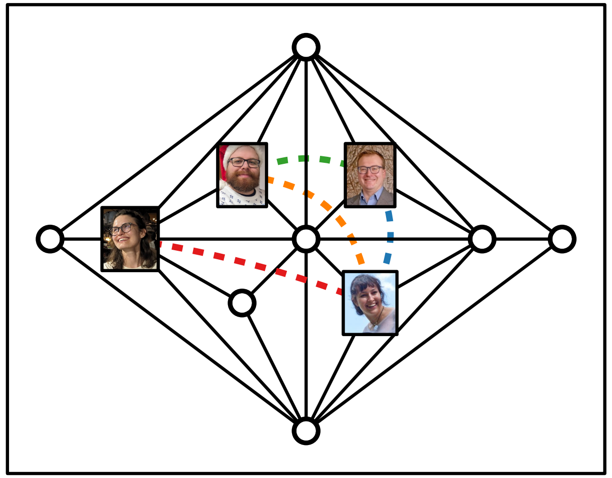


plane graph G



edges E' btw. vtcs in G

7 crossings

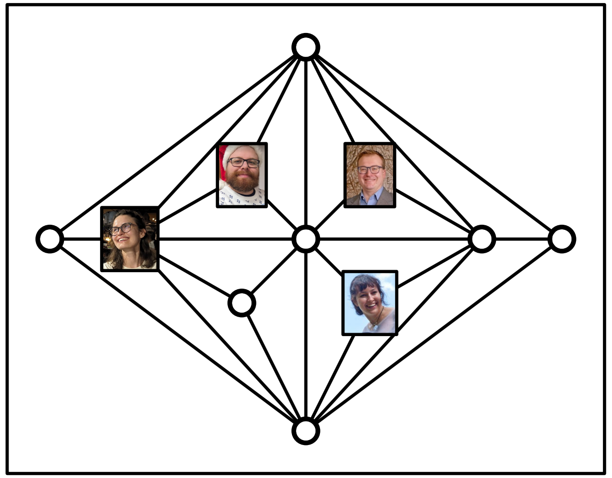


crossing-min. drawing of $G + E'$ that keeps the embedding of G

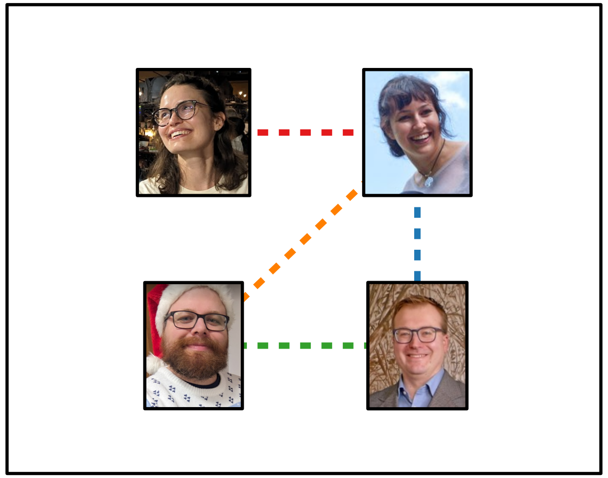
This problem is NP-hard. [Ziegler '01]
... even if G is biconnected.

This problem is in FPT parameterized by #crossings. [Hamm & Hliněný '22]

Inserting **Edges** Into a Plane Graph

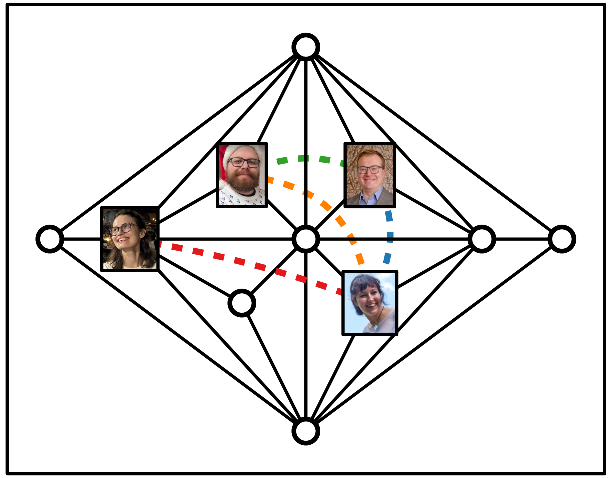


plane graph G



edges E' btw. vtcs in G

7 crossings

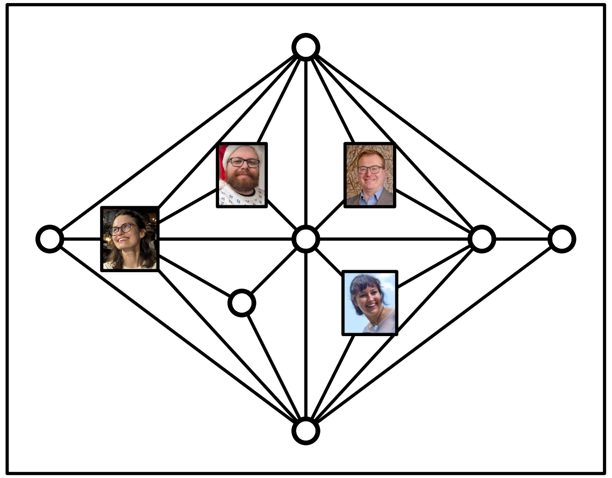


crossing-min. drawing of $G + E'$ that keeps the embedding of G

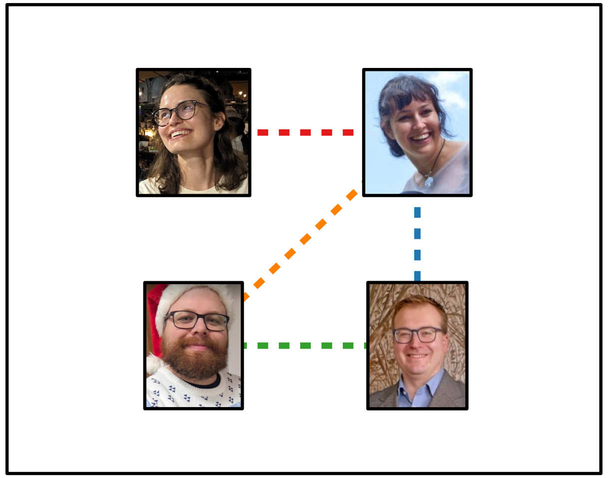
This problem is NP-hard. [Ziegler '01]
... even if G is biconnected.

This problem is in FPT parameterized by #crossings. [Hamm & Hliněný '22]
... even if G is non-planar (or drawn with crossings)

Inserting **Edges** Into a Plane Graph

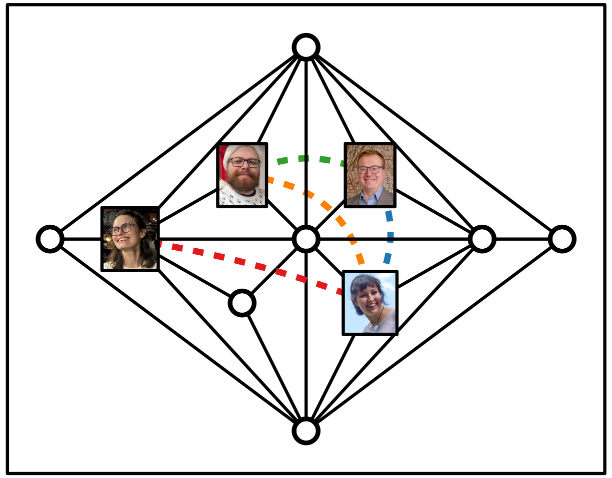


plane graph G



edges E' btw. vtcs in G

7 crossings



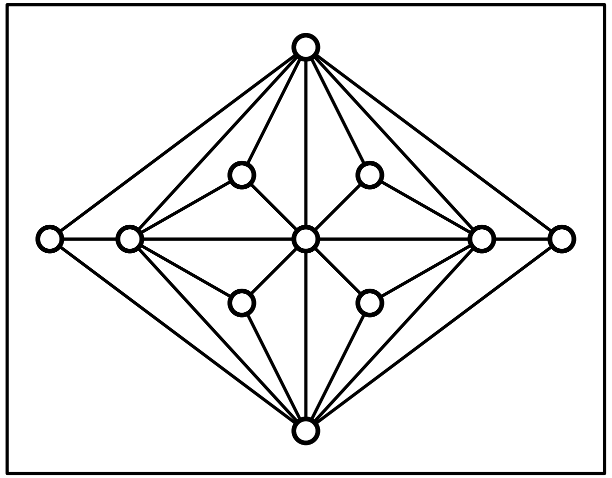
crossing-min. drawing of $G + E'$ that keeps the embedding of G

This problem is NP-hard. [Ziegler '01]
… even if G is biconnected.

This problem is in FPT parameterized by #crossings. [Hamm & Hliněný '22]
… even if G is non-planar (or drawn with crossings)

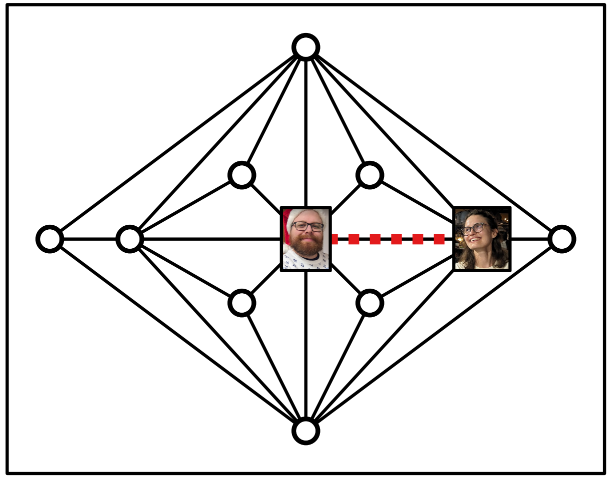
This problem is in FPT parameterized by $|E'|$ [Chimani & Hliněný '23]
if G is biconnected or all cutvertices have constant degree.

Partial Embedding – General Definition

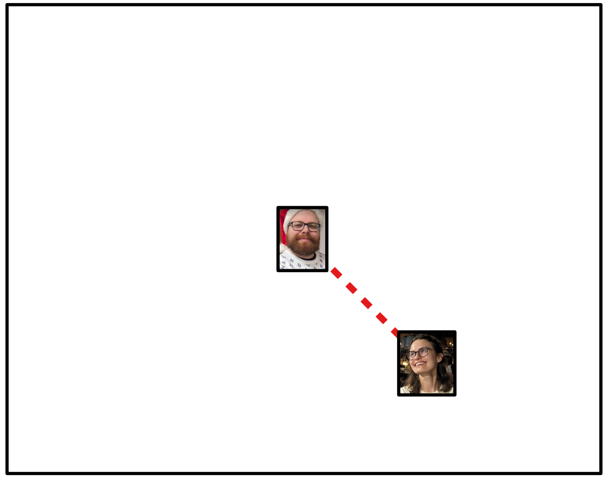


graph G +
drawing style Φ
(e.g., straight-line planar)

Partial Embedding – General Definition

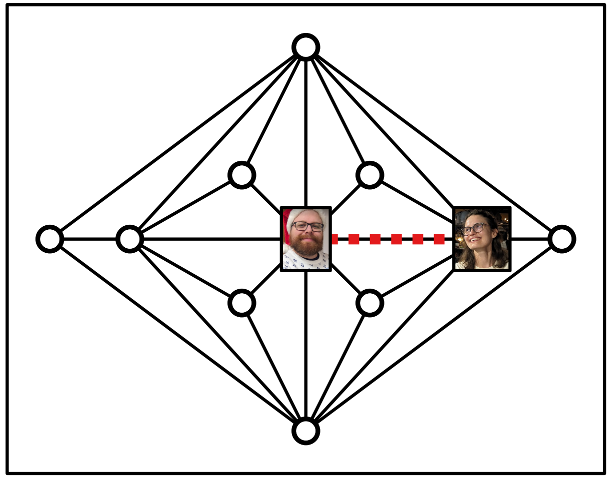


graph G +
drawing style Φ
(e.g., straight-line planar)

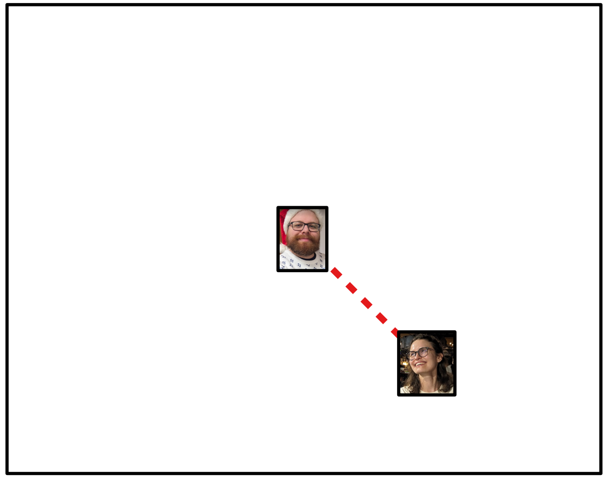


drawing with style Φ of
a subgraph $H \subseteq G$

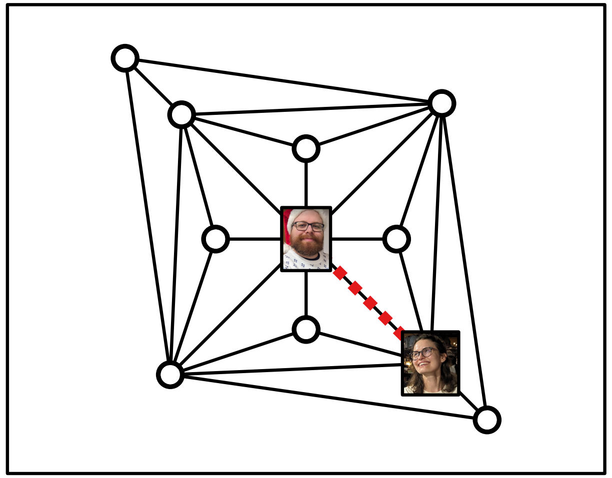
Partial Embedding – General Definition



graph G +
drawing style Φ
(e.g., straight-line planar)

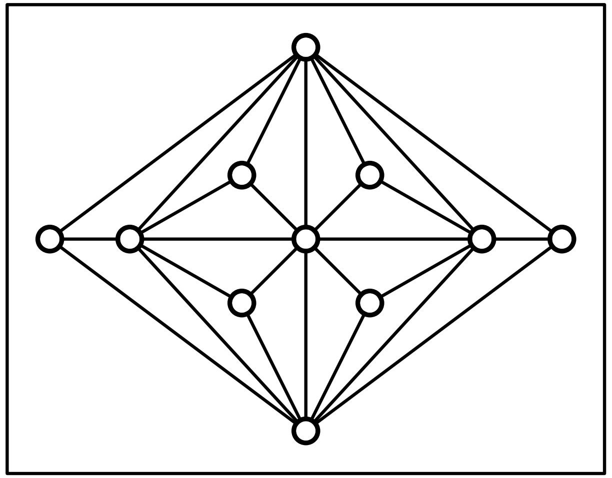


drawing with style Φ of
a subgraph $H \subseteq G$



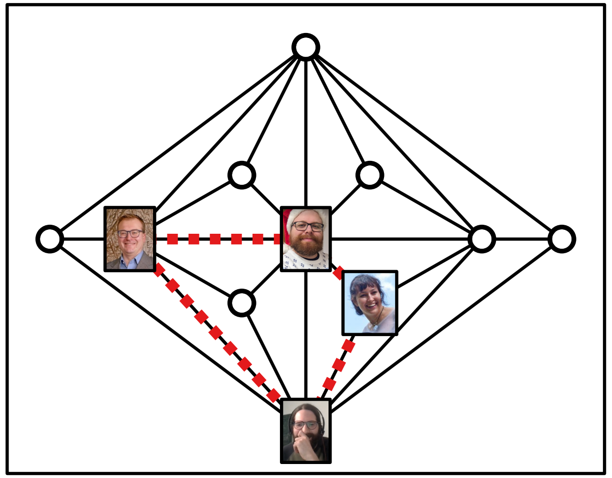
drawing with style Φ of G
s.t. H **keeps its drawing**

Partial Embedding of **Planar** Graphs



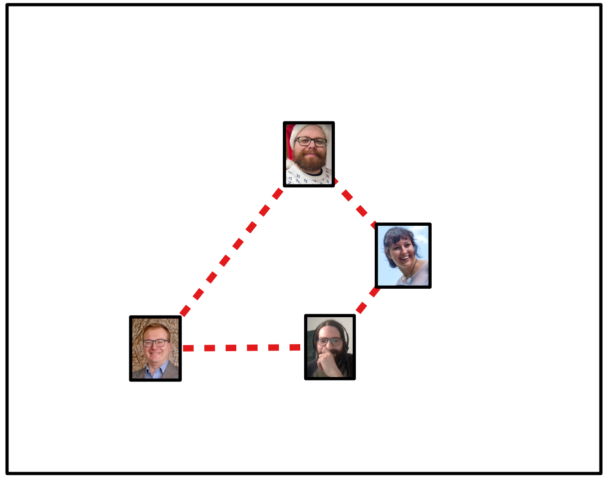
planar graph G

Partial Embedding of **Planar** Graphs



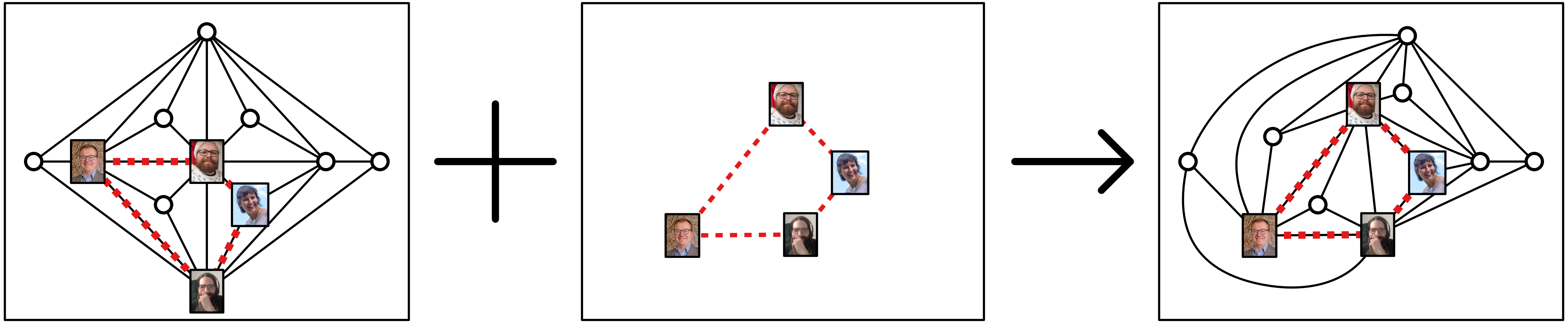
planar graph G

+



planar drawing of a subgraph $H \subseteq G$

Partial Embedding of **Planar** Graphs

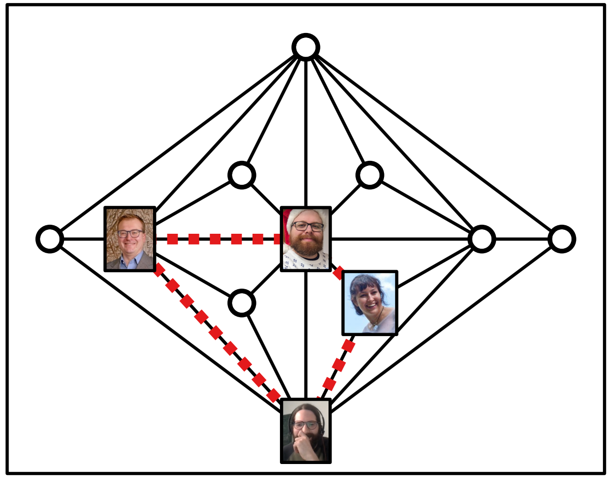


planar graph G

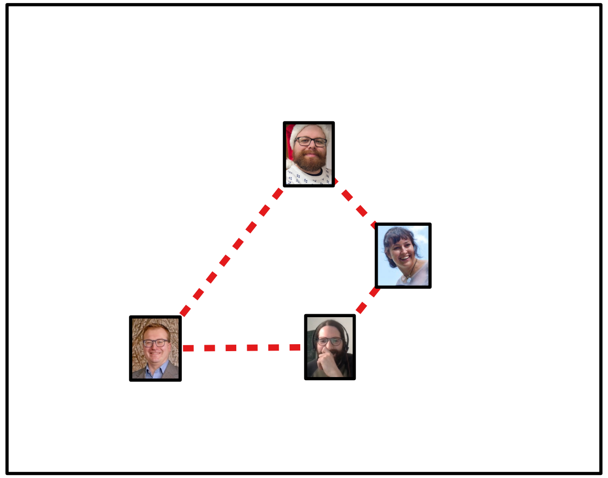
planar drawing of a subgraph $H \subseteq G$

planar drawing of G s.t. H keeps its drawing

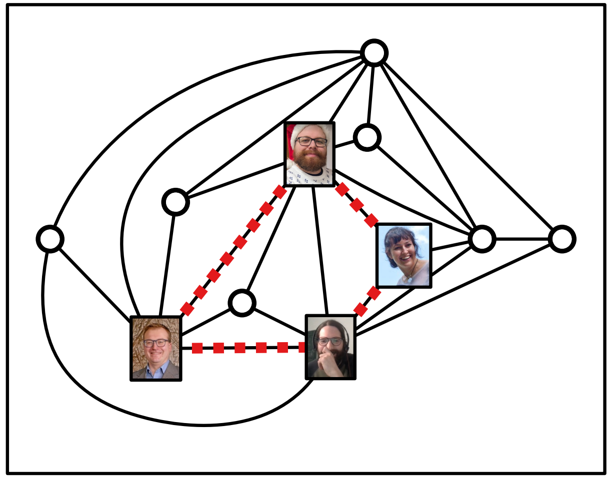
Partial Embedding of **Planar** Graphs



planar graph G



planar drawing of a subgraph $H \subseteq G$

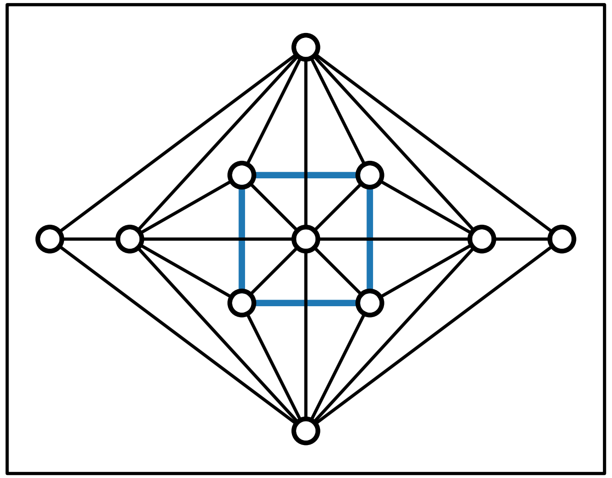


planar drawing of G s.t. H keeps its drawing

This problem can be solved in $\mathcal{O}(n)$ time.

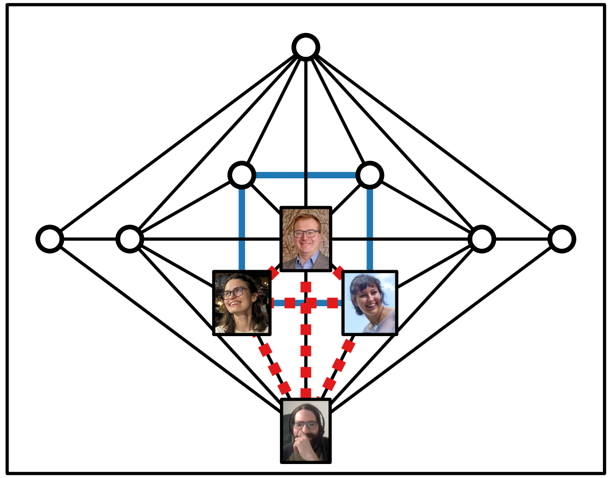
[Angelini, Di Battista, Frati, Jelínek, Kratochvíl, Patrignani, Rutter ’10]

Partial Embedding of 1-Planar Graphs



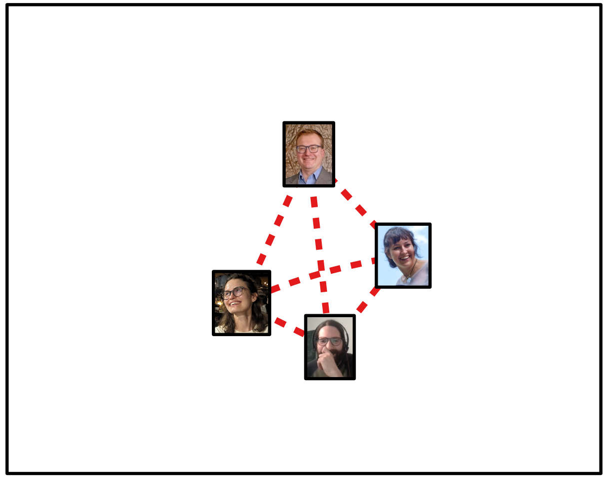
1-planar graph G
(can be drawn s.t.
every edge is crossed
at most once)

Partial Embedding of **1-Planar** Graphs



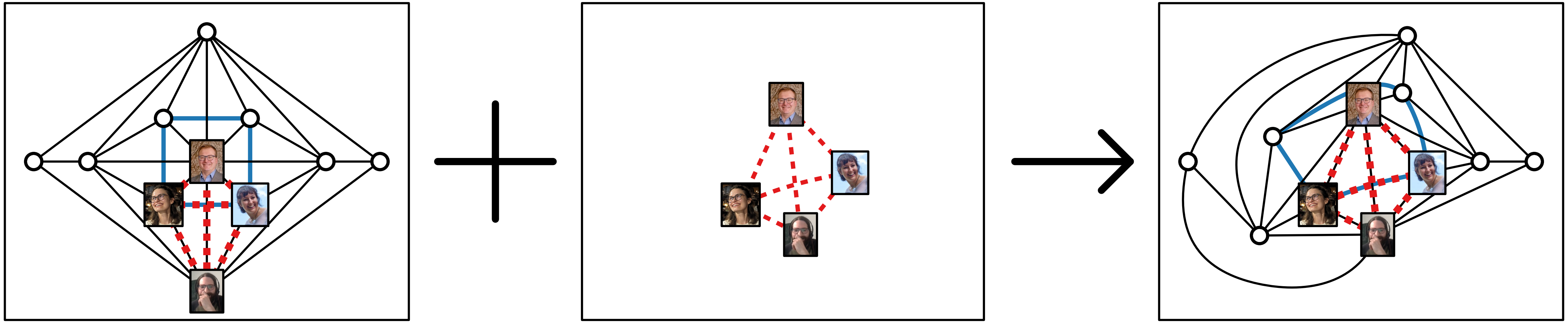
1-planar graph G
(can be drawn s.t. every edge is crossed at most once)

+



1-planar drawing of a subgraph $H \subseteq G$

Partial Embedding of **1-Planar** Graphs

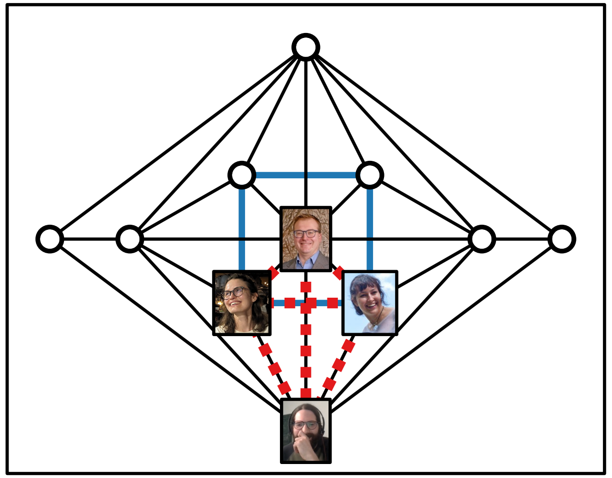


1-planar graph G (can be drawn s.t. every edge is crossed at most once)

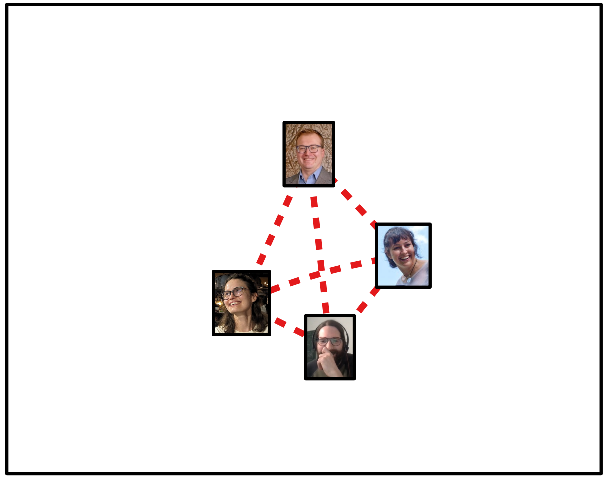
1-planar drawing of a subgraph $H \subseteq G$

1-planar drawing of G s.t. H keeps its drawing

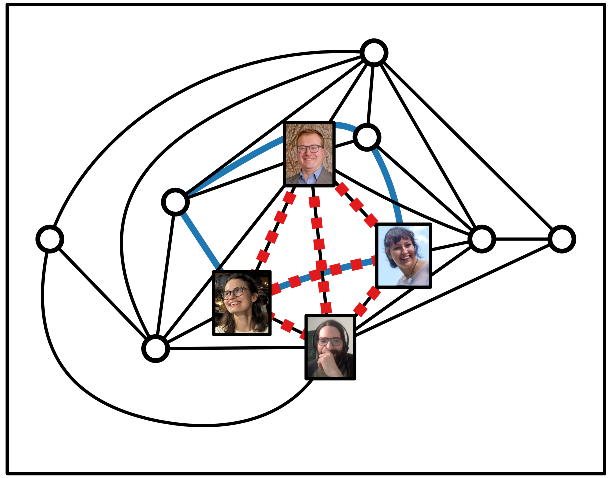
Partial Embedding of **1-Planar** Graphs



1-planar graph G (can be drawn s.t. every edge is crossed at most once)



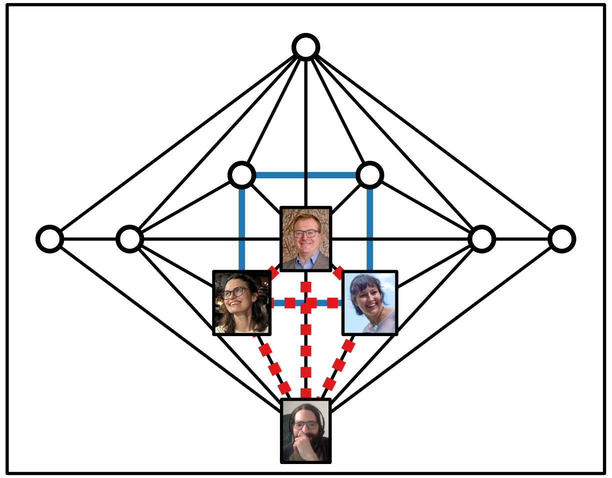
1-planar drawing of a subgraph $H \subseteq G$



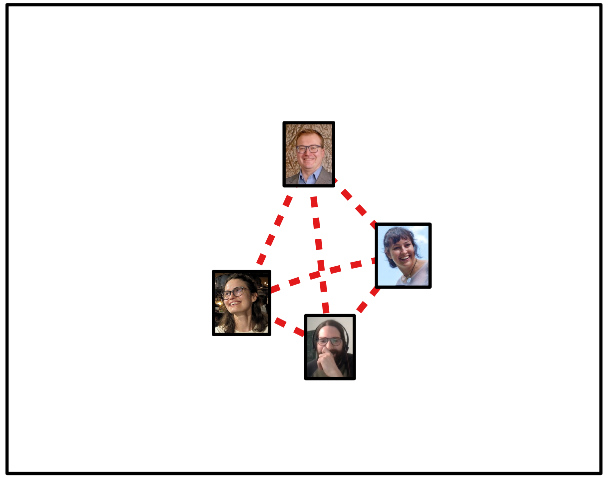
1-planar drawing of G s.t. H keeps its drawing

This problem is NP-hard even if $H = \emptyset$. [Grigoriev & Bodlaender '07]

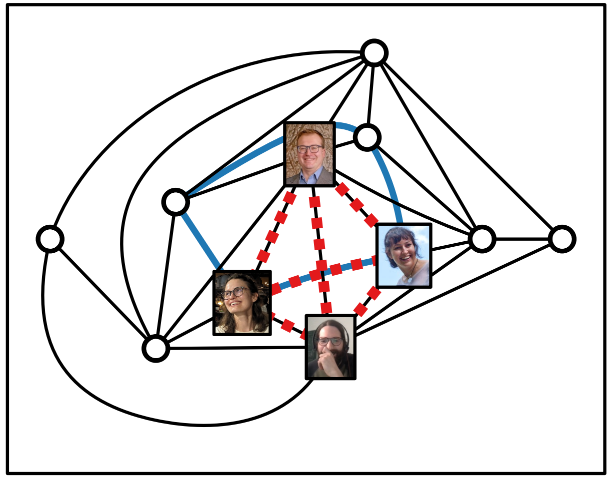
Partial Embedding of **1-Planar** Graphs



1-planar graph G
(can be drawn s.t. every edge is crossed at most once)



1-planar drawing of a subgraph $H \subseteq G$

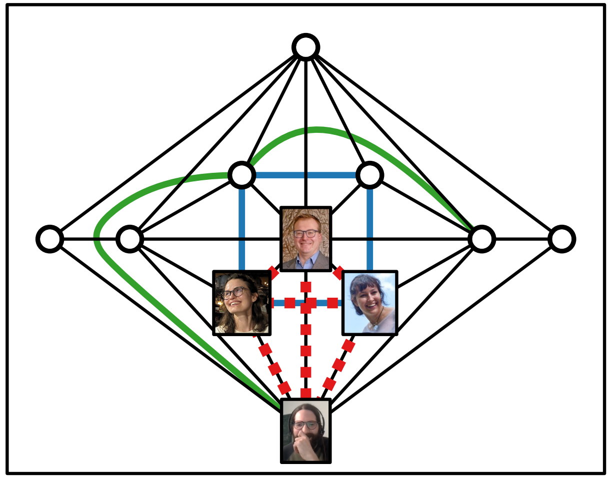


1-planar drawing of G s.t. H keeps its drawing

This problem is NP-hard even if $H = \emptyset$. [Grigoriev & Bodlaender '07]

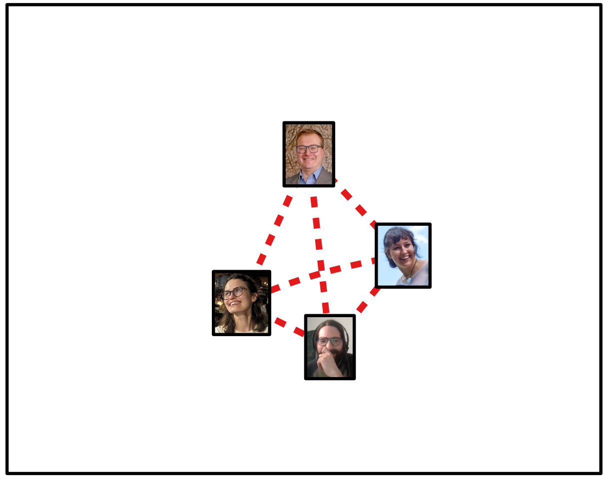
This problem is in FPT parameterized by the vertex+edge deletion distance between G and H. [Eiben, Ganian, Hamm, Klute & Nöllenburg '20]

Partial Embedding of *k*-**Planar** Graphs



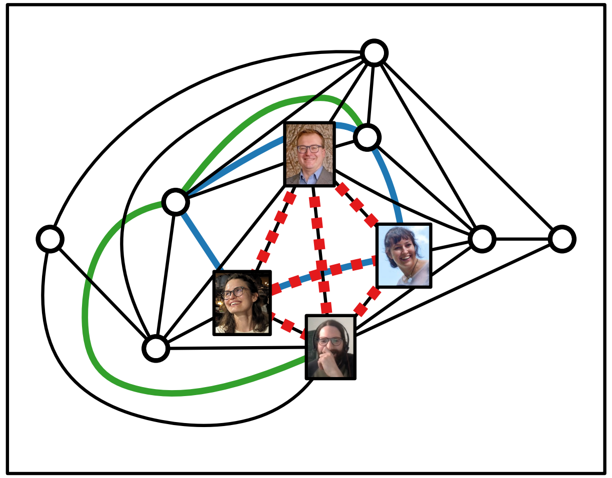
k-**planar** graph G
(can be drawn s.t. every edge is crossed at most k times)

+



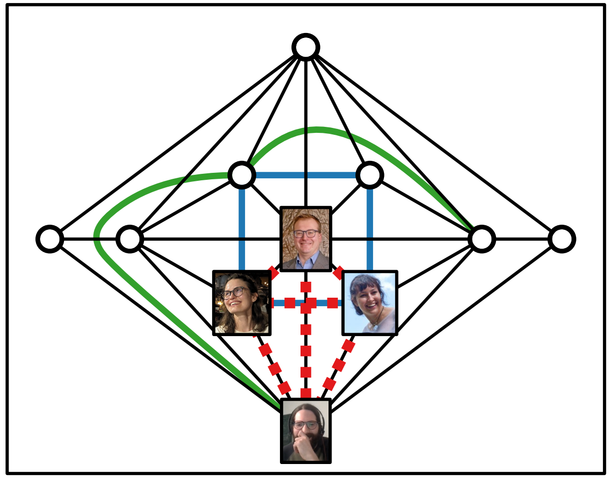
k-**planar** drawing of a subgraph $H \subseteq G$

→

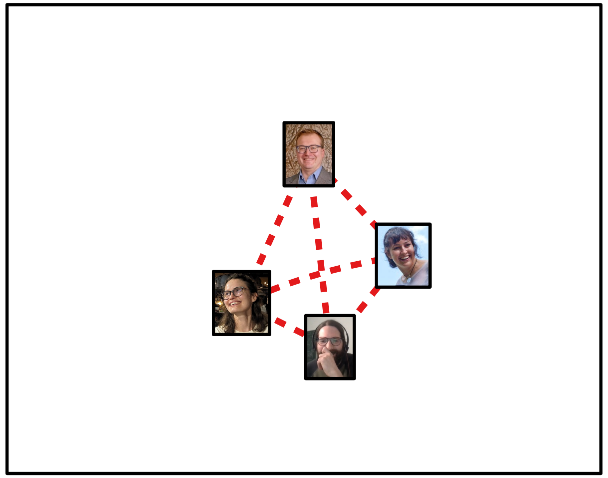


k-**planar** drawing of G s.t. H keeps its drawing

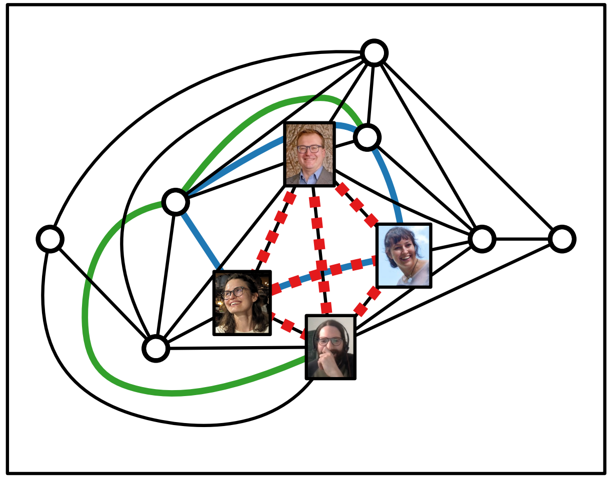
Partial Embedding of *k*-**Planar** Graphs



k-**planar** graph G (can be drawn s.t. every edge is crossed at most k times)



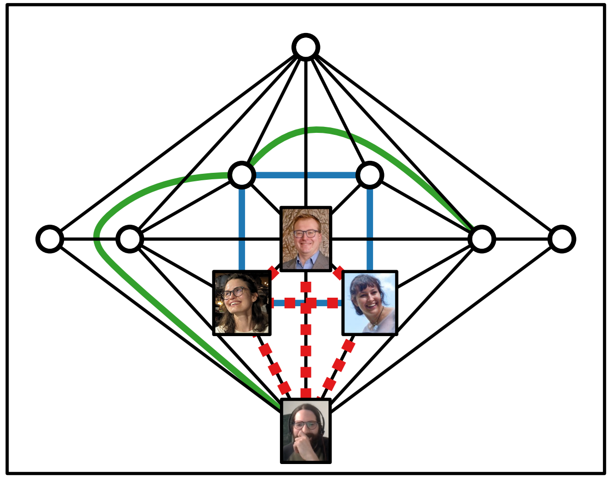
k-**planar** drawing of a subgraph $H \subseteq G$



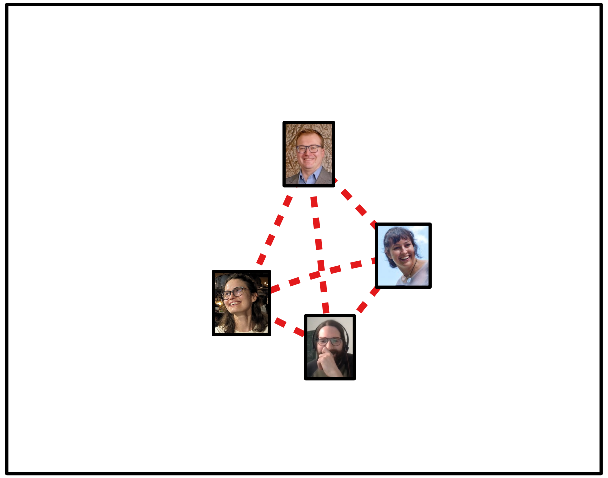
k-**planar** drawing of G s.t. H keeps its drawing

This problem is NP-hard for any constant k even if $H = \emptyset$. [Urschel & Wellens '21]

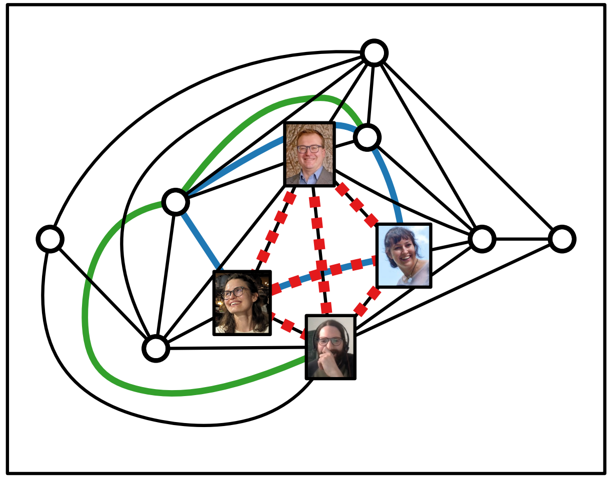
Partial Embedding of *k*-**Planar** Graphs



k-**planar** graph G (can be drawn s.t. every edge is crossed at most k times)



k-**planar** drawing of a subgraph $H \subseteq G$

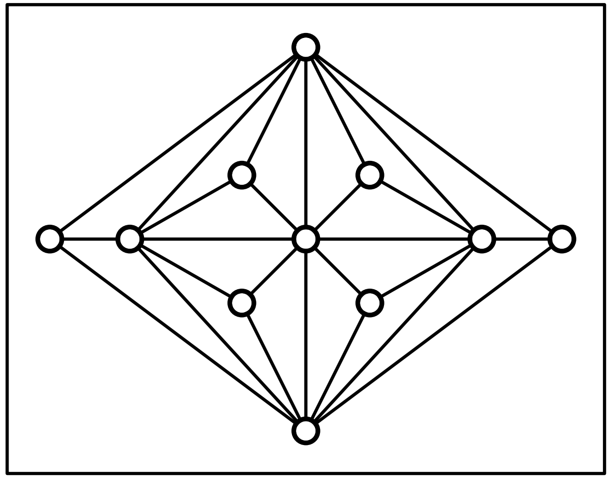


k-**planar** drawing of G s.t. H keeps its drawing

This problem is NP-hard for any constant k even if $H = \emptyset$. [Urschel & Wellens '21]

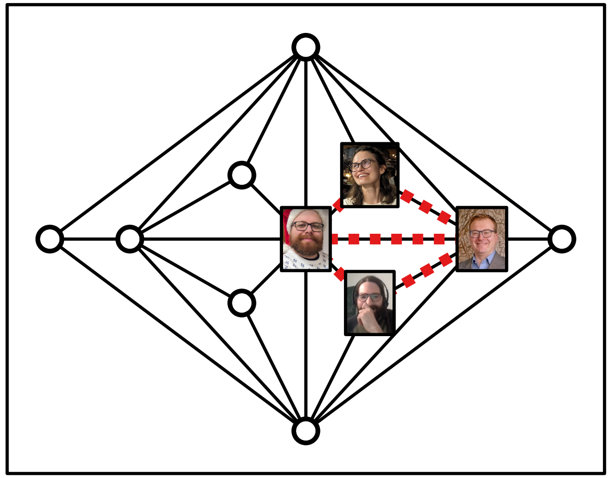
This problem is in FPT parameterized by k + #edges in $G - H$.
[Ganian, Hamm, Klute, Parada & Vogtenhuber '21]

Generalization of Partial Embedding

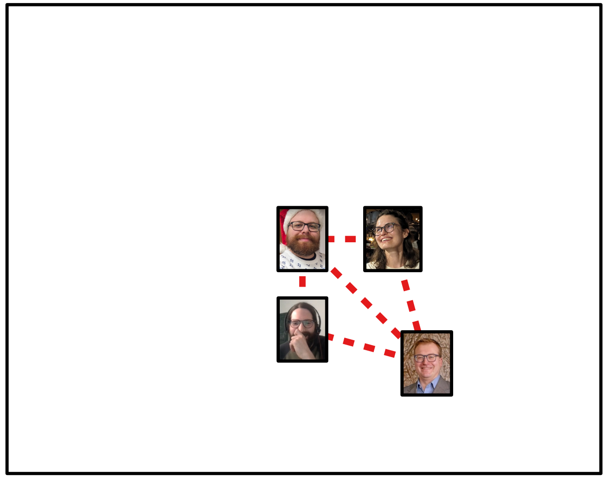


Graph G +
drawing style Φ
(e.g., planar)

Generalization of Partial Embedding

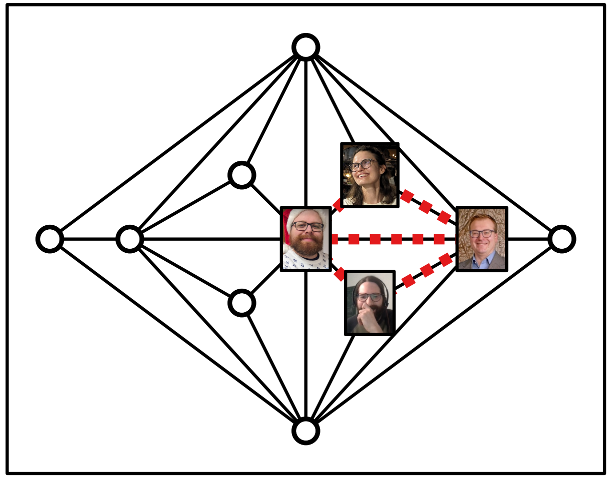


Graph G +
drawing style Φ
(e.g., planar)

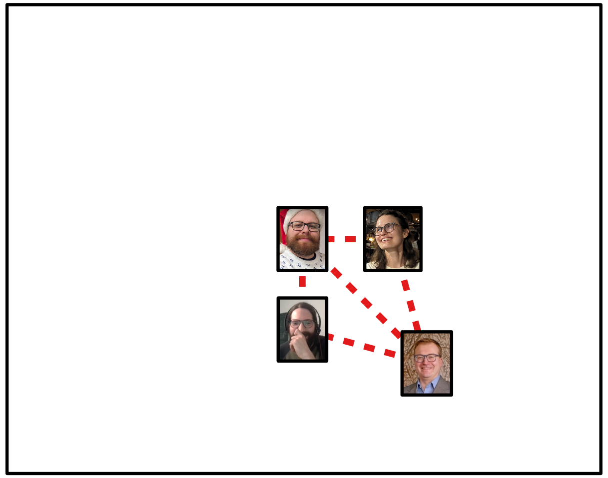


Drawing with style
$\Phi' \subseteq \Phi$
(e.g., straight-line planar)
of a subgraph $H \subseteq G$

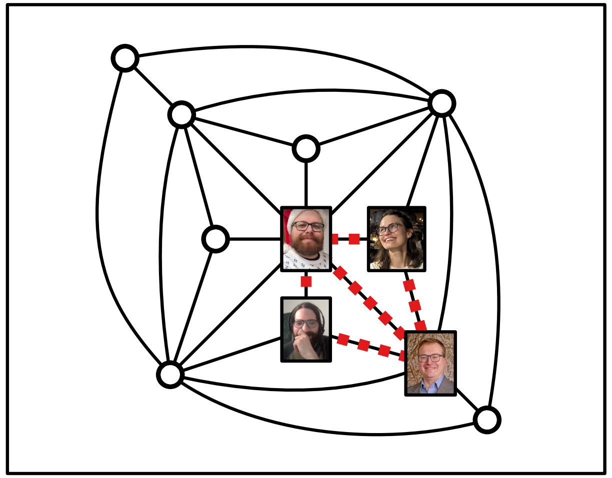
Generalization of Partial Embedding



Graph G +
drawing style Φ
(e.g., planar)

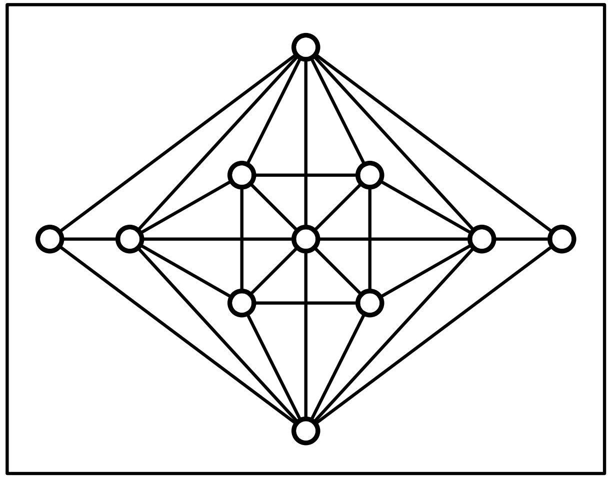


Drawing with style
$\Phi' \subseteq \Phi$
(e.g., straight-line planar)
of a subgraph $H \subseteq G$



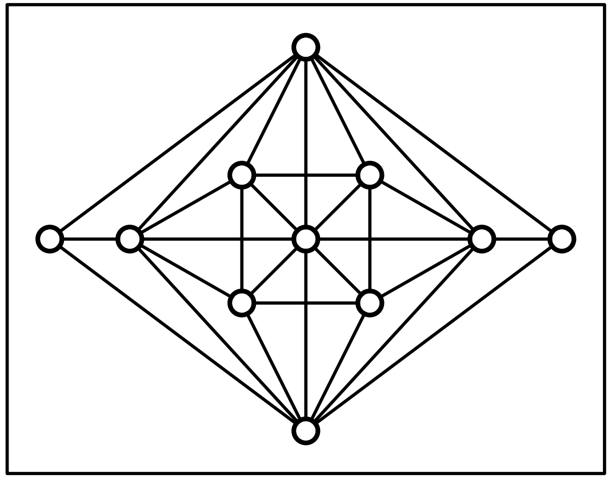
Drawing with style Φ of G
s.t. H **keeps its drawing**

1-Plane Insertion Into a Plane Graph

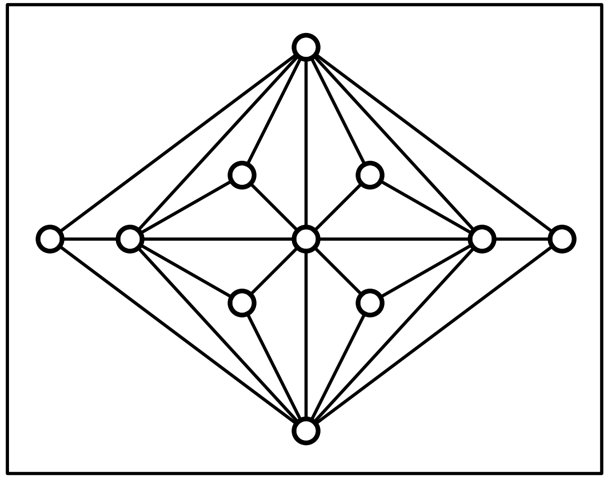


1-planar graph G

1-Plane Insertion Into a Plane Graph

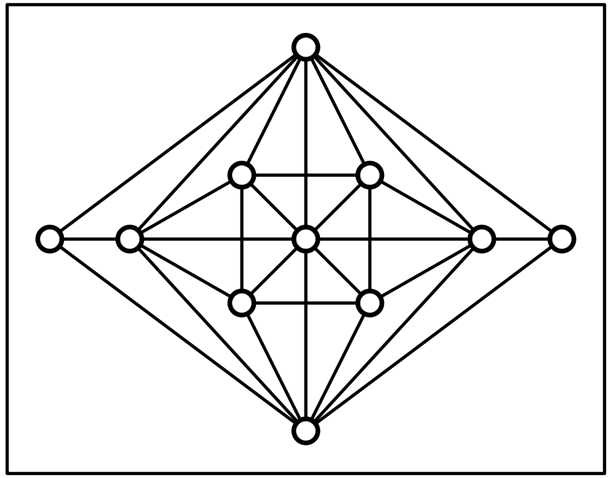


1-planar graph G

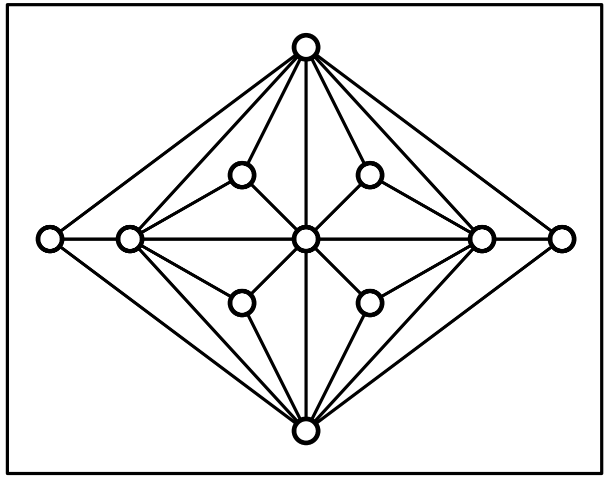


planar drawing of a
spanning subgraph $H \subseteq G$

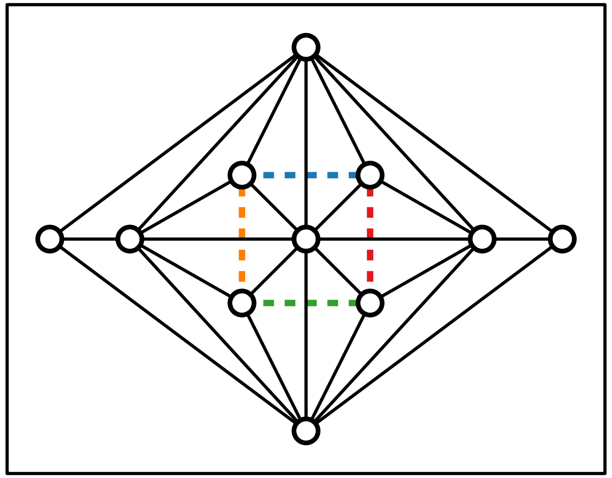
1-Plane Insertion Into a Plane Graph



1-planar graph G

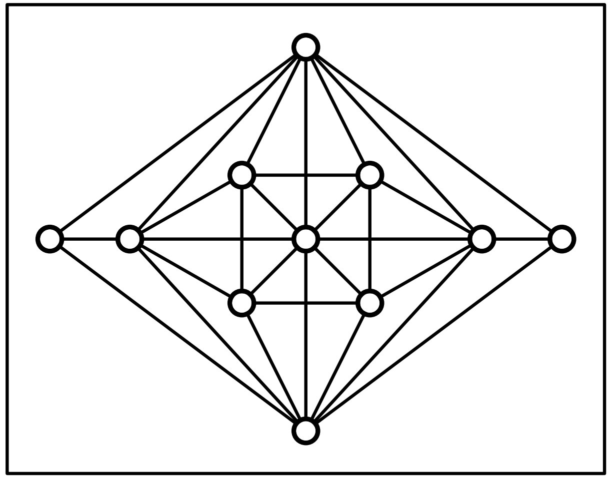


planar drawing of a **spanning** subgraph $H \subseteq G$

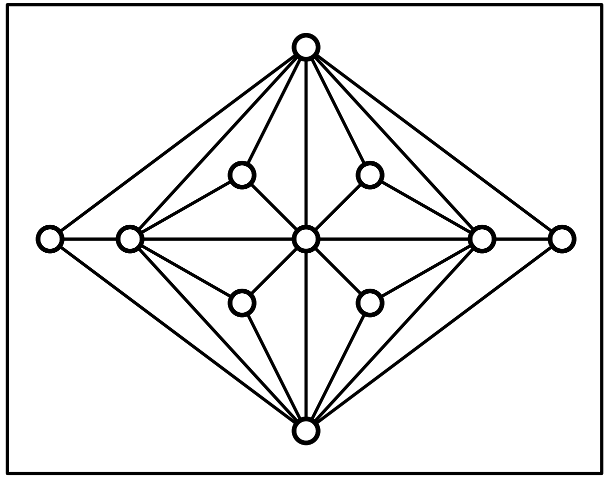


1-planar drawing of G that keeps the drawing of H

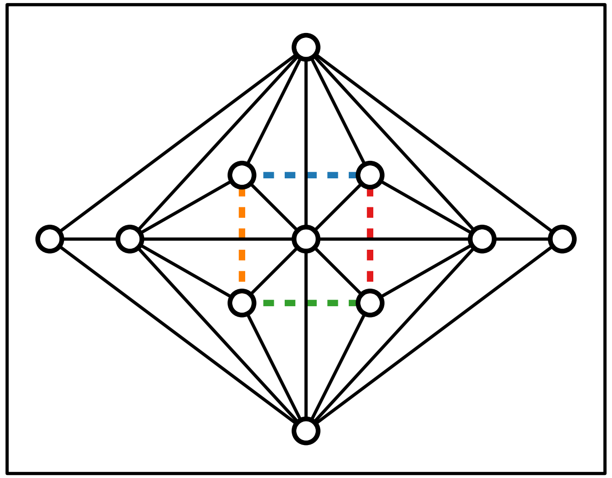
1-Plane Insertion Into a Plane Graph



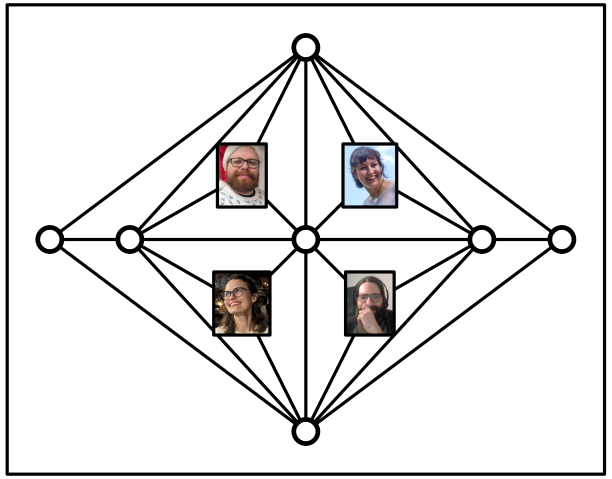
1-planar graph G



planar drawing of a spanning subgraph $H \subseteq G$

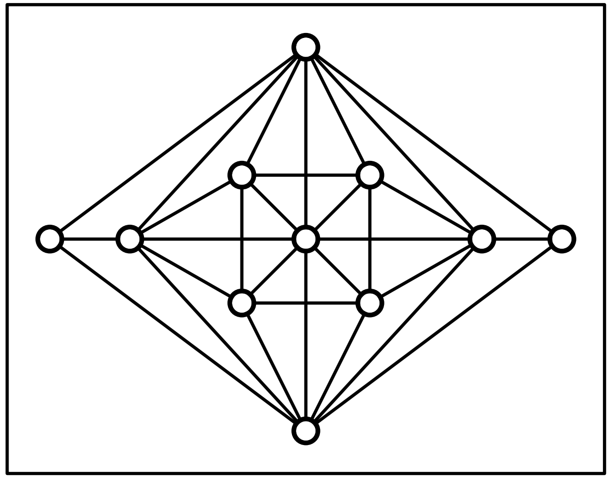


1-planar drawing of G that keeps the drawing of H

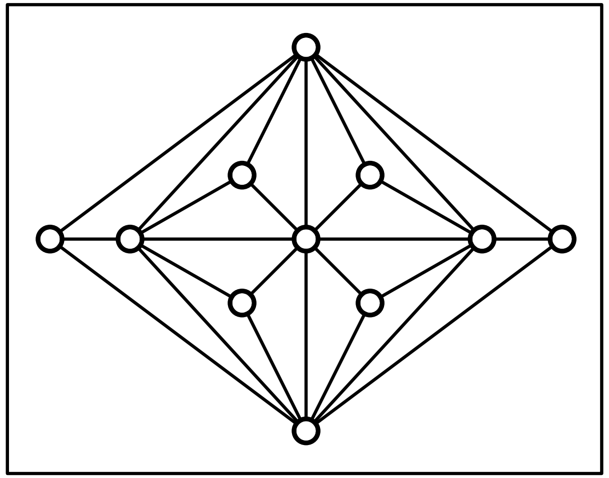


plane graph G

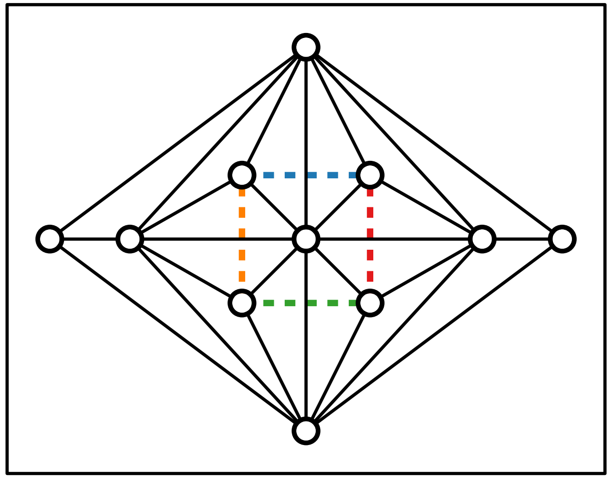
1-Plane Insertion Into a Plane Graph



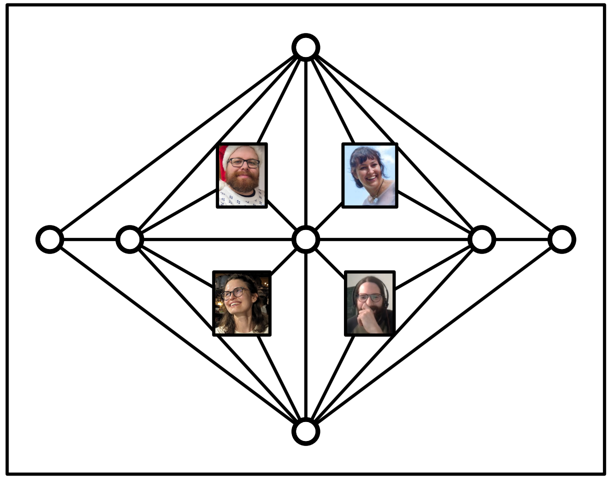
1-planar graph G



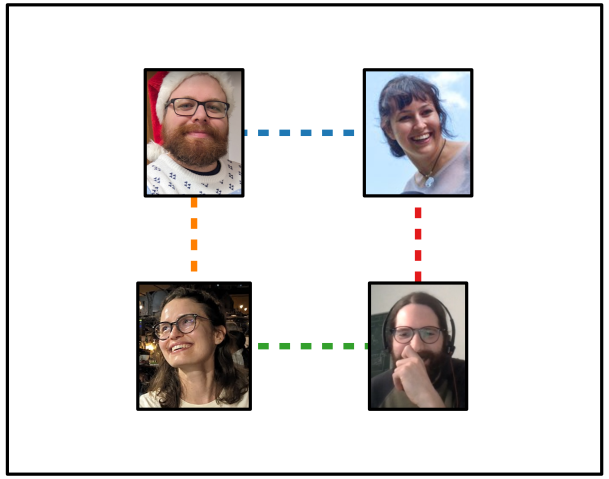
planar drawing of a
spanning subgraph $H \subseteq G$



1-planar drawing of G
that keeps the drawing of H

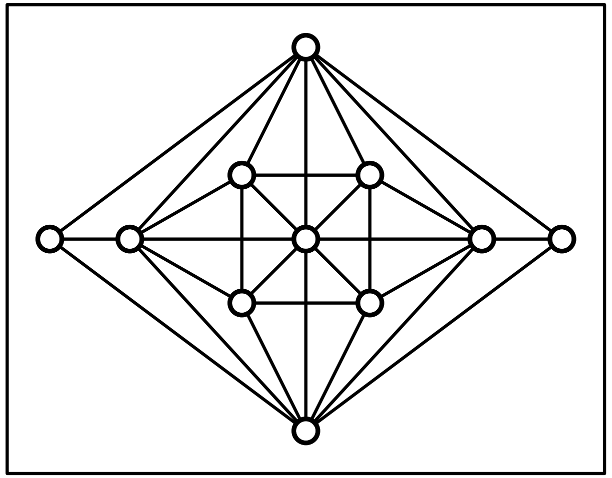


plane graph G

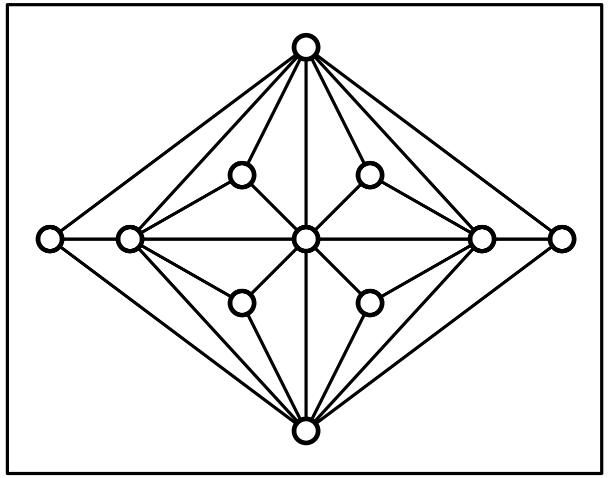


edges E' btw. vtcs in G

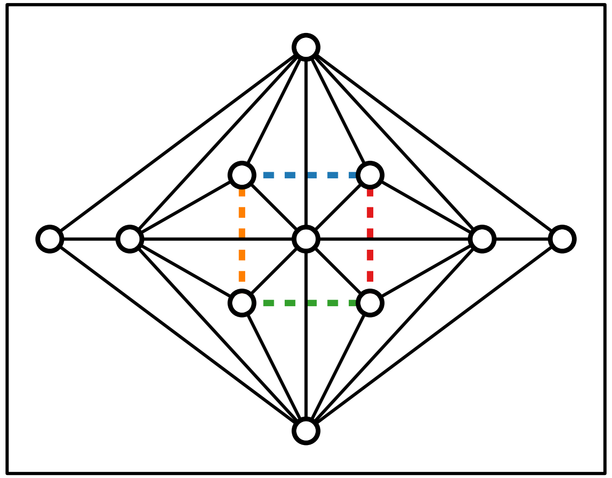
1-Plane Insertion Into a **Plane** Graph



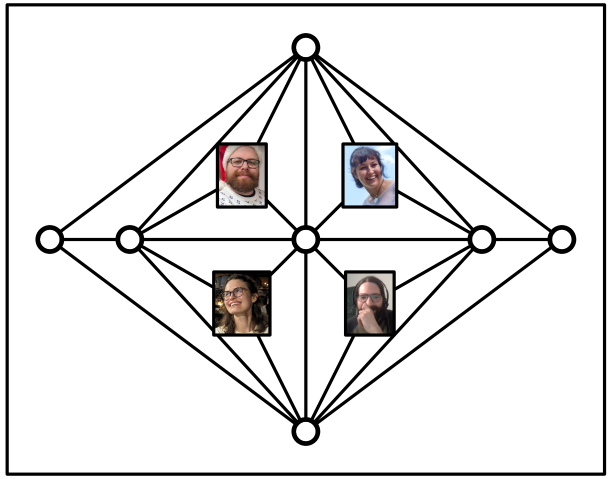
1-planar graph G



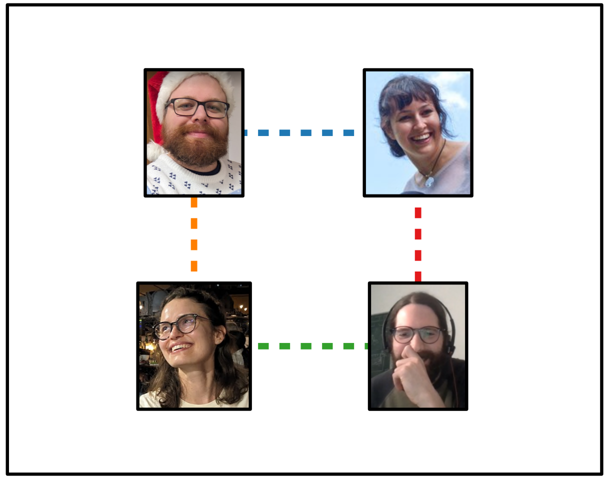
planar drawing of a **spanning** subgraph $H \subseteq G$



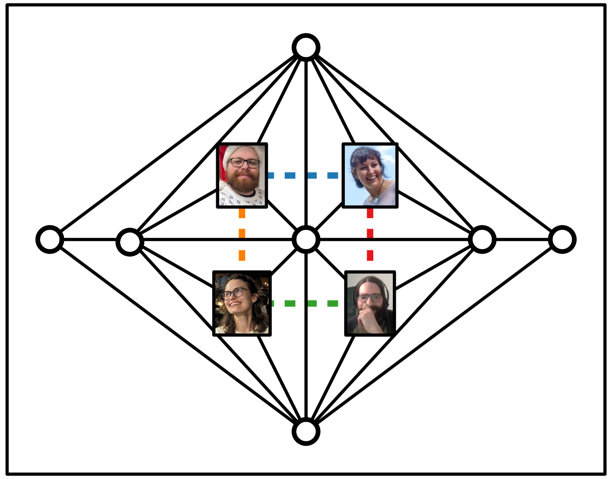
1-planar drawing of G that keeps the drawing of H



plane graph G

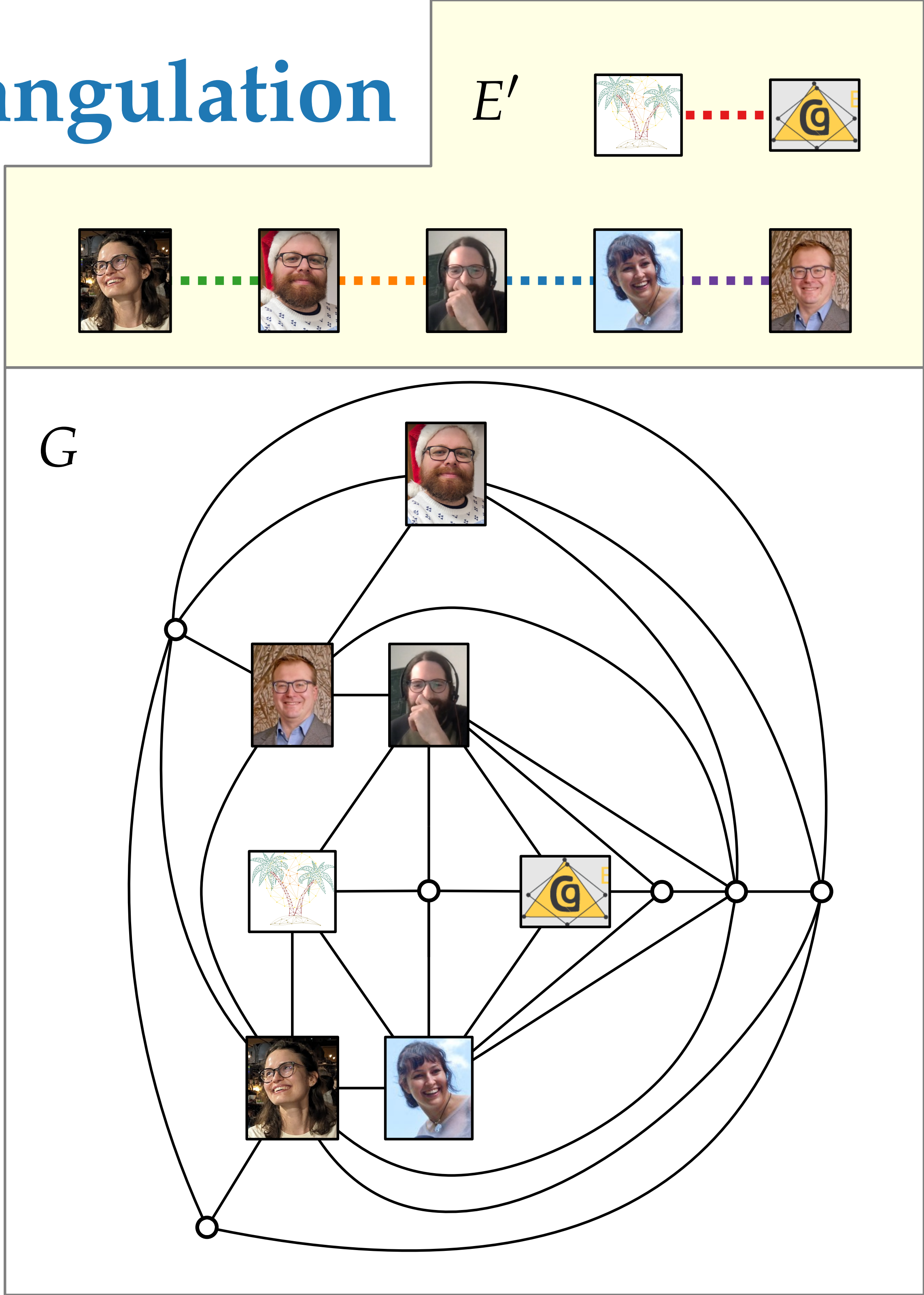


edges E' btw. vtcs in G



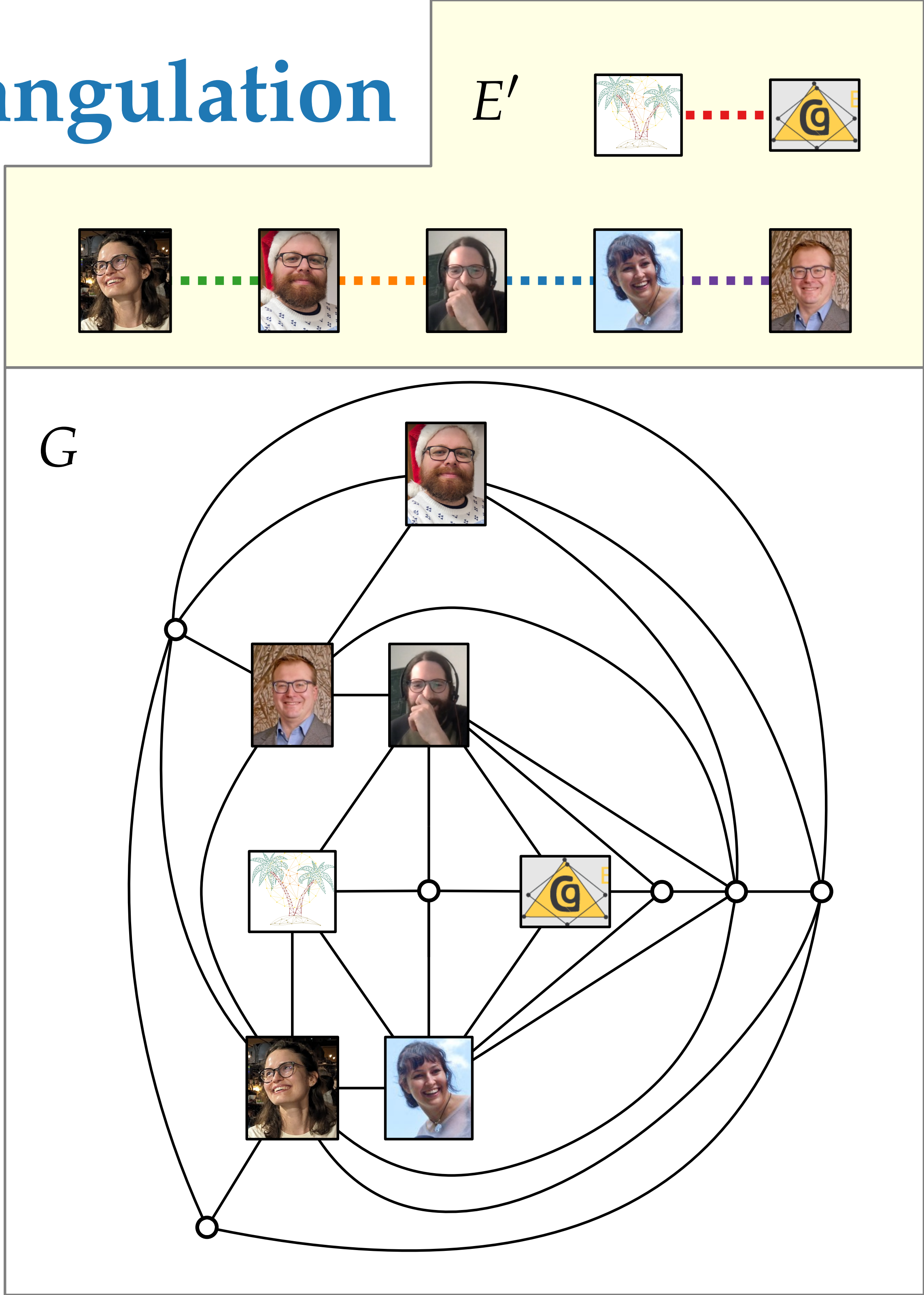
1-planar drawing of $G + E'$ that keeps the embedding of G

1-Plane Insertion Into a Plane Triangulation



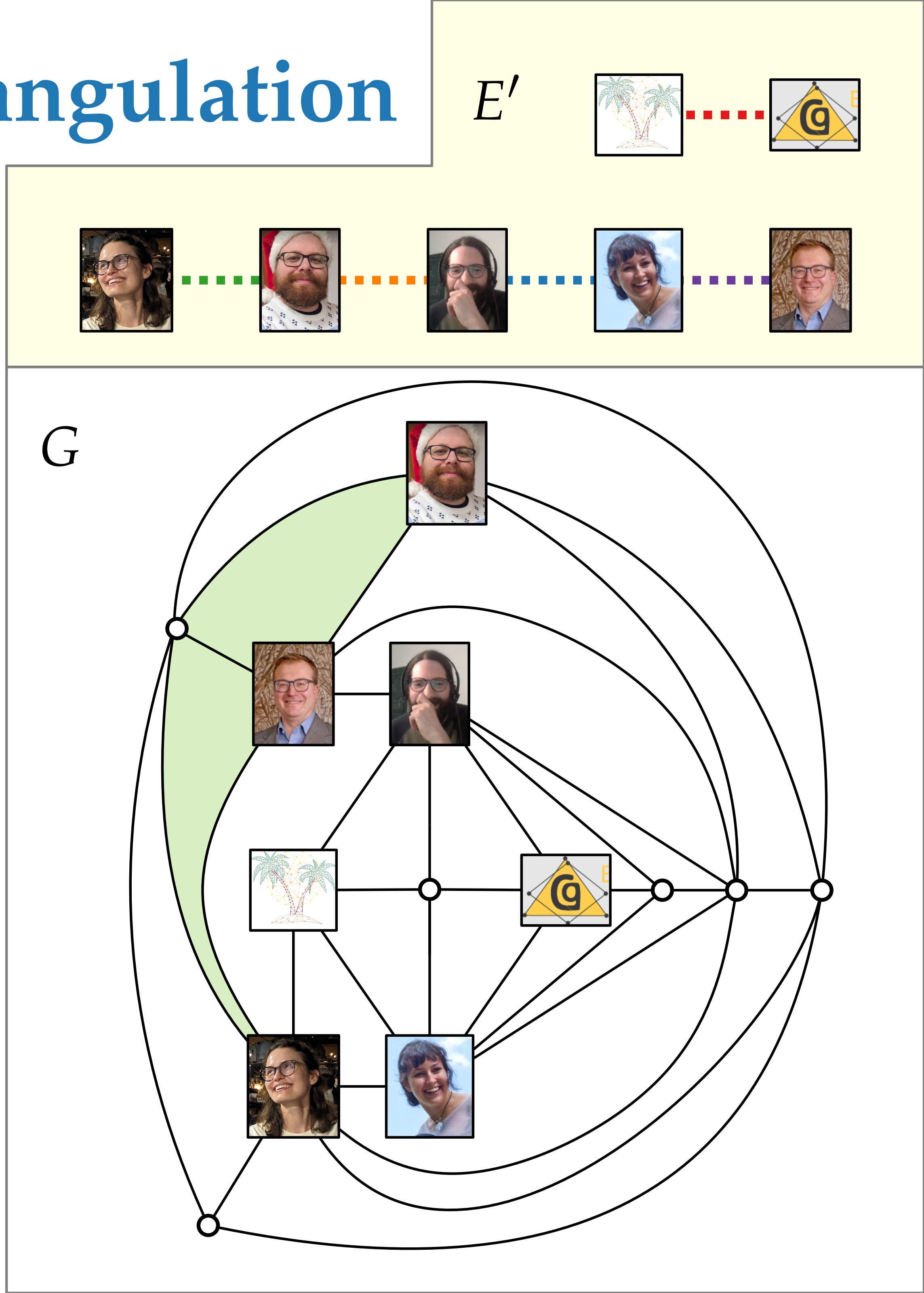
1-Plane Insertion Into a Plane Triangulation

1. For each edge, find all possibilities to route it



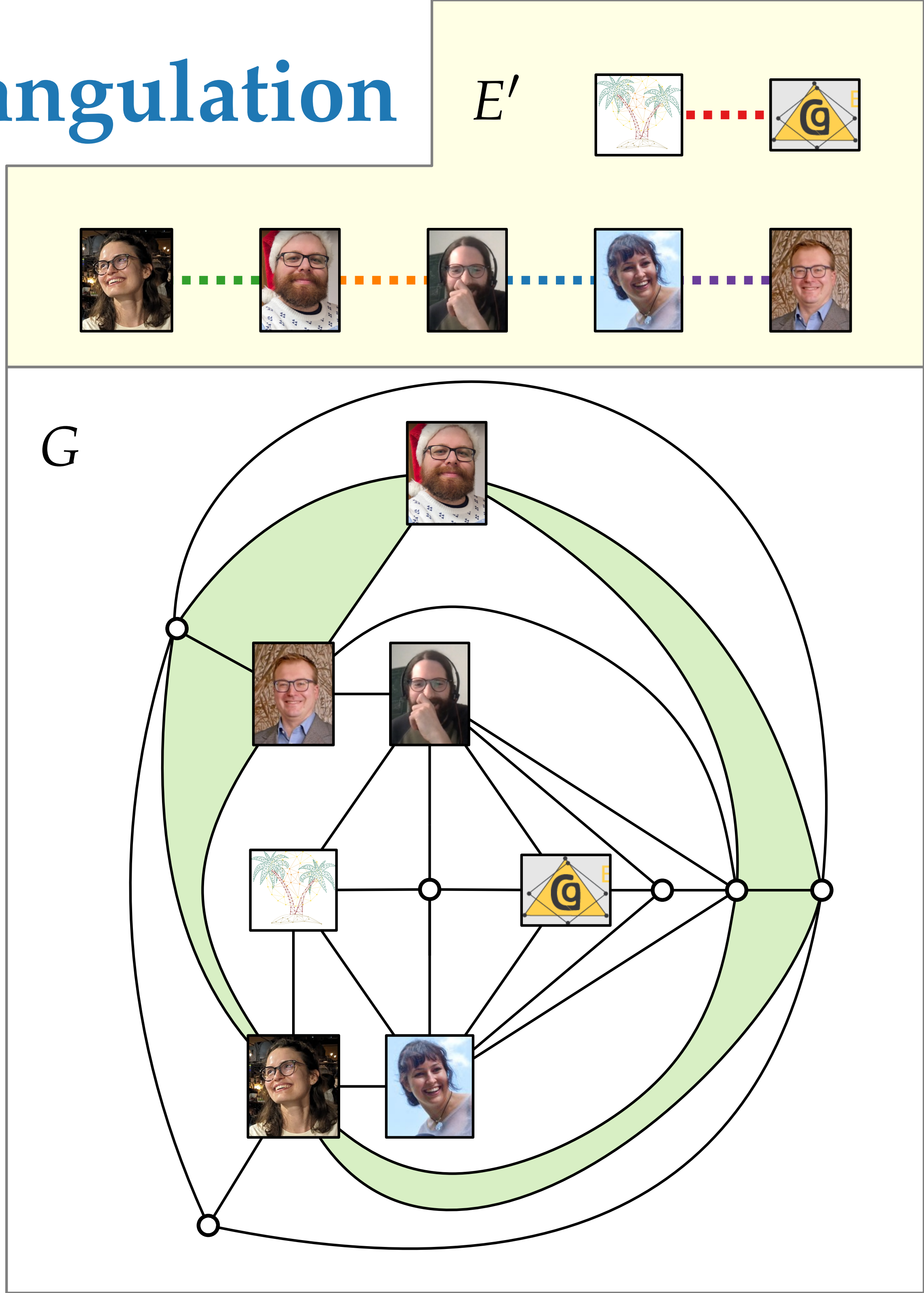
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it



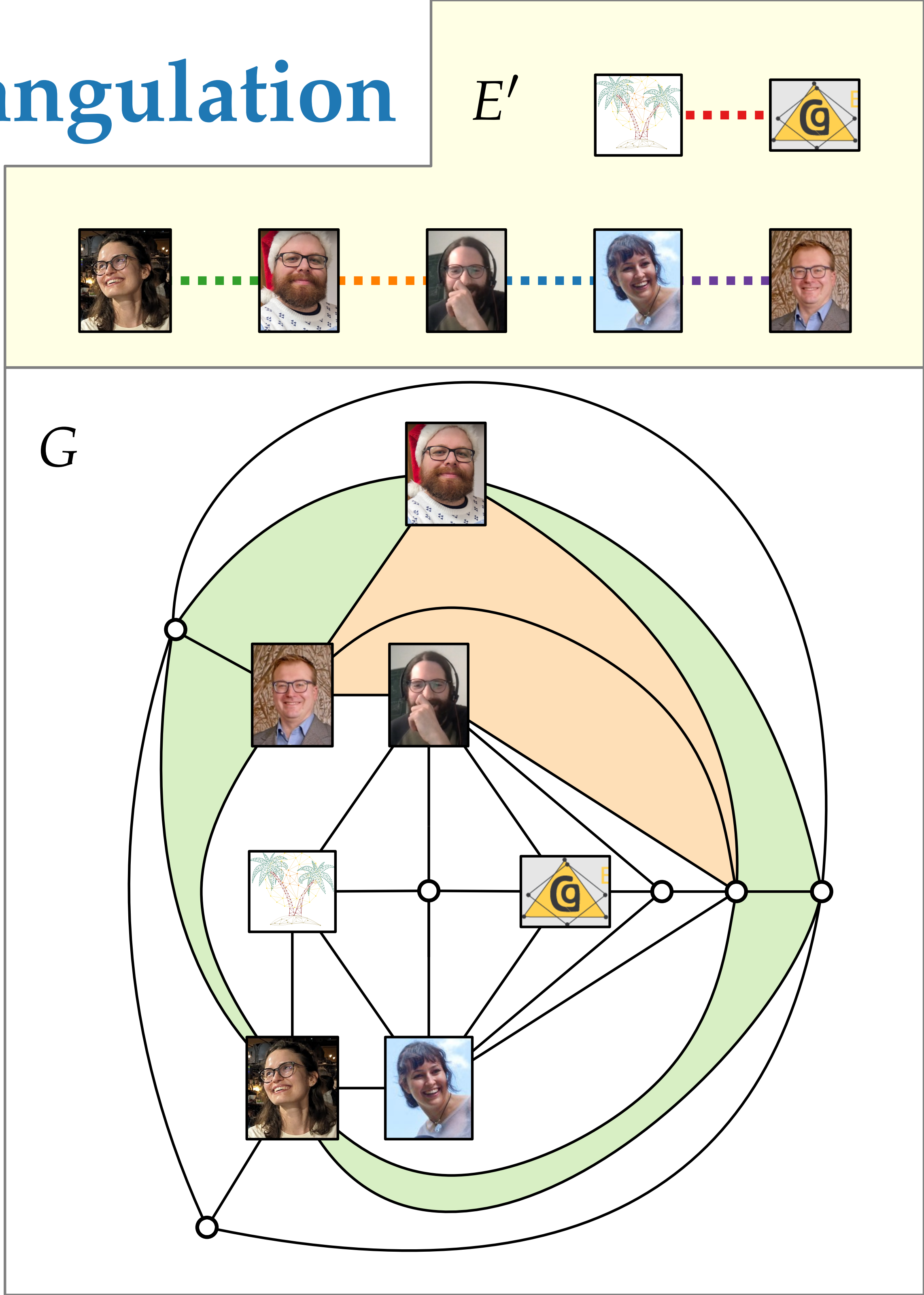
1-Plane Insertion Into a Plane Triangulation

1. For each edge, find all possibilities to route it



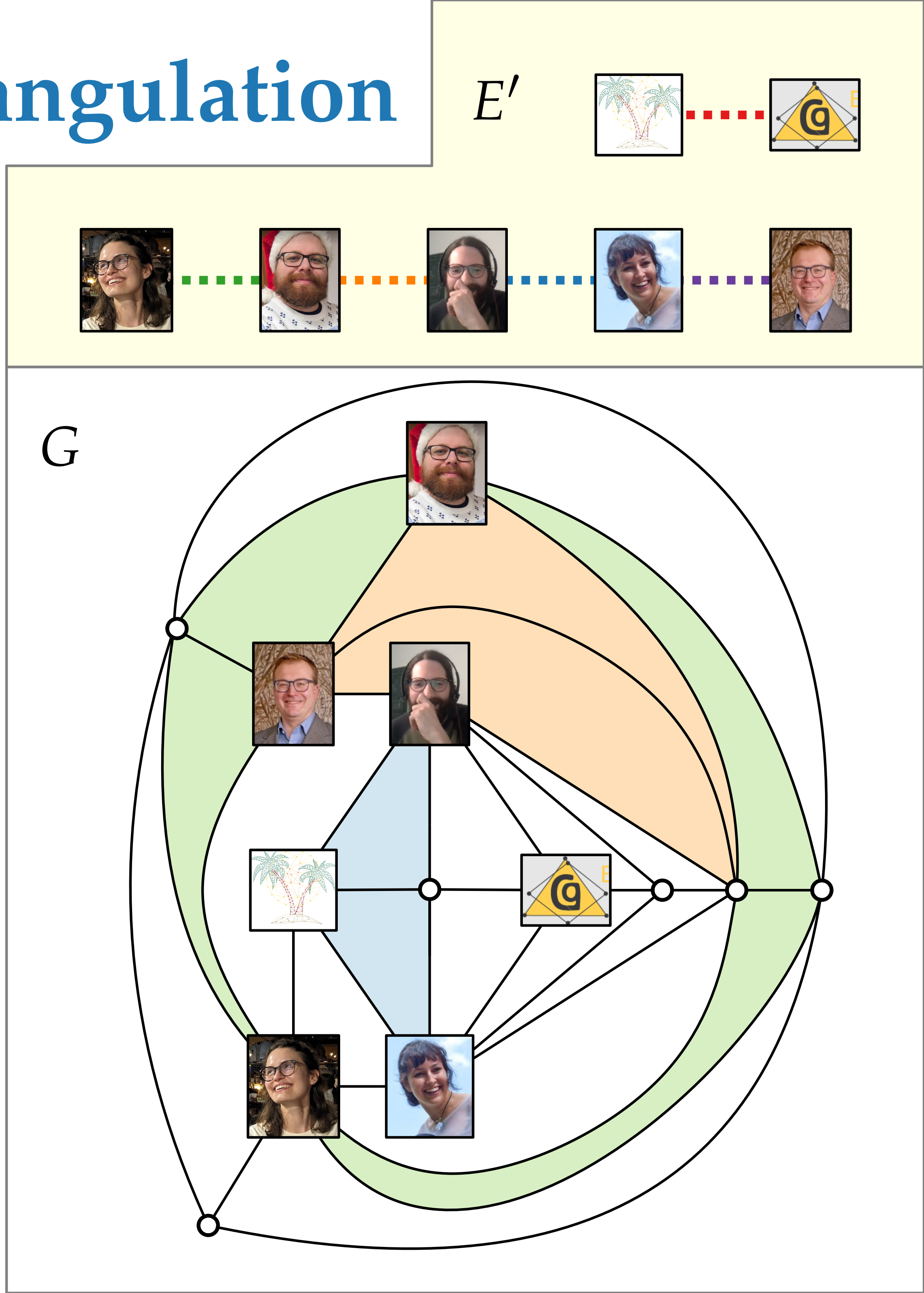
1-Plane Insertion Into a Plane Triangulation

1. For each edge, find all possibilities to route it



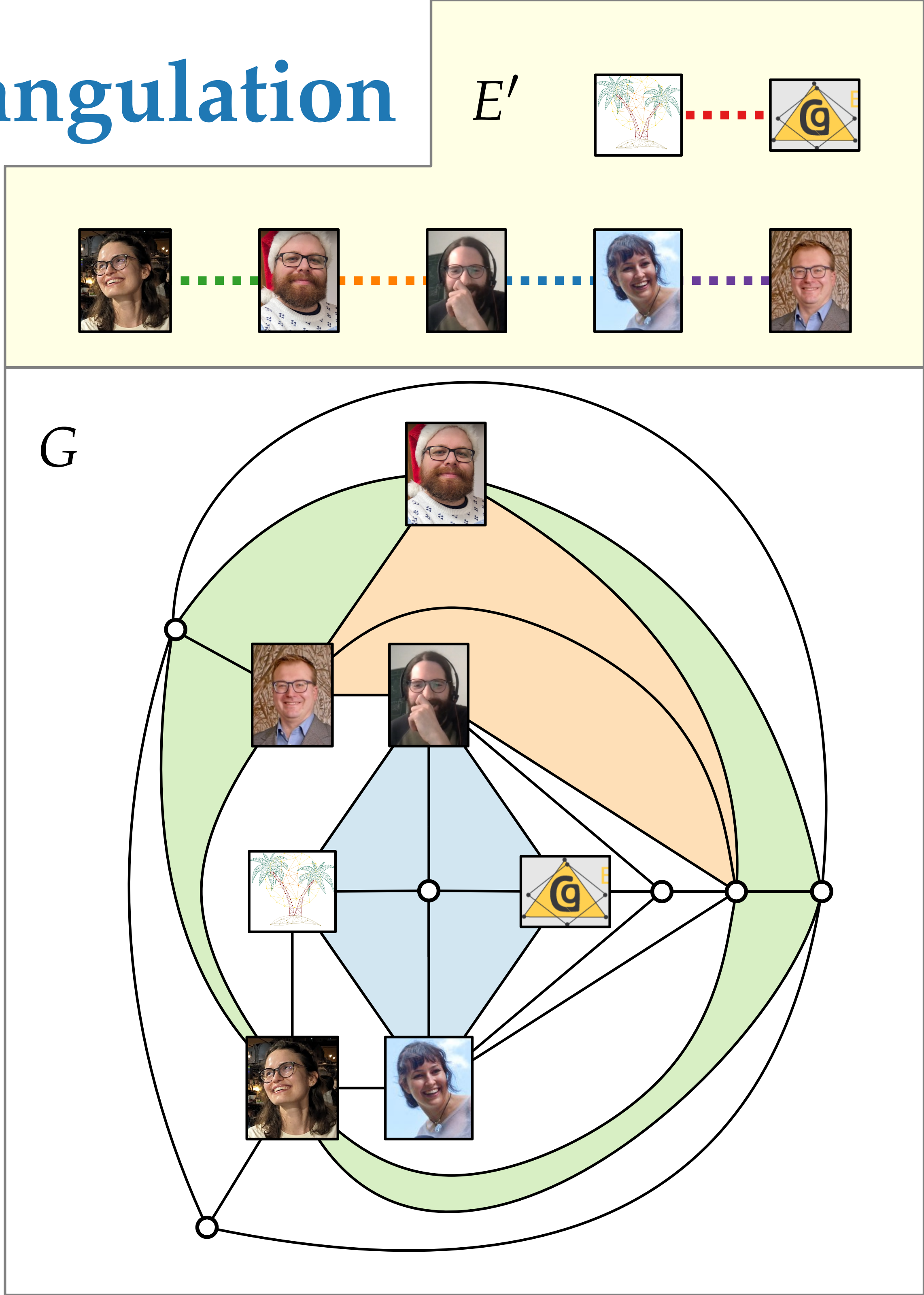
1-Plane Insertion Into a Plane Triangulation

1. For each edge, find all possibilities to route it



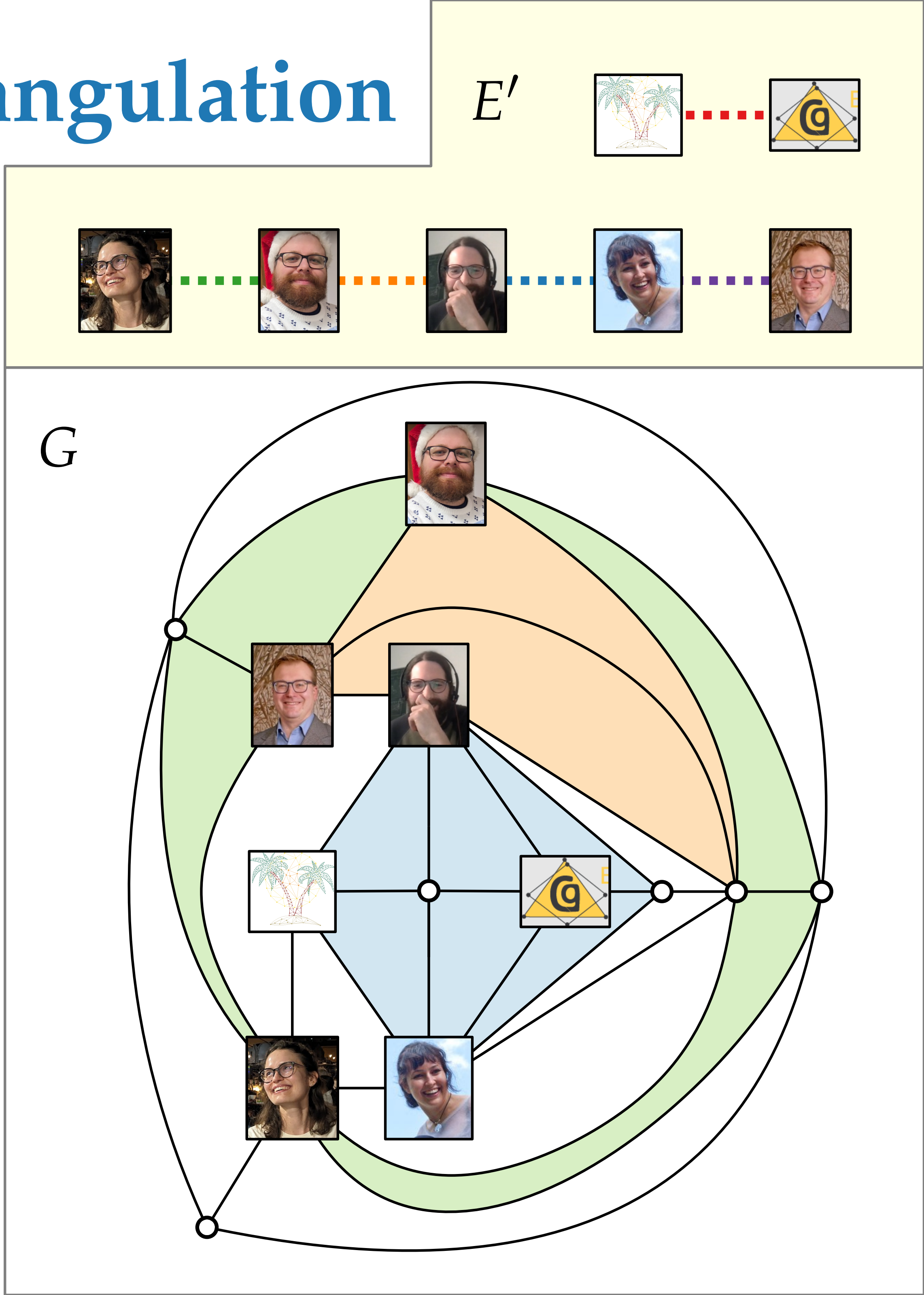
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it



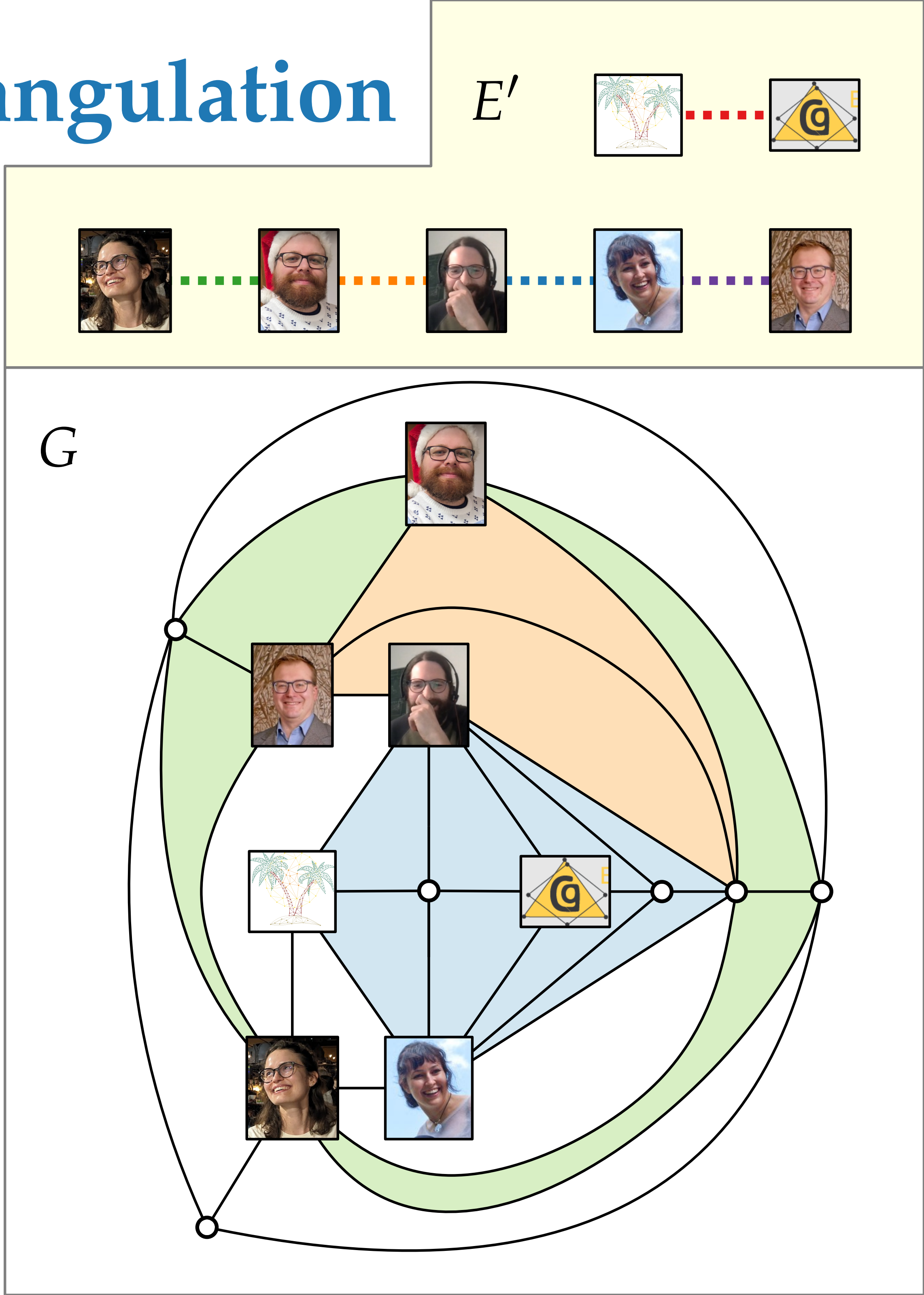
1-Plane Insertion Into a Plane Triangulation

1. For each edge, find all possibilities to route it



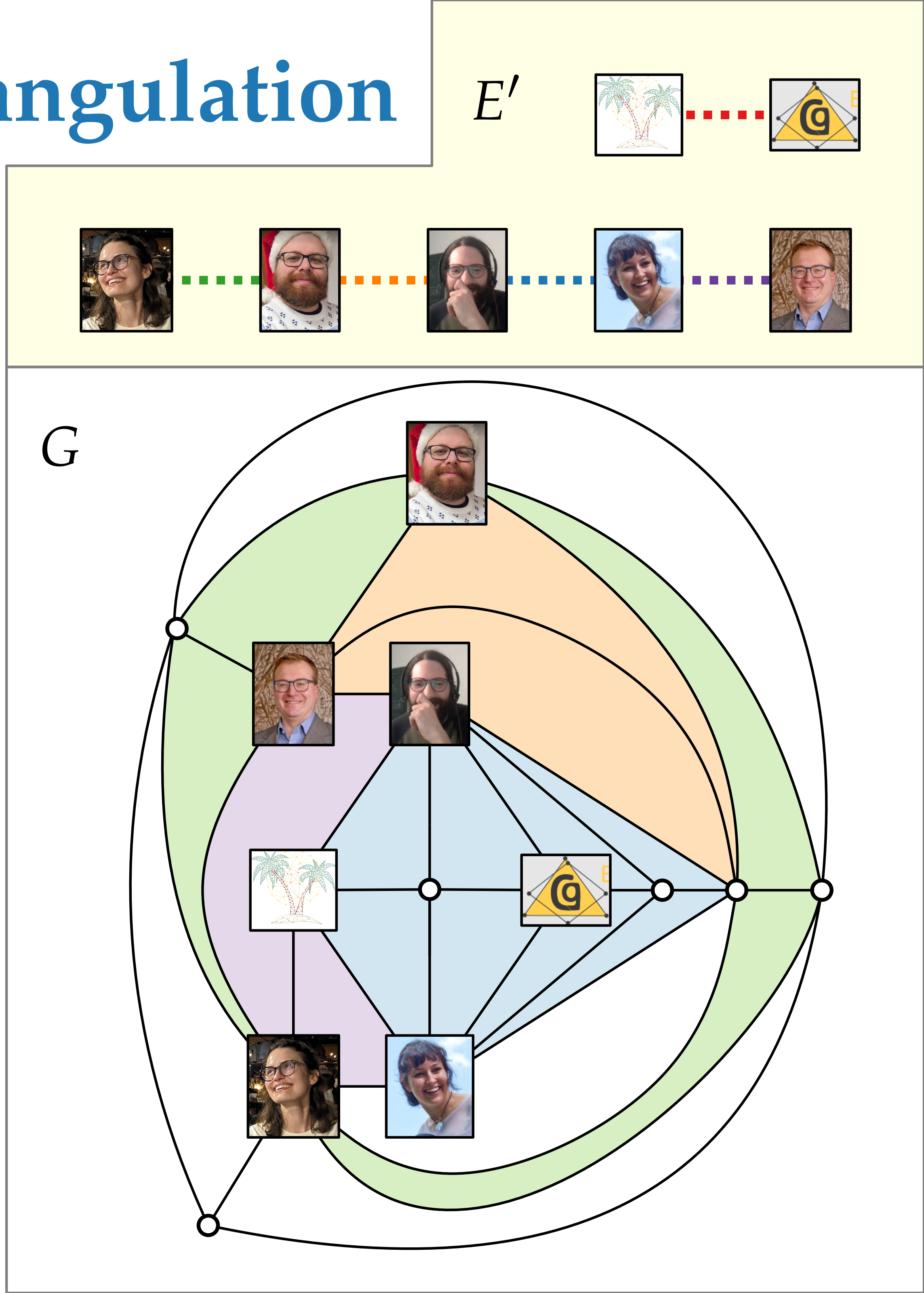
1-Plane Insertion Into a Plane Triangulation

1. For each edge, find all possibilities to route it



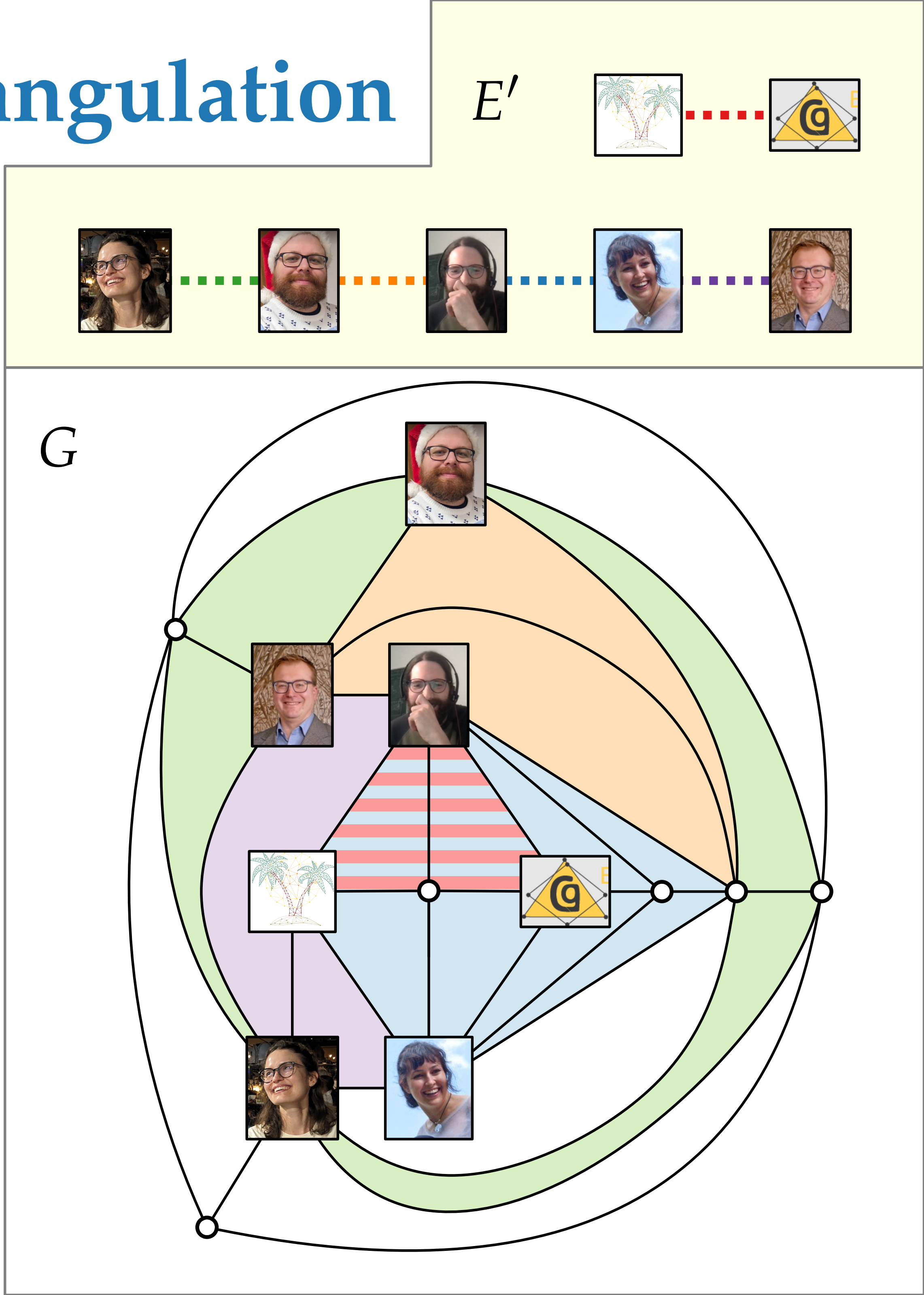
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it



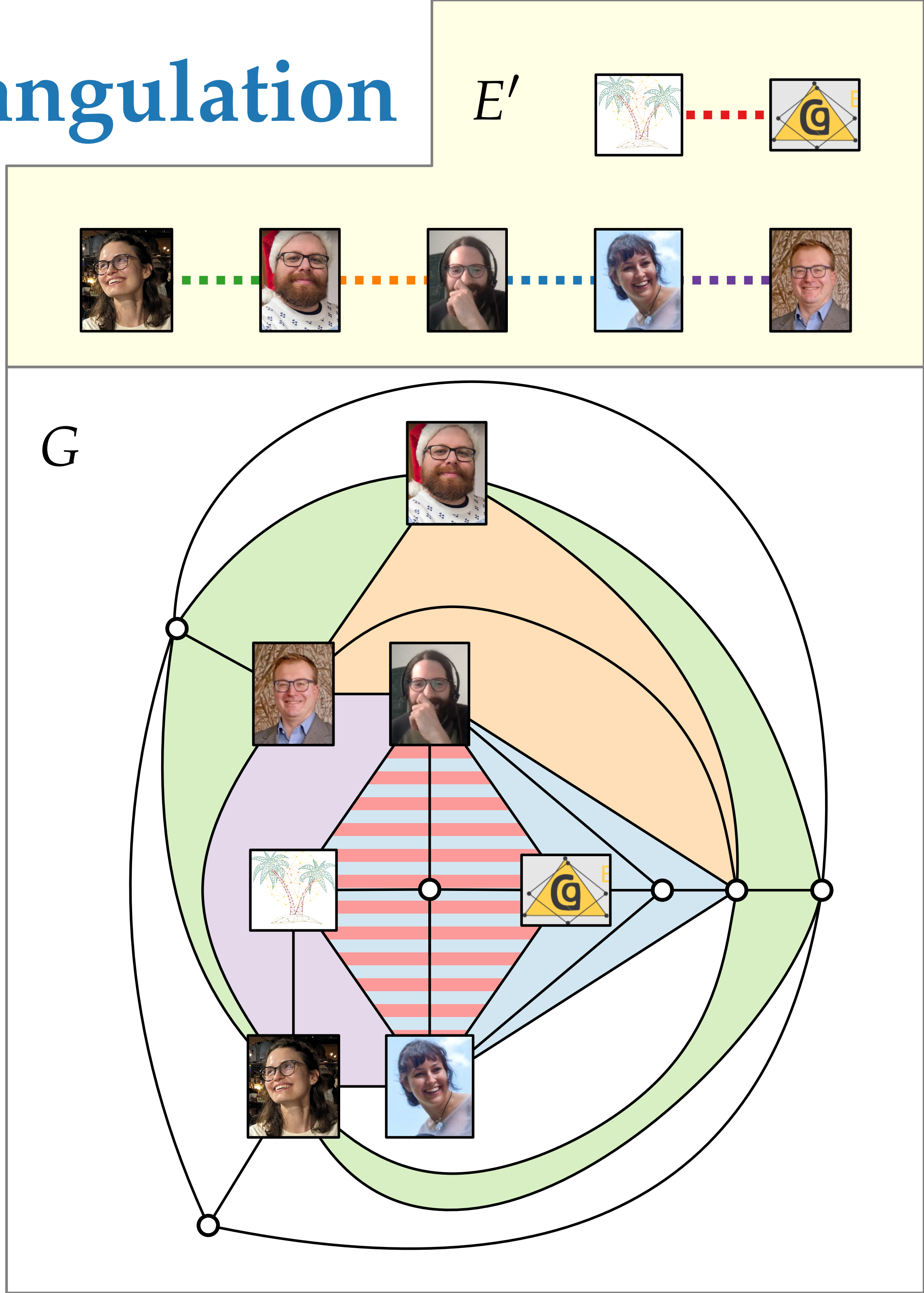
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it



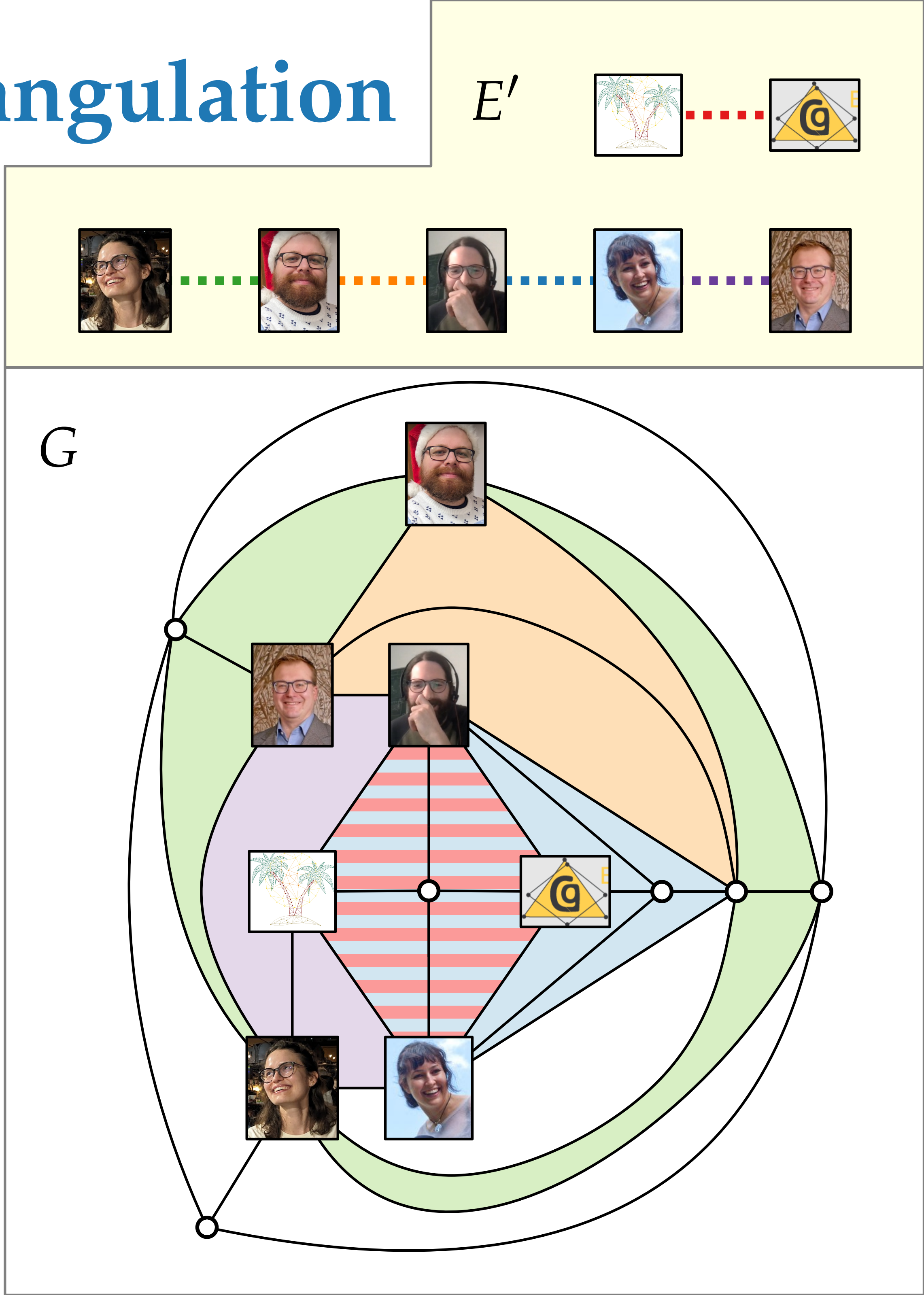
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it



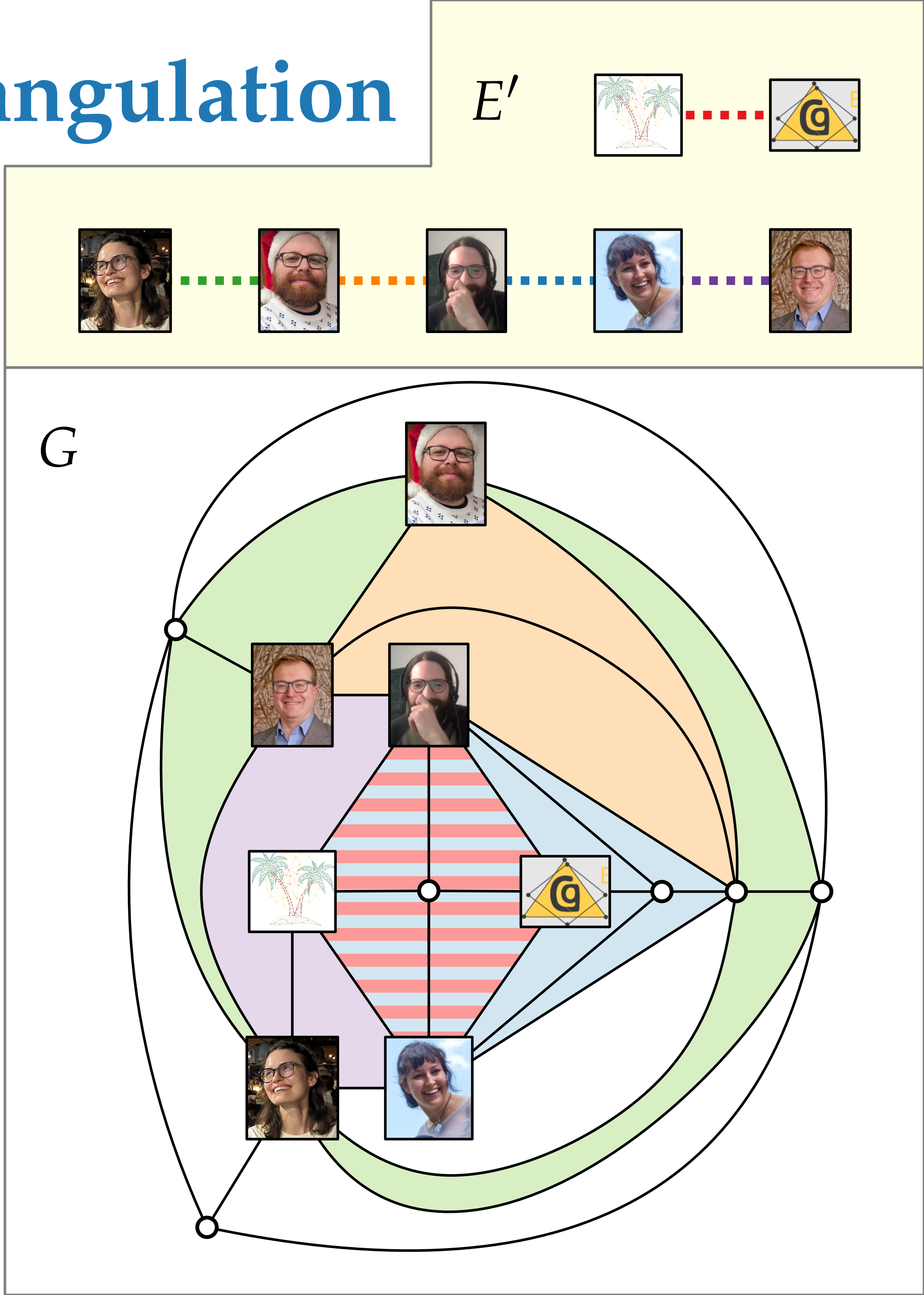
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance



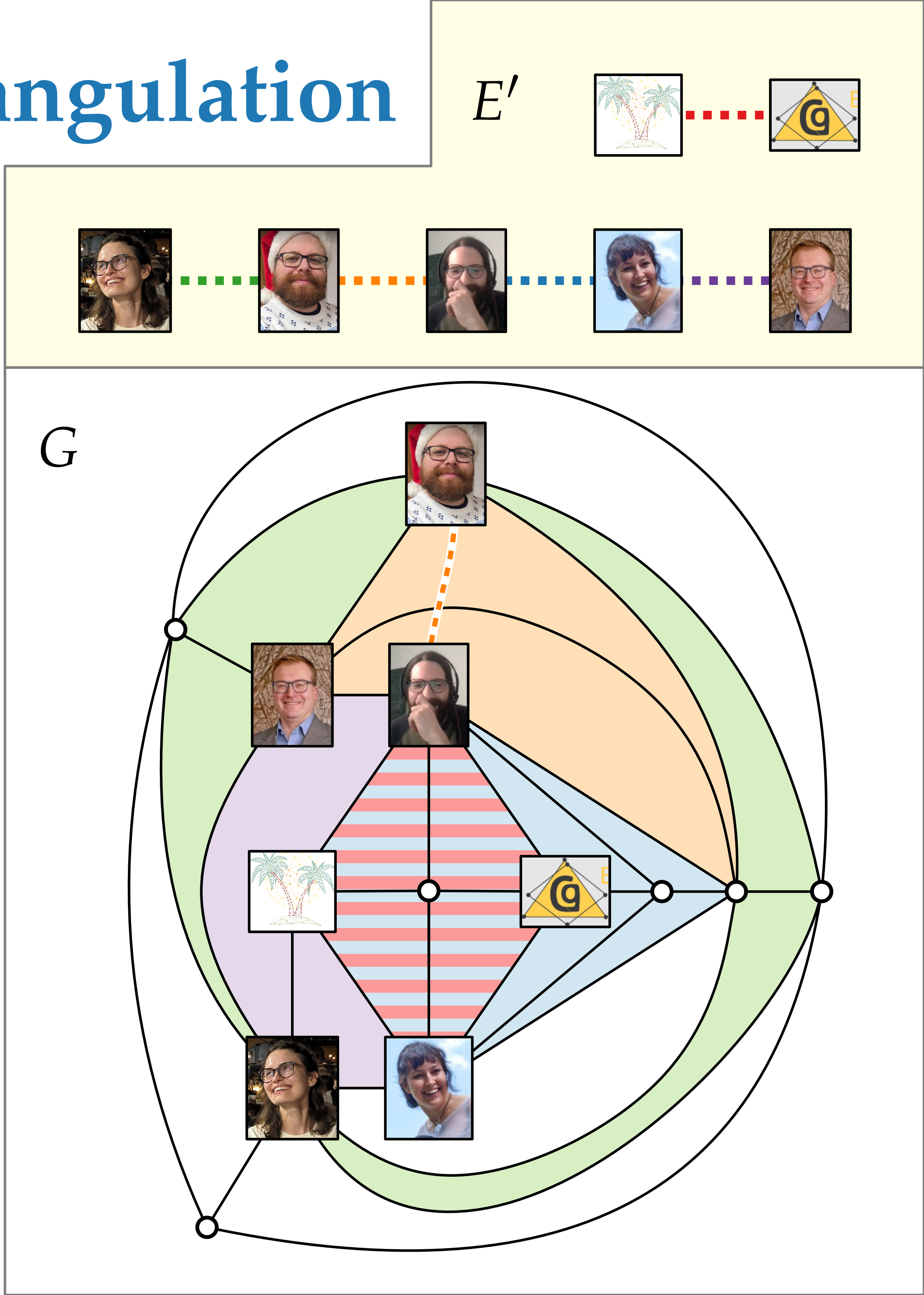
1-Plane Insertion Into a Plane Triangulation

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it



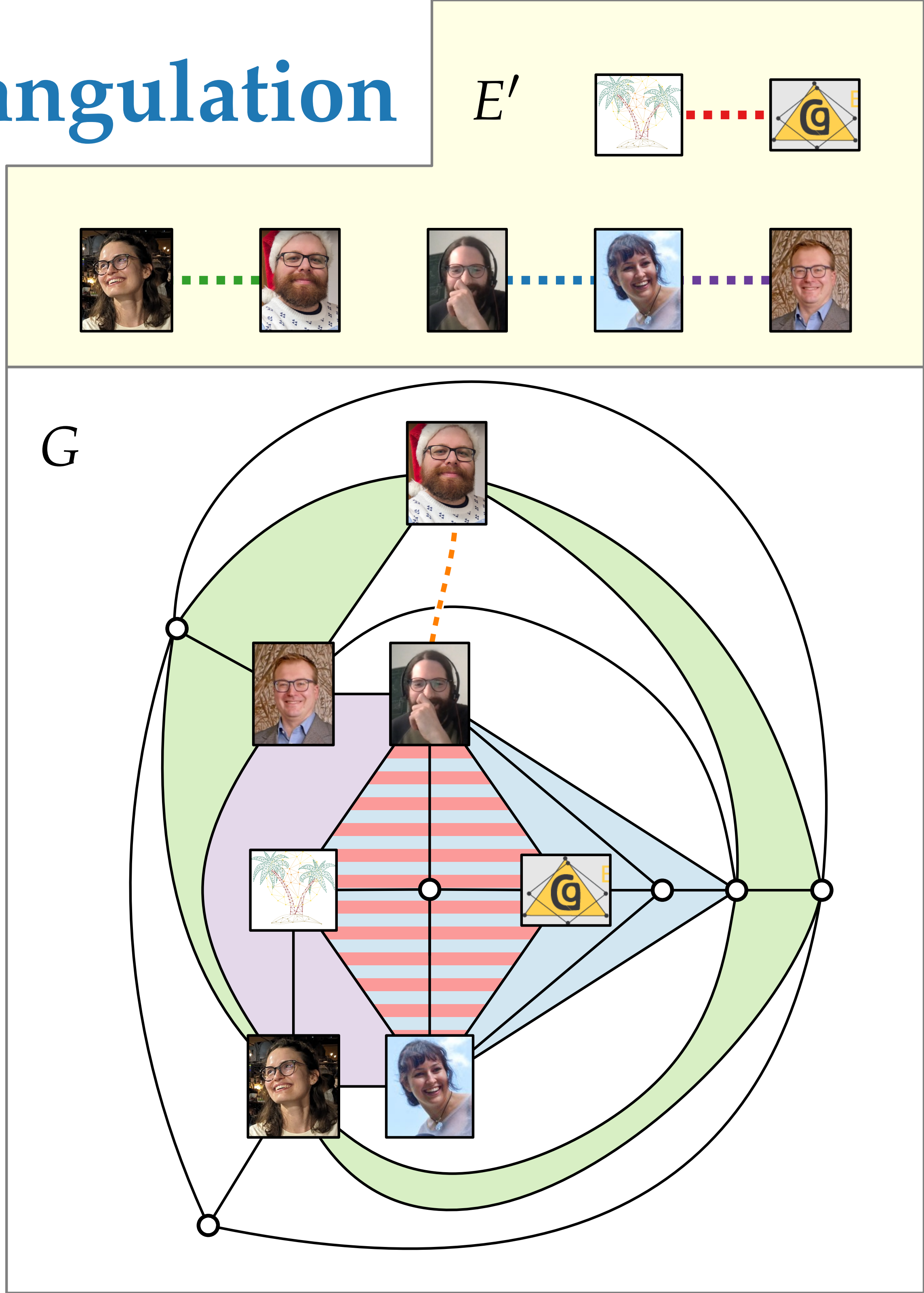
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it



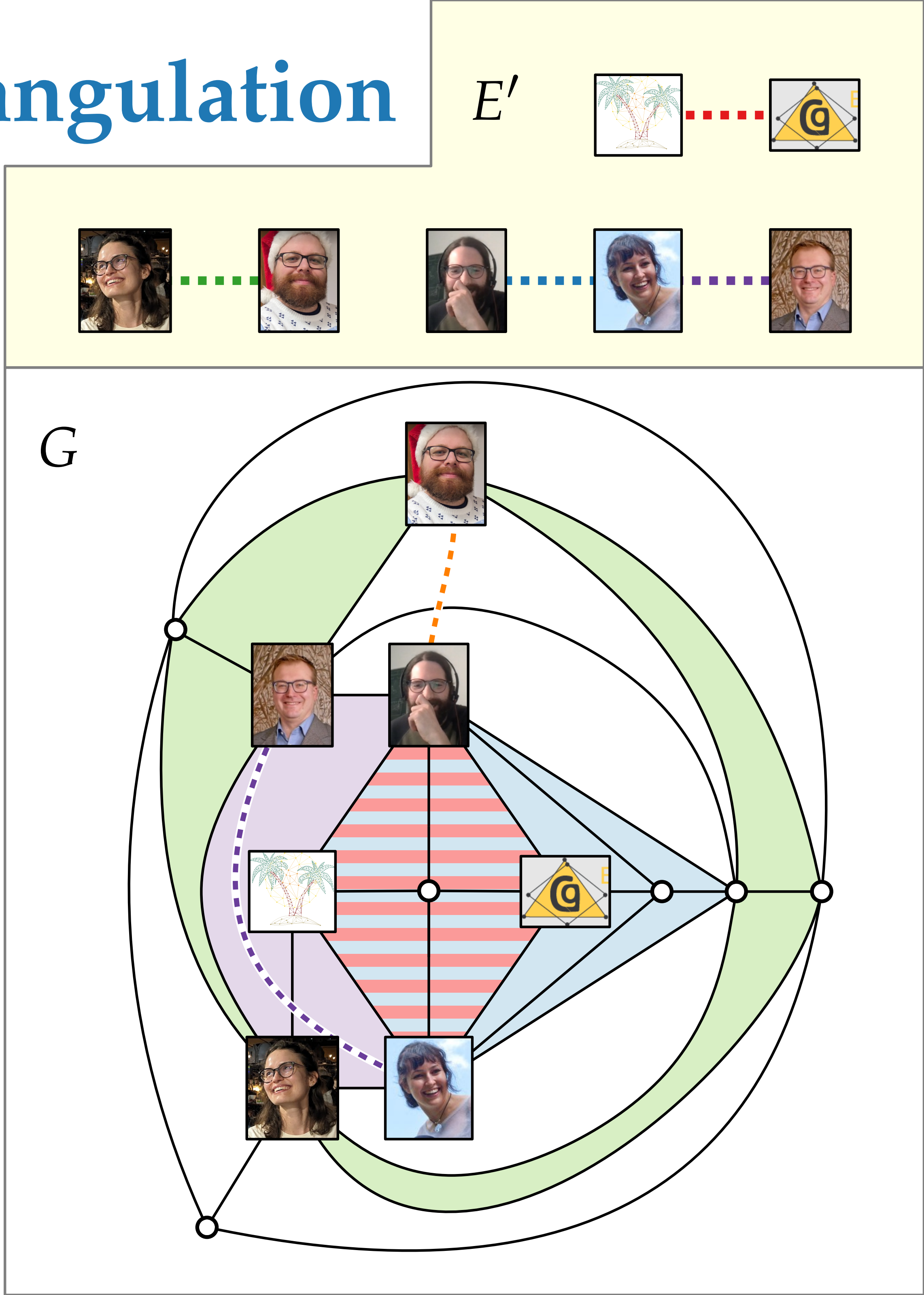
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it



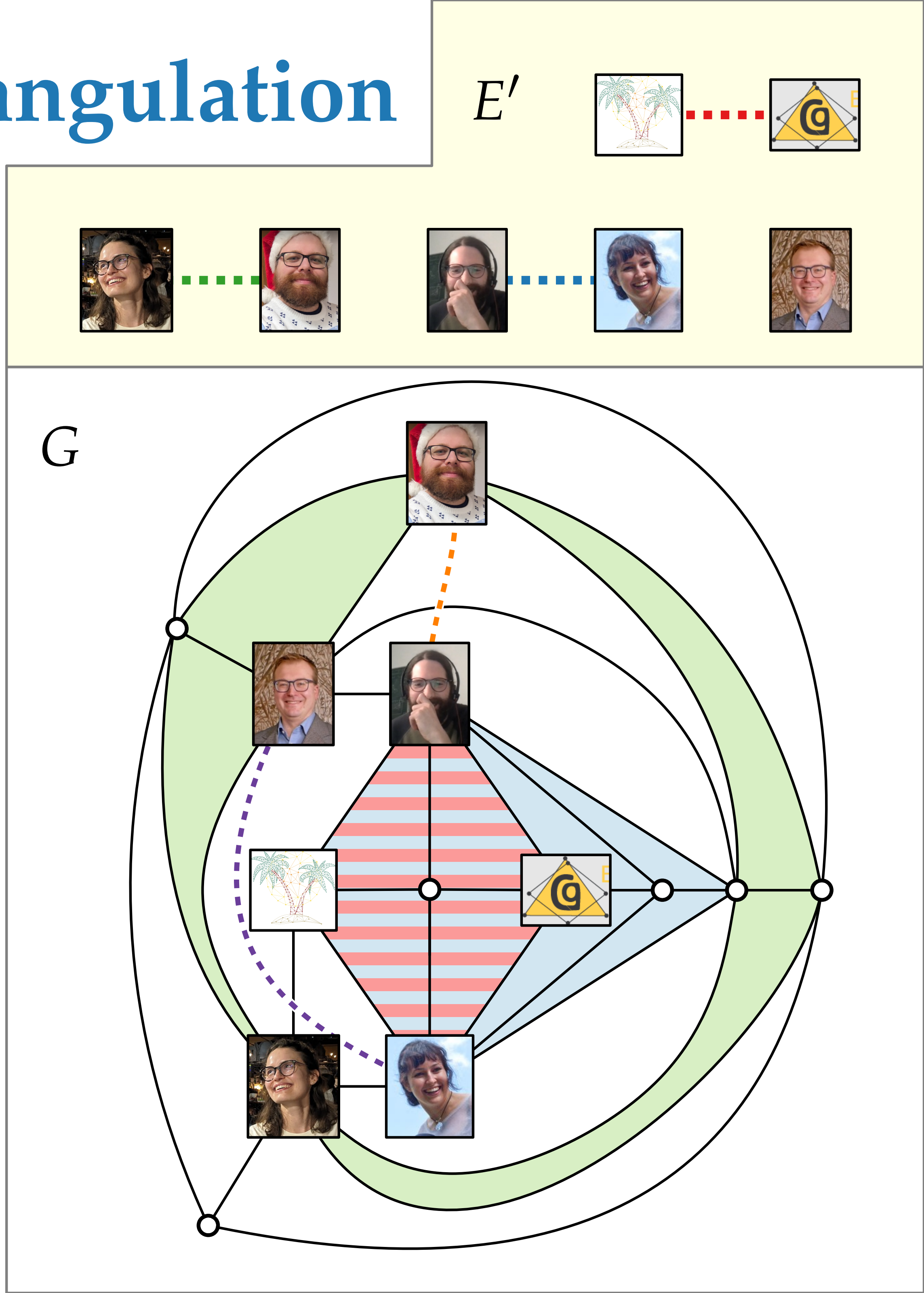
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it



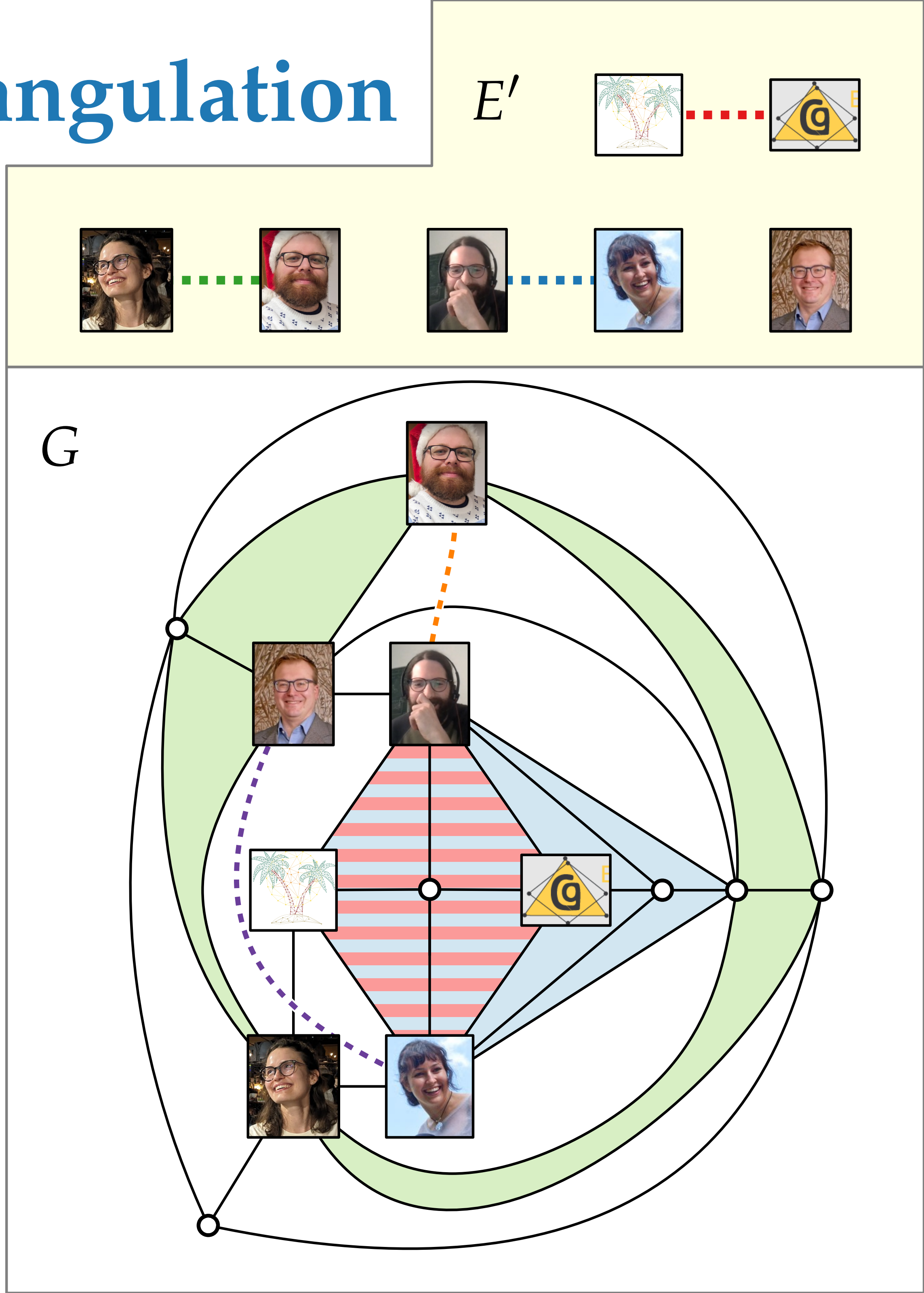
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it



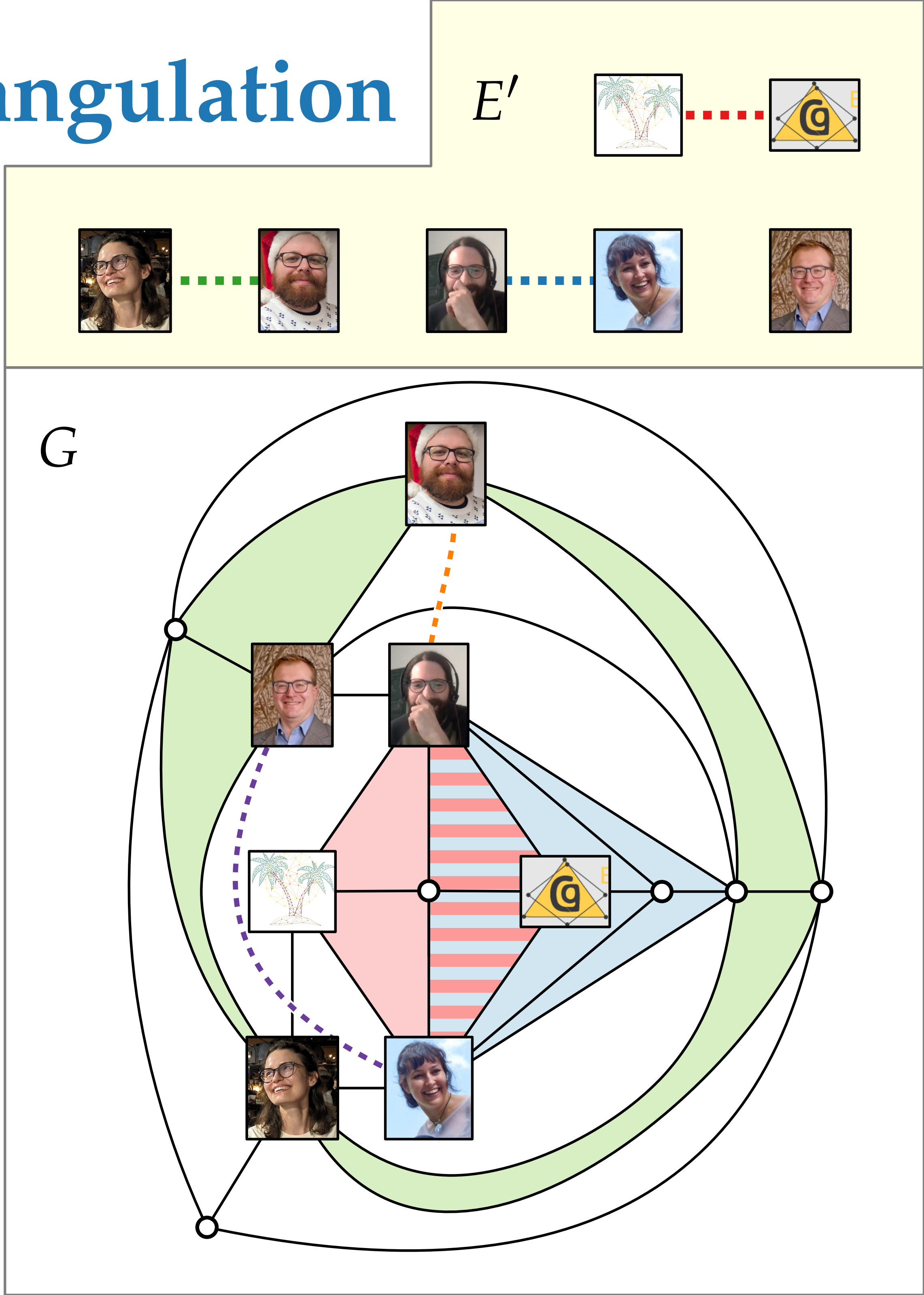
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with $\geq$ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option



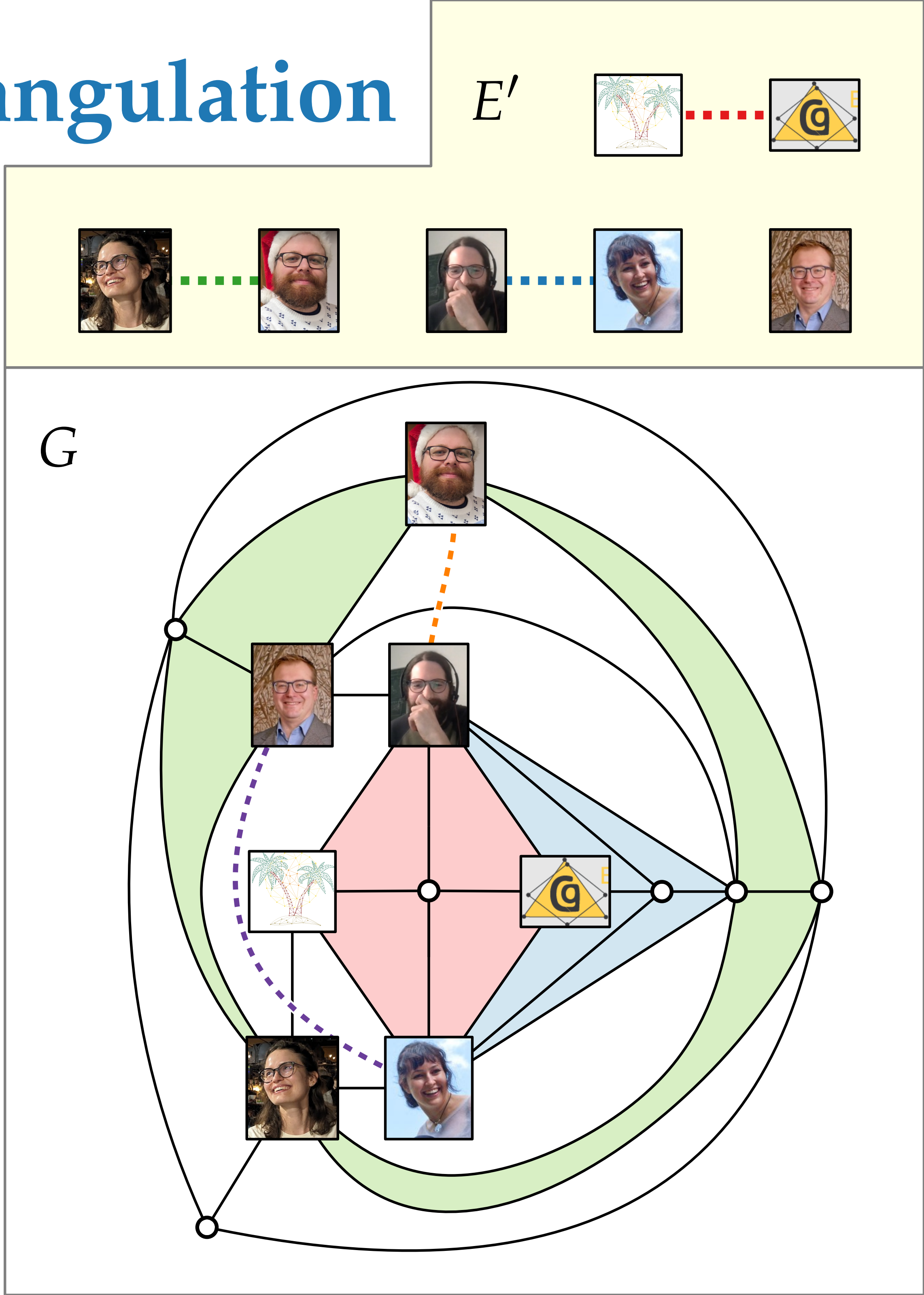
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with $\geq$ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option



1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with $\geq$ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option



1-Plane Insertion Into a Plane **Triangulation**

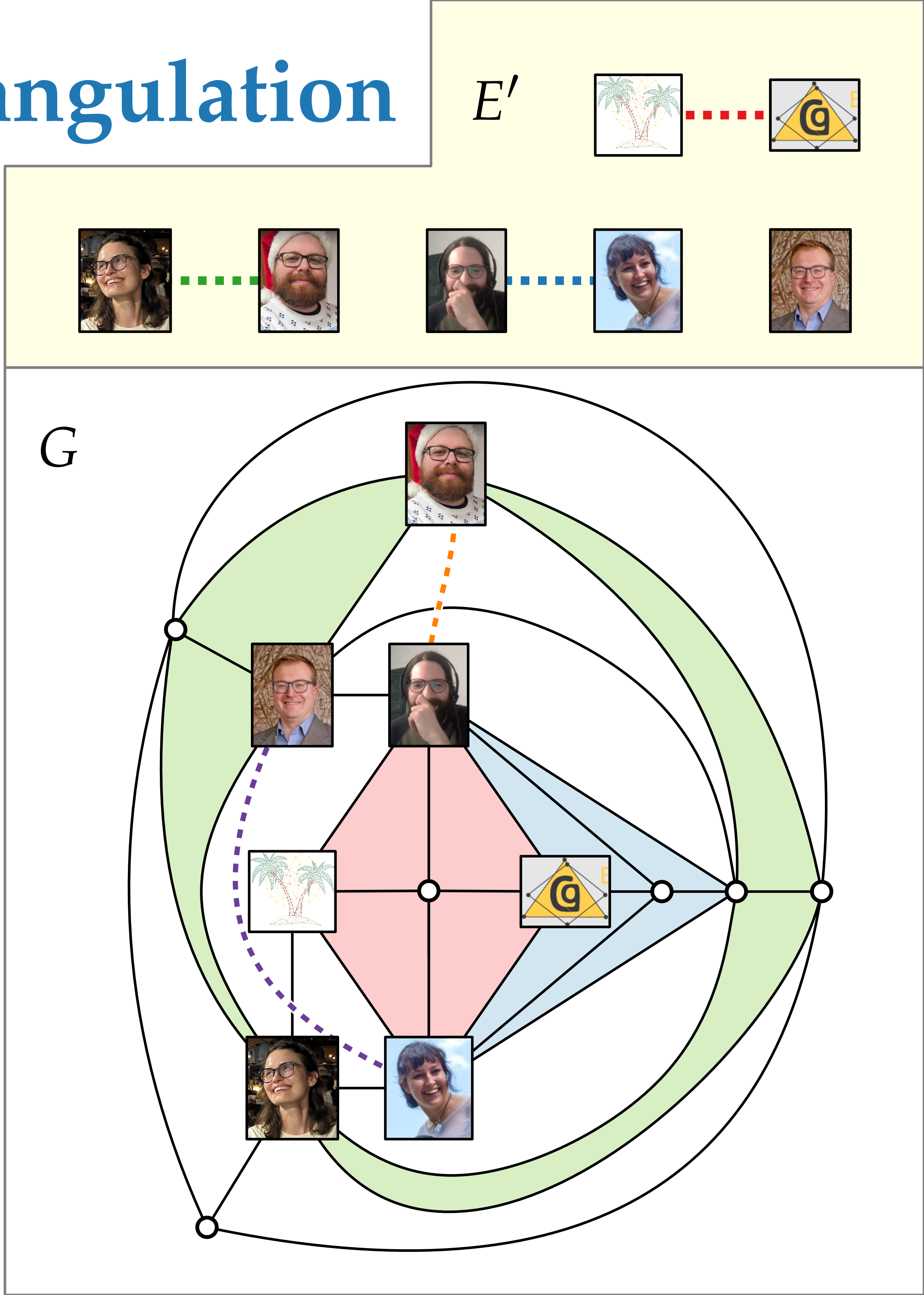
1. For each edge, find all possibilities to route it

2. Edge with 0 options ➡ no-instance

3. Edge with 1 option ➡ pick it

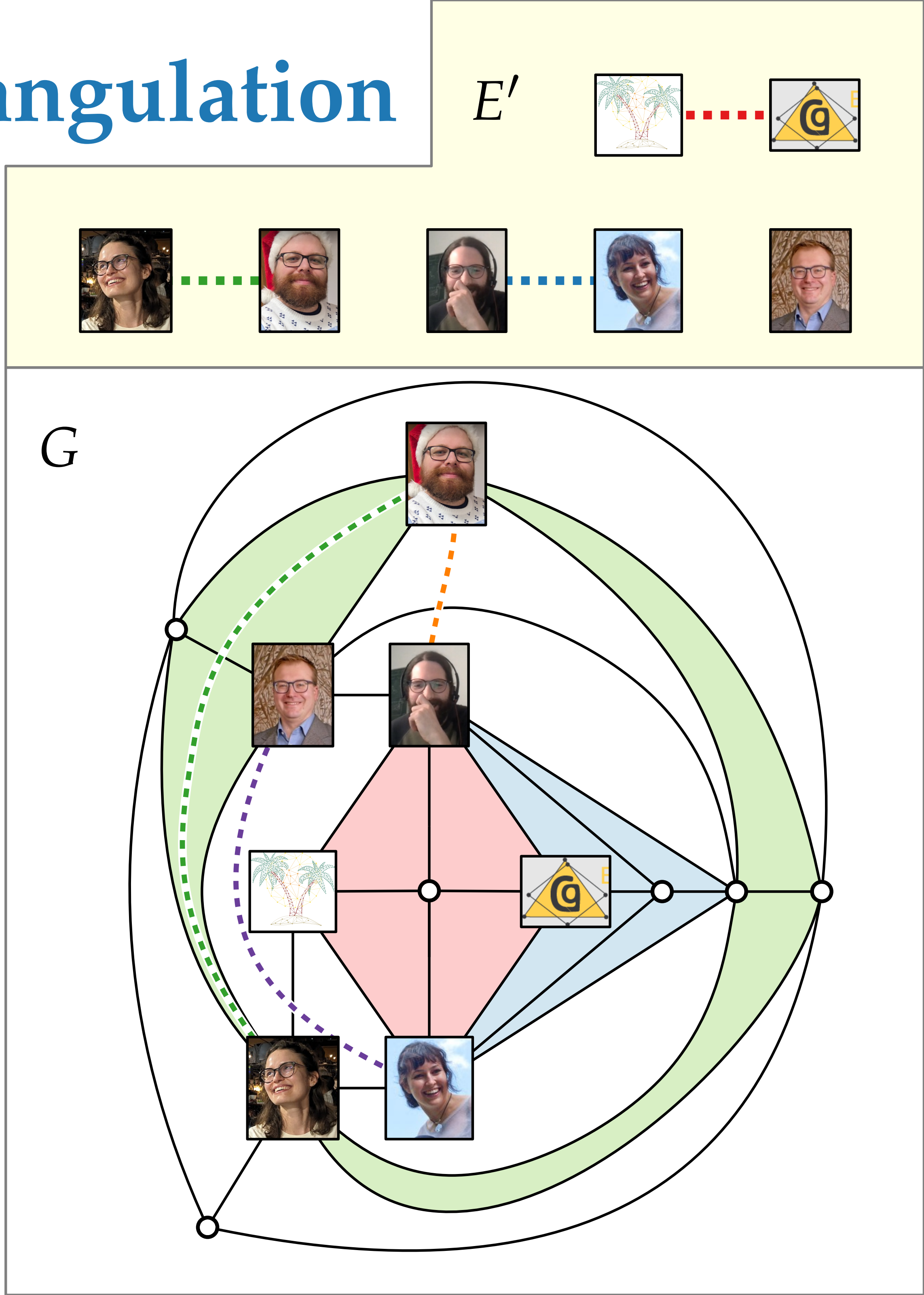
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option

5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



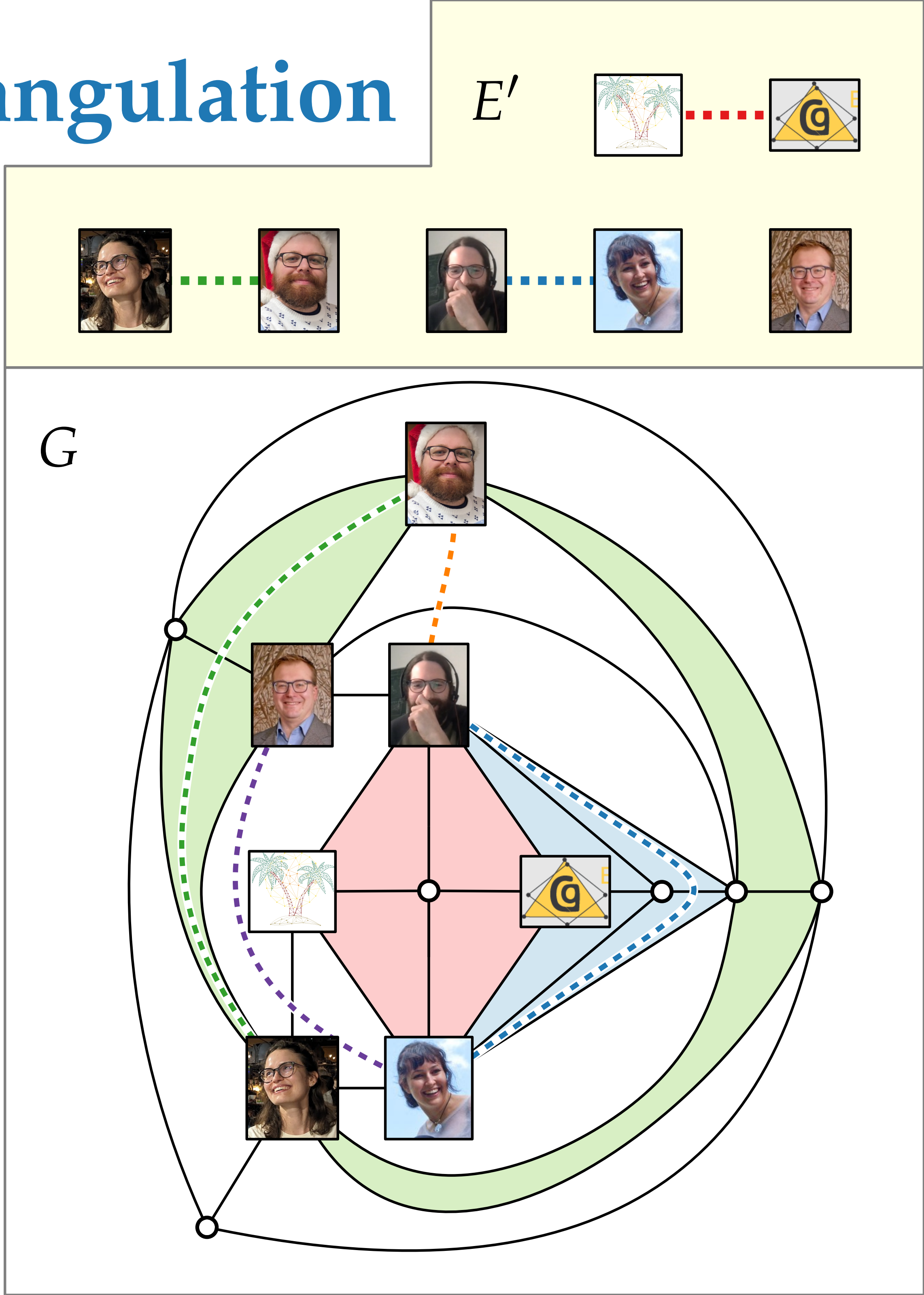
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with $\geq$ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



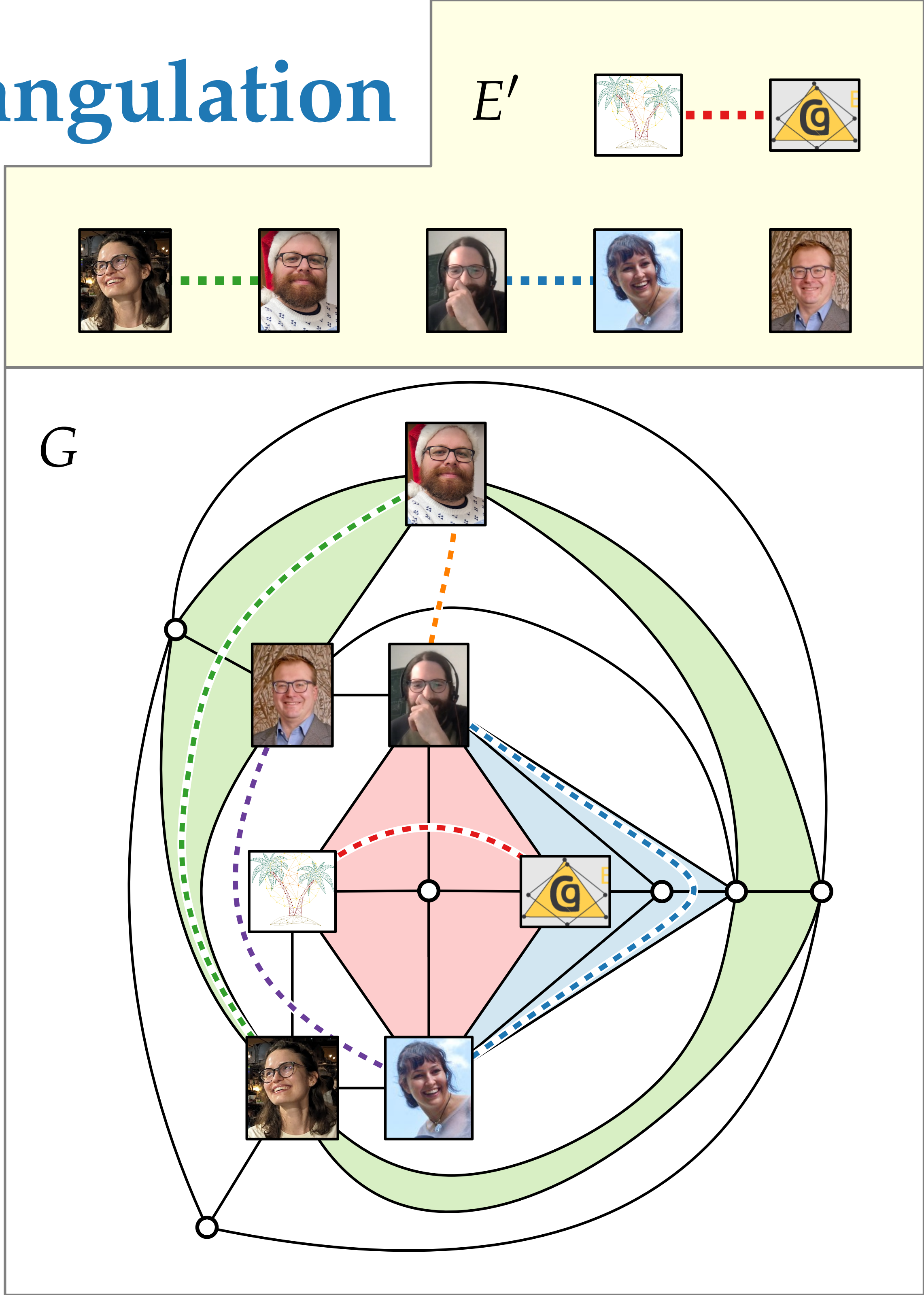
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with $\geq$ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph

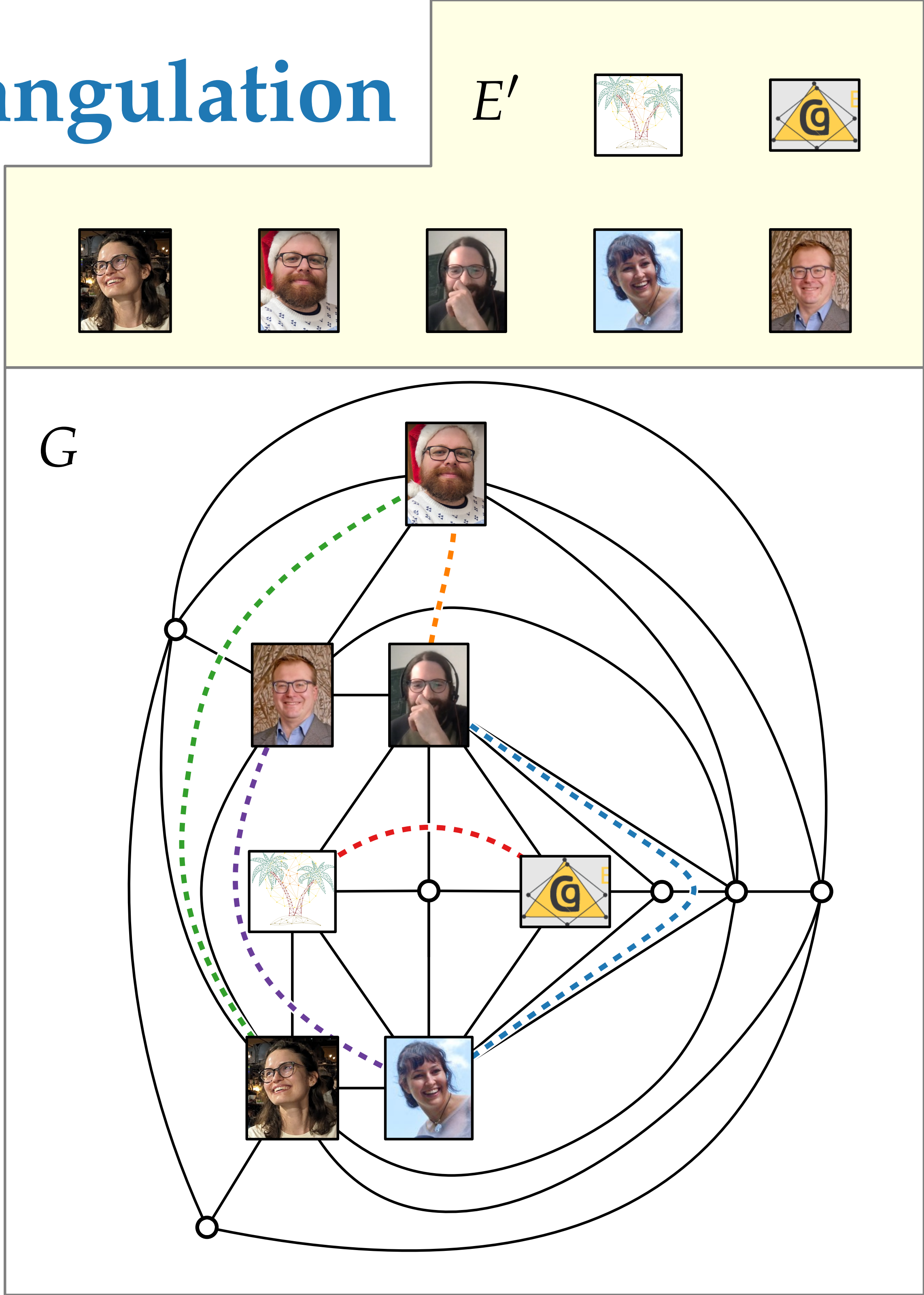


1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with $\geq$ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



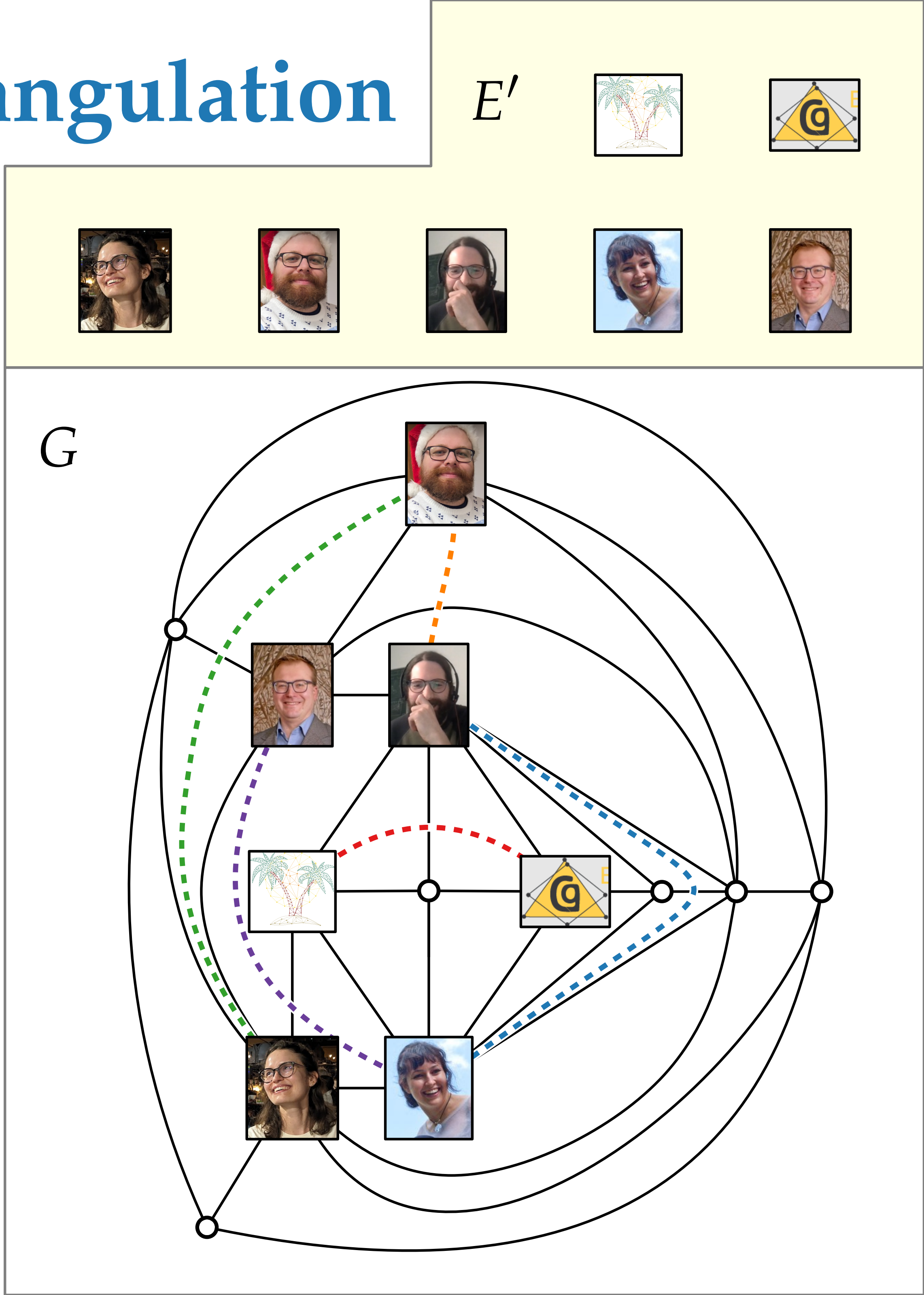
1-Plane Insertion Into a Plane **Triangulation**



1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with $\geq$ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph

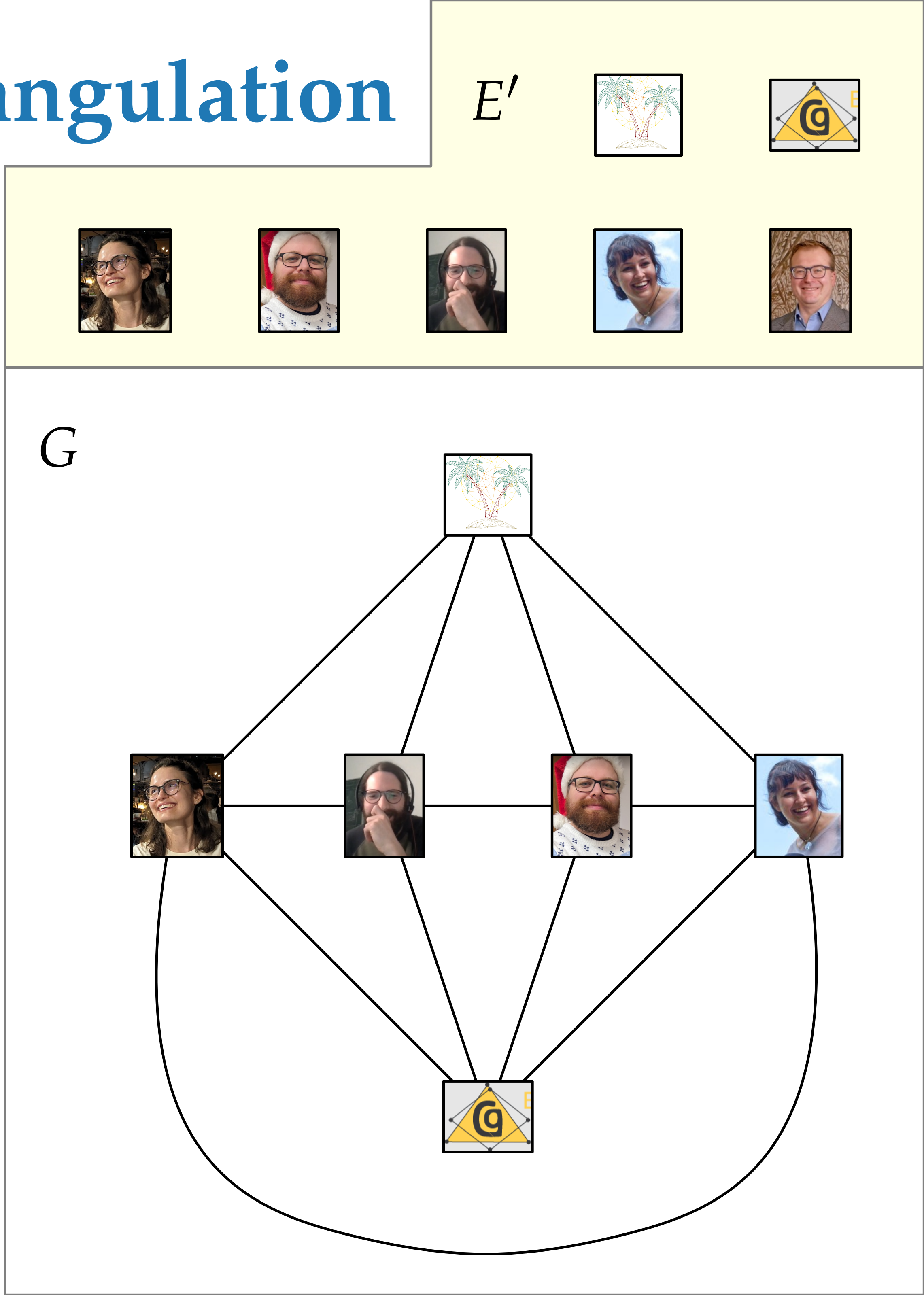
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with $\geq$ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



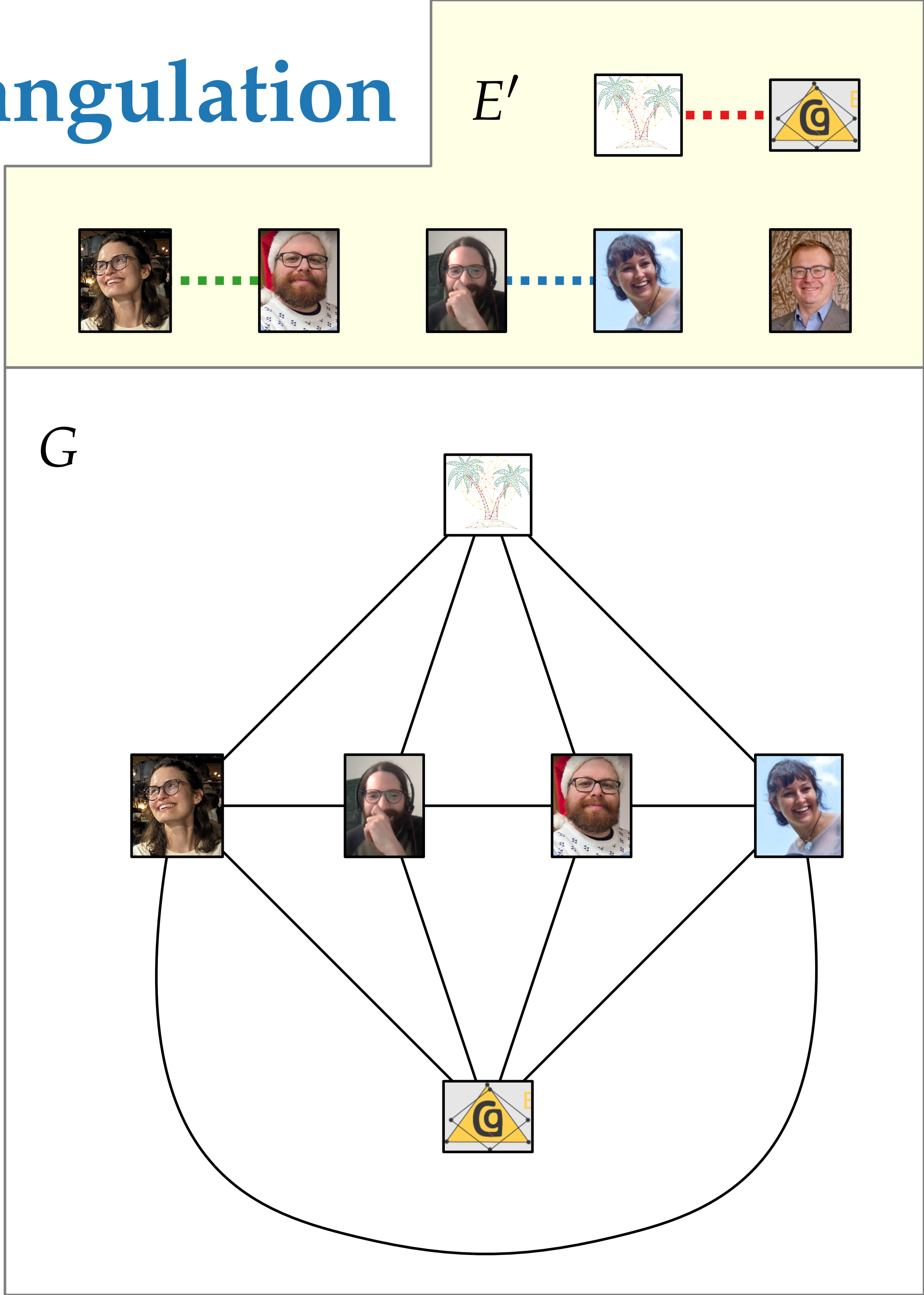
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



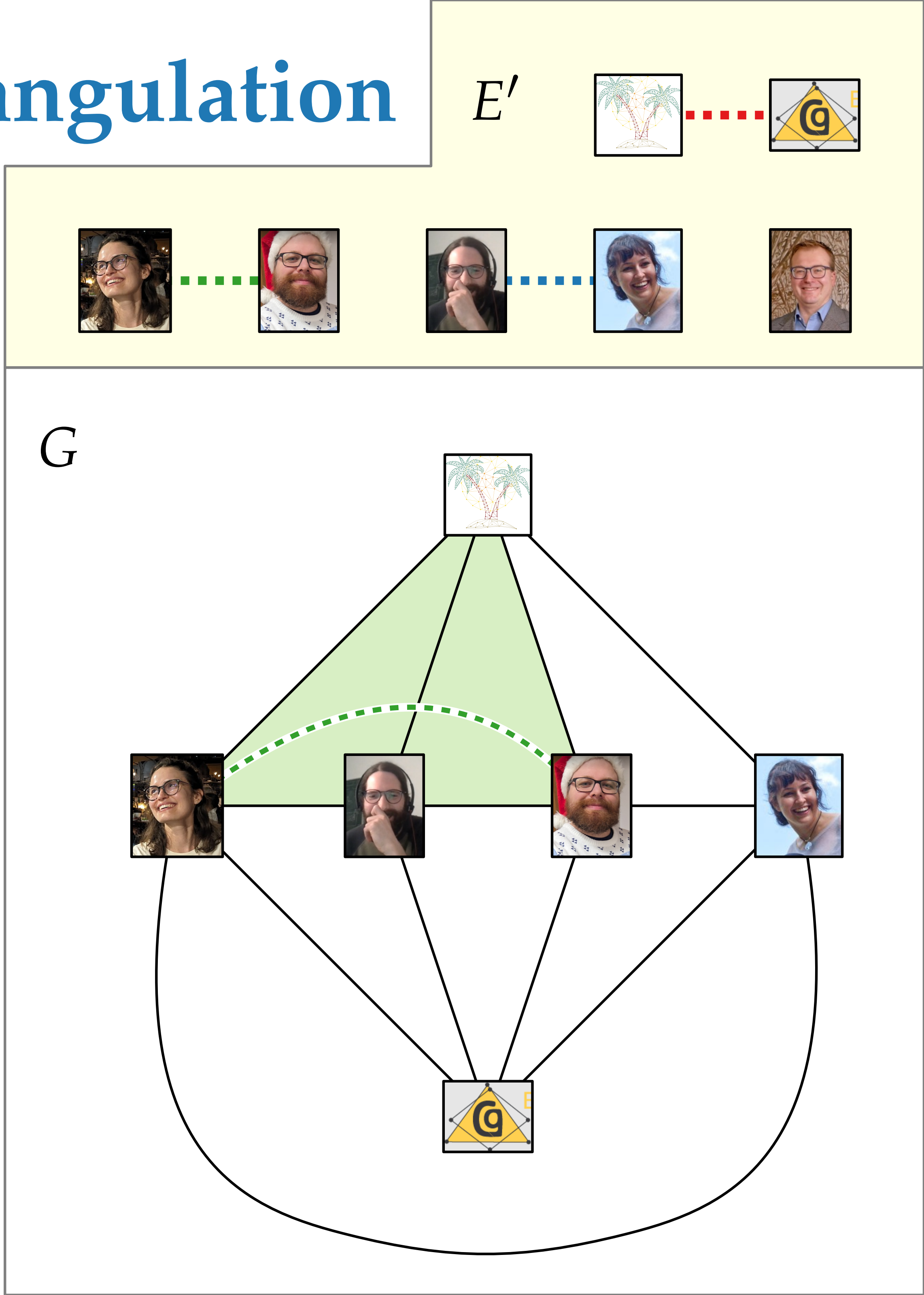
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with $\geq$ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



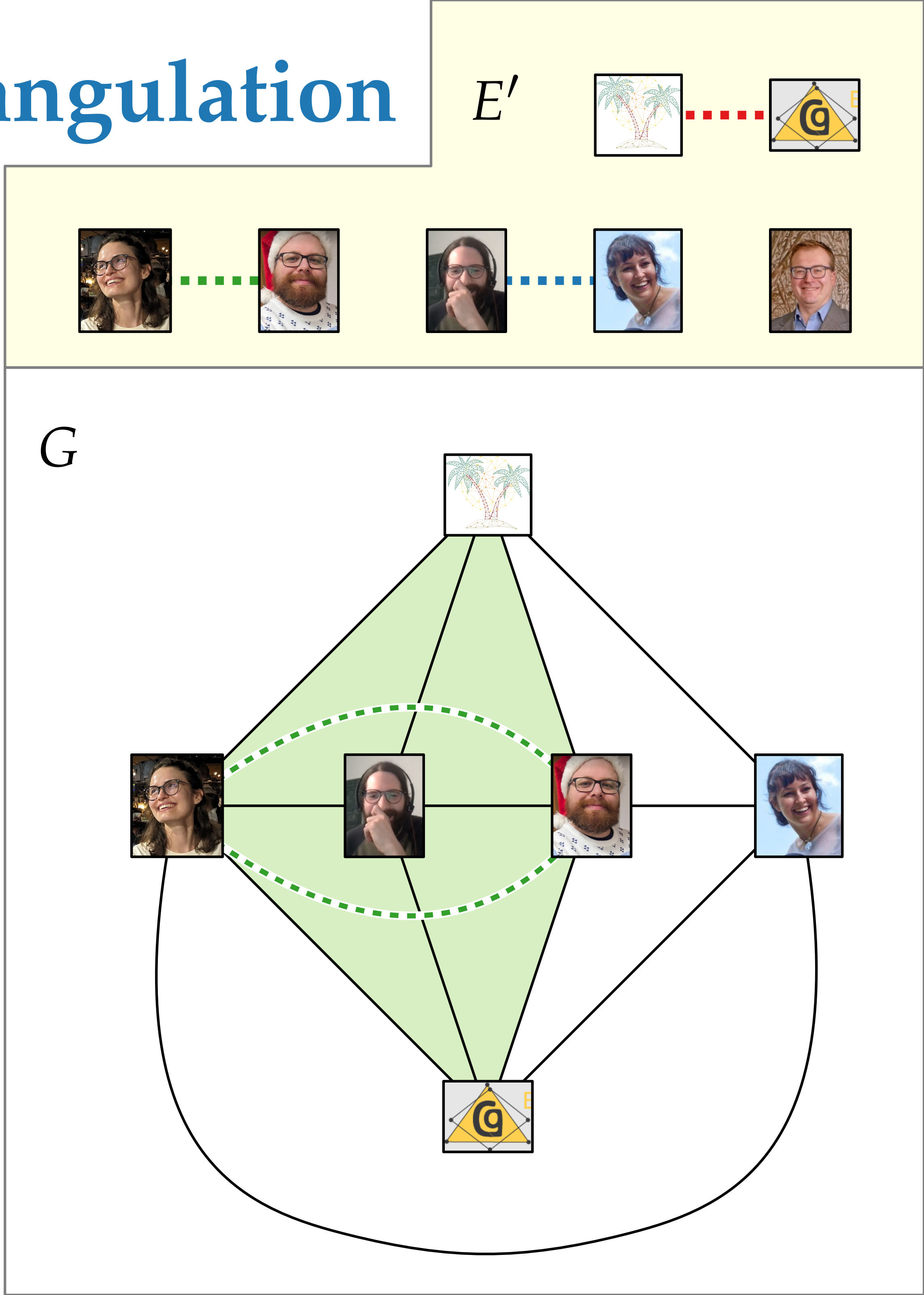
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



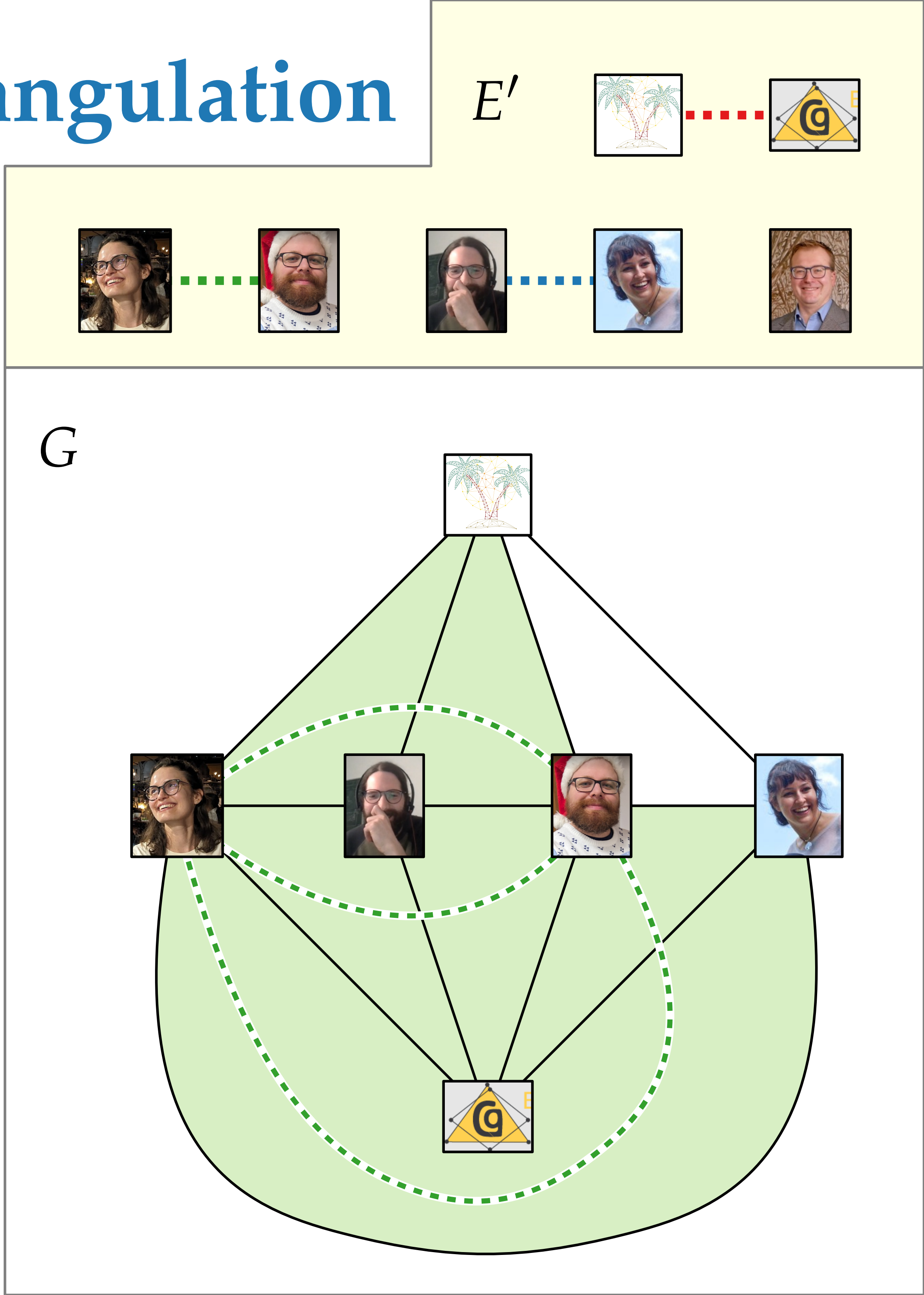
1-Plane Insertion Into a Plane Triangulation

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



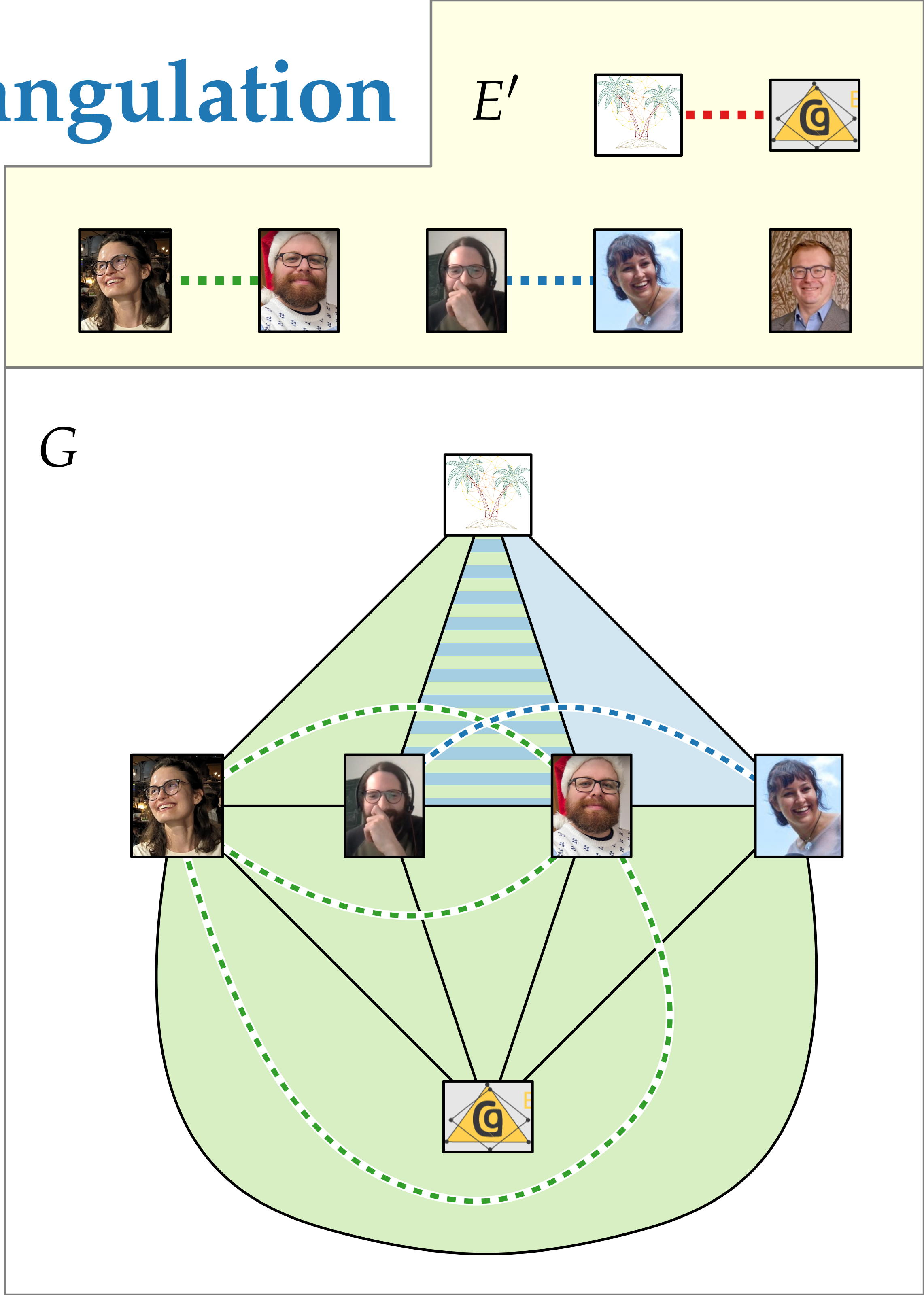
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



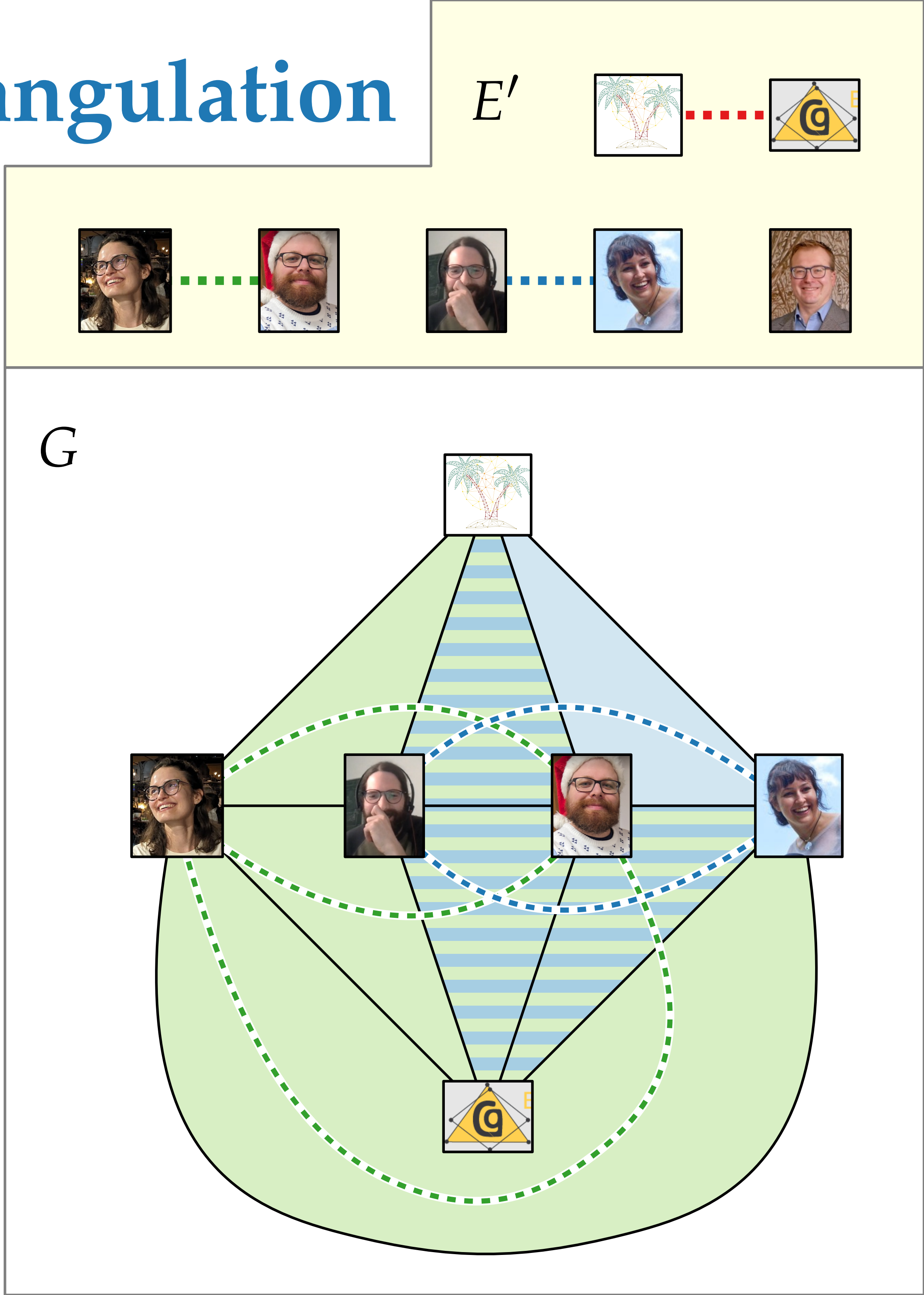
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



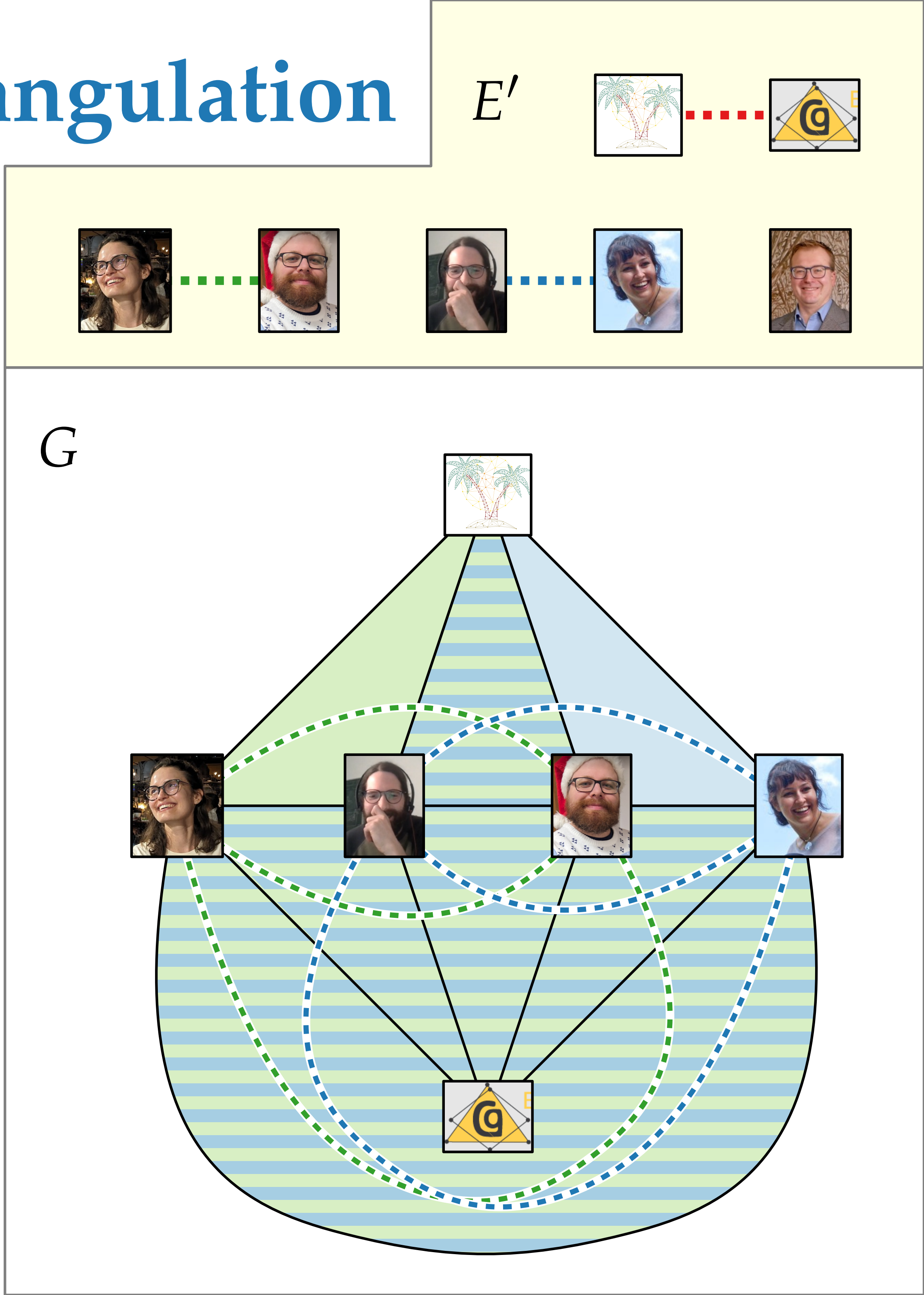
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



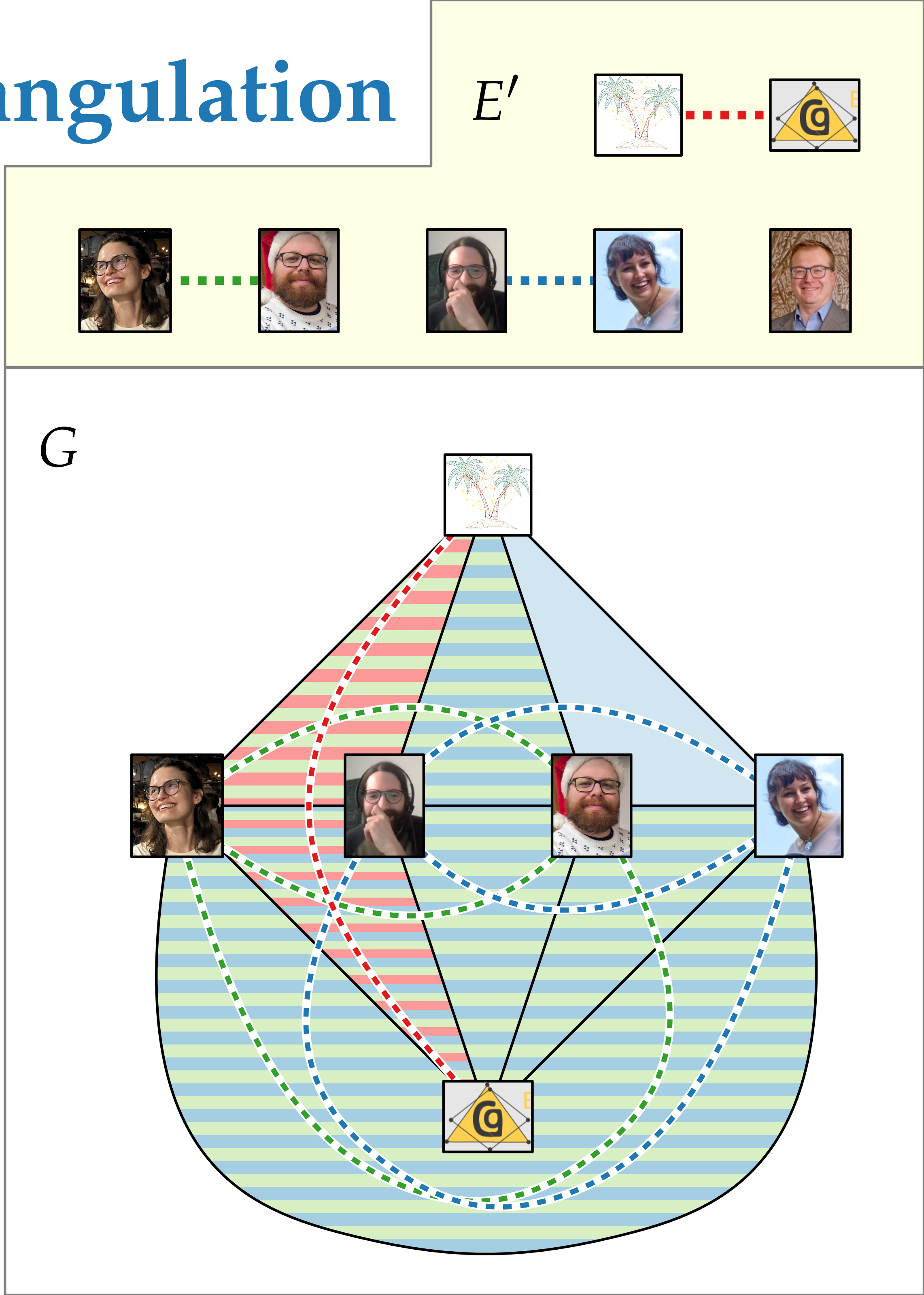
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



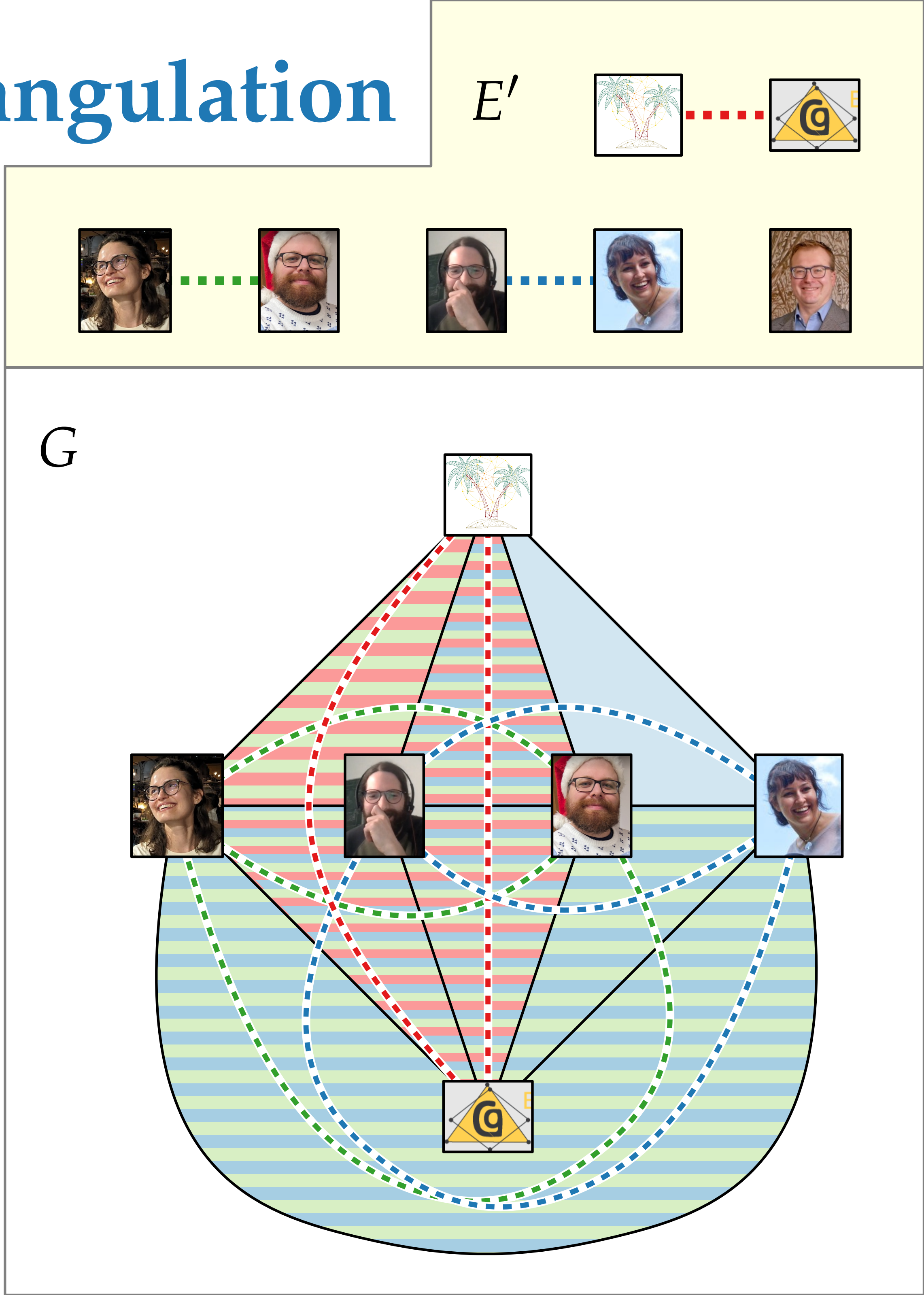
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



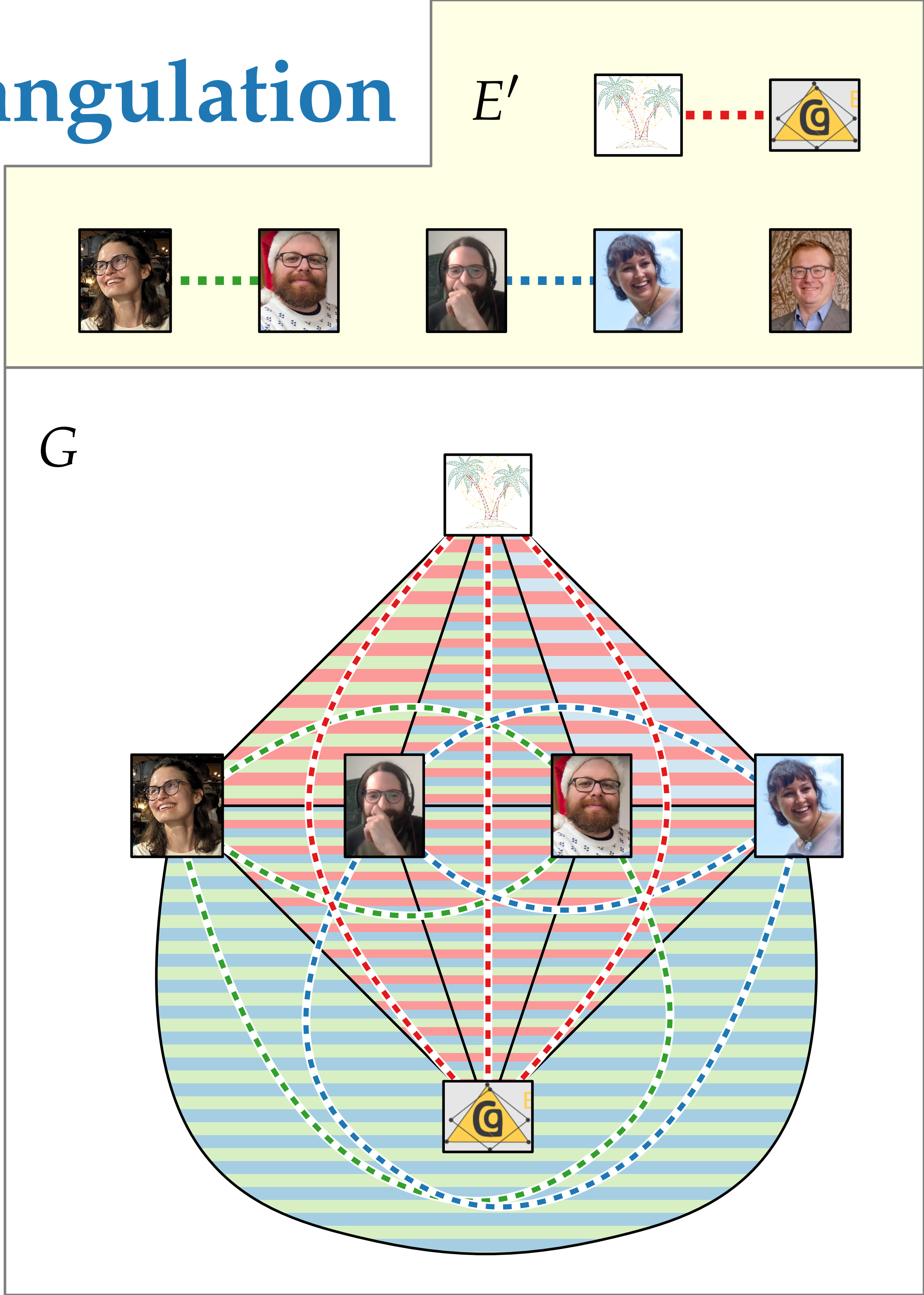
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



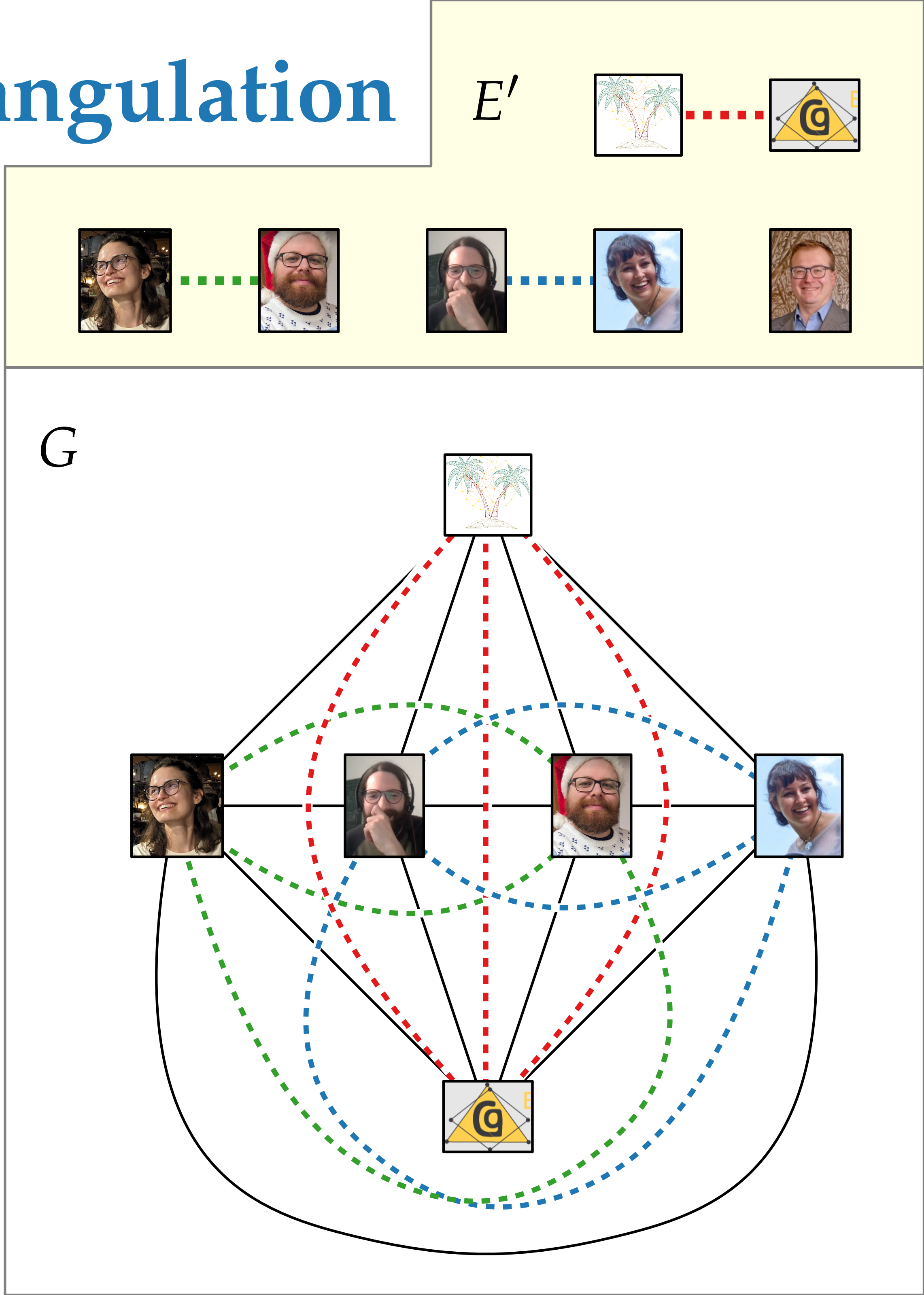
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



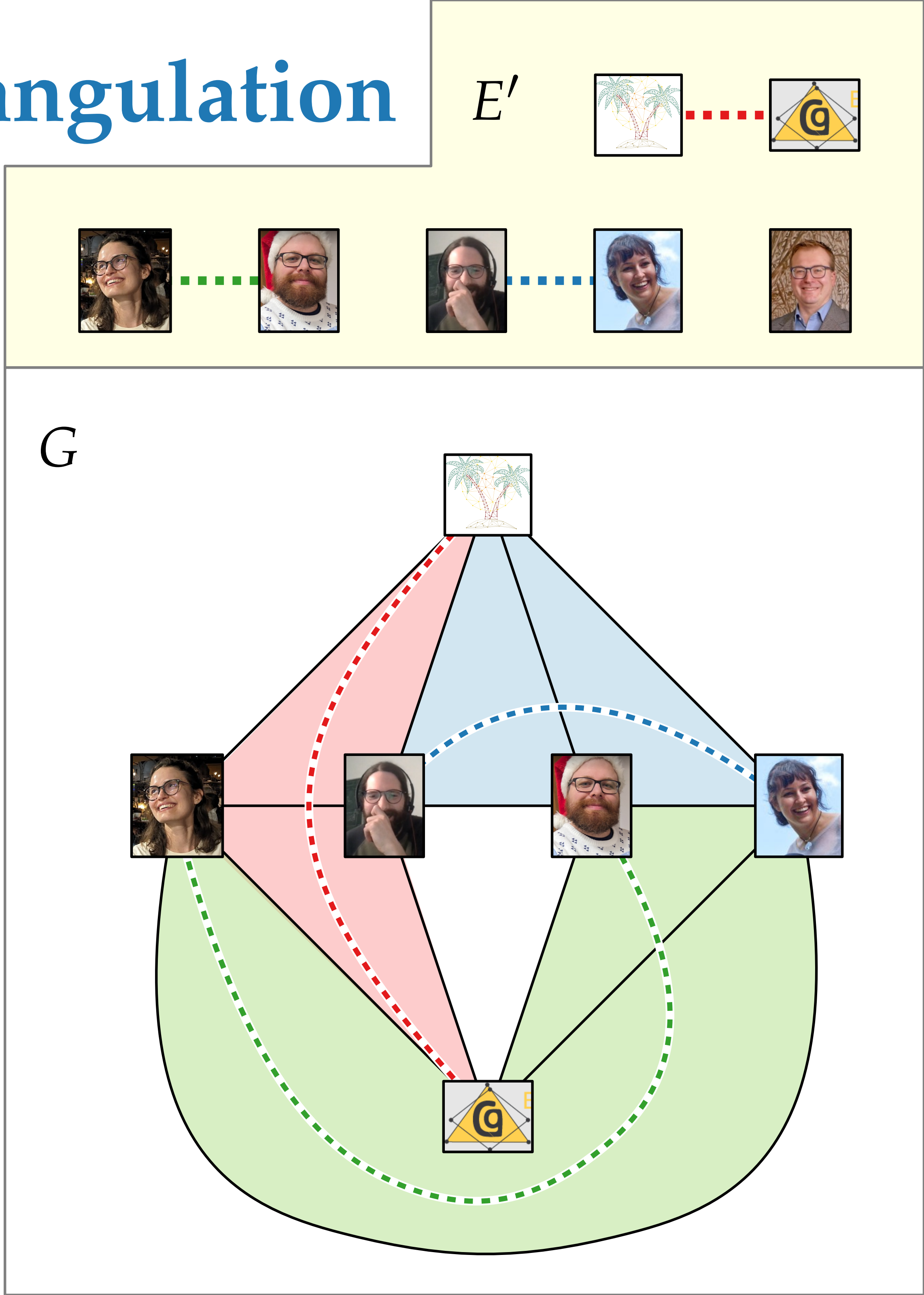
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with $\geq$ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



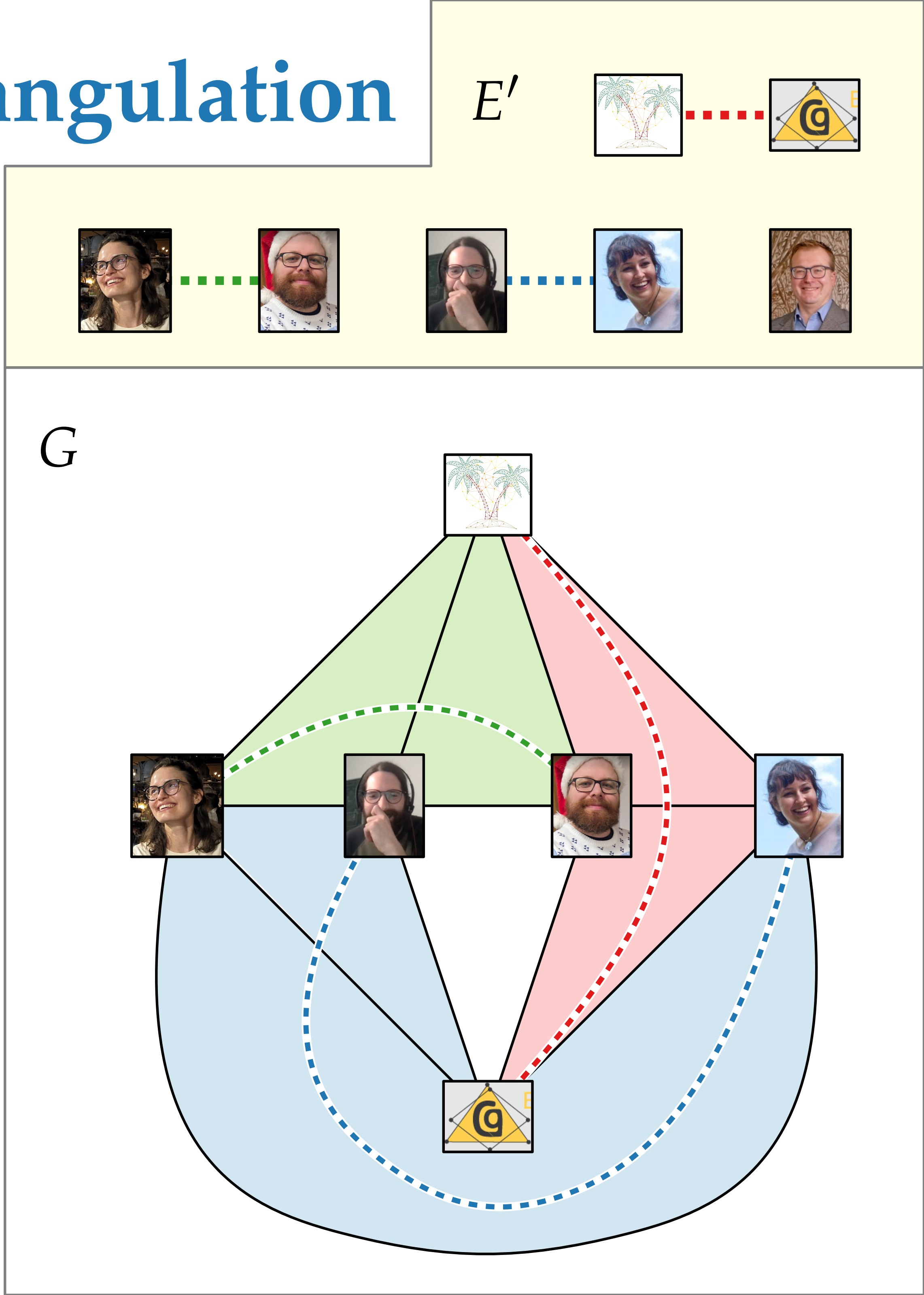
1-Plane Insertion Into a Plane **Triangulation**

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



1-Plane Insertion Into a Plane **Triangulation**

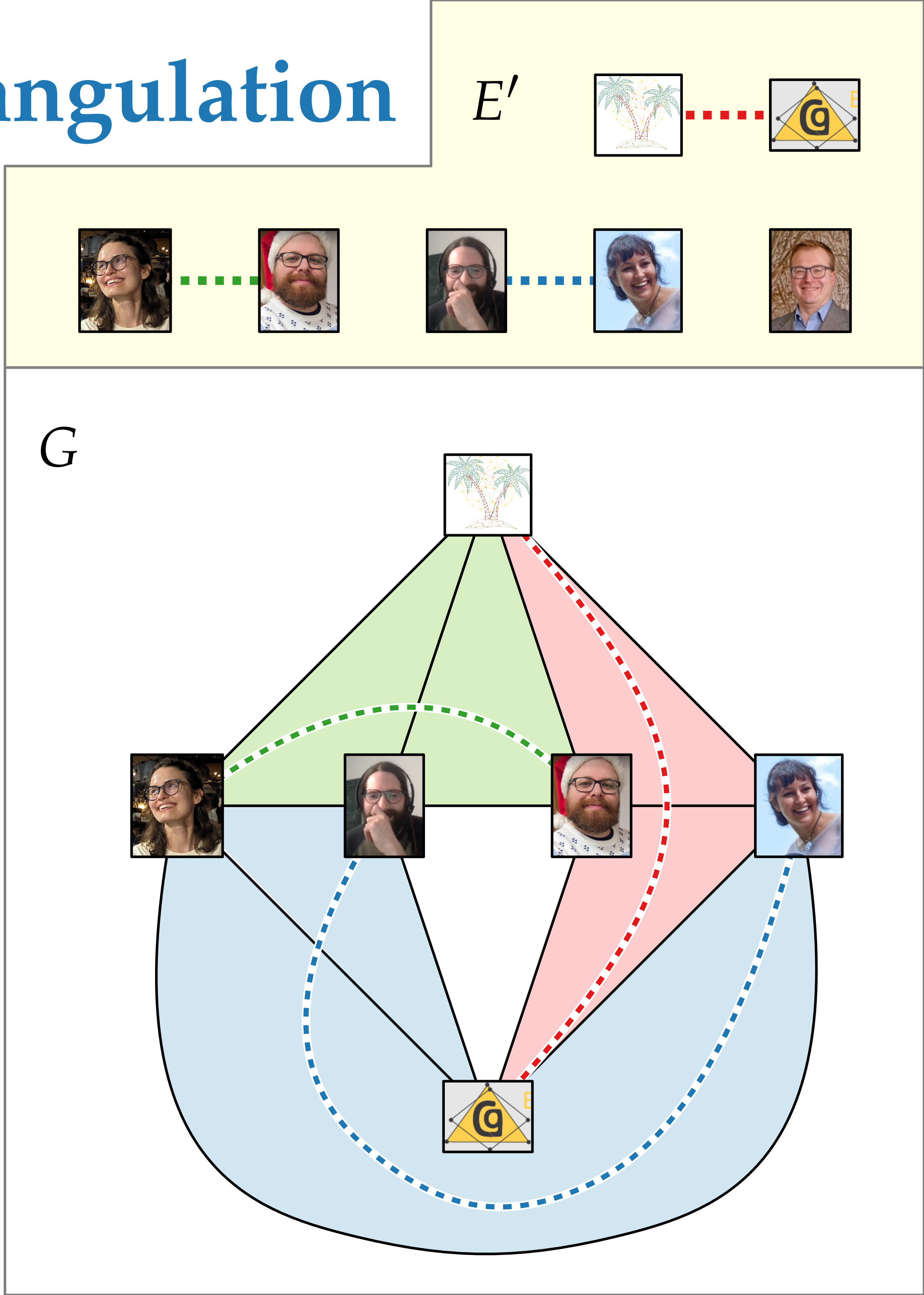
1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph



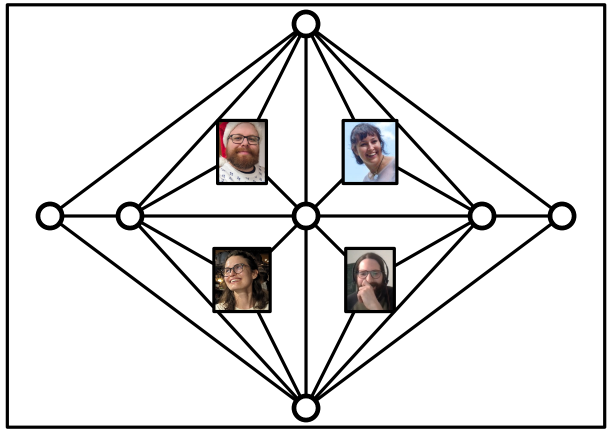
1-Plane Insertion Into a Plane Triangulation

1. For each edge, find all possibilities to route it
2. Edge with 0 options ➡ no-instance
3. Edge with 1 option ➡ pick it
4. Edge with ≥ 3 options
   ➡ there is always a **safe** or an **impossible** option
   ➡ either pick or remove that option
5. All edges have 2 options
   ➡ solve 2-SAT on conflict graph

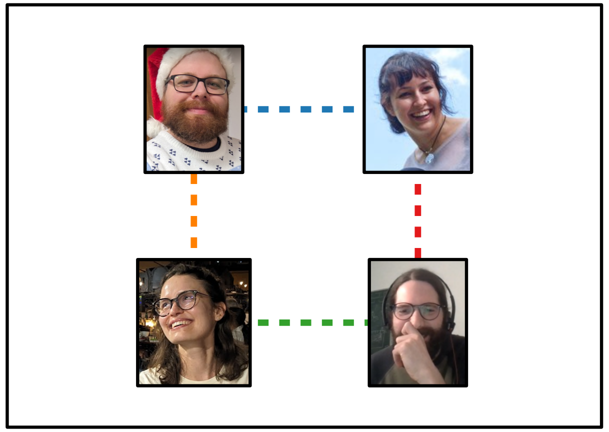
Theorem.
1-plane insertion into a plane triangulation can be solved in $O(n)$ time.



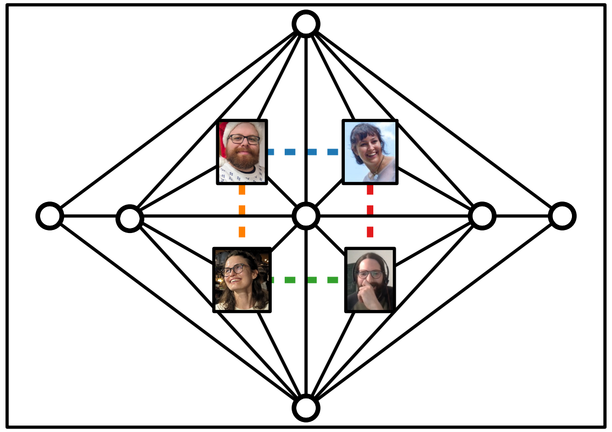
Conclusion



plane graph G

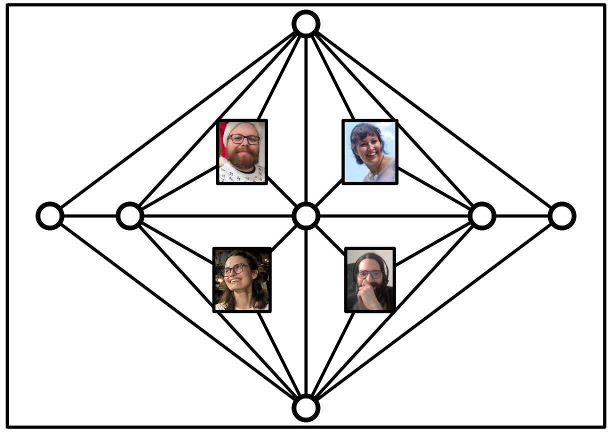


edges E' btw. vtcs in G

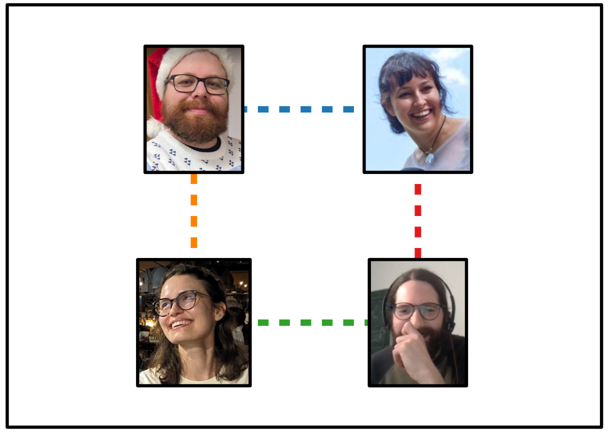


1-planar drawing of $G + E'$
that keeps the embedding of G

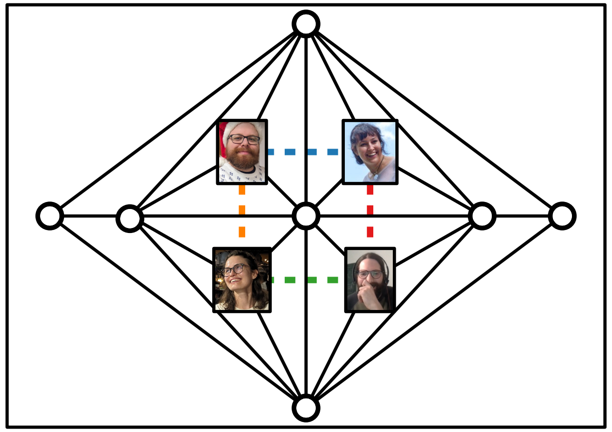
Conclusion



plane graph G



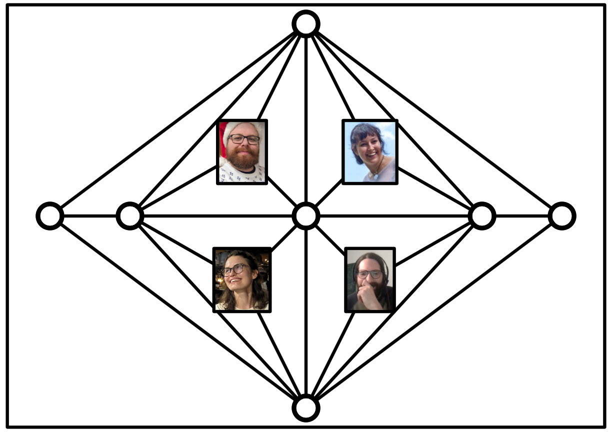
edges E' btw. vtcs in G



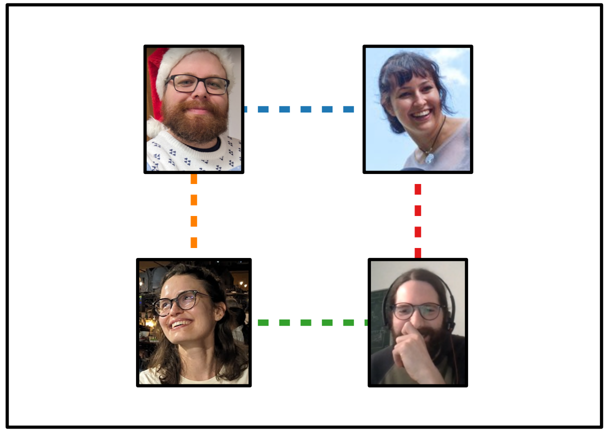
1-planar drawing of $G + E'$ that keeps the embedding of G

- G triangulated ➡ $\mathcal{O}(n)$ time

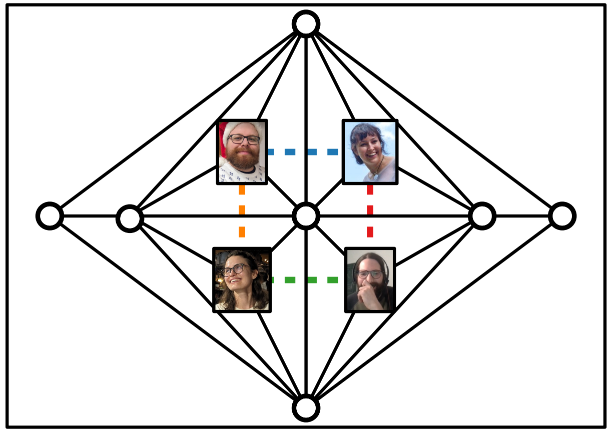
Conclusion



plane graph G



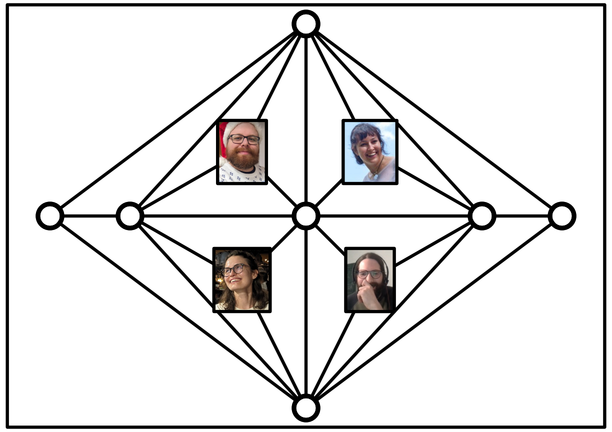
edges E' btw. vtcs in G



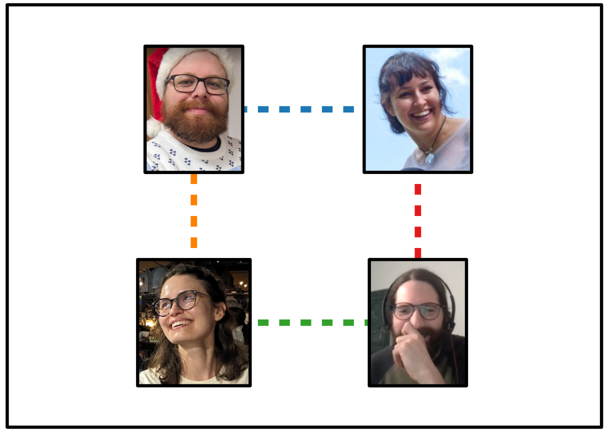
1-planar drawing of $G + E'$ that keeps the embedding of G

- G triangulated ➡ $\mathcal{O}(n)$ time
- G biconnected ➡ NP-complete

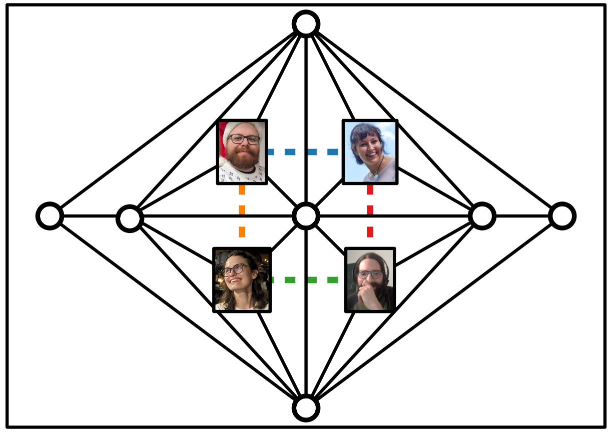
Conclusion



plane graph G



edges E' btw. vtcs in G

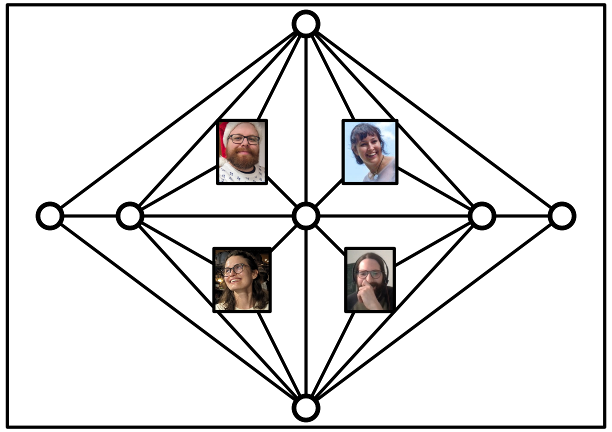


1-planar drawing of $G + E'$ that keeps the embedding of G

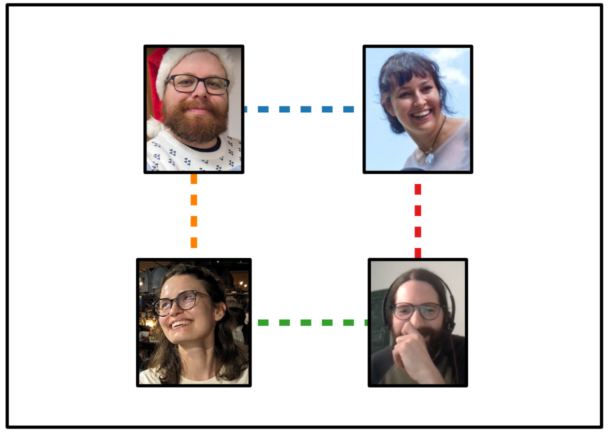
- G triangulated ➡ $\mathcal{O}(n)$ time
- G biconnected ➡ NP-complete

Open Problems

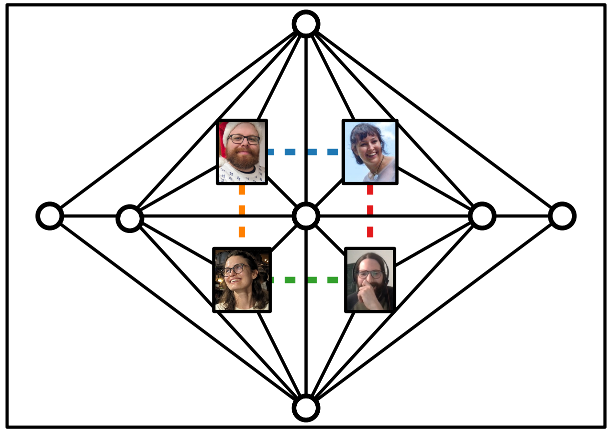
Conclusion



plane graph G



edges E' btw. vtcs in G



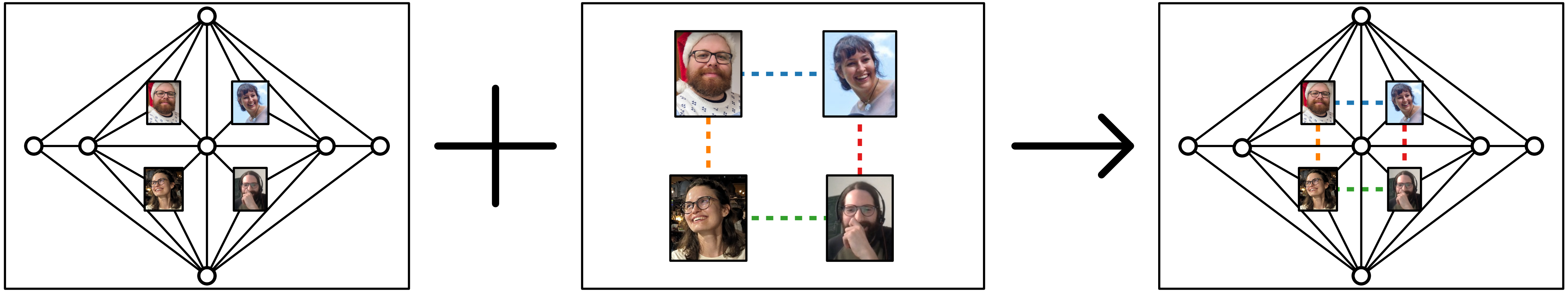
1-planar drawing of $G + E'$ that keeps the embedding of G

- G triangulated ➡ $\mathcal{O}(n)$ time
- G biconnected ➡ NP-complete

Open Problems

- G triconnected?

Conclusion



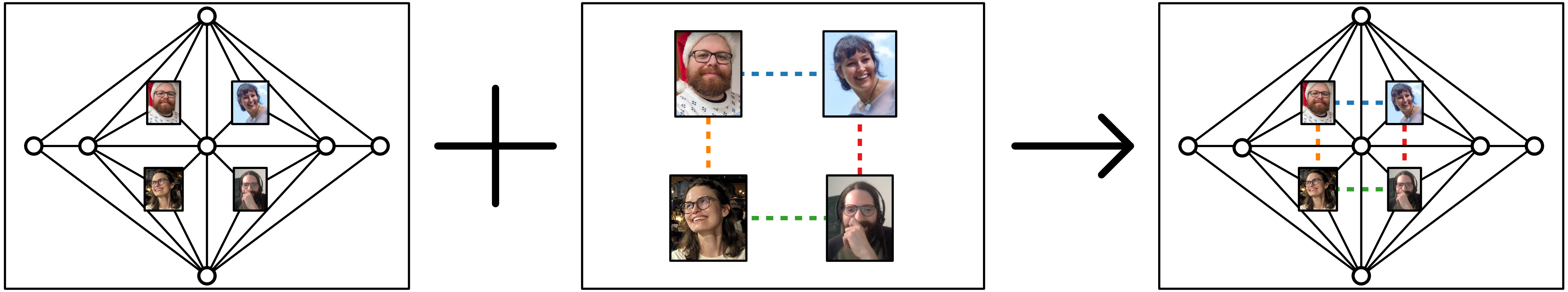
plane graph G edges E' btw. vtcs in G **1-planar** drawing of $G + E'$ that keeps the embedding of G

- G triangulated ➡ $\mathcal{O}(n)$ time
- G biconnected ➡ NP-complete

Open Problems

- G triconnected?
- Other drawing styles? For example

Conclusion

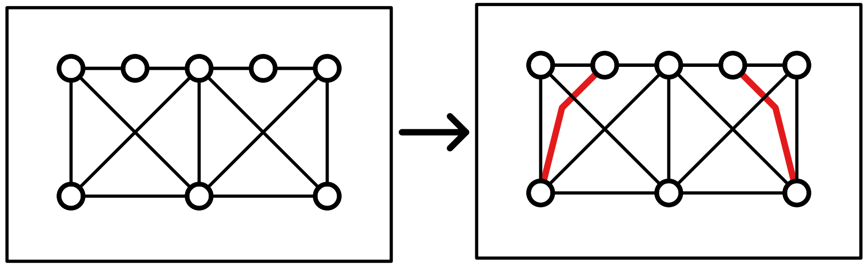


plane graph G — edges E' btw. vtcs in G — **1-planar** drawing of $G + E'$ that keeps the embedding of G

- G triangulated ➡ $\mathcal{O}(n)$ time
- G biconnected ➡ NP-complete

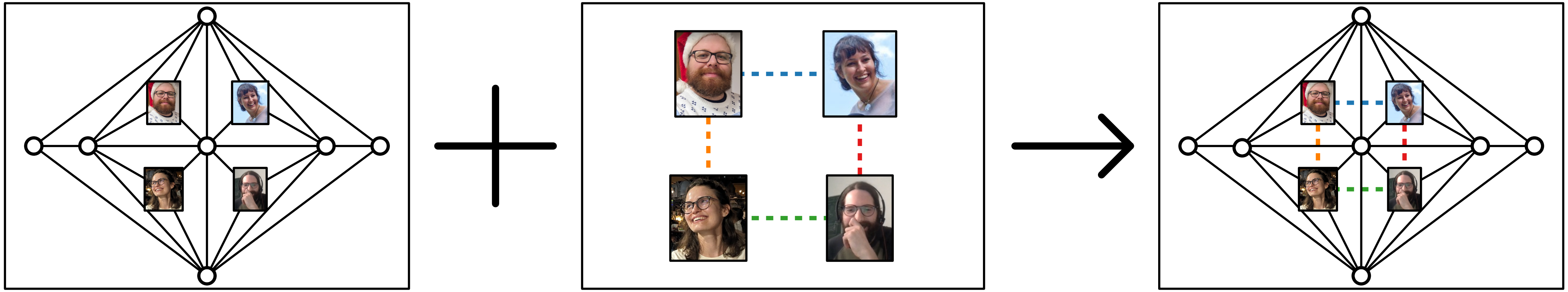
Open Problems

- G triconnected?
- Other drawing styles? For example



RAC → 1-bend RAC

Conclusion

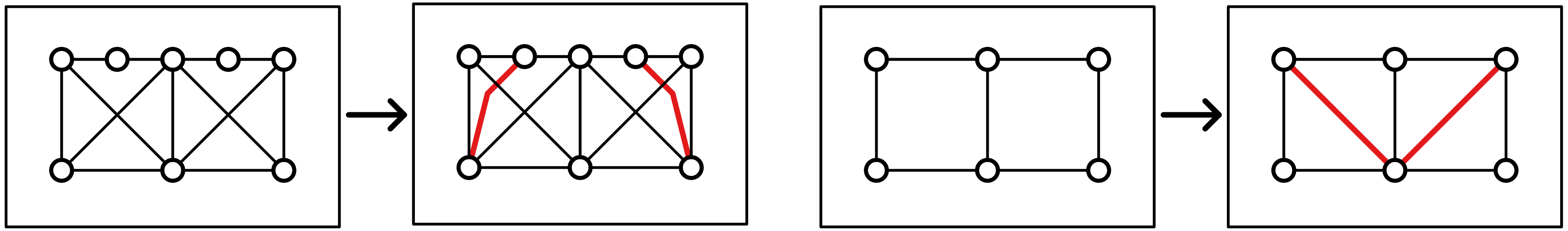


plane graph G | edges E' btw. vtcs in G | **1-planar** drawing of $G + E'$ that keeps the embedding of G

- G triangulated ➡ $\mathcal{O}(n)$ time
- G biconnected ➡ NP-complete

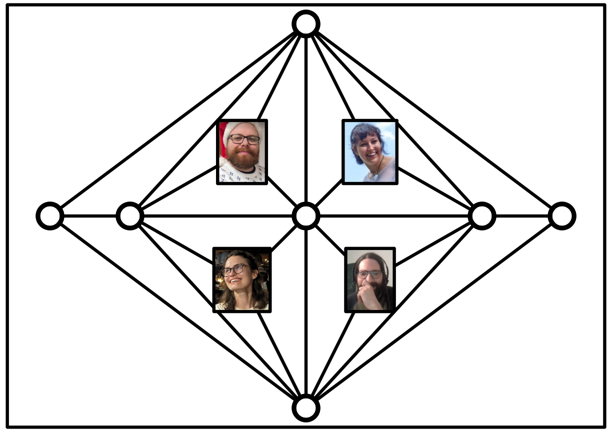
Open Problems

- G triconnected?
- Other drawing styles? For example

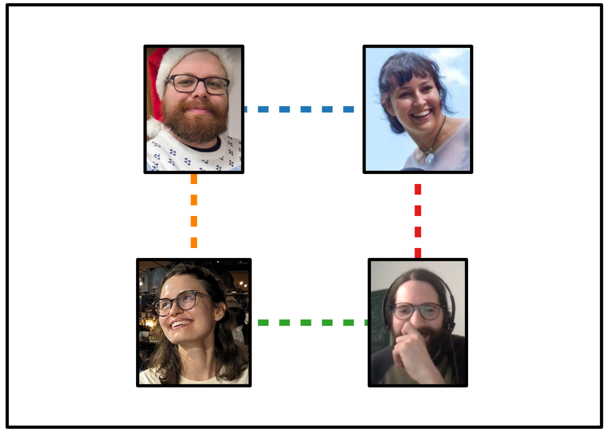


RAC → 1-bend RAC   orthogonal → octilinear

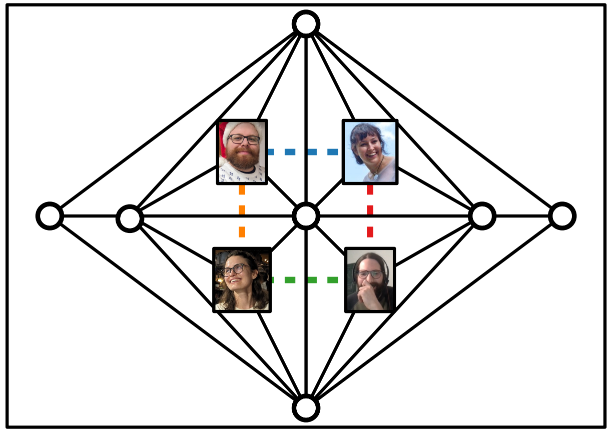
Conclusion



plane graph G



edges E' btw. vtcs in G

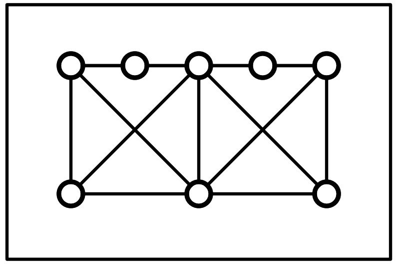


1-planar drawing of $G + E'$ that keeps the embedding of G

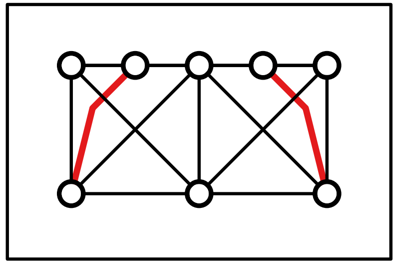
- G triangulated ➡ $\mathcal{O}(n)$ time
- G biconnected ➡ NP-complete

Open Problems

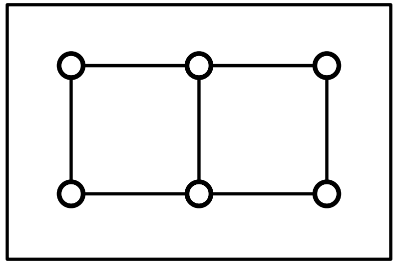
- G triconnected?
- Other drawing styles? For example



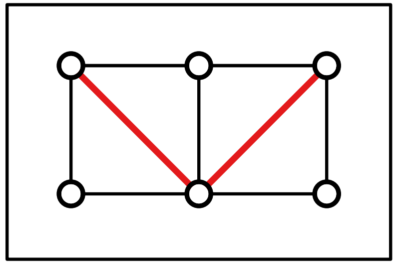
RAC



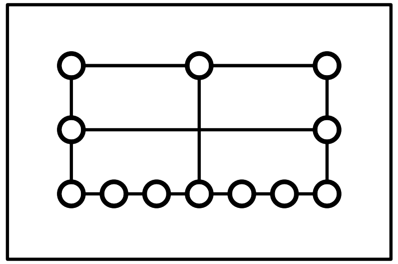
1-bend RAC



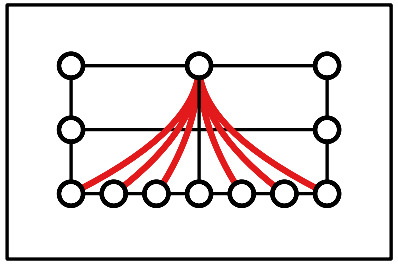
orthogonal



octilinear



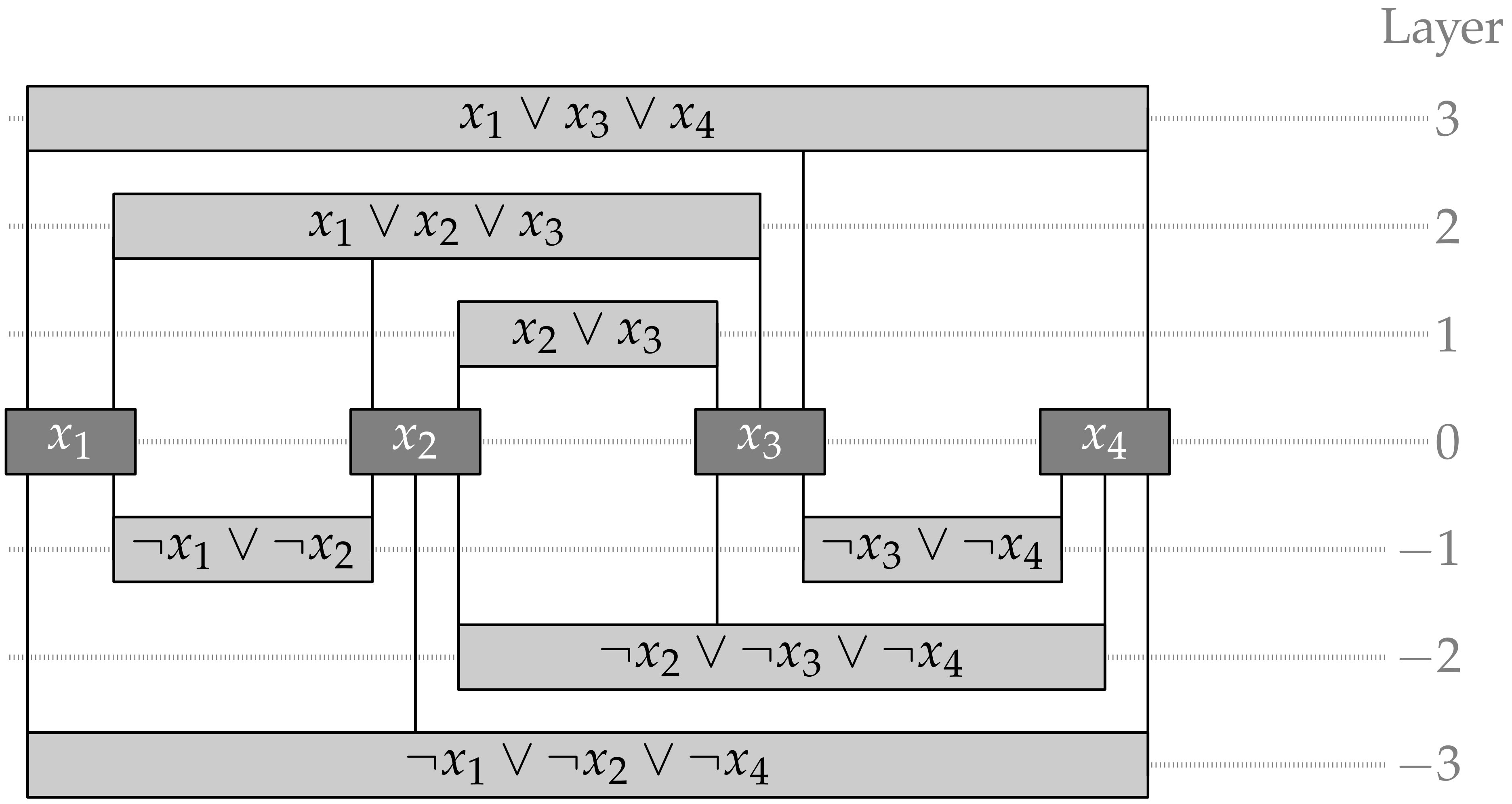
1-planar



fan-planar

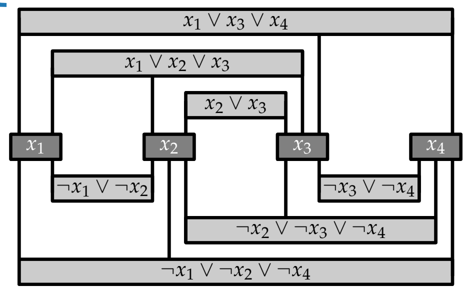
1-Plane Insertion Into a Plane **Biconnected Graph**

1-Plane Insertion Into a Plane **Biconnected Graph**

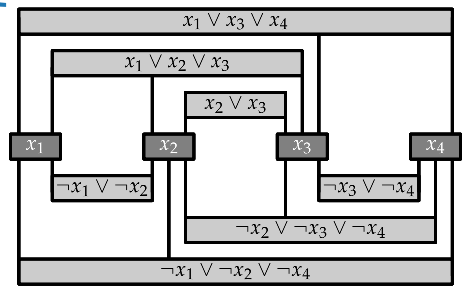


Planar Monotone 3-SAT

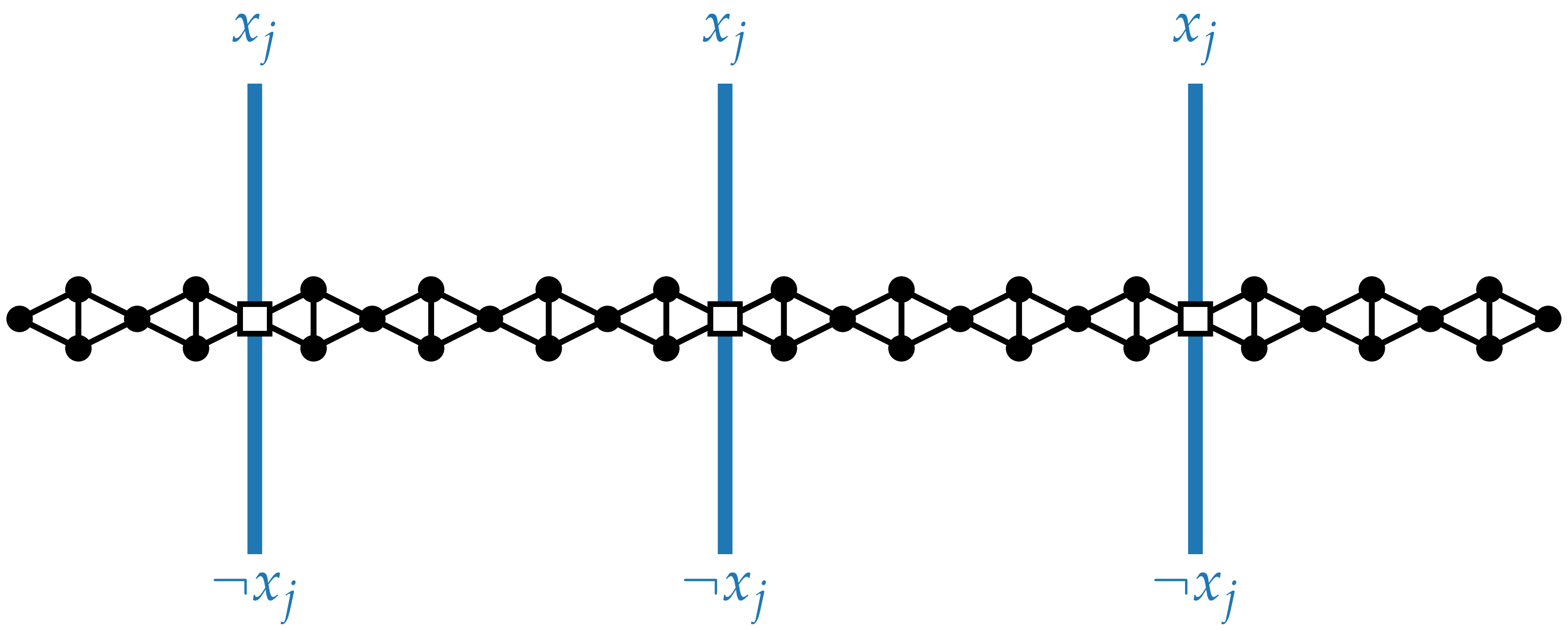
1-Plane Insertion Into a Plane **Biconnected Graph**



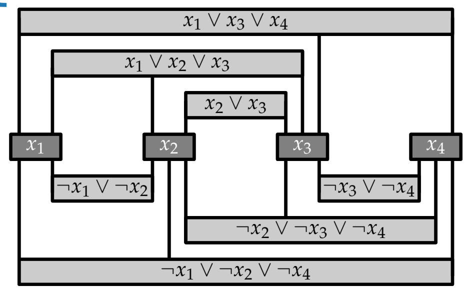
1-Plane Insertion Into a Plane **Biconnected Graph**



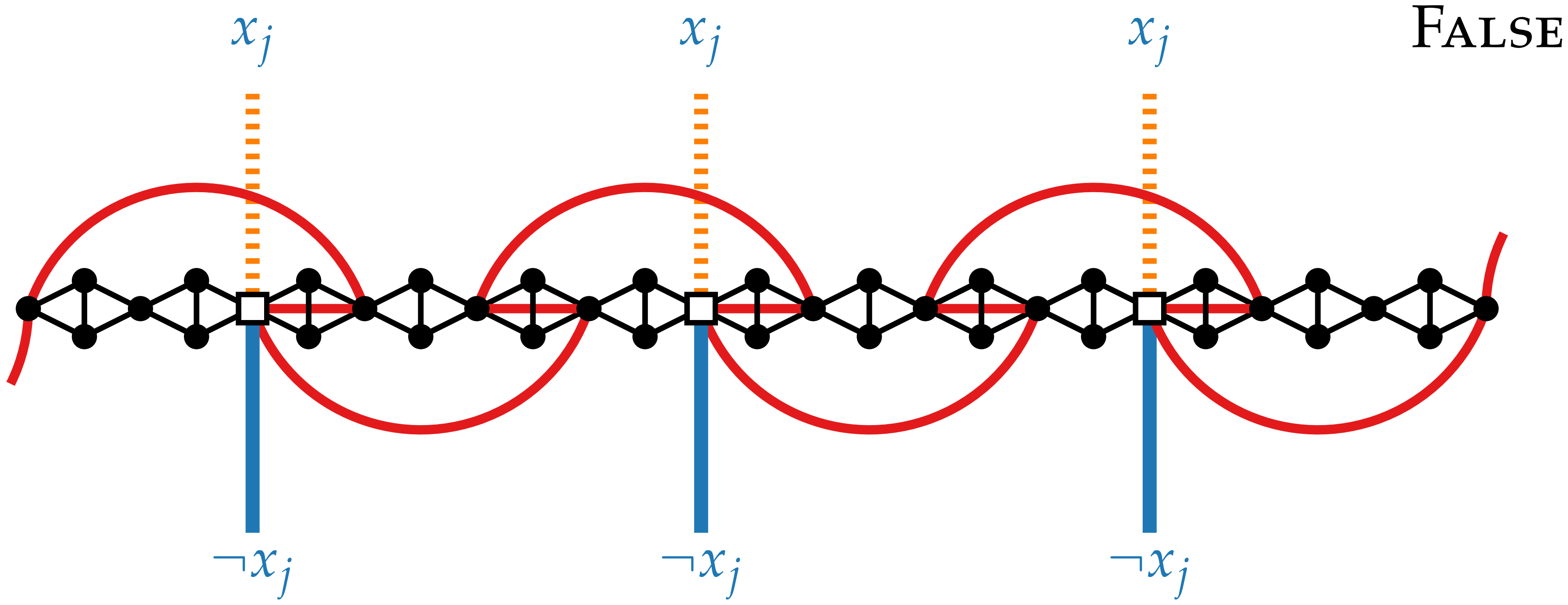
Variable Gadget.



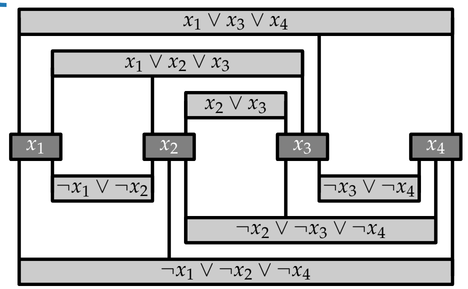
1-Plane Insertion Into a Plane **Biconnected Graph**



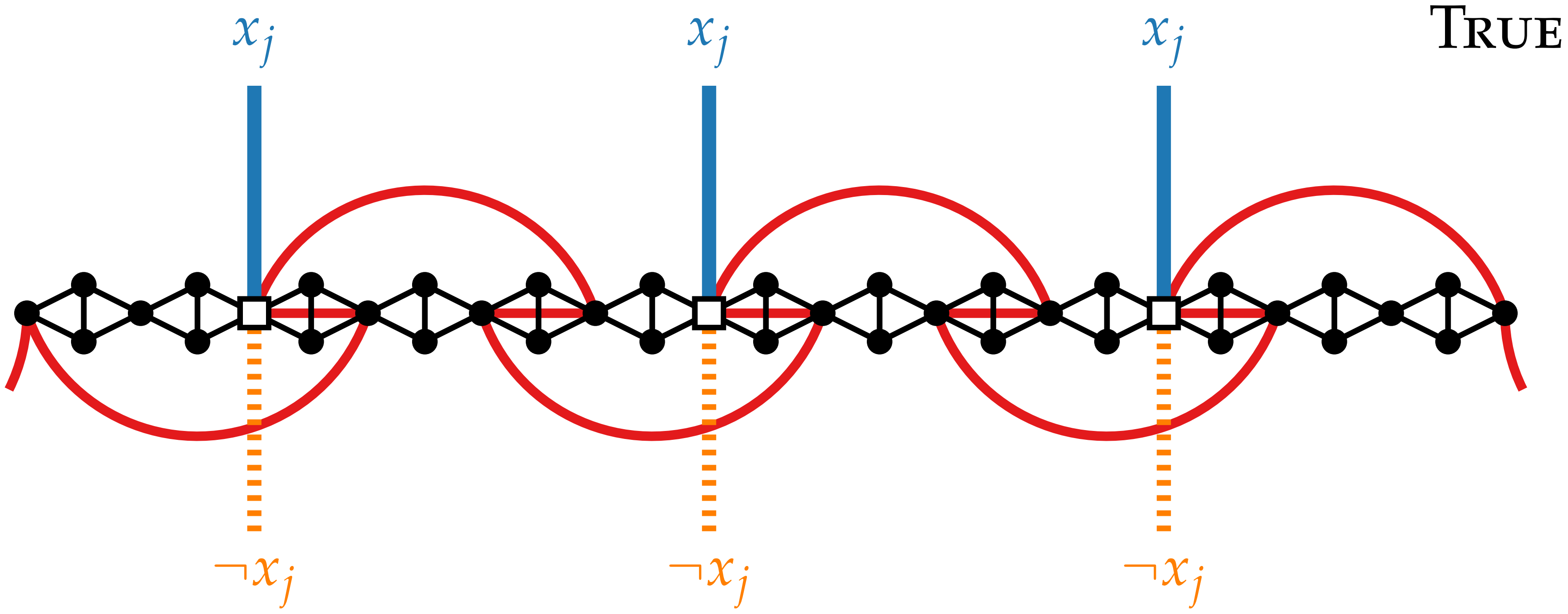
Variable Gadget.



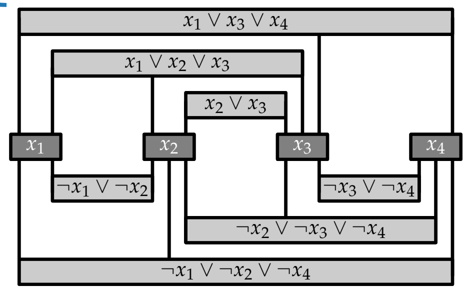
1-Plane Insertion Into a Plane **Biconnected Graph**



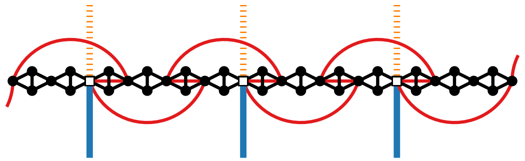
Variable Gadget.



1-Plane Insertion Into a Plane **Biconnected Graph**

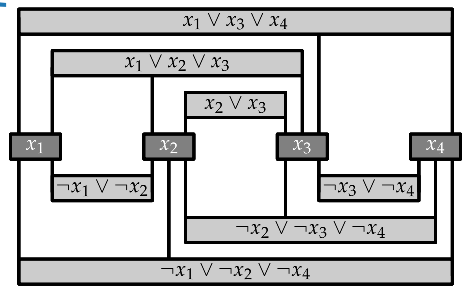


Variable Gadget.

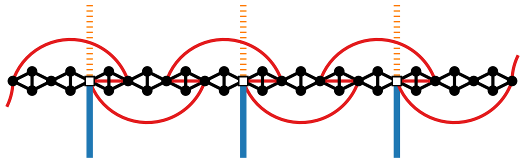


1-Plane Insertion Into a Plane **Biconnected Graph**

Clause Gadget.

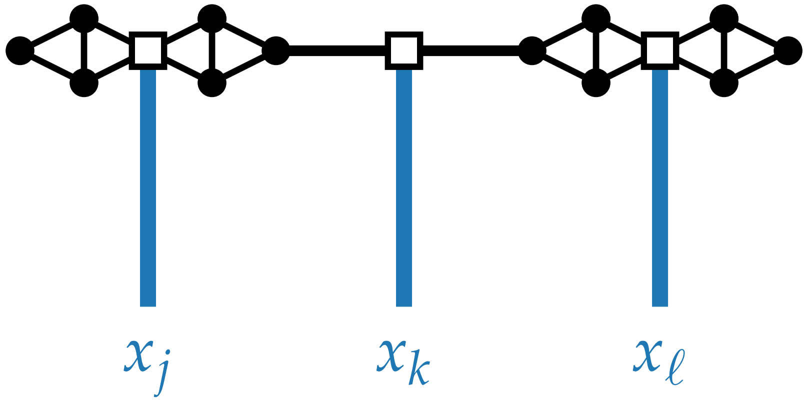


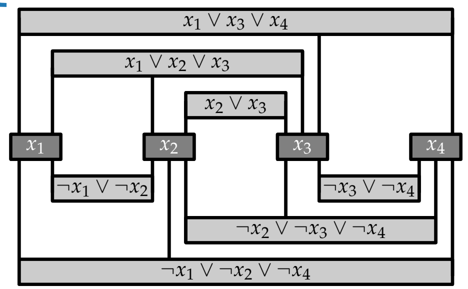
Variable Gadget.



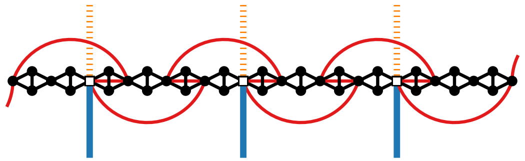
1-Plane Insertion Into a Plane **Biconnected Graph**

Clause Gadget.

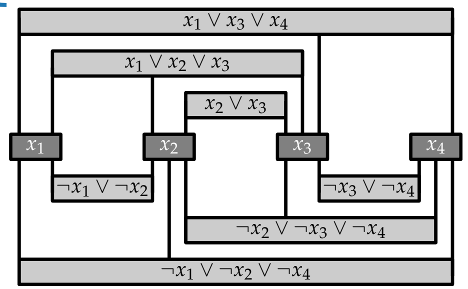




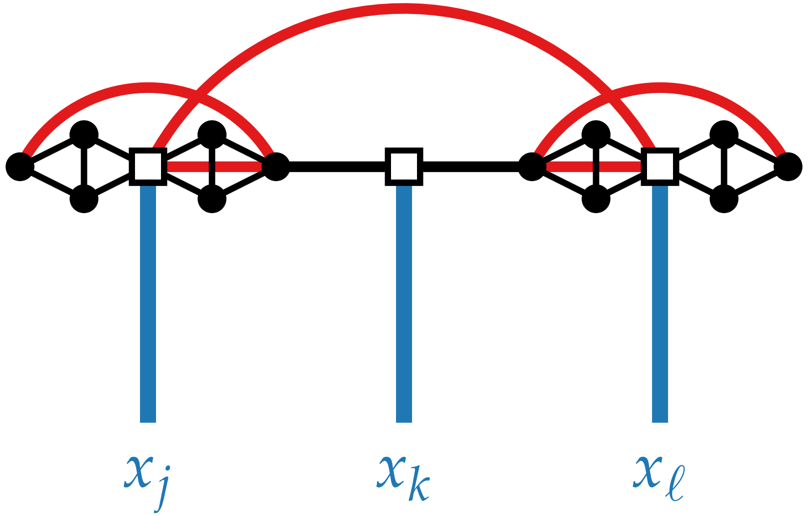
Variable Gadget.



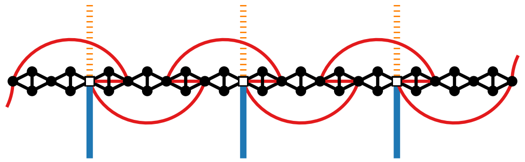
1-Plane Insertion Into a Plane **Biconnected Graph**



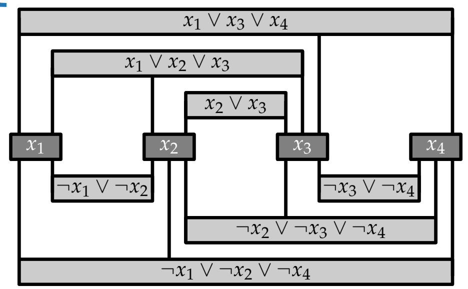
Clause Gadget.



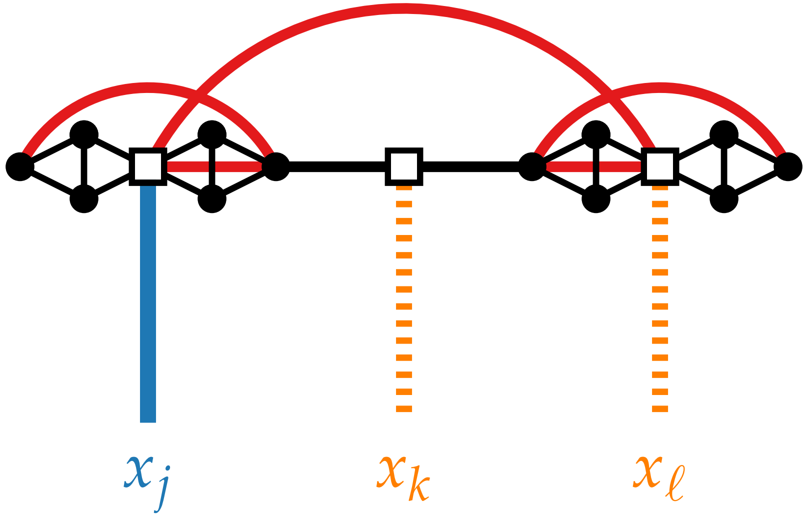
Variable Gadget.



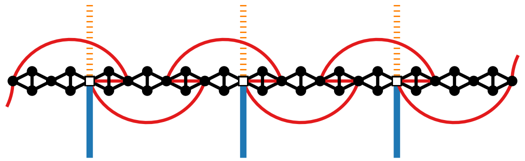
1-Plane Insertion Into a Plane **Biconnected Graph**



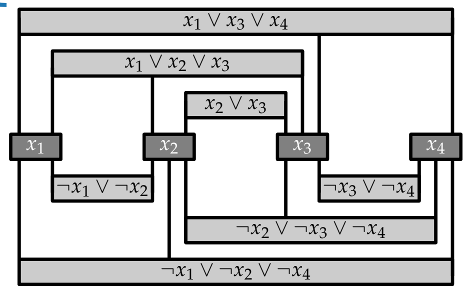
Clause Gadget.



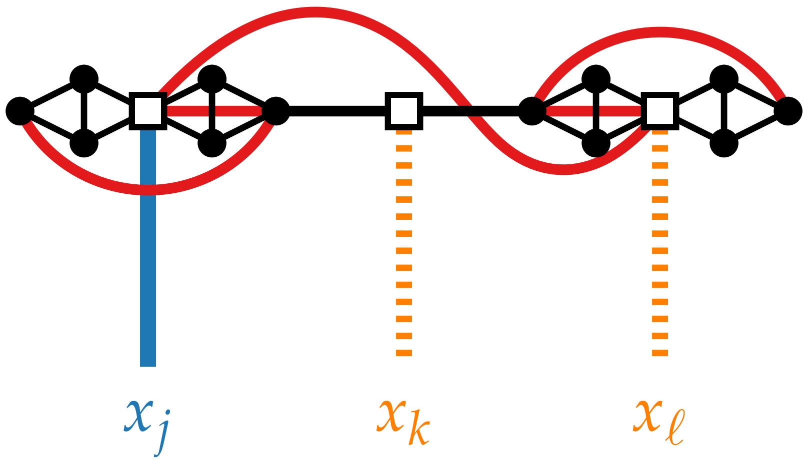
Variable Gadget.



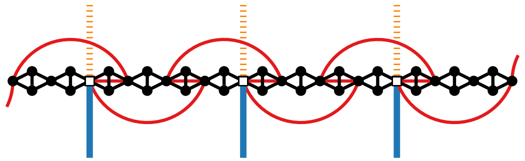
1-Plane Insertion Into a Plane **Biconnected Graph**



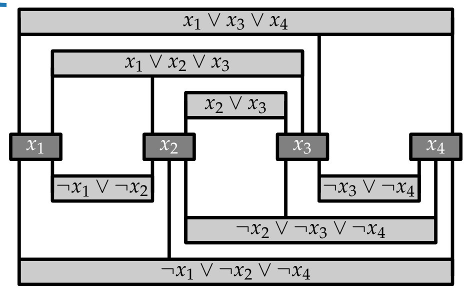
Clause Gadget.



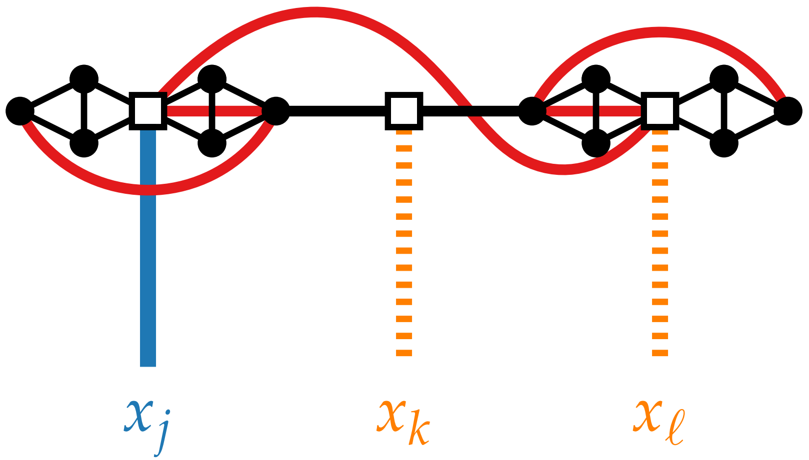
Variable Gadget.

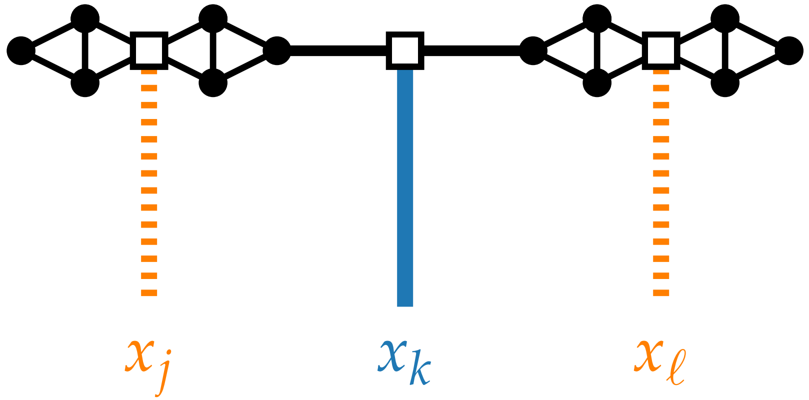


1-Plane Insertion Into a Plane **Biconnected Graph**

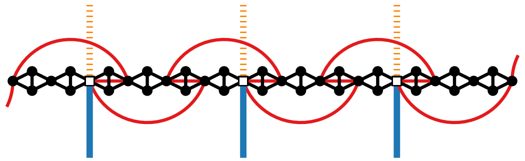


Clause Gadget.

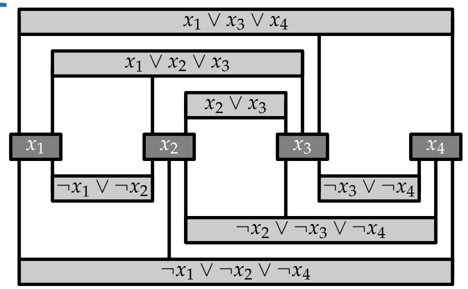




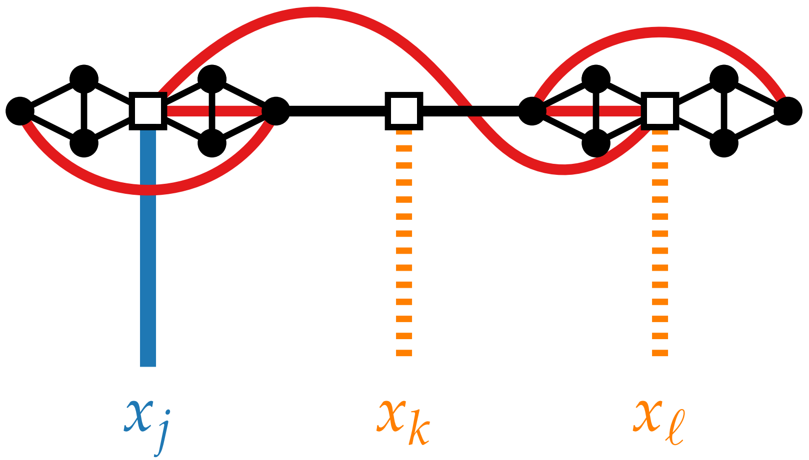
Variable Gadget.

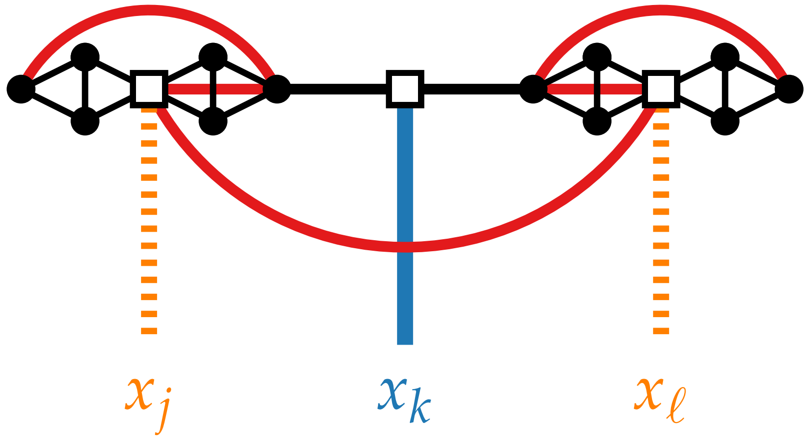


1-Plane Insertion Into a Plane **Biconnected Graph**

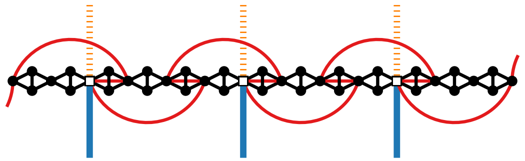


Clause Gadget.

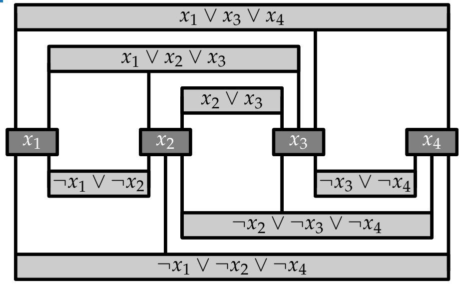




Variable Gadget.

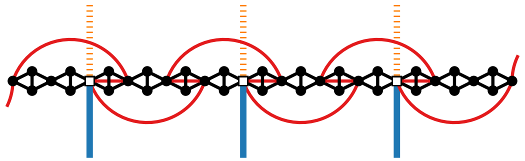


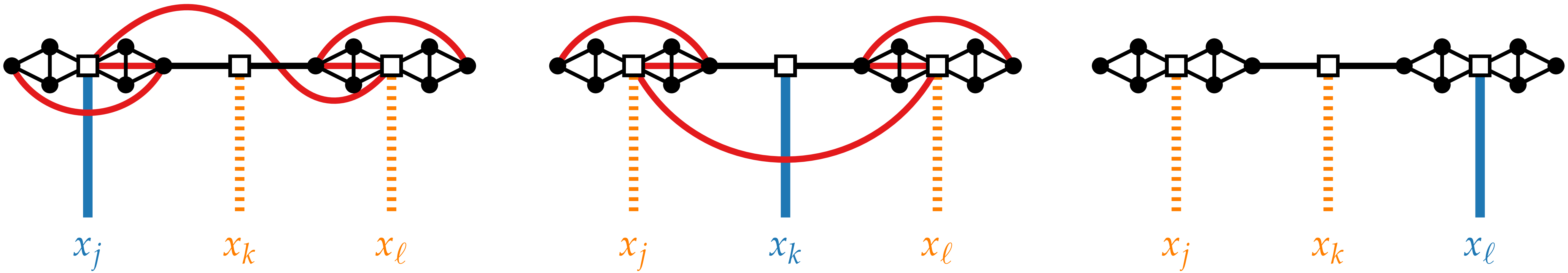
1-Plane Insertion Into a Plane **Biconnected Graph**



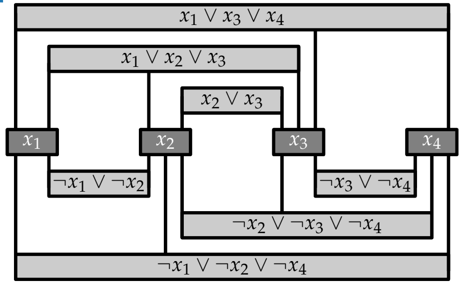
Clause Gadget.

Variable Gadget.



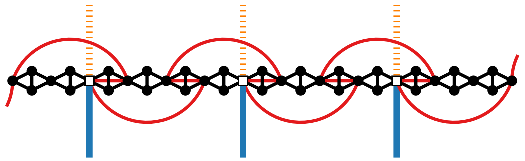


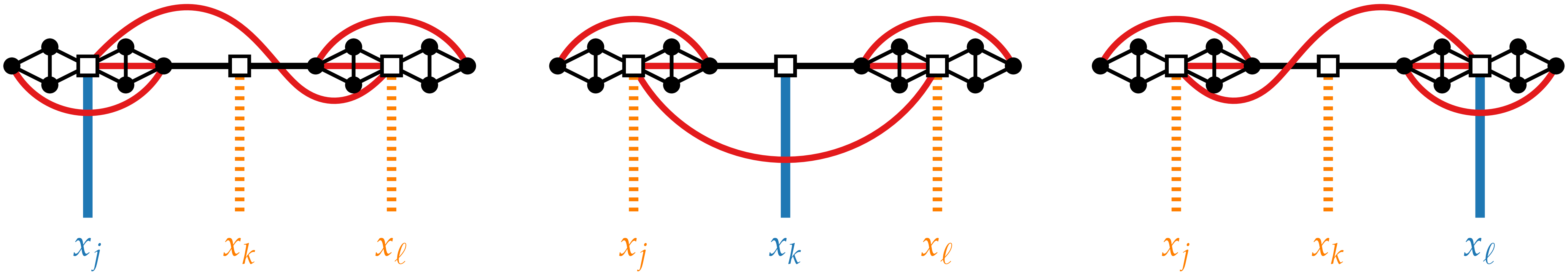
1-Plane Insertion Into a Plane **Biconnected Graph**



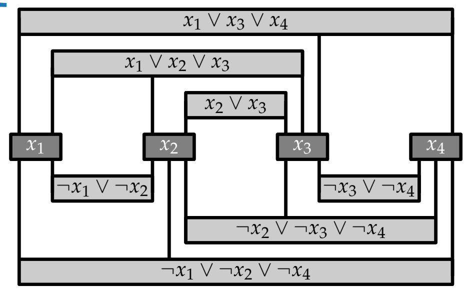
Clause Gadget.

Variable Gadget.

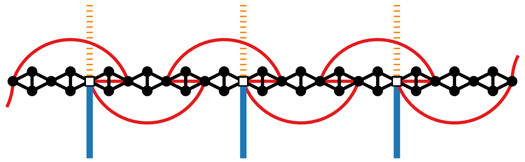




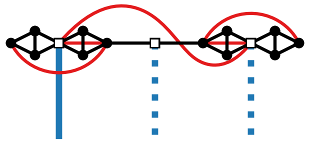
1-Plane Insertion Into a Plane **Biconnected Graph**



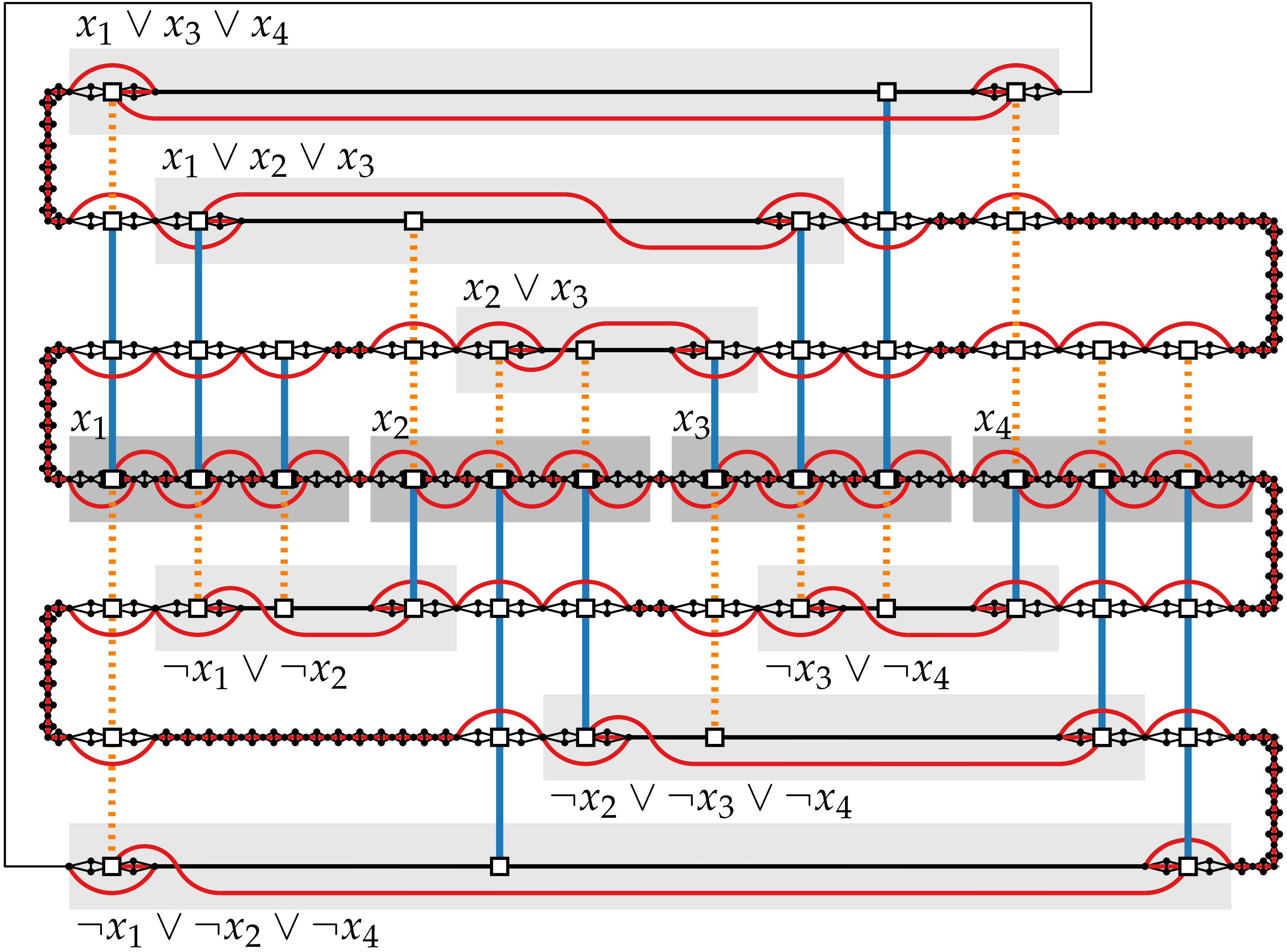
Variable Gadget.

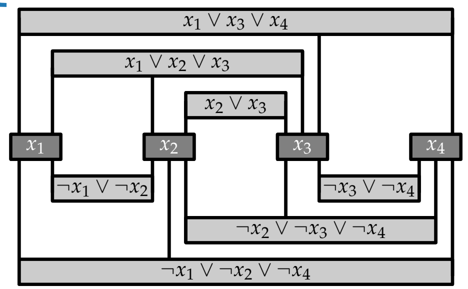


Clause Gadget.

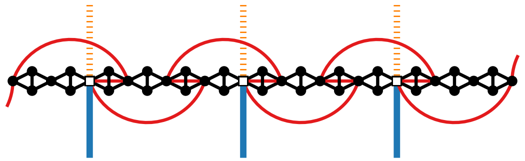


1-Plane Insertion Into a Plane **Biconnected Graph**

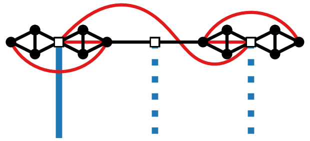




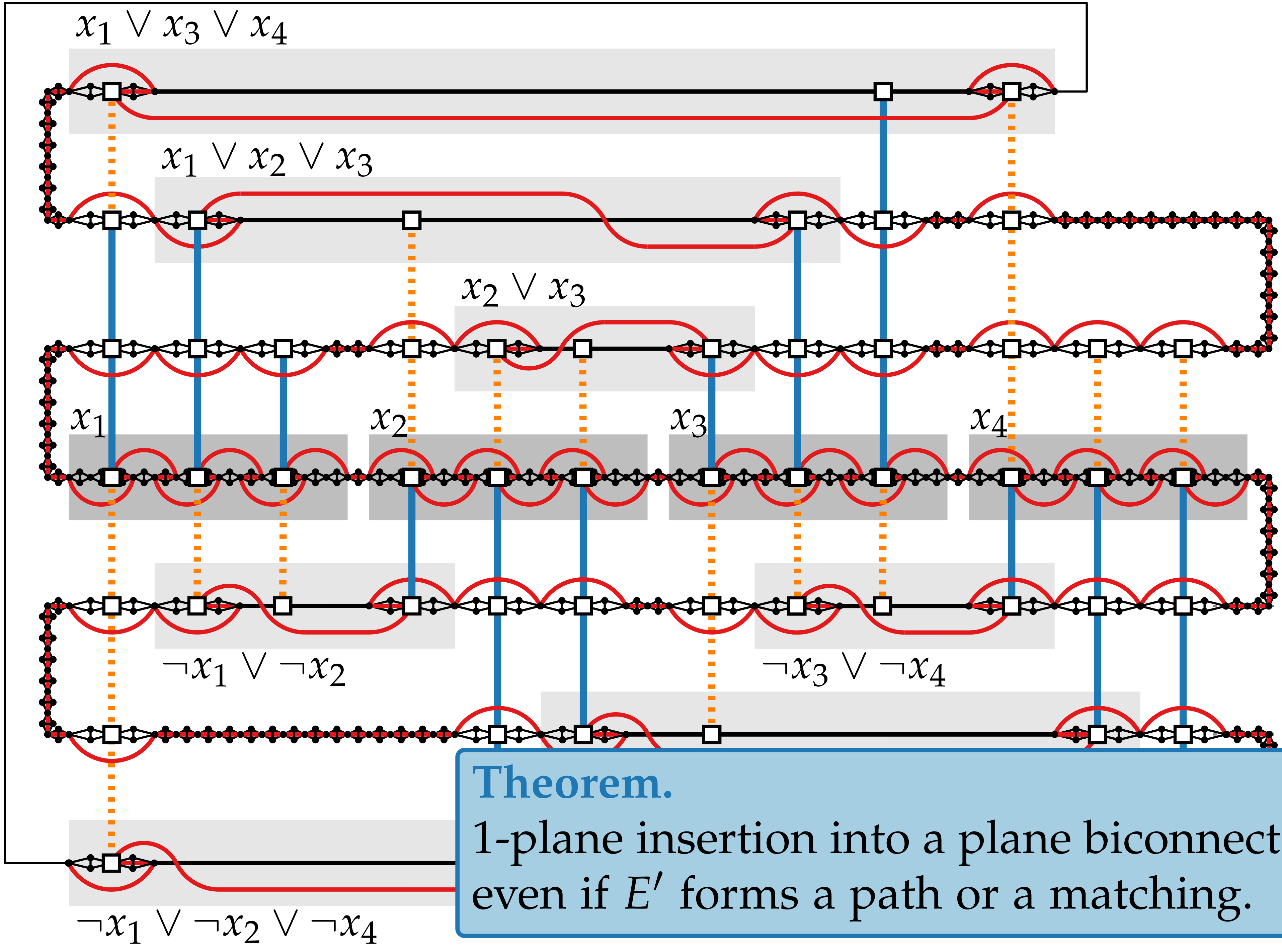
Variable Gadget.

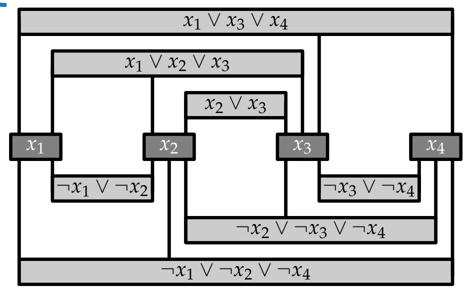


Clause Gadget.

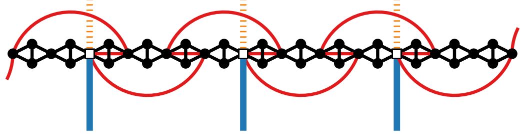


1-Plane Insertion Into a Plane **Biconnected Graph**

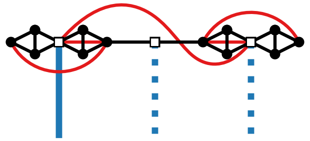




Variable Gadget.



Clause Gadget.



Theorem.
1-plane insertion into a plane biconnected graph is NP-complete, even if E' forms a path or a matching.