

Proving lower bounds in the LOCAL model

Juho Hirvonen,
IRIF, CNRS, and Université Paris Diderot

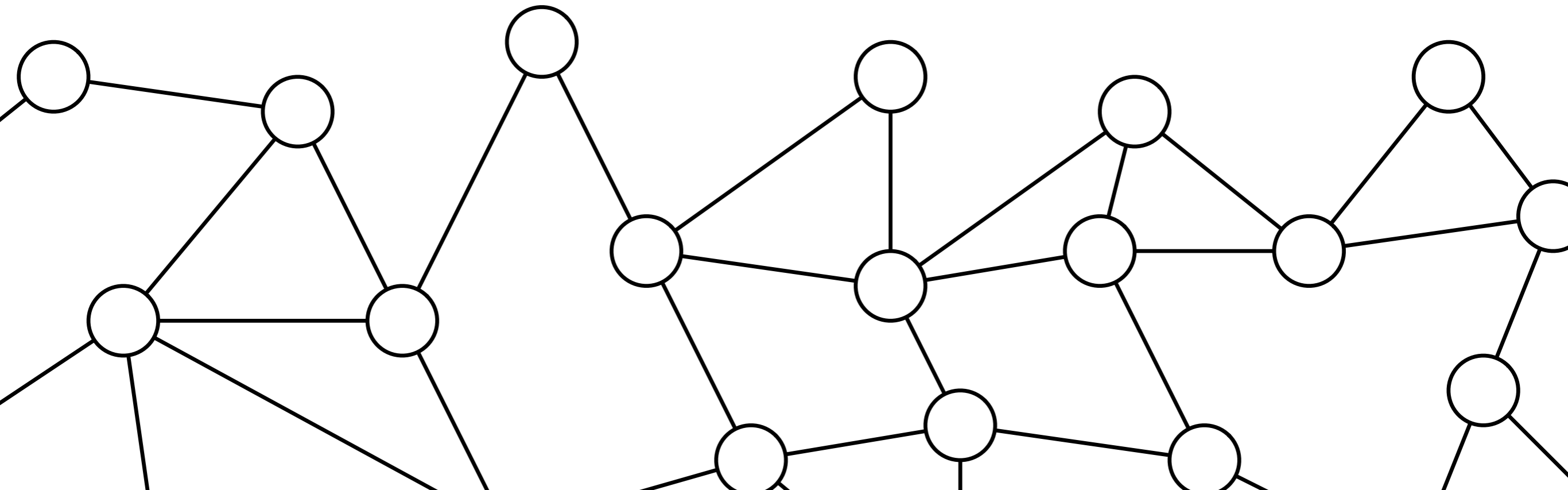
2 October, 2017

Talk outline

- Sketch a lower bound **proof technique** for distributed graph algorithms
- In general, **simulation** is a very powerful tool for lower bounds
- We have the beginnings of a **complexity theory**: can use heavy hammers in lower bound proofs

LOCAL model

- **input** = the communication network
- **output** = every computer produces **local** output
- **global output** = the union of **local outputs**



LOCAL model

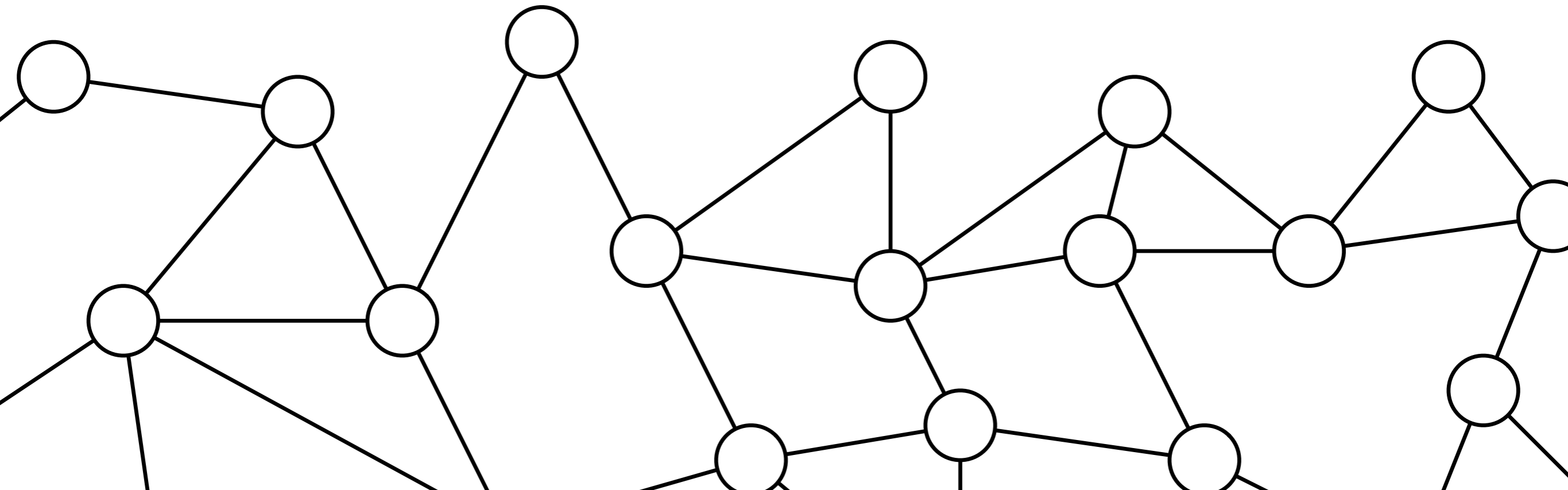
Nodes are running a **synchronous loop**:

1. **exchange** messages
2. **update** state

No bounds on computation, messages, no failures:
trying to abstract away all challenges **except locality**

LOCAL model

- **complexity** = the number of synchronous communication rounds until all nodes have stopped and announced output



The model

- each node has a unique name in **poly(n)**
- graph has bounded maximum degree $\Delta = O(1)$
- for lower bounds, assume **graph size n** is known

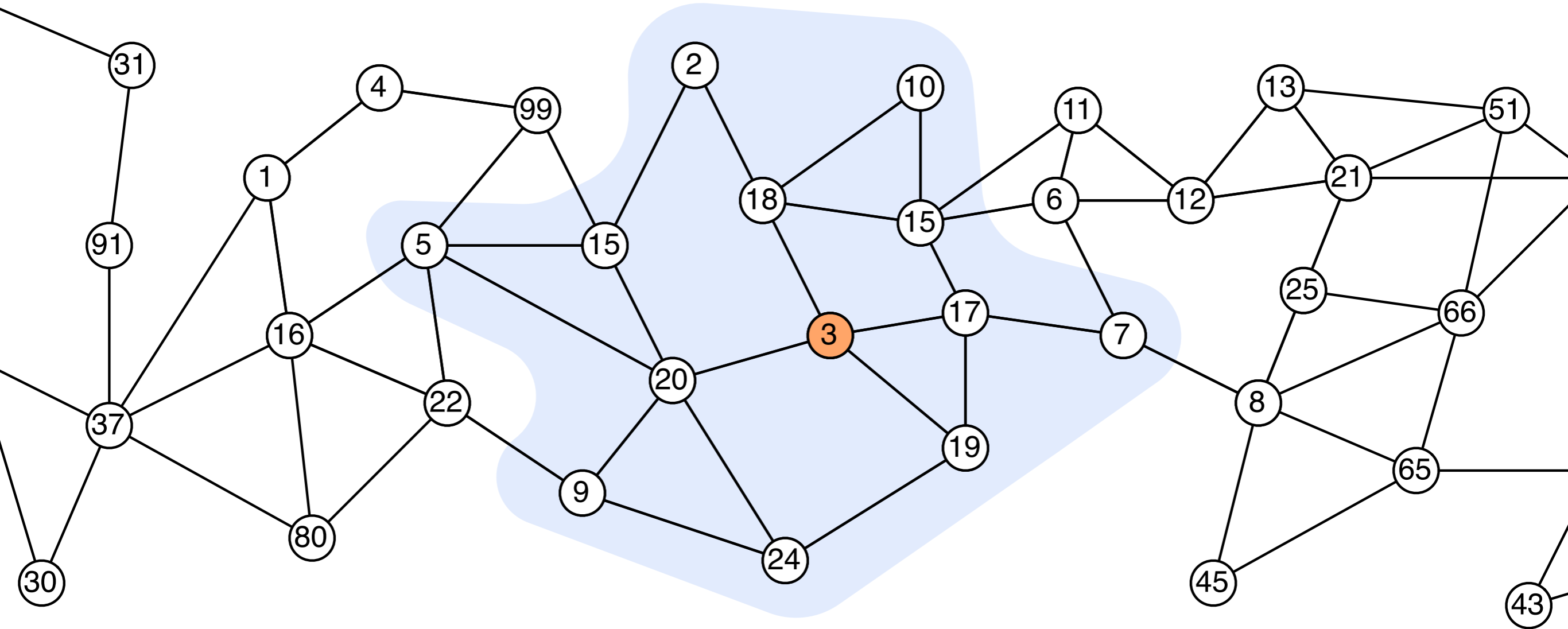
Information-limited

- In **t rounds**, information can propagate at most **t hops** in the network
- After **t rounds**, output cannot depend on input that is more than **t hops** away
- Gathering the **t-hop neighborhood** is all an algorithm can do!

time \approx distance

Information-limited

Assume: $t = 2$



Information-limited

algorithm

=

function on **t-hop neighbourhoods**

The lower bound

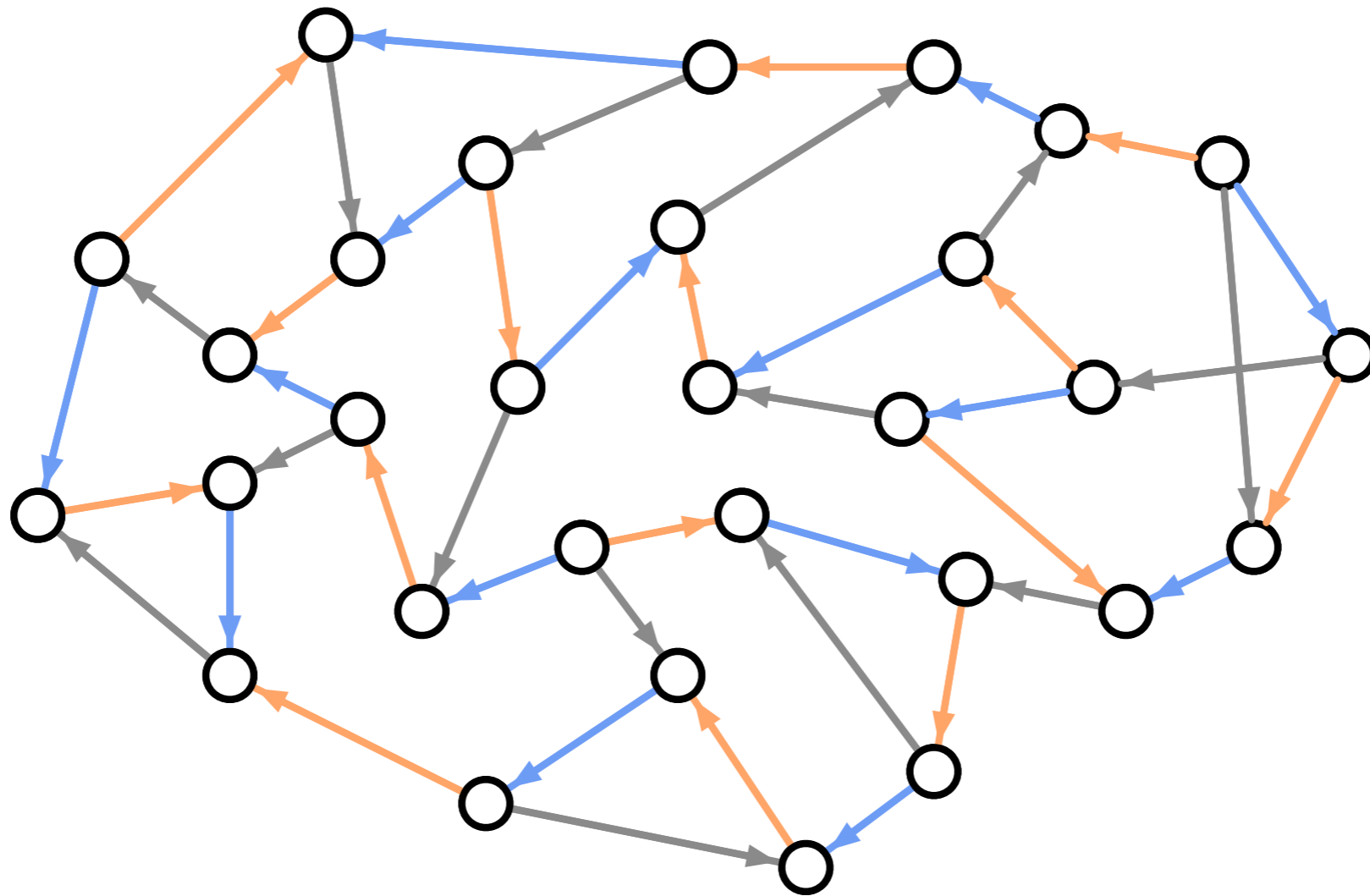
Based on*

A lower bound for the distributed Lovász local lemma,
Brandt, Fischer, Hirvonen, Keller, Lempääinen, Rybicki,
Suomela, and Uitto, STOC 2016

*An exponential separation between randomized and
deterministic complexity in the LOCAL model*
Chang, Kopelowitz, and Pettie, FOCS 2016

*The Complexity of Distributed Edge Coloring with Small
Palettes,* Chang, He, Li, Pettie, and Uitto, 2017

Sinkless orientation



All edges are oriented with no sinks

The lower bound

Sinkless orientation requires $\Omega(\log_{\Delta} n)$ deterministic time

Sinkless orientation requires $\Omega(\log_{\Delta} \log n)$ randomized time

Why do we care?

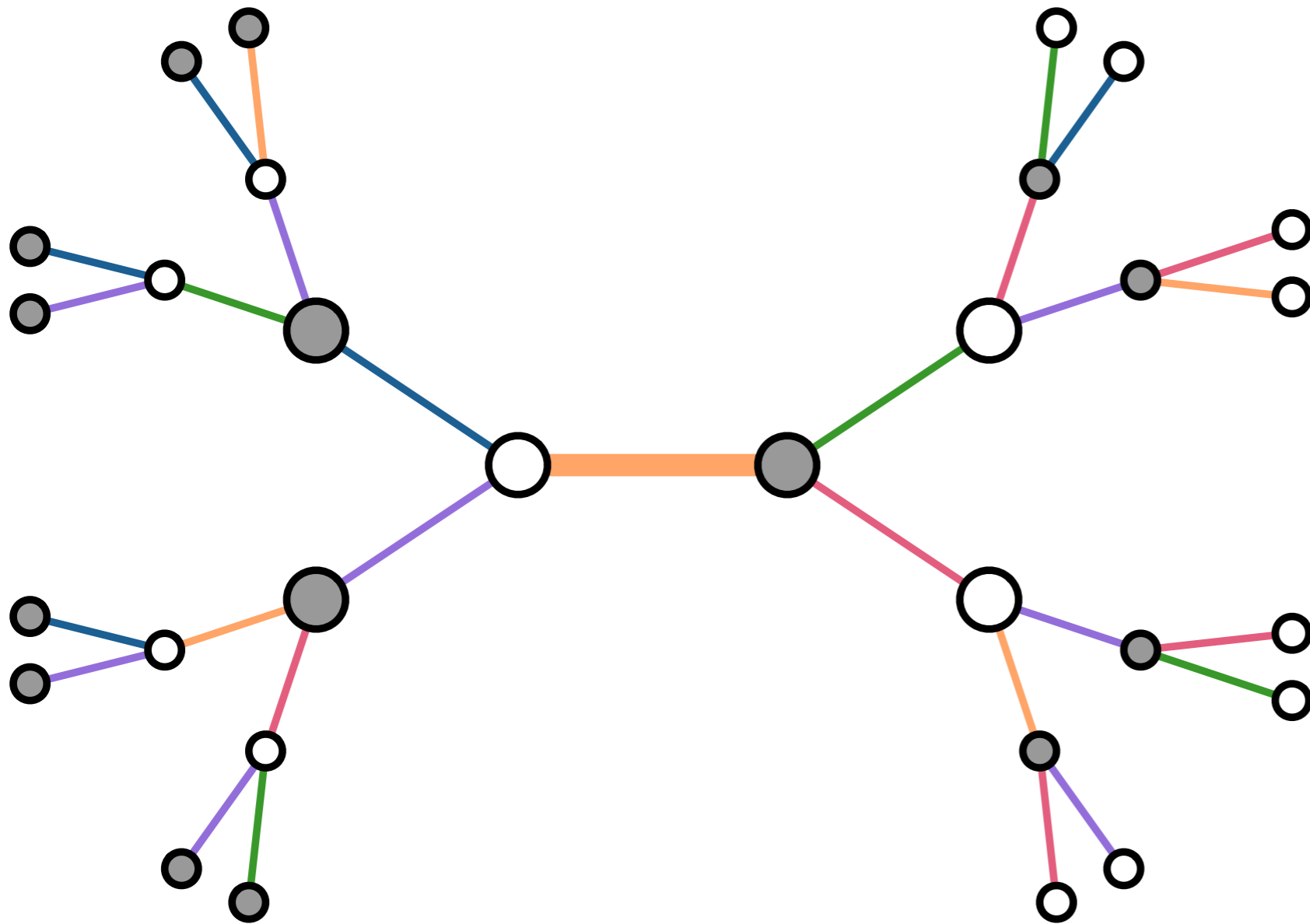
- First example of a problem with **intermediate complexity**
- In **edge-coloured** graphs **Δ -colouring** is at least as hard as **sinkless orientation**
- In **2-vertex-coloured** and **d-edge-coloured** graphs **$(2\Delta-2)$ -edge coloring** is at least as hard as **sinkless orientation**
- Sinkless orientation is a useful primitive for finding **balanced edge splits** (which are at least as hard)

A (simple) deterministic lower bound

We will start by proving a simple lower bound for a
simple deterministic model:

Finding a **sinkless orientation** requires $\Omega(\log_{\Delta} n)$
communication rounds

Lower bound: sinkless orientation



(simple) model:
d-regular graphs,
2-vertex col.
c-edge col.
(for $c > d$)

graphs have large
(logarithmic) girth

Lower bound: sinkless orientation

For **algorithm A**, define **running time profile**

$$\mathbf{t} = (t_1, t_2, \dots, t_c)$$

=

Edges of color **i** must halt after **t_i** rounds*

Lower bound: sinkless orientation

Assume algorithm has running time profile

$$\mathbf{t} = (t, t, \dots, t)$$

=

Edges of **all colors** halt in t communication rounds

Lower bound: sinkless orientation

For example, assume $d=3$ and $c=5$

$$\mathbf{t} = (t, t, t, t, t)$$



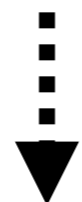
speed up color 5 by simulation

$$\mathbf{t}^{(1)} = (t, t, t, t, t-1)$$



speed up color 4 by simulation

$$\mathbf{t}^{(2)} = (t, t, t, t-1, t-1)$$



Lower bound: sinkless orientation

For example, assume $d=3$ and $c=5$

$$\mathbf{t} = (t, t, t, t, t)$$



speed up each color

$$\mathbf{t-1} = (t-1, t-1, t-1, t-1, t-1)$$



repeat \mathbf{t} times

$$\mathbf{0} = (0, 0, 0, 0, 0)$$

Lower bound: sinkless orientation

For example, assume $d=3$ and $c=5$

algorithm with running time profile

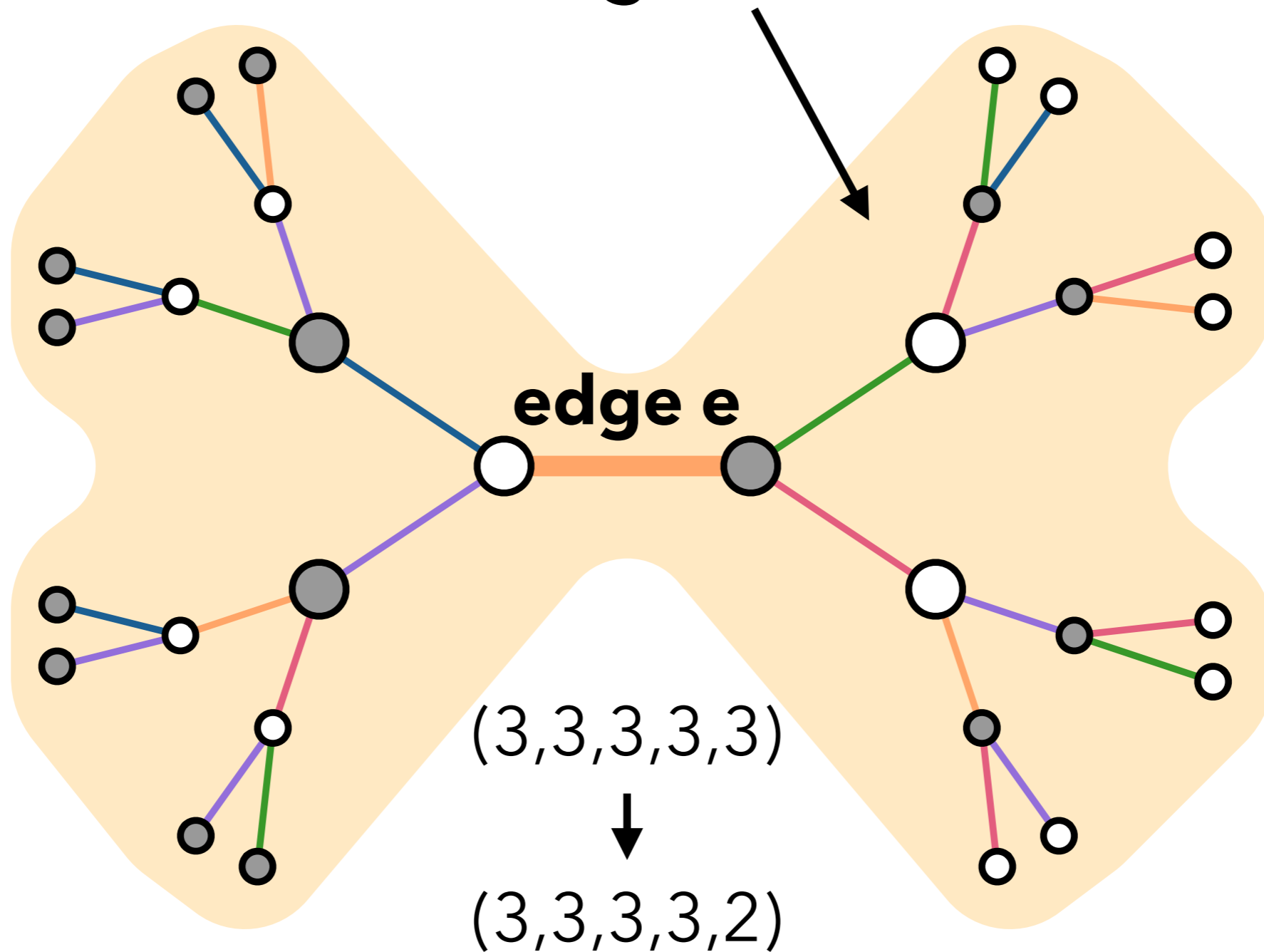
$$\mathbf{0} = (0,0,0,0,0)$$

easy to show that this is impossible!

We can apply argument if initial $\mathbf{t} = \mathbf{o}(\log_{\Delta} n)$

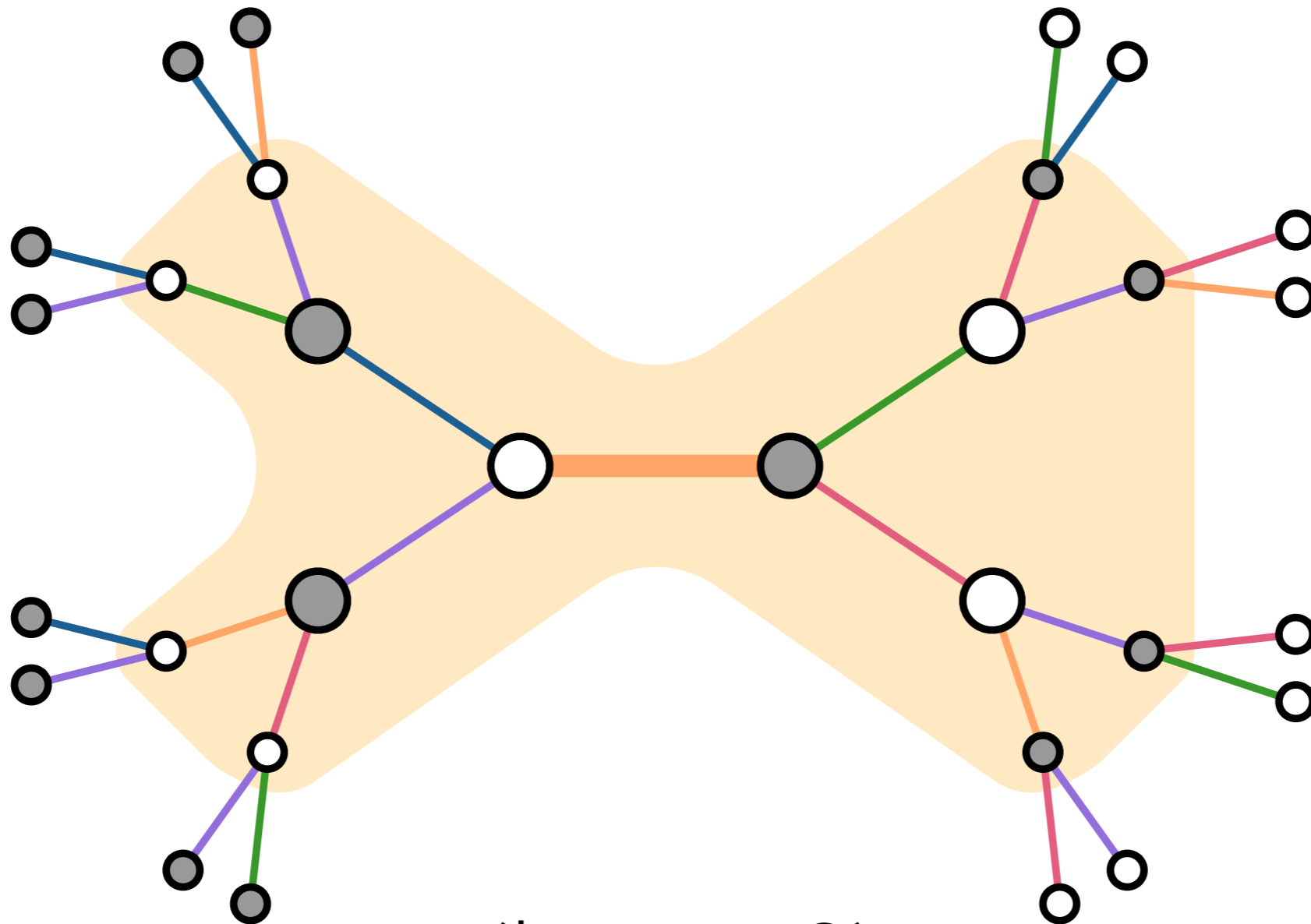
Simulation

3-neighbourhood of orange edge



Simulation

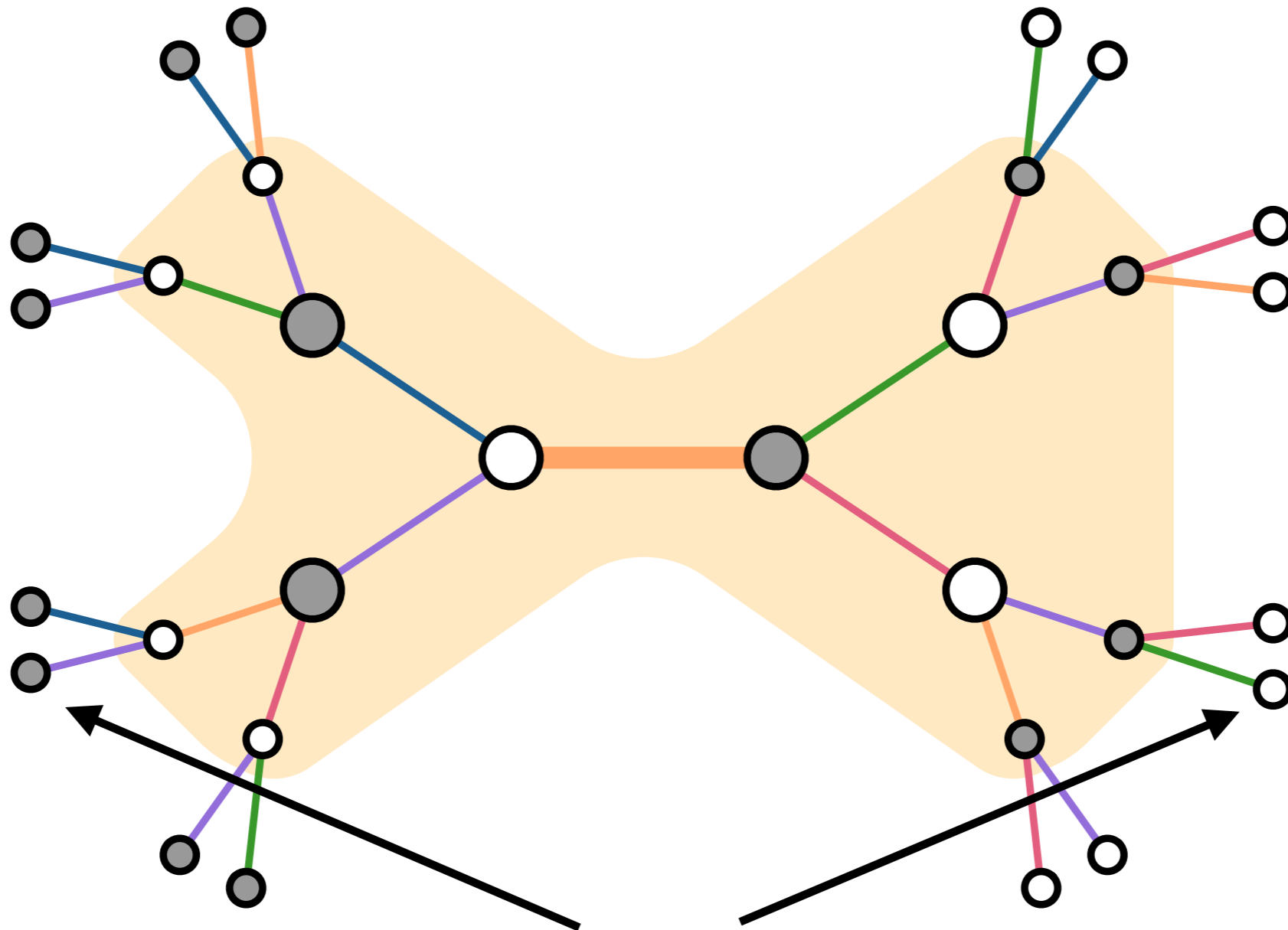
possible outputs given **2-neighbourhood**?



(here **t = 3**)

Simulation

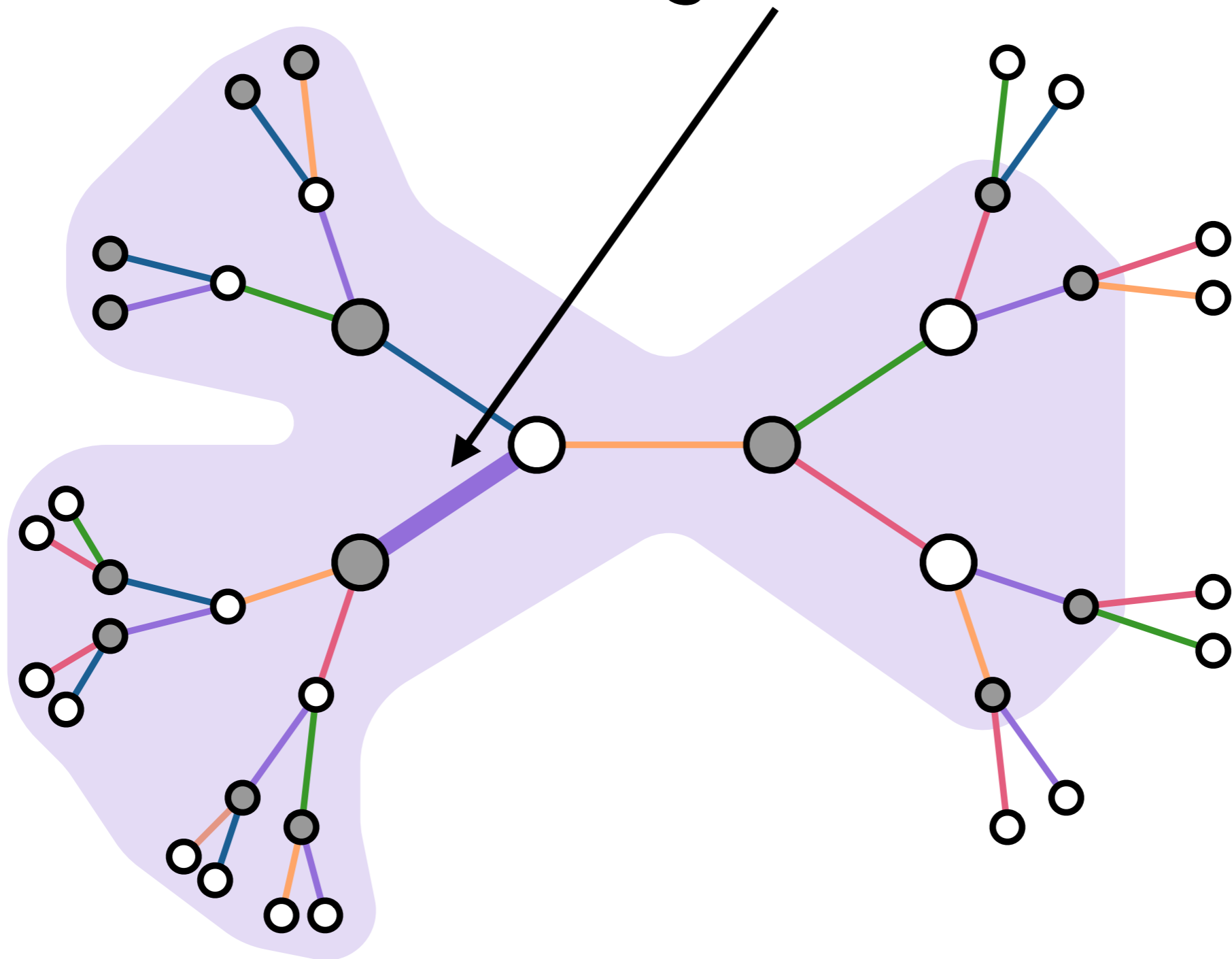
possible outputs given **2-neighbourhood**?



inputs independent

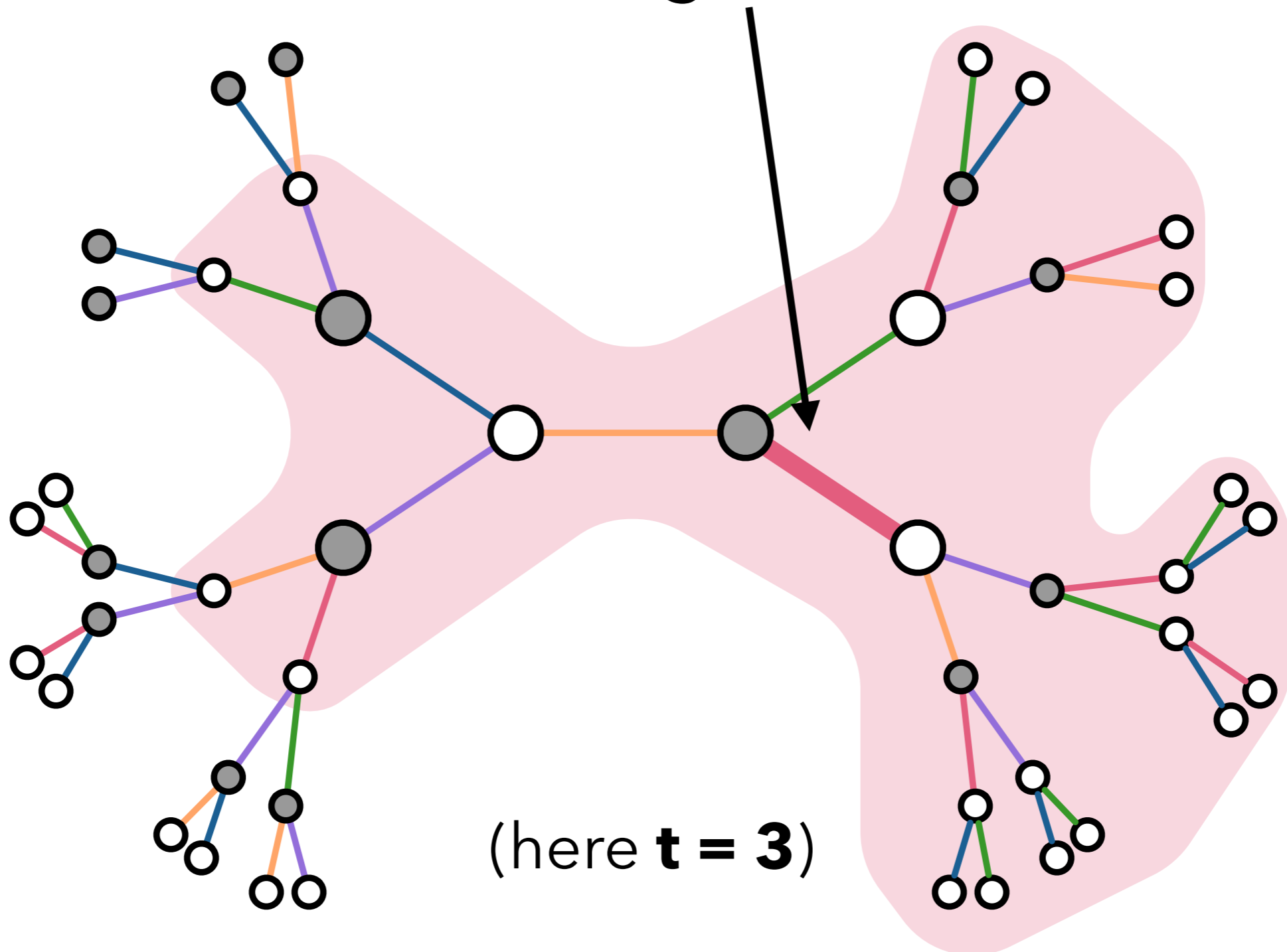
Outputs of incident edges

3-neighbourhood of violet edge



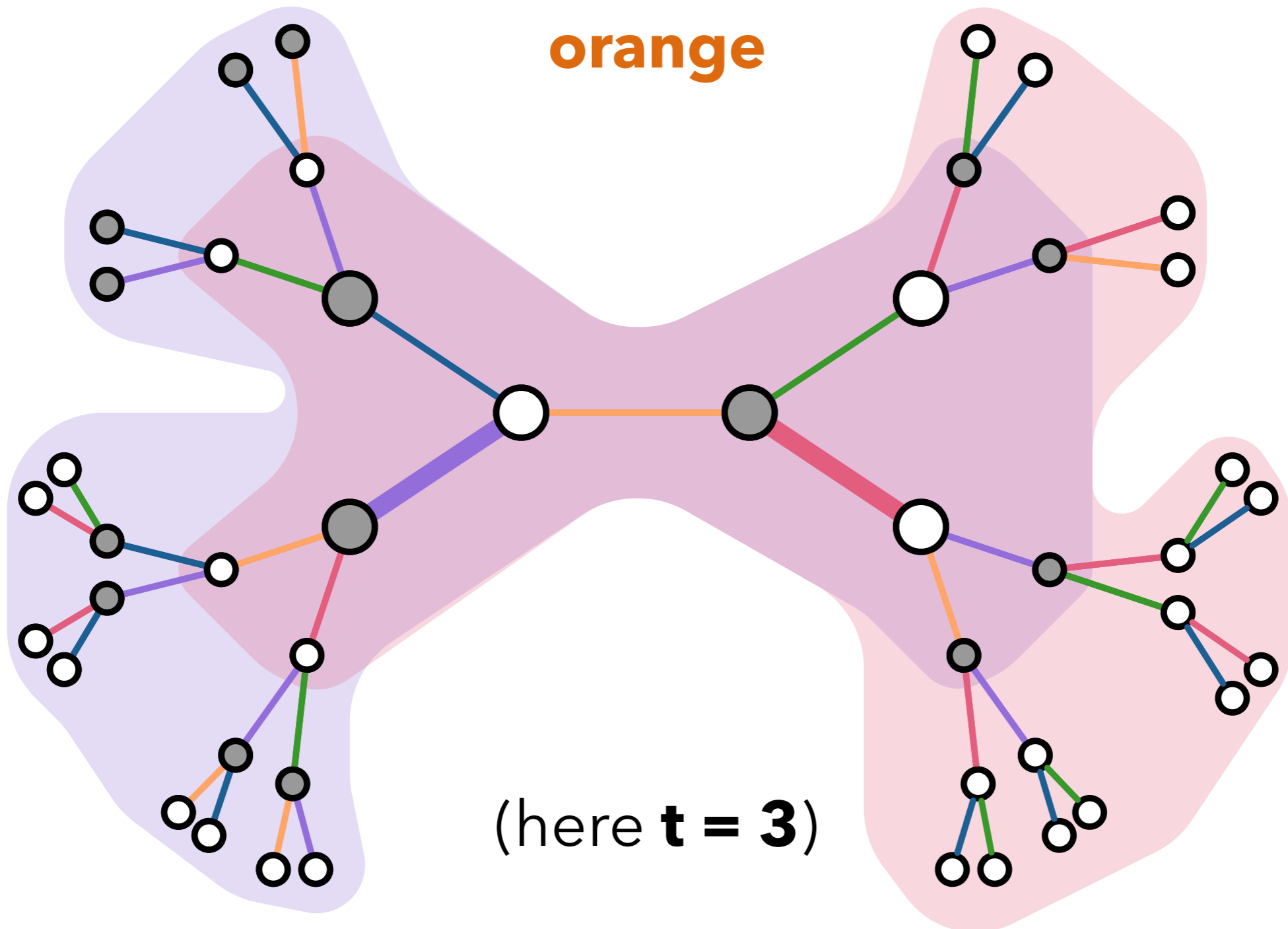
Outputs of incident edges

3-neighbourhood of red edge



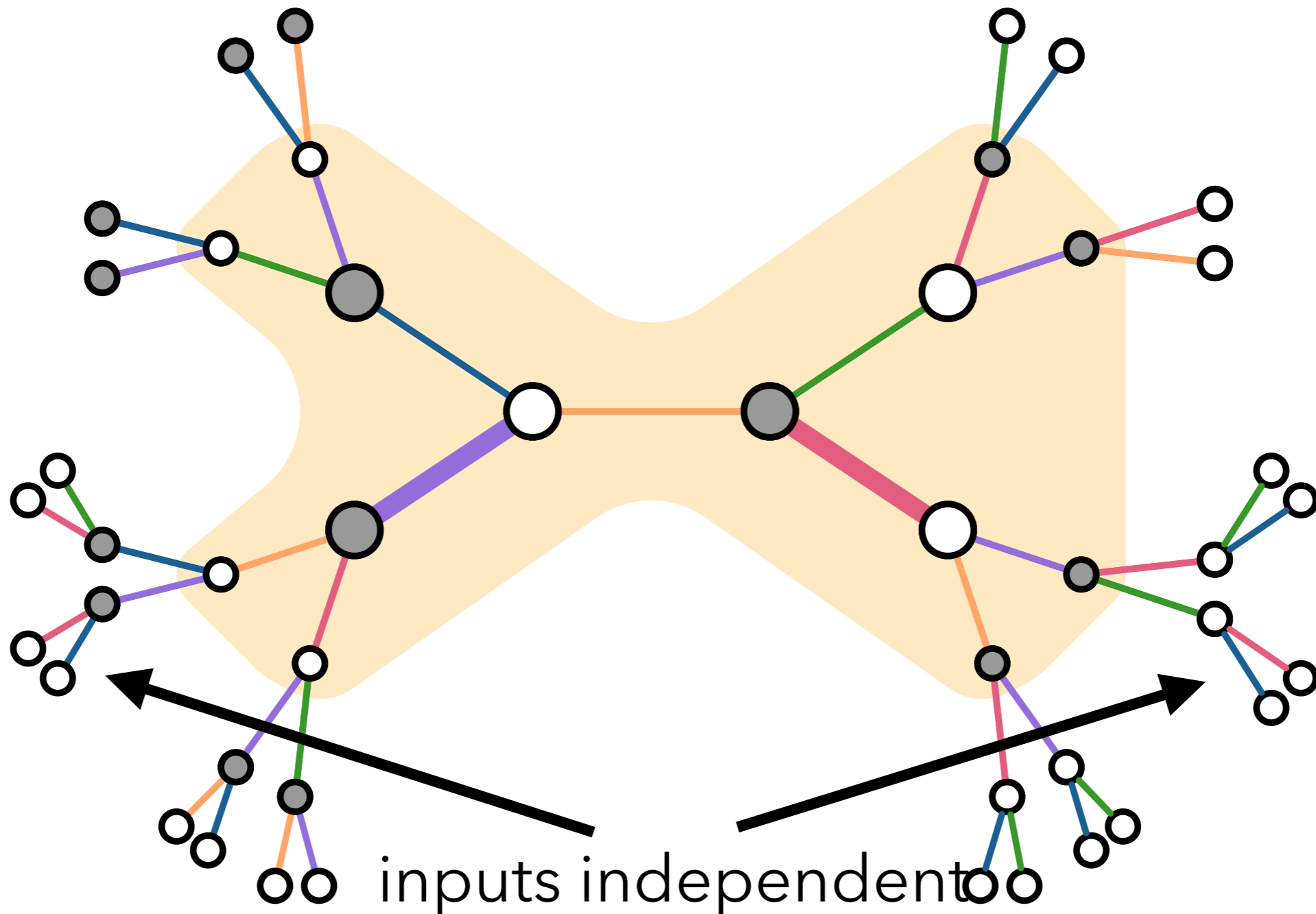
Outputs of incident edges

intersection of **3-neighbourhoods** = **2-neighbourhood** of



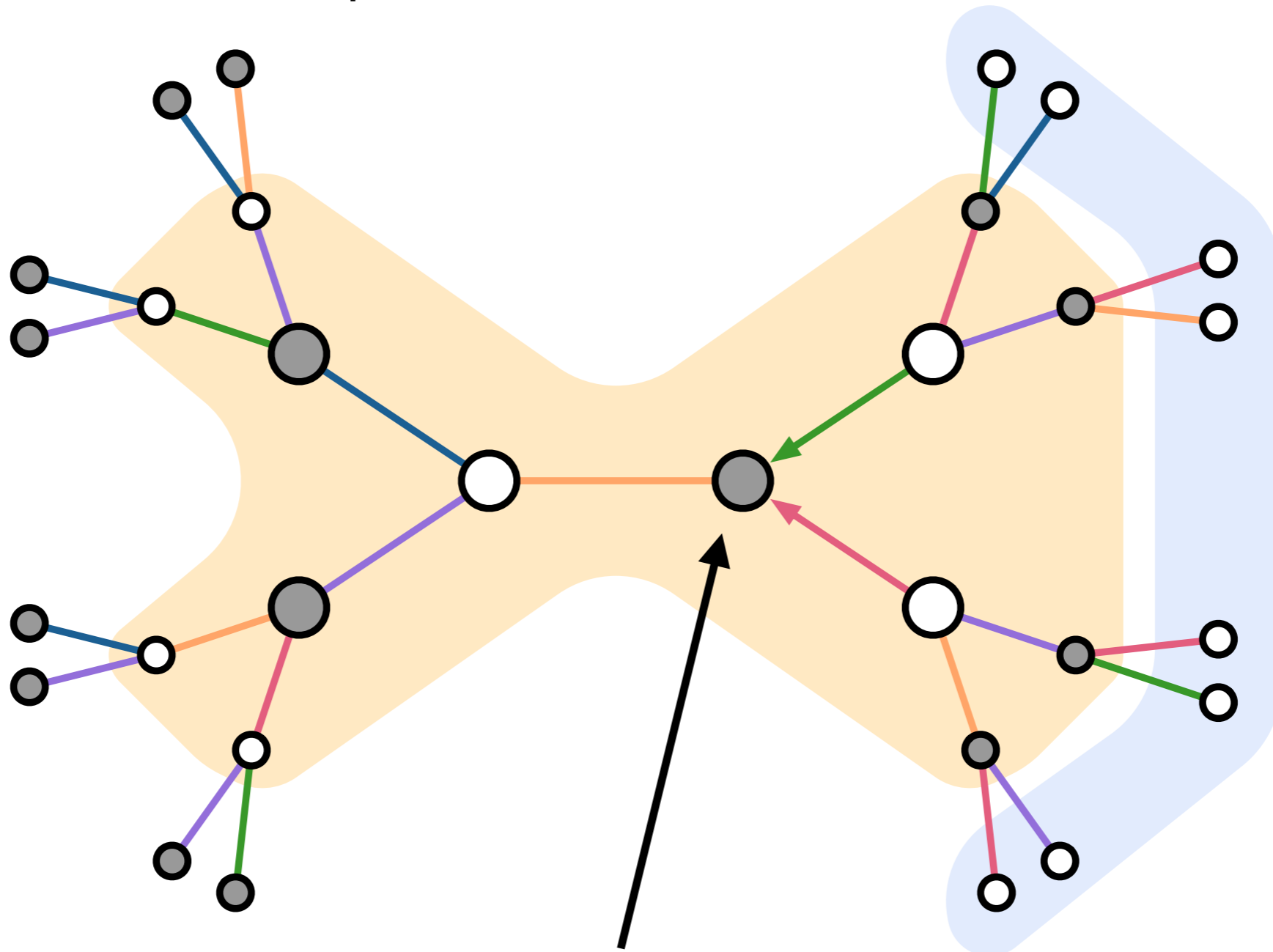
Outputs of incident edges

outputs on the two sides are **independent** given **orange**



Outputs of incident edges

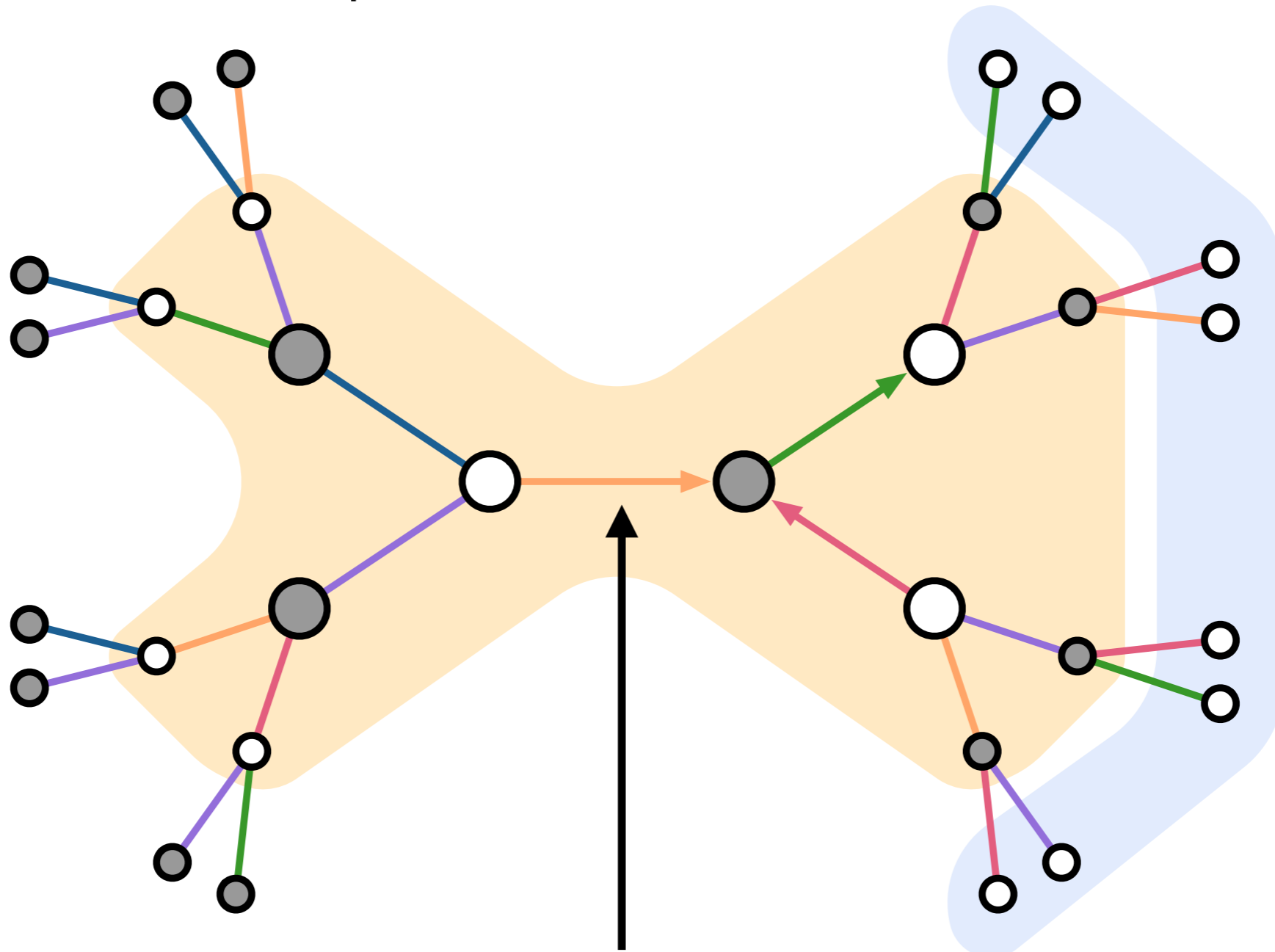
is it possible for endpoint to be a **sink** for the other edges?



\exists input s.t. other edges pointed towards node?

Outputs of incident edges

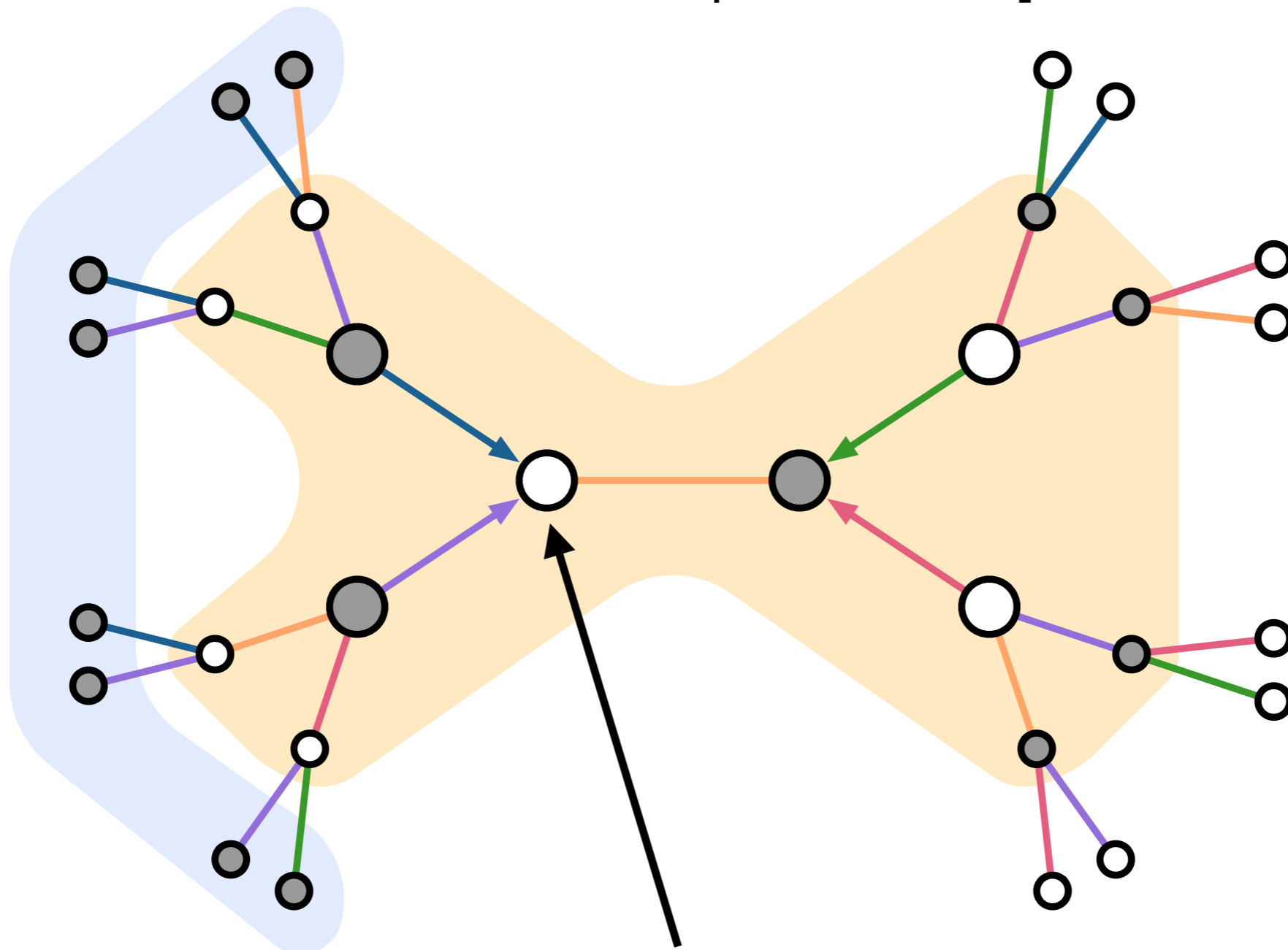
is it possible for endpoint to be a **sink** for the other edges?



if not, we can safely orient towards node

Other endpoint a sink

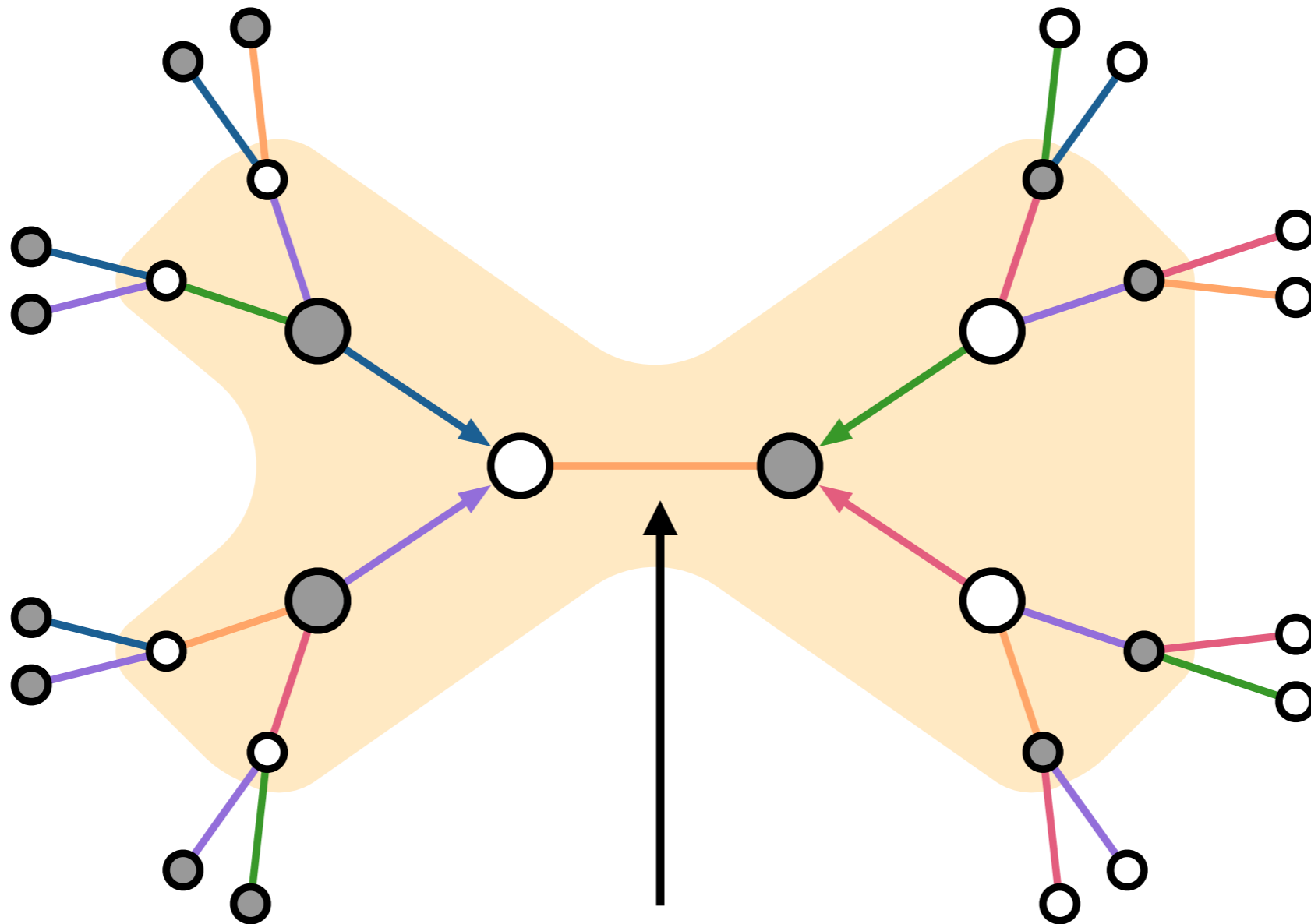
now assume the first endpoint is a **potential sink**



\exists input s.t. other edges pointed towards node?

Other endpoint a sink

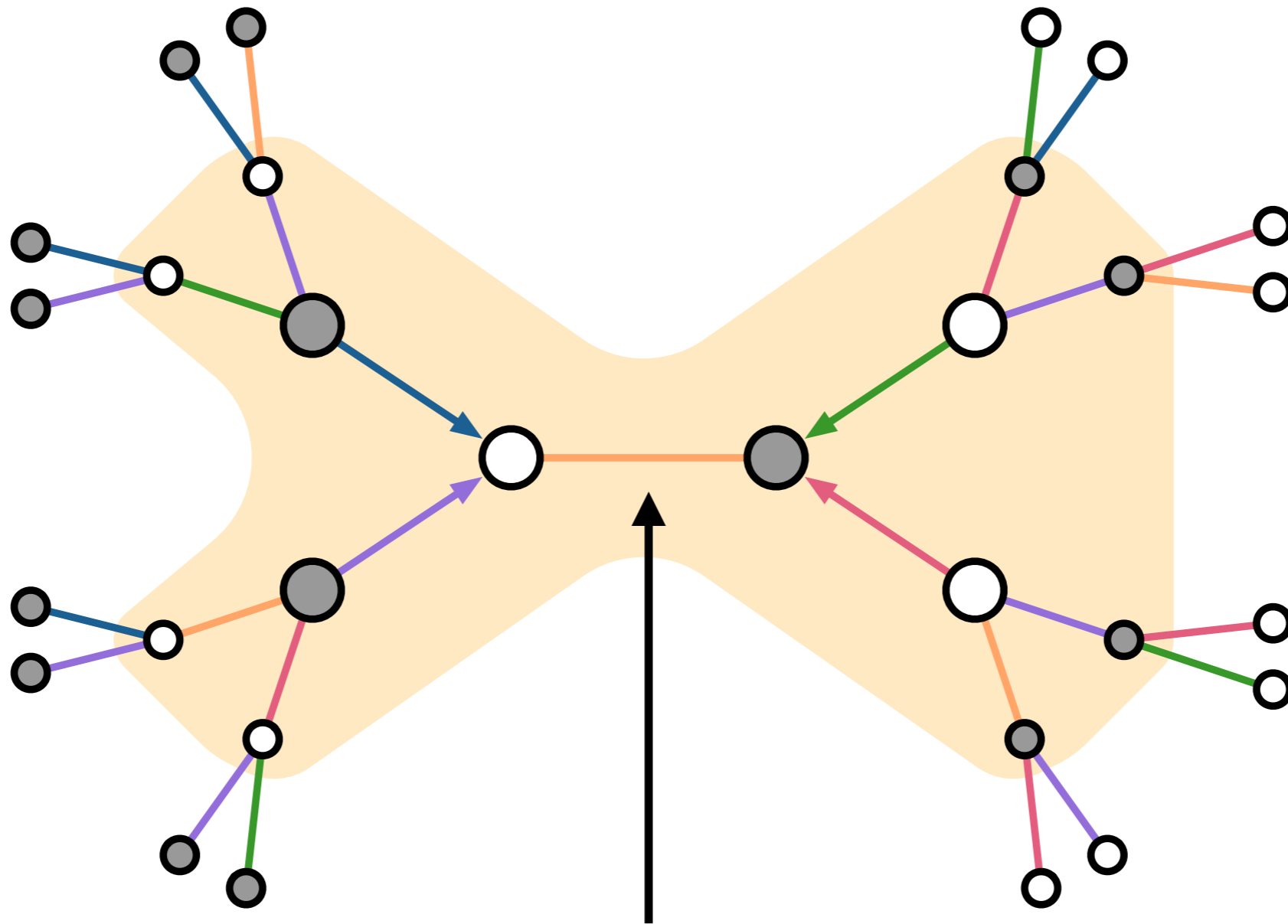
now assume both endpoints potential sinks



if yes, can engineer input such that this happens
(independence)

Other endpoint a sink

now assume both endpoints potential sinks



no feasible output left for middle edge

Lower bound: sinkless orientation

For example, assume $d=3$ and $c=5$

$$\mathbf{t} = (t, t, t, t, t)$$



speed up each color

$$\mathbf{t-1} = (t-1, t-1, t-1, t-1, t-1)$$



repeat \mathbf{t} times

$$\mathbf{0} = (0, 0, 0, 0, 0)$$

Problem with standard model

- Unique identifiers induce dependencies between possible inputs
- Argument that we can force a sink unless one endpoint is safe is no longer true

Roundabout solution: randomize

- Now consider the randomized setting
- In addition to the colouring, nodes have access to u.a.r. real number

Theorem: sinkless orientation requires
 $\Omega(\Delta^{-1} \log_{\Delta} \log n)$ rounds

Lower bound: updated strategy

For example, assume $d=3$ and $c=5$

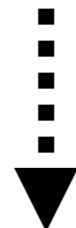
A : $\mathbf{t} = (t, t, t, t, t)$ *error with prob. $< p$*

↓ *speed up color 5 by simulation*

A' : $\mathbf{t}^{(1)} = (t, t, t, t, t-1)$ *error with prob. $< 3p^{1/3}$*

↓ *speed up color 4 by simulation*

A'' : $\mathbf{t}^{(2)} = (t, t, t, t-1, t-1)$



Lower bound: updated strategy

For example, assume $d=3$ and $c=5$

\mathbf{A}_t : $\mathbf{t} = (t, t, t, t, t)$ *error with prob. $< p$*

↓ *speed up each color*

\mathbf{A}_{t-1} : $\mathbf{t-1} = (t-1, t-1, t-1, t-1, t-1)$ *error with prob. $< \mathbf{O}(p^{-3^{(2d-1)}})$*

↓ *repeat \mathbf{t} times*

\mathbf{A}_0 : $\mathbf{0} = (0, 0, 0, 0, 0)$ *error with prob. $< \mathbf{O}(p^{-3^{(t(2d-1))}})$*

Lower bound: updated strategy

start with alg. **A**, running time **t**, error prob. **p₀**



algorithm **A'** with running time **0**, error prob. $< \mathbf{O}(p^{-3^{t(2d-1)}})$

0 rounds: must have error probability $\mathbf{p} > \mathbf{1/8^d}$



$$\mathbf{t} = \mathbf{\Omega}(\Delta^{-1} \log \log n)$$

Lower bound: updated strategy

For example, assume $d=3$ and $c=5$

A : $\mathbf{t} = (t, t, t, t, t)$ error with prob. $< p$

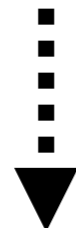


speed up a color by simulation

A' : $\mathbf{t}^{(1)} = (t, t, t, t, t-1)$ error with prob. $< 3p^{1/3}$

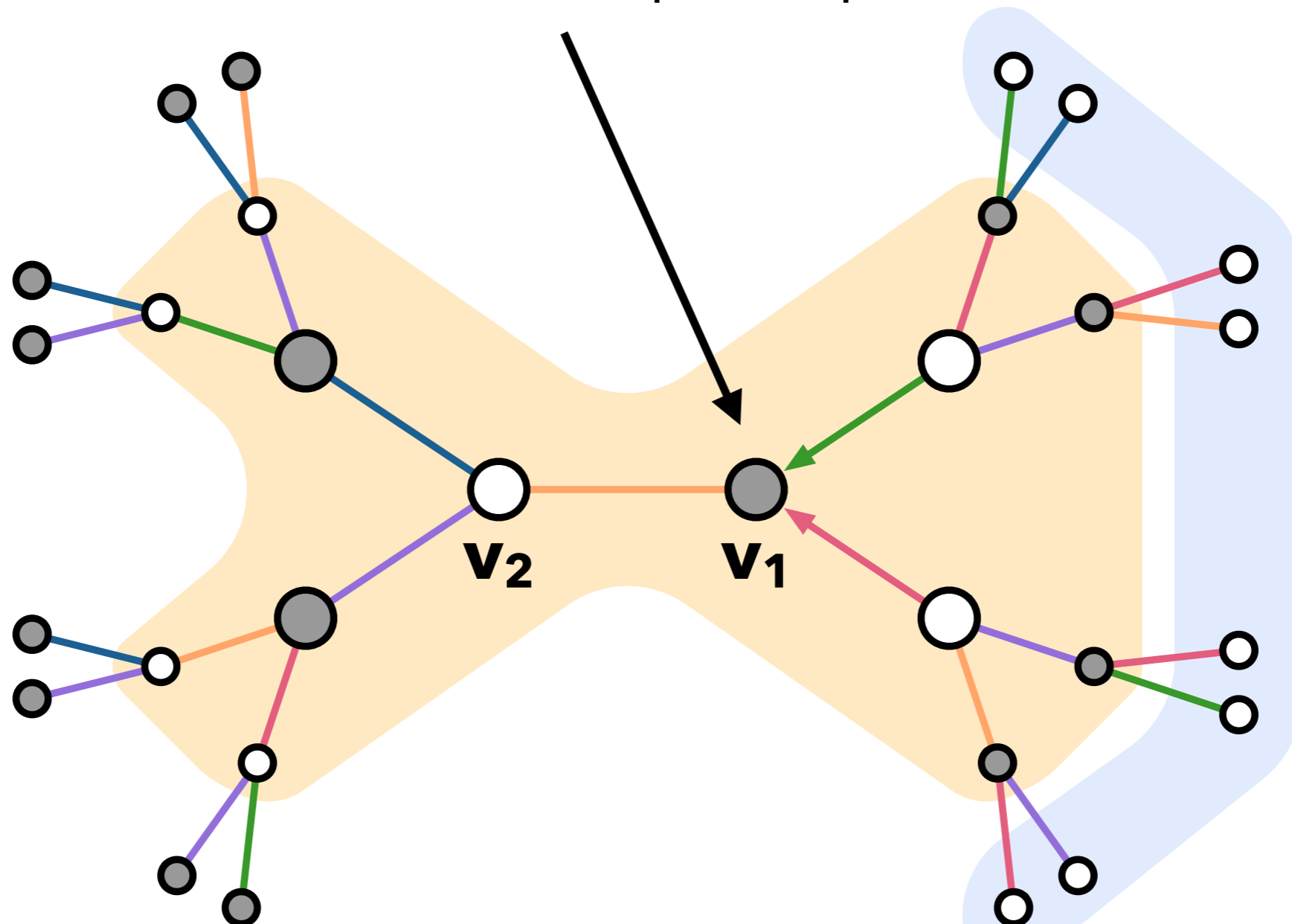


$\mathbf{t}^{(2)} = (t, t, t, t-1, t-1)$



Outputs of incident edges

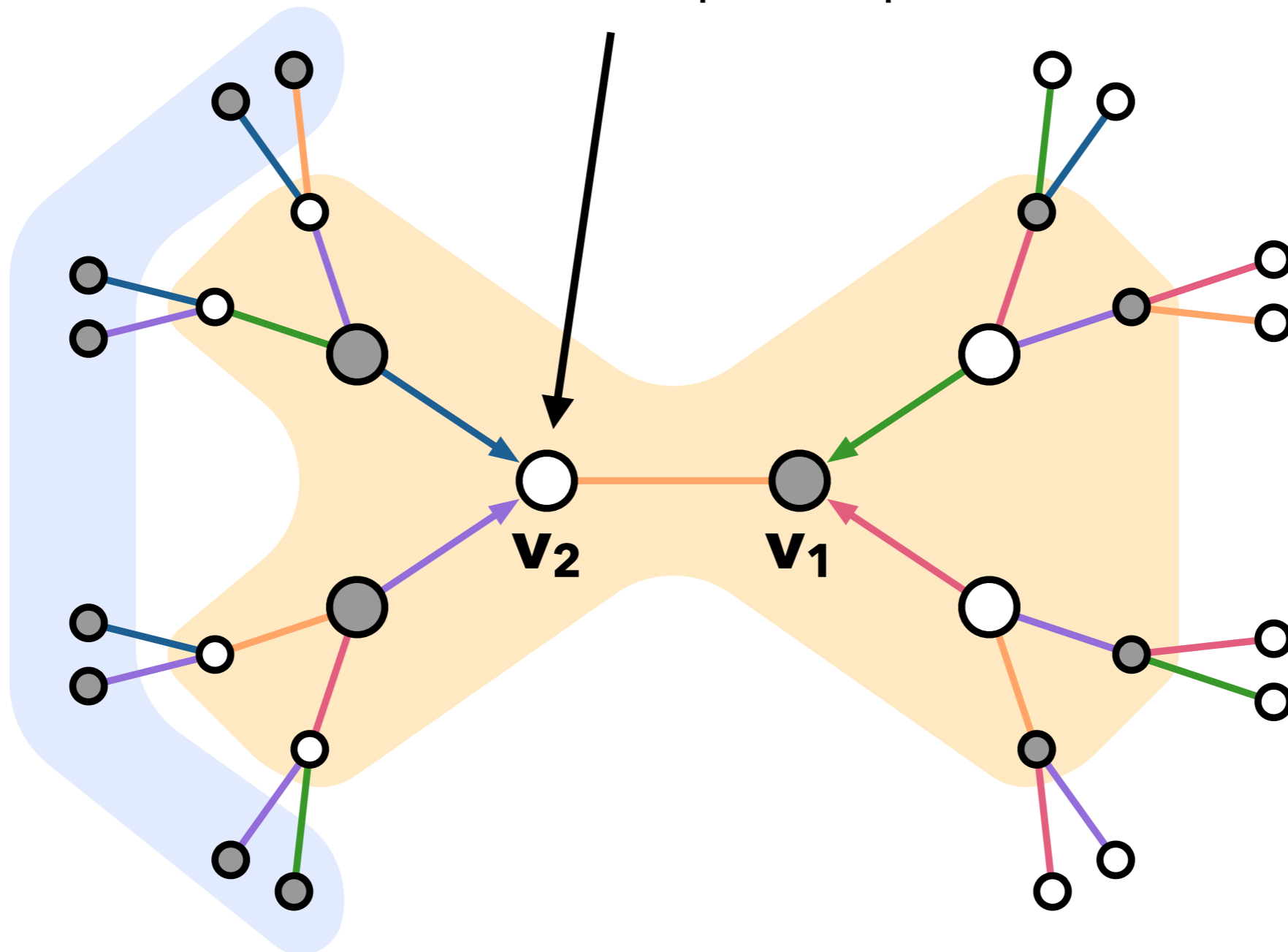
event \mathbf{E}_1 : black endpoint potential **sink**



random input s.t. other edges pointed toward
black endpoint

Outputs of incident edges

event \mathbf{E}_2 : white endpoint potential **sink**



Simulation

Compute probabilities for alg. **A**:

If \mathbf{v}_1 likely potential sink, orient \mathbf{e} from \mathbf{v}_1 to \mathbf{v}_2 and
otherwise from \mathbf{v}_2 to \mathbf{v}_1

(Remember other incident edges work
as in original algorithm)

Simulation

Compute probabilities for alg. **A**:

If \mathbf{v}_1 likely potential sink, orient \mathbf{e} from \mathbf{v}_1 to \mathbf{v}_2 and
otherwise from \mathbf{v}_2 to \mathbf{v}_1

Due to independence, both endpoints are usually not
potential sinks

Lower bound

A : $\mathbf{t} = (t, t, t, t, t)$ error with prob. $< \mathbf{p}$

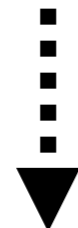


speed up a color by simulation

A' : $\mathbf{t}^{(1)} = (t, t, t, t, t-1)$ error with prob. $< \mathbf{3p}^{1/3}$



$\mathbf{t}^{(2)} = (t, t, t, t-1, t-1)$



Back to deterministic

Theorem (Chang et al., FOCS 2016): Assume that for LCL L there exists an algorithm with running time $t = o(\log_{\Delta} n)$, then there exists an algorithm with running time $t' = O(\log^* n)$



Corollary: sinkless orientation requires $\Omega(\log_{\Delta} n)$ deterministic time

Automatic speed-up

- Another **black box simulation**
- A given **algorithm A** is "fooled" to run faster: compute locally unique "identifiers" (a colouring) and run **A** on those
- Efficient solving of LCLs reduces to **coloring + constant time**

Back to randomized

Theorem (Chang et al., FOCS 2016): randomized complexity of an LCL on instances of **size n** is at least the deterministic complexity on instances of **size $(\log n)^{1/2}$**



Corollary: sinkless orientation requires **$\Omega(\log_{\Delta} \log n)$** randomized time

What just happened?

deterministic: $\Omega(\log_{\Delta} n)$
*Proof technique doesn't
work for identifiers*

automatic connection
randomized:
 $\Omega(\log \log n)$

IDs \rightarrow randomness
randomized:
 $\Omega(\Delta^{-1} \log \log n)$

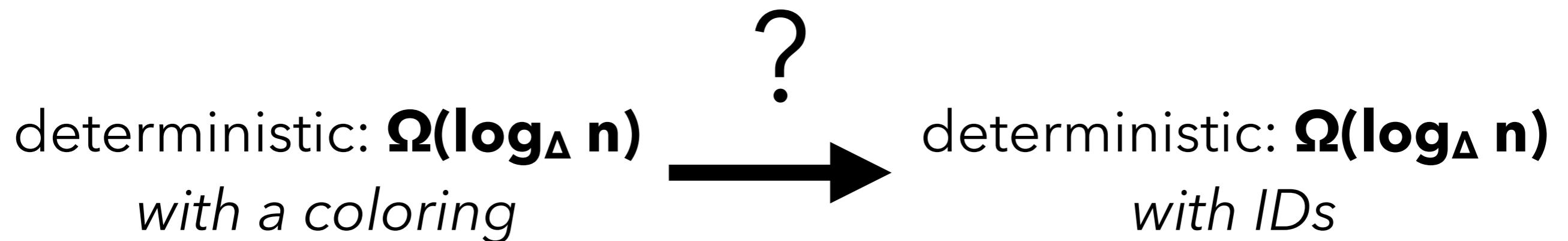


automatic speed-up
deterministic:
 $\Omega(\log n)$



Open questions

Trouble with IDs



Other applications?

A hammer but no nails?

- Identify other **intermediate** problems
- Lower bounds as **function of Δ (maximum degree)?**
- Best algorithms for maximal matching and maximal independent set **linear in Δ**
- Weaker lower bounds exist