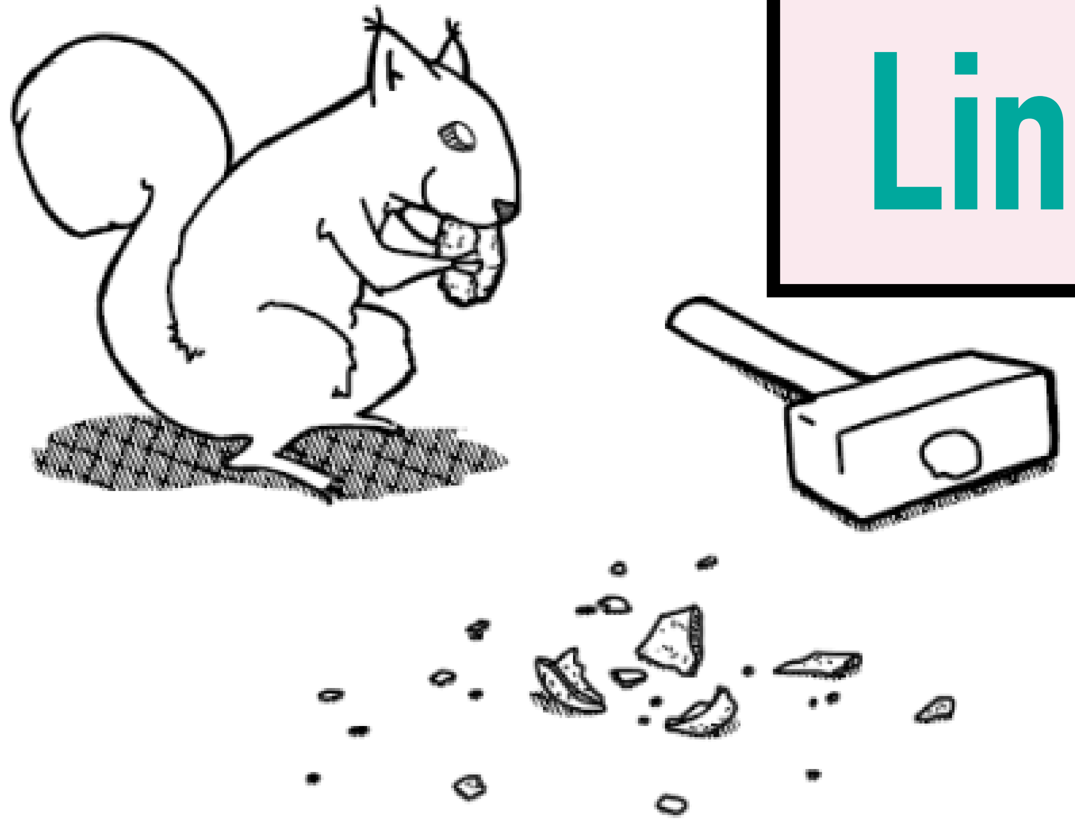
# Protrusion Decompositions Revisited:
# Uniform Lossy Kernels for Reducing Treewidth and Linear Kernels for Hitting Disconnected Minors



**Roohani Sharma**
DIMAG, Institute for Basic Science, South Korea

Michał Włodarczyk
University of Warsaw, Poland

43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026)

March 11, 2026

*Pictures of squirrels by Felix Reidl

# Polynomial Kernelization
# for
# $\mathcal{F}$-MINOR-FREE DELETION

(when $\mathcal{F}$ contains a planar graph)

- What is known?

  Polynomial kernels, tight lower bounds

- Our contribution

  Bypass the lower bounds at the expense of an arbitrary small loss in solution quality.

# Problem Definition

**$\mathscr{F}$-MINOR-FREE DELETION** Here $\mathscr{F}$ is a fixed finite family of graphs.

**Input:** A graph $G$, a positive integer $k$

**Output:** Does there exist a set $S \subseteq V(G)$ of size at most $k$
such that $G - S$ is $\mathscr{F}$-free
(that is has no graph from $\mathscr{F}$ as a minor)?

$\{K_2\}$-MINOR-FREE DELETION = Vertex Cover/ Deletion to Treewidth-0

$\{K_3\}$-MINOR-FREE DELETION = Feedback Vertex Set/ Deletion to Treewidth-1

$\{K_4\}$-MINOR-FREE DELETION = Deletion to Treewidth-2

$\{K_5, X_{126}, X_{174}, 3K_2\}$-MINOR-FREE DELETION = Deletion to Treewidth-3

…

$\mathscr{F}_t^{tw}$-MINOR-FREE DELETION = Deletion to Treewidth-t for some $\mathscr{F}_t^{tw}$

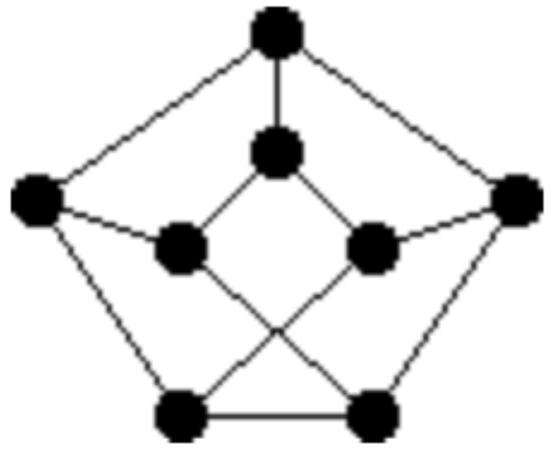
$\mathscr{F}_t^{pw}$-MINOR-FREE DELETION = Deletion to Pathwidth-t for some $\mathscr{F}_t^{pw}$

$\mathscr{F}_t^{td}$-MINOR-FREE DELETION = Deletion to Treedepth-t for some $\mathscr{F}_t^{td}$

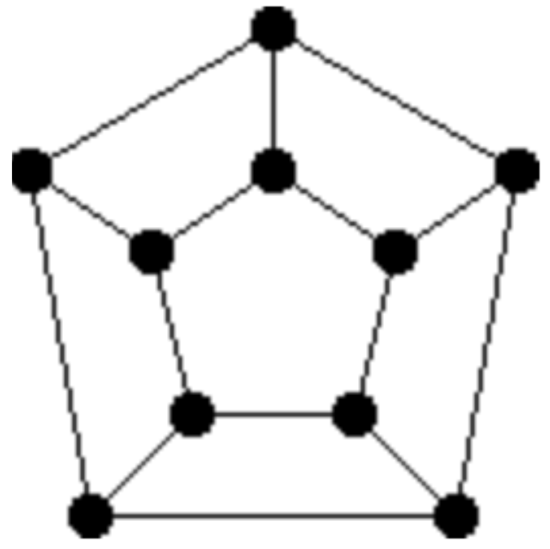
$\{K_{3,3}, K_5\}$-MINOR-FREE DELETION = Deletion to Planarity

…

$X_{126}$



$X_{174}$



# Kernelization

## Can we shrink the graph, in polynomial time, to one of size f(k)?

Kernelization algorithm for a parameterized problem $\Pi$:

**Input**: $(G, k)$
**Time**: polynomial in $|G|$
**Output**: $(G', k')$
such that $(G, k) \equiv (G', k')$ for $\Pi$
and $|G'| \leq f(k)$, $k' \leq k$, for some function $f$.

- $f(k)$ is called the size of the kernel.
- If $f(k)$ is a polynomial function of $k$, then the problem $\Pi$ is said to have a polynomial kernel.

# $\mathscr{F}$-MINOR-FREE DELETION: Two worlds

- When $\mathscr{F}$ contains a planar graph, then for any $\mathscr{F}$-minor-free deletion solution $S$, $\text{tw}(G - S) = O_{\mathscr{F}}(1)$ (because of the grid minor theorem).

  Prototypical (class of) problems in this case is **DELETION TO TREEWIDTH-t.**

- When $\mathscr{F}$ contains no planar graph, the polynomial kernelization complexity of **$\mathscr{F}$-MINOR-FREE DELETION** remains wide open.

  The simplest open case is when $\mathscr{F} = \{K_{3,3}, K_5\}$and the corresponding problem is **DELETION TO PLANARITY.**

# $\mathscr{F}$-MINOR-FREE DELETION when $\mathscr{F}$ contains a planar graph
# (DELETION TO TREEWIDTH-$t$)

**DELETION TO TREEWIDTH**-0 = **VERTEX COVER** = Deletion to graphs without edges
**DELETION TO TREEWIDTH**-1 = **FEEDBACK VERTEX SET** = Deletion to graphs without cycles

Both **VERTEX COVER** and **FEEDBACK VERTEX SET** have a kernel of size $O(k^2)$, which is also tight, unless NP ⊆ coNP/poly.

*extensively studied problems in kernelization*

# $\mathscr{F}$-Minor-Free Deletion when $\mathscr{F}$ contains a planar graph
# (Deletion To Treewidth-$t$)

**Deletion To Treewidth-0** = **Vertex Cover** = Deletion to graphs without edges
**Deletion To Treewidth-1** = **Feedback Vertex Set** = Deletion to graphs without cycles

Both **Vertex Cover** and **Feedback Vertex Set** have a kernel of size $O(k^2)$, which is also tight, unless NP ⊆ coNP/poly.

*extensively studied problems in kernelization*

Fomin, Lokshtanov, Misra, Saurabh [FOCS 2012]

**Deletion To Treewidth-$t$**

(**$\mathscr{F}$-Minor-Free Deletion** when $\mathscr{F}$ contains a planar graph )

polynomial kernel of size

$$g(t) \cdot k^{O(t)}$$

*non-uniform kernel*

# $\mathcal{F}$-MINOR-FREE DELETION when $\mathcal{F}$ contains a planar graph (DELETION TO TREEWIDTH-$t$)

**DELETION TO TREEWIDTH**-0 = **VERTEX COVER** = Deletion to graphs without edges
**DELETION TO TREEWIDTH**-1 = **FEEDBACK VERTEX SET** = Deletion to graphs without cycles

Both **VERTEX COVER** and **FEEDBACK VERTEX SET** have a kernel of size $O(k^2)$, which is also tight, unless NP ⊆ coNP/poly.

*extensively studied problems in kernelization*

Fomin, Lokshtanov, Misra, Saurabh [FOCS 2012]
**DELETION TO TREEWIDTH-$t$**
(**$\mathcal{F}$-MINOR-FREE DELETION** when $\mathcal{F}$ contains a planar graph )
polynomial kernel of size

$$g(t) \cdot k^{O(t)}$$

$$g(t) \cdot k^{100} \; ?$$

*non-uniform kernel*

# $\mathscr{F}$-MINOR-FREE DELETION when $\mathscr{F}$ contains a planar graph
# (DELETION TO TREEWIDTH-$t$)

**DELETION TO TREEWIDTH**-0 = **VERTEX COVER** = Deletion to graphs without edges
**DELETION TO TREEWIDTH**-1 = **FEEDBACK VERTEX SET** = Deletion to graphs without cycles

Both **VERTEX COVER** and **FEEDBACK VERTEX SET** have a kernel of size $O(k^2)$, which is also tight, unless NP ⊆ coNP/poly.

*extensively studied problems in kernelization*

Fomin, Lokshtanov, Misra, Saurabh [FOCS 2012]
**DELETION TO TREEWIDTH**-$t$
($\mathscr{F}$-**MINOR-FREE DELETION** when $\mathscr{F}$ contains a planar graph )
polynomial kernel of size

$$g(t) \cdot k^{O(t)}$$

*non-uniform kernel*

$$g(t) \cdot k^{100} \ ?$$

Giannapoulou, Jansen, Lokshtanov, Saurabh [ACM TALG 2017]
**DELETION TO TREEWIDTH**-$t$
unless NP ⊆ coNP/poly, no kernel of size

$$g(t) \cdot k^{\frac{t}{2}-\delta} \quad \text{for any } \delta > 0$$

*non-uniform size is unavoidable!*

# $\mathscr{F}$-MINOR-FREE DELETION when $\mathscr{F}$ contains a planar graph :
## Restricted to input graphs that are $H$-topological-minor-free

Kim, Langer, Paul, Reidl, Rossmanith, Sau, Sikdar [ACM TALG 2015]
**$\mathscr{F}$-MINOR-FREE DELETION**
when $\mathscr{F}$ contains a planar graph,
**all graphs in $\mathscr{F}$ are connected**
**and the input graph $G$ is $H$-topological-minor-free for some graph $H$,**
has a linear-sized kernel.

$$O_{H,\mathscr{F}}(k)$$

# $\mathcal{F}$-MINOR-FREE DELETION when $\mathcal{F}$ contains a planar graph :

## Restricted to input graphs that are $H$-topological-minor-free

Kim, Langer, Paul, Reidl, Rossmanith, Sau, Sikdar [ACM TALG 2015]
$\mathcal{F}$-MINOR-FREE DELETION
when $\mathcal{F}$ contains a planar graph,
~~all graphs in $\mathcal{F}$ are connected~~
**and the input graph $G$ is $H$-topological-minor-free for some graph $H$,**
has a linear-sized kernel.

Our result 1

$$O_{H,\mathcal{F}}(k)$$

For example,
**DELETION TO EMBEDABILITY ON TORUS** is an $\mathcal{F}$-**MINOR-FREE DELETION** problem
where $\mathcal{F}$ contains the disconnected graph $2K_5$.

# $\mathcal{F}$-MINOR-FREE DELETION when $\mathcal{F}$ contains a planar graph

# (DELETION TO TREEWIDTH-$t$)

DELETION TO TREEWIDTH-0 = VERTEX COVER = Deletion to graphs without edges

DELETION TO T

Can we avoid
the non-uniformity in the polynomial kernel size,
at the expense of only preserving approximate solutions?

Both VERTE

which is also

blems in kernelization

Fomin, Loksl

DELETION TO TREEWIDTH-$t$

($\mathcal{F}$-MINOR-FREE DELETION when $\mathcal{F}$ contains a planar graph)

polynomial kernel of size

$$g(t) \cdot k^{O(t)}$$

$$g(t) \cdot k^{100} \ ?$$

*non-uniform kernel*

Giannapoulou, Jansen, Lokshtanov, Saurabh [ACM TALG 2017]

DELETION TO TREEWIDTH-$t$

unless NP $\subseteq$ coNP/poly, no kernel of size

$$g(t) \cdot k^{\frac{t}{2}-\delta}$$ for any $\delta > 0$

*non-uniform size is unavoidable!*

# How good solutions can we obtain in polynomial time?

## Polynomial-time approximation

**DELETION TO TREEWIDTH**-$0$ = **VERTEX COVER**
**DELETION TO TREEWIDTH**-$1$ = **FEEDBACK VERTEX SET**

- Both **VERTEX COVER** and **FEEDBACK VERTEX SET** admit a $2$-approximation algorithm in polynomial time.
- No $(2 - \delta)$-approximation in polynomial time, for any $\delta > 0$, under the Unique Games Conjecture.
- **DELETION TO TREEWIDTH**-$t$ admits an $O(\log t)$-approximation in polynomial time by Gupta, Lee, Li, Manurangsi, Włodarczyk [SODA 2019].

*classical results in polynomial-time approximation*

# How good solutions can we obtain in polynomial time?

## Polynomial-time approximation

**DELETION TO TREEWIDTH**-$0$ = **VERTEX COVER**
**DELETION TO TREEWIDTH**-$1$ = **FEEDBACK VERTEX SET**

- Both **VERTEX COVER** and **FEEDBACK VERTEX SET** admit a $2$-approximation algorithm in polynomial time.
- No $(2-\delta)$-approximation in polynomial time, for any $\delta > 0$, under the Unique Games Conjecture.
- **DELETION TO TREEWIDTH**-$t$ admits an $O(\log t)$-approximation in polynomial time by Gupta, Lee, Li, Manurangsi, Włodarczyk [SODA 2019].

*classical results in polynomial-time approximation*

Can we get a $2$-approximate solution for **DELETION TO TREEWIDTH**-$t$ using uniform polynomial kernelization?

Can we get a $(1+\epsilon)$-approximate solution for **DELETION TO TREEWIDTH**-$t$ using uniform polynomial kernelization, for any $\epsilon > 0$?

# How good solutions can we obtain in polynomial time?

## Polynomial-time approximation

**DELETION TO TREEWIDTH**-$0$ = **VERTEX COVER**
**DELETION TO TREEWIDTH**-$1$ = **FEEDBACK VERTEX SET**

- Both **VERTEX COVER** and **FEEDBACK VERTEX SET** admit a $2$-approximation algorithm in polynomial time.
- No $(2-\delta)$-approximation in polynomial time, for any $\delta > 0$, under the Unique Games Conjecture.
- **DELETION TO TREEWIDTH**-$t$ admits an $O(\log t)$-approximation in polynomial time by Gupta, Lee, Li, Manurangsi, Włodarczyk [SODA 2019].

*classical results in polynomial-time approximation*

Can we get a $2$-approximate solution for **DELETION TO TREEWIDTH**-$t$ using uniform polynomial kernelization?

Our result 2: Yes!

Can we get a $(1+\epsilon)$-approximate solution for **DELETION TO TREEWIDTH**-$t$ using uniform polynomial kernelization, for any $\epsilon > 0$?

Our result 3: 'Yes!'

# Technical overview

Special case of Our Result 2:

**Deletion To Treewidth-$t$**

admits a $2$-approximate kernelization

of size (uniform polynomial) $\cancel{g(t) \cdot k^6}$. $O(vc^3)$ vertices

$vc$ denotes the size of a minimum vertex cover of the input graph $G$.

$$k \leq vc$$

Remark: The non-uniform polynomial kernel lower bound also holds with the vertex cover parameterization.

# Technical overview

Special case of Our Result 2:

**DELETION TO TREEWIDTH-$t$**

admits a $2$-approximate kernelization

of size (uniform polynomial) ~~$g(t) \cdot k^6$~~. $O(vc^3)$ vertices

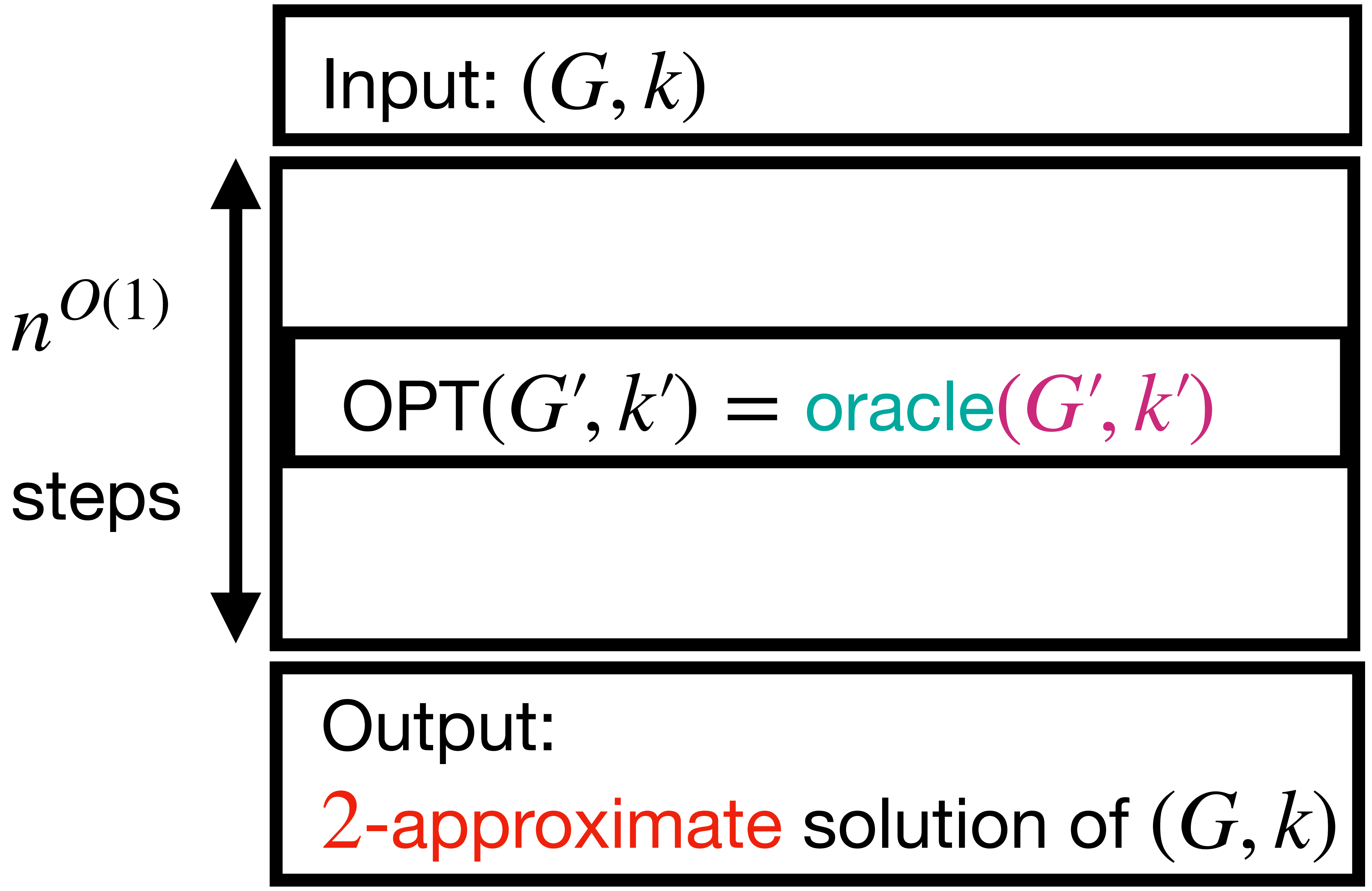
$vc$ denotes the size of a minimum vertex cover of the input graph $G$.

$$k \leq vc$$

Remark: The non-uniform polynomial kernel lower bound also holds with the vertex cover parameterization.

2-approximate kernelization

In polynomial time,
output a 2-approximate solution to the instance $(G, k)$,
if there exists an optimum solution of size at most $k$,
by calling an oracle once.
Oracle takes an instance of size ~~$g(t) \cdot k^{O(1)}$~~ $O(vc^3)$ vertices
and outputs an optimum solution of this instance in unit time.



Input: $(G, k)$

$n^{O(1)}$

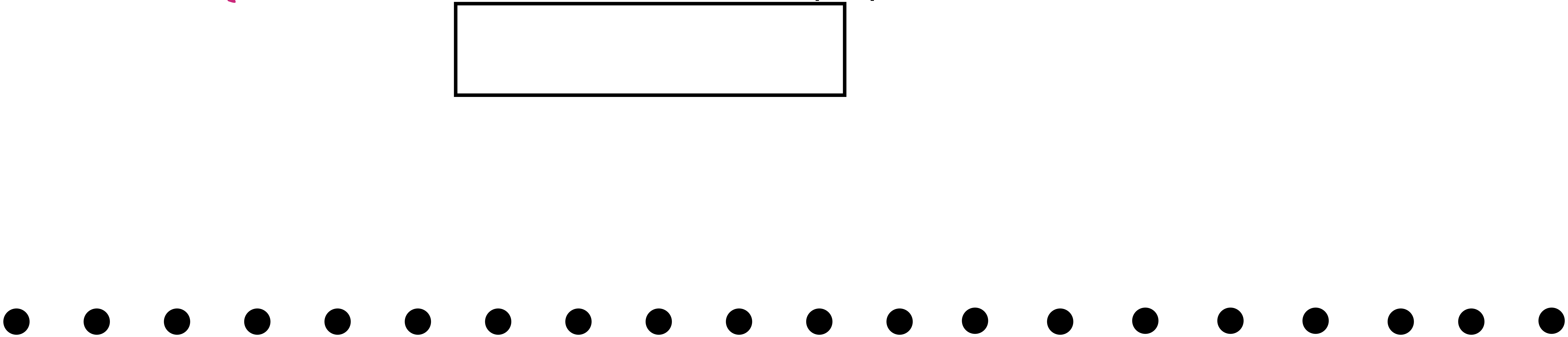
steps

OPT$(G', k')$ = oracle$(G', k')$

Output:
$2$-approximate solution of $(G, k)$

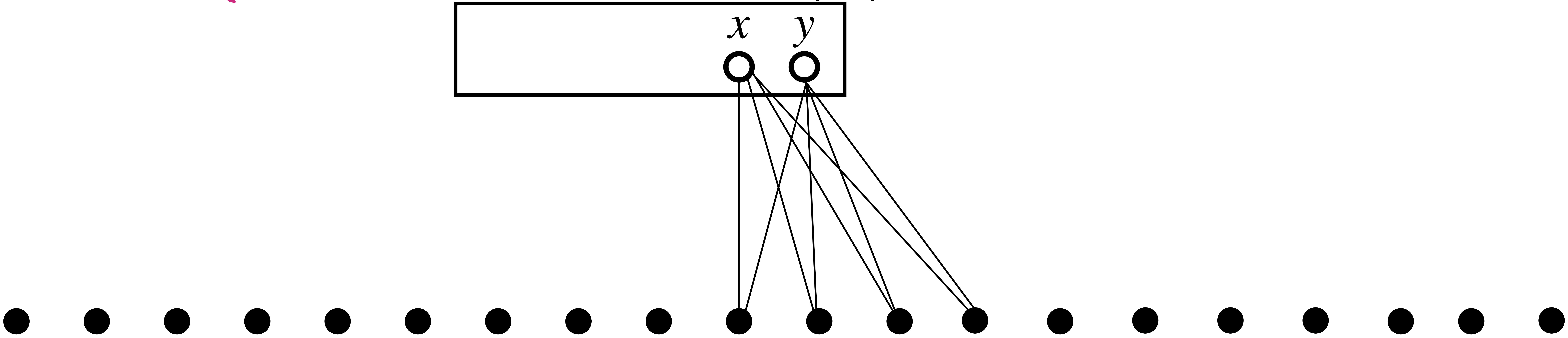
# The 2-approximate kernelization algorithm for Deletion to Treewidth−t/vc

**Step 1:** 2-approximate vertex cover, $|X| \leq 2 \cdot \text{vc}$



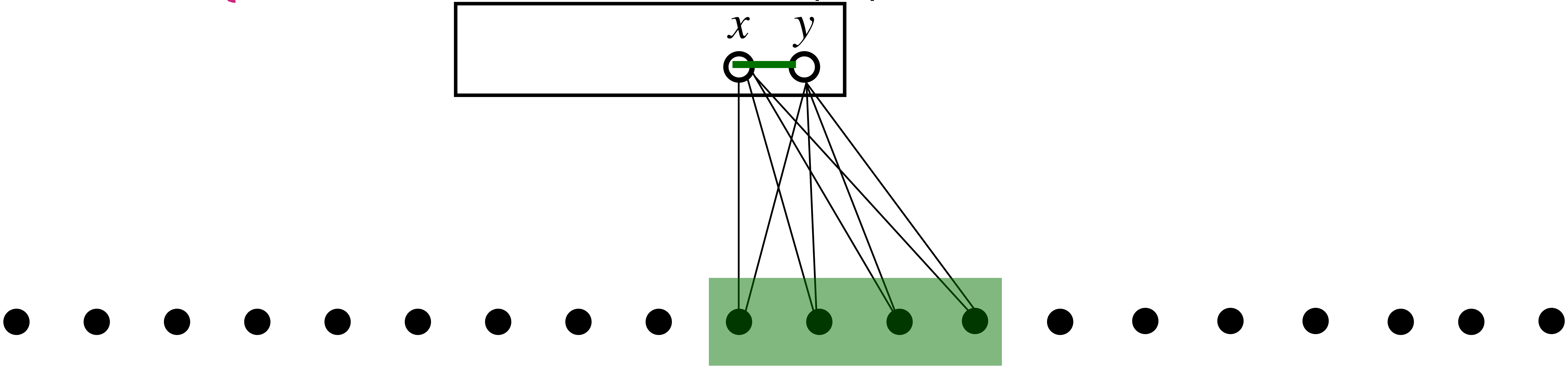
# The 2-approximate kernelization algorithm for DELETION TO TREEWIDTH−t/vc

**Step 1:** 2-approximate vertex cover, $|X| \leq 2 \cdot \text{vc}$



# The 2-approximate kernelization algorithm for DELETION TO TREEWIDTH−t/vc

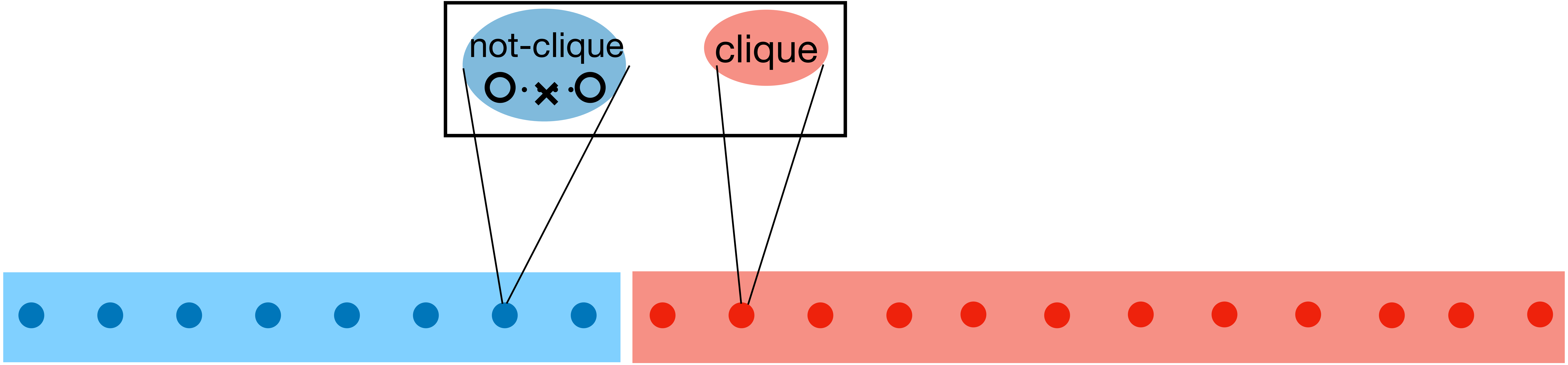
**Step 1:** 2-approximate vertex cover, $|X| \leq 2 \cdot \text{vc}$



**Step 2:** If $|N(x) \cap N(y)| \geq k + t + 2$,
then it is safe to add an edge between $x$ and $y$.

# The 2-approximate kernelization algorithm for DELETION TO TREEWIDTH-t/vc

2-approximate vertex cover, $|X| \leq 2 \cdot \text{vc}$



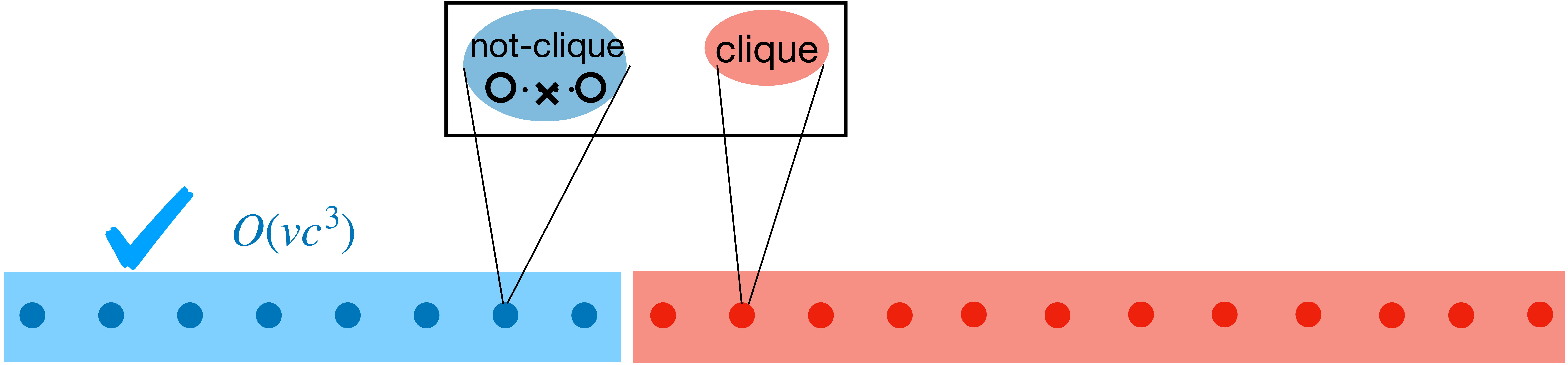
Partition $V(G) - X$ into two sets: blue and red

Blue: vertices whose neighborhood is not a clique

Red: vertices whose neighborhood is a clique

# The 2-approximate kernelization algorithm for Deletion to Treewidth−t/vc

2-approximate vertex cover, $|X| \leq 2 \cdot \text{vc}$



Partition $V(G) - X$ into two sets: blue and red

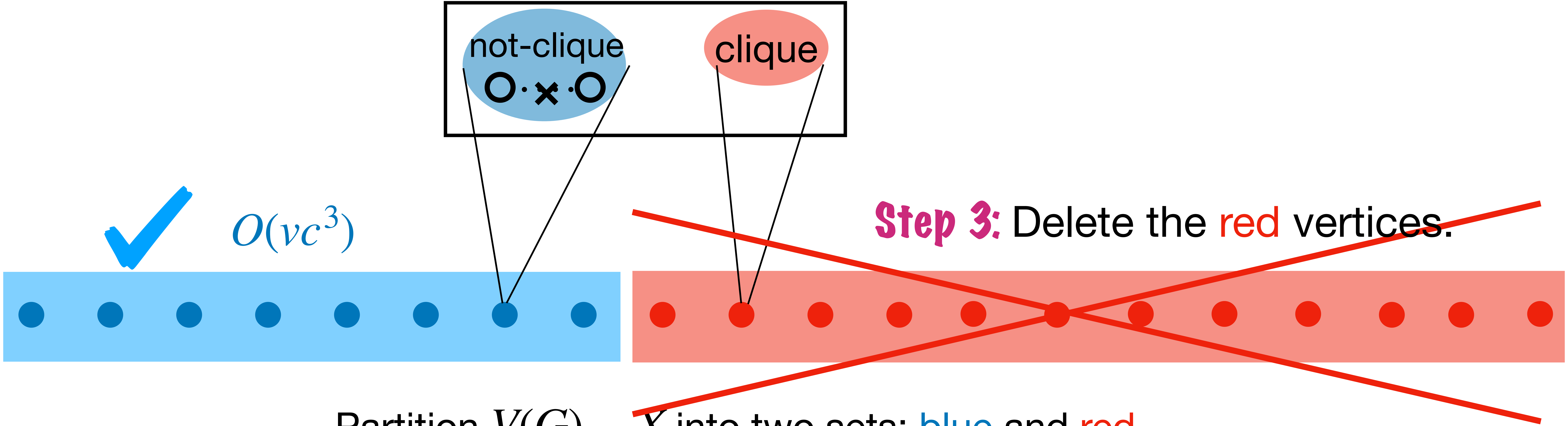
Blue: vertices whose neighborhood is not a clique

Red: vertices whose neighborhood is a clique

$$\boxed{\text{Number of blue vertices is } \binom{|X|}{2} \cdot (k + t + 1) = O(vc^3)}$$

# The 2-approximate kernelization algorithm for DELETION TO TREEWIDTH−t/vc

2-approximate vertex cover, $|X| \leq 2 \cdot \text{vc}$



$O(vc^3)$

**Step 3:** Delete the red vertices.

Partition $V(G) - X$ into two sets: blue and red

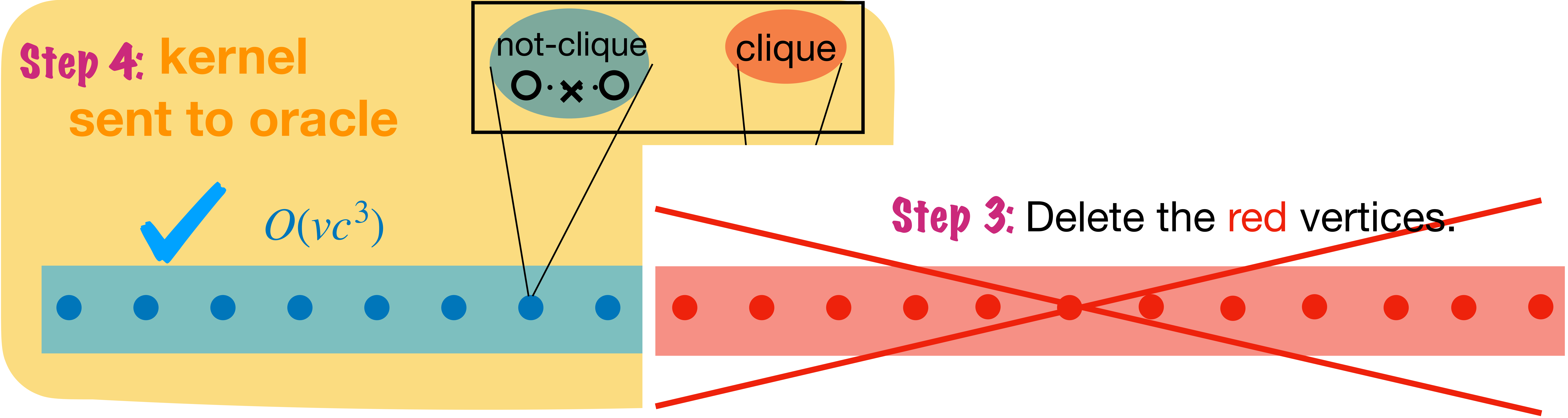
Blue: vertices whose neighborhood is not a clique

Red: vertices whose neighborhood is a clique

Number of blue vertices is $\binom{|X|}{2} \cdot (k + t + 1) = O(vc^3)$

# The 2-approximate kernelization algorithm for DELETION TO TREEWIDTH−t/vc

2-approximate vertex cover, $|X| \leq 2\text{vc}$



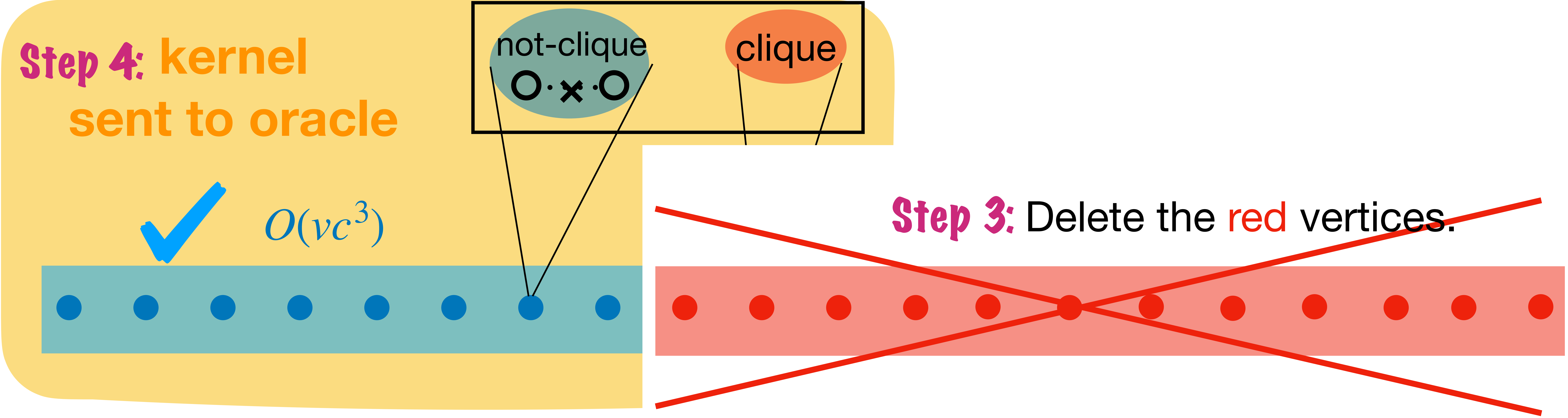
**Step 4:** kernel sent to oracle

✓ $O(vc^3)$

**Step 3:** Delete the red vertices.

# The 2-approximate kernelization algorithm for DELETION TO TREEWIDTH−t/vc

2-approximate vertex cover, $|X| \leq 2$vc



**Step 4:** kernel sent to oracle

**Step 3:** Delete the red vertices.

- Let $G' := G-$red.

**Step 5:** Let $S' :=$ oracle$(G', k)$. That is, tw$(G' - S') \leq t$.

- Claim: tw$(G - S') \leq t + 1$.

**Step 6:** Solve the problem optimally on $G - S'$ in linear time using Courcelle's theorem. Let $S''$ be an optimum solution of $G - S'$.

**Step 7:**
- Output: $S' \cup S''$

# Ideas for 2-approximate kernelization for k parameter

**DELETION TO TREEWIDTH-$t$**
admits a $2$-approximate kernelization
of size (uniform polynomial) $g(t) \cdot k^6$
by calling the oracle once.

1. Start by finding an $O(\log t)$-approximate solution $X$ in polynomial time. $|X| = O_t(k)$.

2. Bound the number of connected components of $G - X$ using ideas discussed for vertex cover parameterization.

3. To bound the size of connected components

- compute a near-protrusion decomposition,
- design reduction rules that preserve an approximate solution leading to a protrusion decompositions,
- then use protrusion replacers.

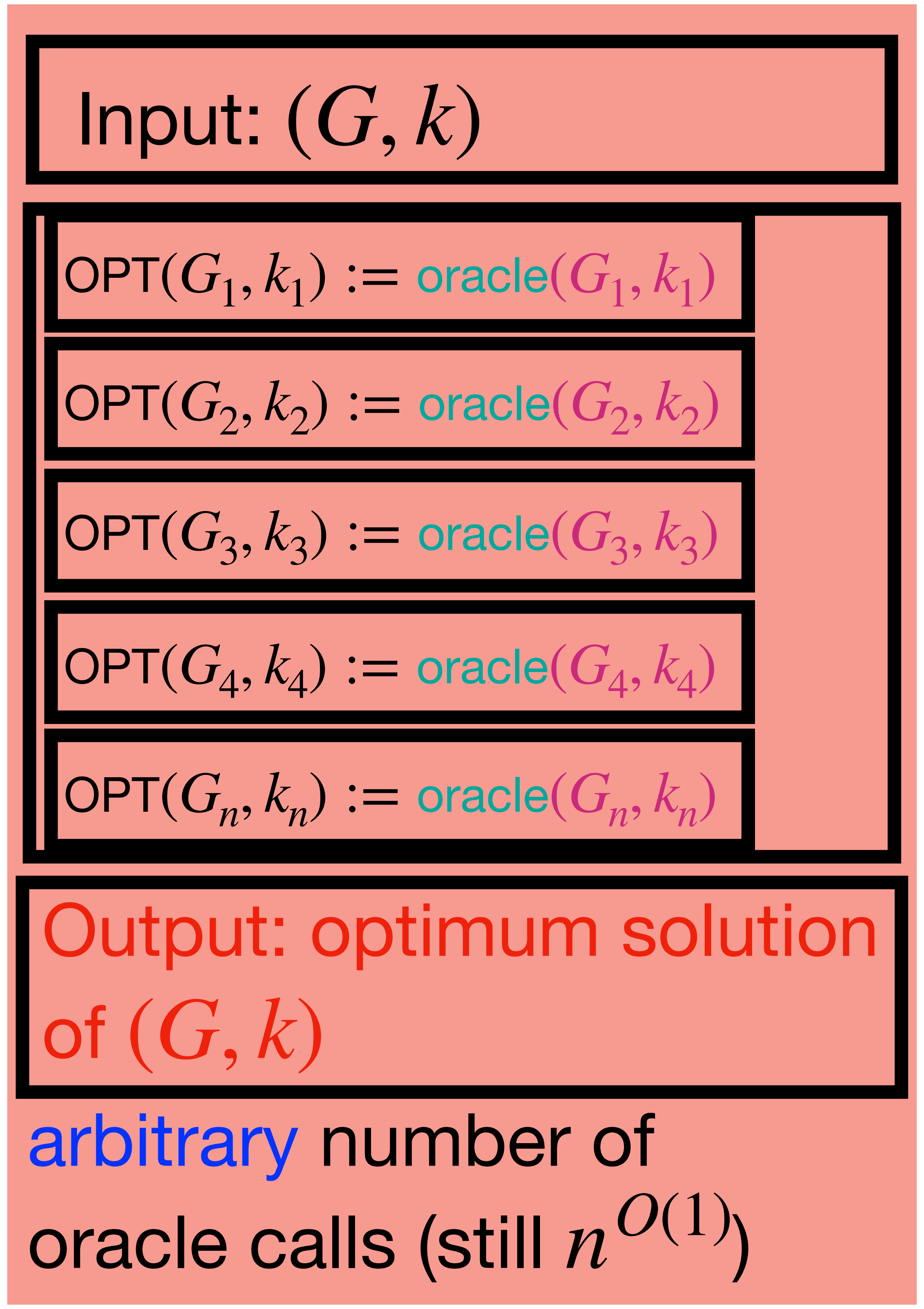
# Summary: DELETION TO TREEWIDTH$-t$ polynomial kernelization landscape

| | | | | |
|---|---|---|---|---|
| | Input: $(G, k)$ | Input: $(G, k)$ | Input: $(G, k)$ | Input: $(G, k)$ |
| $n^{O(1)}$ steps | | OPT$(G', k')$ := oracle$(G', k')$ | OPT$(G', k')$ := oracle$(G', k')$ | OPT$(G', k')$ := oracle$(G', k')$<br>OPT$(G', k')$ := oracle$(G', k')$ |
| | Output: $O(\log t)$-approximate solution of $(G, k)$ | Output: optimal solution of $(G, k)$ | Output: $2$-approximate solution of $(G, k)$ | Output: $(1 + \epsilon)$-approximate solution of $(G, k)$ |
| | # oracle calls: 0<br>NP-hard<br>No $(2 - \epsilon)$-approximation | # oracle calls: 1<br>oracle size: $\geq g(t) \cdot k^{\frac{t}{2}}$ | # oracle calls: 1<br>oracle size: $g(t) \cdot k^6$ | # oracle calls: $O\left(\frac{1}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$<br>oracle size: $g(t) \cdot k^{O\left(\frac{1}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)}$ |

# One open question from this work,
# for DELETION TO TREEWIDTH–t

**Uniform polynomial size with Turing kernelization?**

Is there a Turing Kernelization
of size $g(t) \cdot k^{O(1)}$
(that preserves an optimum solution)?



Input: $(G, k)$

OPT$(G_1, k_1)$ := oracle$(G_1, k_1)$

OPT$(G_2, k_2)$ := oracle$(G_2, k_2)$

OPT$(G_3, k_3)$ := oracle$(G_3, k_3)$

OPT$(G_4, k_4)$ := oracle$(G_4, k_4)$

OPT$(G_n, k_n)$ := oracle$(G_n, k_n)$

Output: optimum solution of $(G, k)$

arbitrary number of oracle calls (still $n^{O(1)}$)

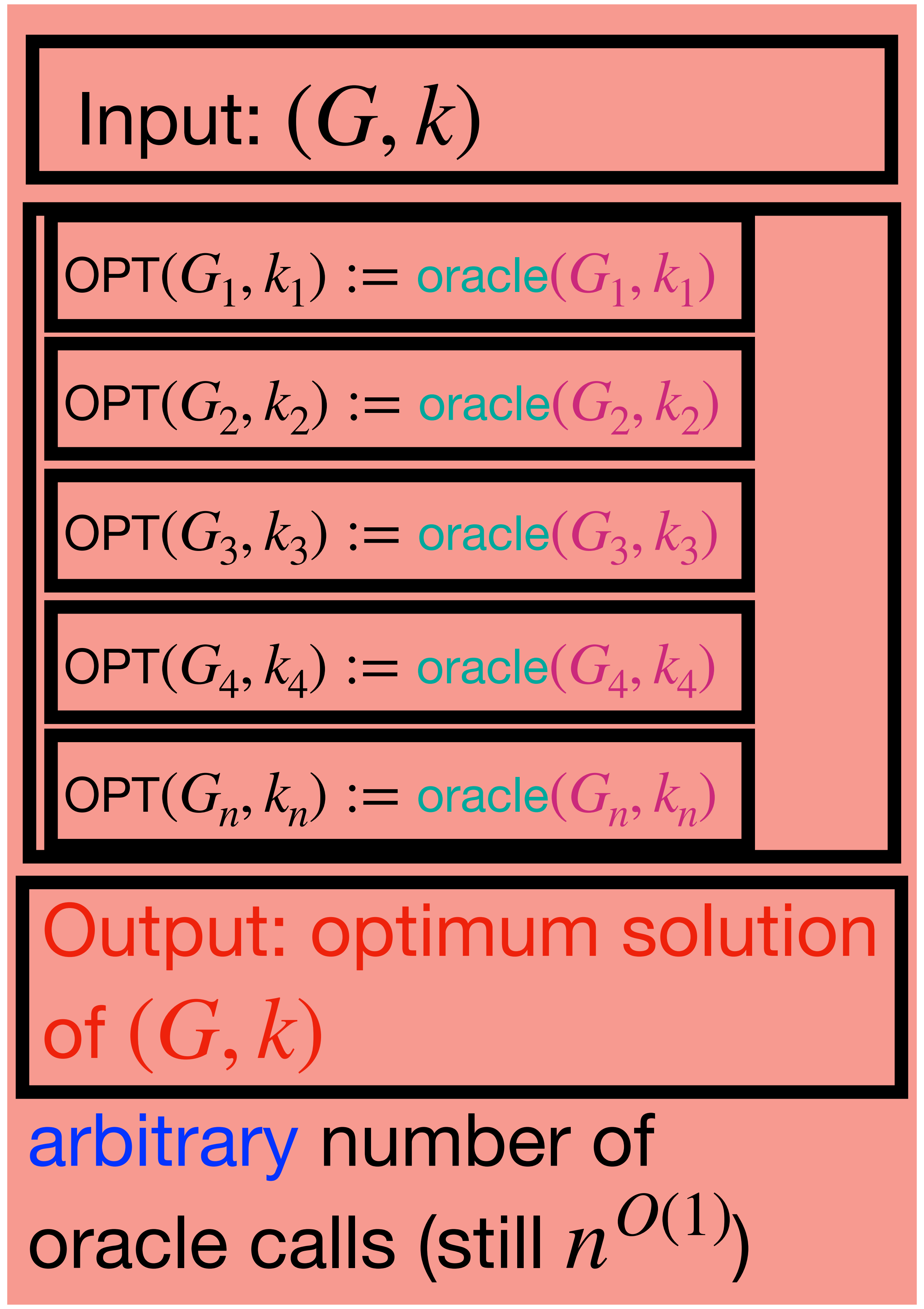
of size $g(t) \cdot k^{O(1)}$

Turing Kernelization

# One open question from this work, for DELETION TO TREEWIDTH-t

**Uniform polynomial size with Turing kernelization?**

Is there a Turing Kernelization of size $g(t) \cdot k^{O(1)}$ (that preserves an optimum solution)?



Input: $(G, k)$

OPT$(G_1, k_1)$ := oracle$(G_1, k_1)$

OPT$(G_2, k_2)$ := oracle$(G_2, k_2)$

OPT$(G_3, k_3)$ := oracle$(G_3, k_3)$

OPT$(G_4, k_4)$ := oracle$(G_4, k_4)$

OPT$(G_n, k_n)$ := oracle$(G_n, k_n)$

Output: optimum solution of $(G, k)$

arbitrary number of oracle calls (still $n^{O(1)}$)

of size $g(t) \cdot k^{O(1)}$

Turing Kernelization

Thank you!