Hitting minors on bounded treewidth graphs

Julien Baste\textsuperscript{1}  Ignasi Sau\textsuperscript{2}  Dimitrios M. Thilikos\textsuperscript{2,3}

\textbf{Fortaleza, Ceará}
February 2019

\textsuperscript{1} Universität Ulm, Ulm, Germany
\textsuperscript{2} CNRS, LIRMM, Université de Montpellier, France
\textsuperscript{3} Dept. of Maths, National and Kapodistrian University of Athens, Greece

[arXiv 1704.07284]
Minors and topological minors

- $H$ is a minor of a graph $G$ if $H$ can be obtained from a subgraph of $G$ by contracting edges.

$G$ is a minor of a graph $G$ if $H$ can be obtained from a subgraph of $G$ by contracting edges.

$H$ is a topological minor of $G$ if $H$ can be obtained from a subgraph of $G$ by contracting edges with at least one endpoint of degree $\leq 2$. Therefore:

$H$ topological minor of $G$ $\Rightarrow$ $H$ minor of $G$
Minors and topological minors

- \( H \) is a minor of a graph \( G \) if \( H \) can be obtained from a subgraph of \( G \) by contracting edges.

- \( H \) is a topological minor of \( G \) if \( H \) can be obtained from a subgraph of \( G \) by contracting edges with at least one endpoint of deg \( \leq 2 \).
Minors and topological minors

\[ G \quad \rightarrow \quad H \]

- \( H \) is a minor of a graph \( G \) if \( H \) can be obtained from a subgraph of \( G \) by contracting edges.

- \( H \) is a topological minor of \( G \) if \( H \) can be obtained from a subgraph of \( G \) by contracting edges with at least one endpoint of \( \text{deg} \leq 2 \).

Therefore: \[ H \text{ topological minor of } G \implies H \text{ minor of } G \]
Minors and topological minors

- $H$ is a **minor** of a graph $G$ if $H$ can be obtained from a subgraph of $G$ by contracting edges.

- $H$ is a **topological minor** of $G$ if $H$ can be obtained from a subgraph of $G$ by contracting edges with at least one endpoint of deg $\leq 2$.

Therefore: $H$ topological minor of $G \not\iff H$ minor of $G$
A \textit{k-tree} is a graph that can be built starting from a \((k + 1)\)-clique and then \textit{iteratively} adding a vertex connected to a \textit{k-clique}.
Treewidth via $k$-trees

A $k$-tree is a graph that can be built starting from a $(k + 1)$-clique and then iteratively adding a vertex connected to a $k$-clique.

Example of a 2-tree:
Treewidth via $k$-trees

A $k$-tree is a graph that can be built starting from a $(k + 1)$-clique and then iteratively adding a vertex connected to a $k$-clique.

Example of a 2-tree:
A $k$-tree is a graph that can be built starting from a $(k + 1)$-clique and then iteratively adding a vertex connected to a $k$-clique.
A $k$-tree is a graph that can be built starting from a $(k + 1)$-clique and then iteratively adding a vertex connected to a $k$-clique.

Example of a 2-tree:
A $k$-tree is a graph that can be built starting from a $(k + 1)$-clique and then iteratively adding a vertex connected to a $k$-clique.

Example of a 2-tree:
A *k-tree* is a graph that can be built starting from a \((k + 1)\)-clique and then *iteratively* adding a vertex connected to a *k-clique*.
A $k$-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a $k$-clique.

Example of a 2-tree:
Treewidth via \(k\)-trees

A \(k\)-tree is a graph that can be built starting from a \((k+1)\)-clique and then iteratively adding a vertex connected to a \(k\)-clique.
Treewidth via $k$-trees

A $k$-tree is a graph that can be built starting from a $(k + 1)$-clique and then iteratively adding a vertex connected to a $k$-clique.

Example of a 2-tree:
Treewidth via $k$-trees

A $k$-tree is a graph that can be built starting from a $(k + 1)$-clique and then iteratively adding a vertex connected to a $k$-clique.

A partial $k$-tree is a subgraph of a $k$-tree.
A *k-tree* is a graph that can be built starting from a \((k + 1)\)-clique and then *iteratively* adding a vertex connected to a *k*-clique.

A partial *k-tree* is a subgraph of a *k*-tree.

**Treewidth** of a graph \(G\), denoted \(\text{tw}(G)\): smallest integer \(k\) such that \(G\) is a partial *k*-tree.
Treewidth via $k$-trees

A $k$-tree is a graph that can be built starting from a $(k + 1)$-clique and then iteratively adding a vertex connected to a $k$-clique.

A partial $k$-tree is a subgraph of a $k$-tree.

**Treewidth** of a graph $G$, denoted $\text{tw}(G)$: smallest integer $k$ such that $G$ is a partial $k$-tree.

Invariant that measures the topological resemblance of a graph to a tree.
A *k-tree* is a graph that can be built starting from a \((k + 1)\)-clique and then *iteratively* adding a vertex connected to a *k-clique*.

A *partial k-tree* is a subgraph of a *k-tree*.

**Treewidth** of a graph \(G\), denoted \(\text{tw}(G)\):

smallest integer \(k\) such that \(G\) is a partial *k-tree*.

Invariant that measures the topological *resemblance* of a graph to a *tree*.

Construction suggests the notion of *tree decomposition*: small separators.
Why treewidth?

Treewidth is important for (at least) 3 different reasons:
Why treewidth?

Treewidth is important for (at least) 3 different reasons:

1. Treewidth is a fundamental *combinatorial tool* in graph theory: key role in the Graph Minors project of Robertson and Seymour.
Why treewidth?

Treewidth is important for (at least) 3 different reasons:

1. Treewidth is a fundamental **combinatorial tool** in graph theory: key role in the Graph Minors project of Robertson and Seymour.

2. In many **practical scenarios**, it turns out that the treewidth of the associated graph is small (programming languages, road networks, ...).
Why treewidth?

Treewidth is **important** for (at least) 3 different reasons:

1. Treewidth is a fundamental **combinatorial tool** in graph theory: key role in the Graph Minors project of Robertson and Seymour.

2. In many **practical scenarios**, it turns out that the treewidth of the associated graph is small (programming languages, road networks, ...).

3. Treewidth behaves very well **algorithmically**...
Monadic Second Order Logic (MSOL): Graph logic that allows quantification over sets of vertices and edges. Example:

$$\text{DomSet}(S) : \forall v \in V(G) \land S, \exists u \in S : \{u, v\} \in E(G)$$

Theorem (Courcelle, 1990) Every problem expressible in MSOL can be solved in time $f(tw) \cdot n$ on graphs on $n$ vertices and treewidth at most $tw$. In parameterized complexity: FPT parameterized by treewidth. Examples: Vertex Cover, Dominating Set, Hamiltonian Cycle, Clique, Independent Set, $k$-Coloring for fixed $k$, ...
Treewidth behaves very well algorithmically

Monadic Second Order Logic (MSOL): Graph logic that allows quantification over sets of vertices and edges.

**Example:** \( \text{DomSet}(S) : \left[ \forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G) \right] \)
Treewidth behaves very well algorithmically

Monadic Second Order Logic (MSOL): Graph logic that allows quantification over sets of vertices and edges.

**Example:** \( \text{DomSet}(S) : \forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G) \]

**Theorem (Courcelle, 1990)**

Every problem expressible in MSOL can be solved in time \( f(tw) \cdot n \) on graphs on \( n \) vertices and treewidth at most \( tw \).

In parameterized complexity: FPT parameterized by treewidth.

**Examples:** Vertex Cover, Dominating Set, Hamiltonian Cycle, Clique, Independent Set, \( k \)-Coloring for fixed \( k \), ...
Typically, Courcelle’s theorem allows to prove that a problem is FPT…

\[ f(tw) \cdot n^{O(1)} \]
Typically, Courcelle’s theorem allows to prove that a problem is FPT... ... but the running time can (and must) be huge!

\[ f(tw) \cdot n^{O(1)} = 2^{45678^{tw}} \cdot n^{O(1)} \]
Typically, Courcelle’s theorem allows to prove that a problem is FPT... ... but the running time can (and must) be huge!

\[ f(tw) \cdot n^{O(1)} = 2^{345678^{tw}} \cdot n^{O(1)} \]

**Major goal** find the smallest possible function \( f(tw) \).

This is a very active area in parameterized complexity.
Typically, Courcelle’s theorem allows to prove that a problem is FPT... ... but the running time can (and must) be huge!

$$f(tw) \cdot n^{O(1)} = 2^{345678^{tw}} \cdot n^{O(1)}$$

Major goal: find the smallest possible function $f(tw)$.

This is a very active area in parameterized complexity.

Remark: Algorithms parameterized by treewidth appear very often as a “black box” in all kinds of parameterized algorithms.
Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.
Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.

For many problems, like \textsc{Vertex Cover} or \textsc{Dominating Set}, the “natural” DP algorithms lead to (optimal) single-exponential algorithms:

$$2^{O(tw)} \cdot n^{O(1)}.$$
Two behaviors for problems parameterized by treewidth

Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.

For many problems, like Vertex Cover or Dominating Set, the “natural” DP algorithms lead to (optimal) single-exponential algorithms:

\[ 2^{O(tw)} \cdot n^{O(1)}. \]

But for the so-called connectivity problems, like Longest Path or Steiner Tree, the “natural” DP algorithms provide only time

\[ 2^{O(tw \cdot \log tw)} \cdot n^{O(1)}. \]
On topologically structured graphs (planar, surfaces, minor-free), it is possible to solve connectivity problems in time $2^{O(tw)} \cdot n^{O(1)}$:

- **Planar graphs:**
  
  [Dorn, Penninkx, Bodlaender, Fomin. 2005]

- **Graphs on surfaces:**
  
  [Dorn, Fomin, Thilikos. 2006]
  
  [Rué, S., Thilikos. 2010]

- **Minor-free graphs:**
  
  [Dorn, Fomin, Thilikos. 2008]
  
  [Rué, S., Thilikos. 2012]
On **topologically structured** graphs (**planar, surfaces, minor-free**), it is possible to solve **connectivity problems** in time $2^{O(tw)} \cdot n^{O(1)}$:

- **Planar graphs:**
  
  - [Dorn, Penninkx, Bodlaender, Fomin. 2005]

- **Graphs on surfaces:**
  
  - [Dorn, Fomin, Thilikos. 2006]
  - [Rué, S., Thilikos. 2010]
  - [Dorn, Fomin, Thilikos. 2008]
  - [Rué, S., Thilikos. 2012]

**Main idea** special type of decomposition with nice topological properties:

- **partial solutions** $\iff$ **non-crossing partitions**

\[
CN(k) = \frac{1}{k+1} \binom{2k}{k} \sim \frac{4^k}{\sqrt{\pi} k^{3/2}} \leq 4^k.
\]
The revolution of single-exponential algorithms

It was believed that, except on sparse graphs (planar, surfaces), algorithms in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$ were optimal for connectivity problems.
The revolution of single-exponential algorithms

It was believed that, except on sparse graphs (planar, surfaces), algorithms in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$ were optimal for connectivity problems.

This was false!!

Cut&Count technique: [Cygan, Nederlof, Pilipczuk, van Rooij, Wojtaszczyk. 2011]
Randomized single-exponential algorithms for connectivity problems.
The revolution of single-exponential algorithms

It was believed that, except on sparse graphs (planar, surfaces), algorithms in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$ were optimal for connectivity problems.

This was false!!

Cut&Count technique: [Cygan, Nederlof, Pilipczuk, van Rooij, Wojtaszczyk. 2011]
Randomized single-exponential algorithms for connectivity problems.

Deterministic algorithms with algebraic tricks: [Bodlaender, Cygan, Kratsch, Nederlof. 2013]

Representative sets in matroids: [Fomin, Lokshtanov, Saurabh. 2014]
Do all connectivity problems admit single-exponential algorithms (on general graphs) parameterized by treewidth?
Do all connectivity problems admit single-exponential algorithms (on general graphs) parameterized by treewidth?

No!

**Cycle Packing**: find the maximum number of vertex-disjoint cycles.
Do all connectivity problems admit single-exponential algorithms (on general graphs) parameterized by treewidth?

No!

**Cycle Packing**: find the maximum number of vertex-disjoint cycles.

An algorithm in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$ is optimal under the ETH.

[Cygan, Nederlof, Pilipczuk, Pilipczuk, van Rooij, Wojtaszczyk. 2011]

**ETH**: The 3-SAT problem on $n$ variables cannot be solved in time $2^{o(n)}$

[Impagliazzo, Paturi. 1999]
Do all connectivity problems admit single-exponential algorithms (on general graphs) parameterized by treewidth?

No!

**Cycle Packing**: find the maximum number of vertex-disjoint cycles.

An algorithm in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$ is optimal under the ETH.

[Cygan, Nederlof, Pilipczuk, Pilipczuk, van Rooij, Wojtaszczyk. 2011]

**ETH**: The **3-SAT** problem on $n$ variables cannot be solved in time $2^{o(n)}$

[Impagliazzo, Paturi. 1999]

There are other examples of such problems...
The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.
The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion

**Input:** A graph $G$ and an integer $k$.

**Parameter:** The treewidth $tw$ of $G$.

**Question:** Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?
The $\mathcal{F}$-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

<table>
<thead>
<tr>
<th>$\mathcal{F}$-Deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> A graph $G$ and an integer $k$.</td>
</tr>
<tr>
<td><strong>Parameter:</strong> The treewidth $tw$ of $G$.</td>
</tr>
<tr>
<td><strong>Question:</strong> Does $G$ contain a set $S \subseteq V(G)$ with $</td>
</tr>
</tbody>
</table>

- $\mathcal{F} = \{K_2\}$: Vertex Cover.
The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

\begin{center}
\fbox{
\begin{tabular}{l}
\textbf{F-M-Deletion} \\
\textbf{Input:} A graph $G$ and an integer $k$. \\
\textbf{Parameter:} The treewidth $tw$ of $G$. \\
\textbf{Question:} Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?
\end{tabular}}
\end{center}

- $\mathcal{F} = \{K_2\}$: Vertex Cover. \\
  Easily solvable in time $2^{O(tw)} \cdot n^{O(1)}$. 

- $\mathcal{F} = \{C_3\}$: Feedback Vertex Set. \\
  "Hardly" solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$. [Cut&Count. 2011]

- $\mathcal{F} = \{K_5, K_3, 3\}$: Vertex Planarization. \\
  Solvable in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$. [Jansen, Lokshtanov, Saurabh. 2014 + Pilipczuk. 2017]
Let $\mathcal{F}$ be a fixed finite collection of graphs.

### $\mathcal{F}$-M-Deletion

**Input:** A graph $G$ and an integer $k$.

**Parameter:** The treewidth $tw$ of $G$.

**Question:** Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

- $\mathcal{F} = \{K_2\}$: **Vertex Cover**.
  Easily solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$.

- $\mathcal{F} = \{C_3\}$: **Feedback Vertex Set**.
The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

\begin{center}
\hspace{1cm} \fbox{\begin{tabular}{l}
\textbf{\(\mathcal{F}\)-M-Deletion}\\
\textbf{Input:} A graph $G$ and an integer $k$. \hfill \textbf{Parameter:} The treewidth $tw$ of $G$. \\
\textbf{Question:} Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?
\end{tabular}}
\end{center}

- $\mathcal{F} = \{K_2\}$: **Vertex Cover.**
  Easily solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$.

- $\mathcal{F} = \{C_3\}$: **Feedback Vertex Set.**
  “Hardly” solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$. 

[Cut&Count. 2011]
The \( F\)-M-Deletion problem

Let \( F \) be a fixed finite collection of graphs.

\[ F\text{-M-Deletion} \]
Input: A graph \( G \) and an integer \( k \).
Parameter: The treewidth \( tw \) of \( G \).
Question: Does \( G \) contain a set \( S \subseteq V(G) \) with \( |S| \leq k \) such that \( G - S \) does not contain any of the graphs in \( F \) as a minor?

- \( F = \{K_2\} \): Vertex Cover.
  Easily solvable in time \( 2^{\Theta(tw)} \cdot n^{O(1)} \).

- \( F = \{C_3\} \): Feedback Vertex Set.
  “Hardly” solvable in time \( 2^{\Theta(tw)} \cdot n^{O(1)} \). [Cut&Count. 2011]

- \( F = \{K_5, K_{3,3}\} \): Vertex Planarization.
The $\mathcal{F}$-M-Deletion problem

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion

Input: A graph $G$ and an integer $k$.
Parameter: The treewidth $tw$ of $G$.
Question: Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any of the graphs in $\mathcal{F}$ as a minor?

- $\mathcal{F} = \{K_2\}$: **Vertex Cover.**
  Easily solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$.

- $\mathcal{F} = \{C_3\}$: **Feedback Vertex Set.**
  “Hardly” solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$. [Cut&Count. 2011]

- $\mathcal{F} = \{K_5, K_{3,3}\}$: **Vertex Planarization.**
  Solvable in time $2^{\Theta(tw \cdot \log tw)} \cdot n^{O(1)}$. [Jansen, Lokshnanov, Saurabh. 2014 + Pilipczuk. 2017]
Covering topological minors

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion

**Input:** A graph $G$ and an integer $k$.

**Parameter:** The treewidth $tw$ of $G$.

**Question:** Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a minor?
Covering topological minors

Let $\mathcal{F}$ be a fixed finite collection of graphs.

\textbf{$\mathcal{F}$-M-Deletion}

\textbf{Input:} A graph $G$ and an integer $k$.
\textbf{Parameter:} The treewidth $\text{tw}$ of $G$.
\textbf{Question:} Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a minor?

\textbf{$\mathcal{F}$-TM-Deletion}

\textbf{Input:} A graph $G$ and an integer $k$.
\textbf{Parameter:} The treewidth $\text{tw}$ of $G$.
\textbf{Question:} Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a topol. minor?
Covering topological minors

Let $\mathcal{F}$ be a fixed finite collection of graphs.

$\mathcal{F}$-M-Deletion

Input: A graph $G$ and an integer $k$.
Parameter: The treewidth $tw$ of $G$.
Question: Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a minor?

$\mathcal{F}$-TM-Deletion

Input: A graph $G$ and an integer $k$.
Parameter: The treewidth $tw$ of $G$.
Question: Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a topol. minor?

Both problems are NP-hard if $\mathcal{F}$ contains some edge. [Lewis, Yannakakis. 1980]
Covering topological minors

Let $\mathcal{F}$ be a fixed finite collection of graphs.

**$\mathcal{F}$-M-Deletion**

**Input:** A graph $G$ and an integer $k$.

**Parameter:** The treewidth $\text{tw}$ of $G$.

**Question:** Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a minor?

**$\mathcal{F}$-TM-Deletion**

**Input:** A graph $G$ and an integer $k$.

**Parameter:** The treewidth $\text{tw}$ of $G$.

**Question:** Does $G$ contain a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a topol. minor?

Both problems are **NP-hard** if $\mathcal{F}$ contains some edge. [Lewis, Yannakakis. 1980]

**FPT** by Courcelle’s Theorem.
Objective

Determine, for every fixed $\mathcal{F}$, the (asymptotically) smallest function $f_{\mathcal{F}}$ such that $\mathcal{F}$-M-Deletion/$\mathcal{F}$-TM-Deletion can be solved in time

$$f_{\mathcal{F}}(tw) \cdot n^{O(1)}$$

on $n$-vertex graphs.
Goal of this project

Objective

Determine, for every fixed $\mathcal{F}$, the (asymptotically) smallest function $f_{\mathcal{F}}$ such that $\mathcal{F}$-$M$-Deletion/$\mathcal{F}$-$TM$-Deletion can be solved in time

$$f_{\mathcal{F}}(tw) \cdot n^{O(1)}$$

on $n$-vertex graphs.

- We do not want to optimize the degree of the polynomial factor.
- We do not want to optimize the constants.
- Our hardness results hold under the ETH.
Summary of our results

For every $F$:
- $F$-M/TM-Deletion in time $2^{O\left(tw \cdot \log tw\right)} \cdot n^{O(1)}$.

$F$ connected and planar:
- $F$-M-Deletion in time $2^{O\left(tw \cdot \log tw\right)} \cdot n^{O(1)}$.

$G$ planar + $F$ connected:
- $F$-M-Deletion in time $2^{O\left(tw\right)} \cdot n^{O(1)}$.

(For $F$-TM-Deletion we need: $F$ contains a subcubic planar graph.)

$F$ (connected):
- $F$-M/TM-Deletion not in time $2^{o\left(tw \cdot \log tw\right)}$ unless the ETH fails, even if $G$ planar.

$F = \{H\}$, $H$ connected and planar:
- Complete tight dichotomy.

---

1. **Connected** collection $\mathcal{F}$: all the graphs are **connected**.
2. **Planar** collection $\mathcal{F}$: contains at least one **planar** graph.
Summary of our results

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-Deletion in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

---

1. **Connected** collection $\mathcal{F}$: all the graphs are connected.
2. **Planar** collection $\mathcal{F}$: contains at least one planar graph.
Summary of our results

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-DELETION in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

- $\mathcal{F}$ connected$^1$ + planar$^2$: $\mathcal{F}$-M-DELETION in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

---

$^1$ **Connected** collection $\mathcal{F}$: all the graphs are connected.

$^2$ **Planar** collection $\mathcal{F}$: contains at least one planar graph.
Summary of our results

- For every $\mathcal{F}$: \textsc{F-M/TM-Deletion} in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

- $\mathcal{F}$ connected$^1$ $\implies$ planar$^2$: \textsc{F-M-Deletion} in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

---

$^1$\textbf{Connected} collection $\mathcal{F}$: all the graphs are connected.

$^2$\textbf{Planar} collection $\mathcal{F}$: contains at least one planar graph.
Summary of our results

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-Deletion in time $2^{2^{O(tw \cdot \log tw)}} \cdot n^{O(1)}$.

- $\mathcal{F}$ connected$^1$ + planar$^2$: $\mathcal{F}$-M-Deletion in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

- $G$ planar + $\mathcal{F}$ connected: $\mathcal{F}$-M-Deletion in time $2^{O(tw)} \cdot n^{O(1)}$.

---

$^1$Connected collection $\mathcal{F}$: all the graphs are connected.

$^2$Planar collection $\mathcal{F}$: contains at least one planar graph.
Summary of our results

- For every \( \mathcal{F} \): \textbf{F-M/TM-Deletion} in time \( 2^{O(tw \cdot \log tw)} \cdot n^{O(1)} \).

- \( \mathcal{F} \) connected\(^1\) + planar\(^2\): \textbf{F-M-Deletion} in time \( 2^{O(tw \cdot \log tw)} \cdot n^{O(1)} \).

- \( G \) planar + \( \mathcal{F} \) connected: \textbf{F-M-Deletion} in time \( 2^{O(tw)} \cdot n^{O(1)} \).

(For \textbf{F-TM-Deletion} we need: \( \mathcal{F} \) contains a subcubic planar graph.)

---

\(^1\) Connected collection \( \mathcal{F} \): all the graphs are connected.
\(^2\) Planar collection \( \mathcal{F} \): contains at least one planar graph.
Summary of our results

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-Deletion in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

- $\mathcal{F}$ connected$^1 \oplus$ planar$^2$: $\mathcal{F}$-M-Deletion in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

- $G$ planar + $\mathcal{F}$ connected: $\mathcal{F}$-M-Deletion in time $2^{O(tw)} \cdot n^{O(1)}$.

  (For $\mathcal{F}$-TM-Deletion we need: $\mathcal{F}$ contains a subcubic planar graph.)

- $\mathcal{F}$ (connected): $\mathcal{F}$-M/TM-Deletion not in time $2^{o(tw)} \cdot n^{O(1)}$ unless the ETH fails, even if $G$ planar.

---

$^1$Connected collection $\mathcal{F}$: all the graphs are connected.

$^2$Planar collection $\mathcal{F}$: contains at least one planar graph.
Summary of our results

- For every $F$: $F$-$M/TM$-DELETION in time $2^{O(tw\cdot \log tw)} \cdot n^{O(1)}$.

- $F$ connected$^1 \oplus$ planar$^2$: $F$-$M$-DELETION in time $2^{O(tw\cdot \log tw)} \cdot n^{O(1)}$.

- $G$ planar + $F$ connected: $F$-$M$-DELETION in time $2^{O(tw)} \cdot n^{O(1)}$.

  (For $F$-$TM$-DELETION we need: $F$ contains a subcubic planar graph.)

- $F$ (connected): $F$-$M/TM$-DELETION not in time $2^{o(tw)} \cdot n^{O(1)}$ unless the ETH fails, even if $G$ planar.

- $F = \{H\}$, $H$ connected and planar:

---

$^1$Connected collection $F$: all the graphs are connected.

$^2$Planar collection $F$: contains at least one planar graph.
Summary of our results

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-Deletion in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

- $\mathcal{F}$ connected$^1$ + planar$^2$: $\mathcal{F}$-M-Deletion in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

- $G$ planar + $\mathcal{F}$ connected: $\mathcal{F}$-M-Deletion in time $2^{O(tw)} \cdot n^{O(1)}$.

  (For $\mathcal{F}$-TM-Deletion we need: $\mathcal{F}$ contains a subcubic planar graph.)

- $\mathcal{F}$ (connected): $\mathcal{F}$-M/TM-Deletion not in time $2^{o(tw)} \cdot n^{O(1)}$ unless the ETH fails, even if $G$ planar.

- $\mathcal{F} = \{H\}$, $H$ connected and planar: complete tight dichotomy.

---

$^1$Connected collection $\mathcal{F}$: all the graphs are connected.

$^2$Planar collection $\mathcal{F}$: contains at least one planar graph.
Summary of our results

- For every $\mathcal{F}$: $\mathcal{F}$-M/TM-Deletion in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

- $\mathcal{F}$ connected$^1$ + planar$^2$: $\mathcal{F}$-M-Deletion in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

- $G$ planar + $\mathcal{F}$ connected: $\mathcal{F}$-M-Deletion in time $2^{O(tw)} \cdot n^{O(1)}$.

  (For $\mathcal{F}$-TM-Deletion we need: $\mathcal{F}$ contains a subcubic planar graph.)

- $\mathcal{F}$ (connected): $\mathcal{F}$-M/TM-Deletion not in time $2^{o(tw)} \cdot n^{O(1)}$ unless the ETH fails, even if $G$ planar.

- $\mathcal{F} = \{H\}$, $H$ connected and planar: complete tight dichotomy.

---

$^1$Connected collection $\mathcal{F}$: all the graphs are connected.

$^2$Planar collection $\mathcal{F}$: contains at least one planar graph.
Classification of the complexity of $\{H\}$-M-Deletion for all connected simple planar graphs $H$ with $|V(H)| \leq 5$ and $|E(H)| \geq 1$: for the 9 graphs on the left (resp. 20 graphs on the right), the problem is solvable in time $2^{\Theta(tw)} \cdot n^{O(1)}$ (resp. $2^{\Theta(tw \cdot \log tw)} \cdot n^{O(1)}$). For $\{H\}$-TM-Deletion, $K_{1,4}$ should be on the left.
For topological minors, there (at least) one change

\[ 2^{\Theta(tw)} \]

\[ 2^{\Theta(tw \cdot \log tw)} \]

\[ P_5 \]

\[ P_2 \]

\[ P_3 \]

\[ P_4 \]

\[ C_3 \]

\[ C_4 \]

\[ \text{claw} \]

\[ \text{paw} \]

\[ \text{chair} \]

\[ \text{banner} \]

\[ \text{diamond} \]

\[ K_4 \]

\[ C_5 \]

\[ K_{3 \cup 2K_1} \]

\[ K_5 - e \]

\[ W_4 \]

\[ P_3 \cup 2K_1 \]

\[ \text{gem} \]

\[ \text{house} \]

\[ K_5 \]

\[ P_2 \cup P_3 \]

\[ \text{px} \]

\[ \text{kite} \]

\[ \text{dart} \]

\[ K_{2,3} \]

\[ \text{bull} \]

\[ \text{butterfly} \]

\[ \text{cricket} \]

\[ \text{co-banner} \]
A compact statement for small planar minors

All these cases can be succinctly described as follows:
A compact statement for small planar minors

All these cases can be succinctly described as follows:

- All the graphs on the left are minors of \( \square \) (called the banner)
A compact statement for small planar minors

All these cases can be succinctly described as follows:

- All the graphs on the left are minors of \( \square \) (called the banner).
- All the graphs on the right are not minors of \( \square \).
A compact statement for small planar minors

All these cases can be succinctly described as follows:

- All the graphs on the left are minors of \( K_2,3 \) (called the banner).
- All the graphs on the right are not minors of \( K_2,3 \) except \( P_5 \).
A dichotomy for hitting connected minors

We can prove that any connected $H$ with $|V(H)| \geq 6$ is hard: \{H\}-M-Deletion cannot be solved in time $2^{o(tw \cdot \log tw)} \cdot n^{O(1)}$ under the ETH.
A dichotomy for hitting connected minors

We can prove that any connected $H$ with $|V(H)| \geq 6$ is hard: \{H\}-M-Deletion cannot be solved in time $2^{o(tw \cdot \log tw)} \cdot n^{O(1)}$ under the ETH.

**Theorem**

Let $H$ be a connected planar graph.
A dichotomy for hitting connected minors

We can prove that any connected $H$ with $|V(H)| \geq 6$ is hard:
$\{H\}$-M-Deletion cannot be solved in time $2^{o(tw \cdot \log tw)} \cdot n^{O(1)}$ under the ETH.

Theorem

Let $H$ be a connected planar graph.
We can prove that any connected $H$ with $|V(H)| \geq 6$ is hard: \{H\}-\text{M-Deletion} cannot be solved in time $2^{o(tw \cdot \log tw)} \cdot n^{O(1)}$ under the ETH.

**Theorem**

Let $H$ be a connected planar graph. The \{H\}-\text{M-Deletion} problem is solvable in time

- $2^{O(tw)} \cdot n^{O(1)}$, if $H \preceq_m \begin{tikzpicture}[baseline={([yshift=-.5ex]current bounding box.center)}]
\draw (0,0) -- (1,0) -- (1,1) -- (0,1) -- cycle;
\end{tikzpicture}$ and $H \neq P_5$. 


A dichotomy for hitting connected minors

We can prove that any connected $H$ with $|V(H)| \geq 6$ is hard: $\{H\}$-M-Deletion cannot be solved in time $2^{o(tw \cdot \log tw)} \cdot n^{O(1)}$ under the ETH.

**Theorem**

Let $H$ be a connected planar graph.
The $\{H\}$-M-Deletion problem is solvable in time

- $2^{O(tw)} \cdot n^{O(1)}$, if $H \preceq_m \square$ and $H \not= P_5$.

- $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$, otherwise.

In both cases, the running time is asymptotically optimal under the ETH.
Every connected component (with at least 5 vertices) of a graph that excludes the banner as a (topological) minor is either:

- a cycle (of any length),
- or a tree in which some vertices have been replaced by triangles.

Both such types of components can be maintained by a dynamic programming algorithm in single-exponential time.

If the characterization of the allowed connected components is enriched in some way, such as restricting the length of the allowed cycles or forbidding certain degrees, the problem becomes harder.
Why the banner??

- Every **connected component** (with at least 5 vertices) of a graph that excludes the **banner** as a (topological) minor is either:

![Diagram of a graph](image-url)
Every connected component (with at least 5 vertices) of a graph that excludes the banner as a (topological) minor is either:

- a cycle (of any length),
- or a tree in which some vertices have been replaced by triangles.
Every connected component (with at least 5 vertices) of a graph that excludes the banner as a (topological) minor is either:

- a cycle (of any length),
- or a tree in which some vertices have been replaced by triangles.

Both such types of components can be maintained by a dynamic programming algorithm in single-exponential time.
Every connected component (with at least 5 vertices) of a graph that excludes the banner as a (topological) minor is either:

- a cycle (of any length),
- or a tree in which some vertices have been replaced by triangles.

Both such types of components can be maintained by a dynamic programming algorithm in single-exponential time.

If the characterization of the allowed connected components is enriched in some way, such as restricting the length of the allowed cycles or forbidding certain degrees, the problem becomes harder.
We have three types of results

1. General algorithms
   - For every $F$: time $O(tw \cdot \log tw) \cdot n^{O(1)}$.
   - $F$ connected + planar: time $O(tw \cdot \log tw) \cdot n^{O(1)}$.
   - $G$ planar + $F$ connected: time $O(tw) \cdot n^{O(1)}$.

2. Ad-hoc single-exponential algorithms
   - Some use "typical" dynamic programming.
   - Some use the rank-based approach. [Bodlaender, Cygan, Kratsch, Nederlof. 2013]

3. Lower bounds under the ETH
   - $2^{o(tw)}$ is "easy".
   - $2^{o(tw \cdot \log tw)}$ is much more involved and we get ideas from:
     - [Lokshtanov, Marx, Saurabh. 2011]
     - [Marcin Pilipczuk. 2017]
     - [Bonnet, Brettell, Kwon, Marx. 2017]
We have three types of results

1. **General algorithms**

   - For every $\mathcal{F}$: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
   - $\mathcal{F}$ connected + planar: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
   - $\mathcal{F}$ connected + planar: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
   - $G$ planar + $\mathcal{F}$ connected: time $2^{O(tw)} \cdot n^{O(1)}$.

2. **Ad-hoc single-exponential algorithms**

   Some use "typical" dynamic programming.
   Some use the rank-based approach. [Bodlaender, Cygan, Kratsch, Nederlof. 2013]

3. **Lower bounds under the ETH**

   $2^{o(tw)}$ is "easy".
   $2^{o(tw \cdot \log tw)}$ is much more involved and we get ideas from: [Lokshtanov, Marx, Saurabh. 2011] [Marcin Pilipczuk. 2017] [Bonnet, Brettell, Kwon, Marx. 2017]
We have three types of results

1. **General algorithms**

   - For every $\mathcal{F}$: time $2^{2^{O(tw \cdot \log tw)}} \cdot n^{O(1)}$.
   - $\mathcal{F}$ connected + planar: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
   - $\mathcal{F}$ connected $\rightarrow$ planar: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
   - $G$ planar + $\mathcal{F}$ connected: time $2^{O(tw)} \cdot n^{O(1)}$.

2. **Ad-hoc single-exponential algorithms**

   - Some use “typical” dynamic programming.
   - Some use the *rank-based* approach.

   [Bodlaender, Cygan, Kratsch, Nederlof. 2013]
We have three types of results

1. **General algorithms**
   - For every $\mathcal{F}$: time $2^{2^{O(tw \cdot \log tw)}} \cdot n^{O(1)}$.
   - $\mathcal{F}$ connected + planar: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
   - $\mathcal{F}$ connected + planar: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
   - $G$ planar + $\mathcal{F}$ connected: time $2^{O(tw)} \cdot n^{O(1)}$.

2. **Ad-hoc single-exponential algorithms**
   - Some use “typical” dynamic programming.
   - Some use the rank-based approach. [Bodlaender, Cygan, Kratsch, Nederlof. 2013]

3. **Lower bounds under the ETH**
   - $2^{o(tw)}$ is “easy”.
   - $2^{o(tw \cdot \log tw)}$ is much more involved and we get ideas from:
     - Lokshtanov, Marx, Saurabh. 2011
     - Marcin Pilipczuk. 2017
     - Bonnet, Brettell, Kwon, Marx. 2017
Some ideas of the general algorithms

- For every $\mathcal{F}$: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
- $\mathcal{F}$ connected + planar: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
- $G$ planar + $\mathcal{F}$ connected: time $2^{O(tw)} \cdot n^{O(1)}$. 

We build on the machinery of bounded graphs and representatives:

[Bodlaender, Fomin, Lokshtanov, Penninkx, Saurabh, Thilikos. 2009]
[Fomin, Lokshtanov, Saurabh, Thilikos. 2010]
[Kim, Langer, Paul, Reidl, Rossmanith, S., Sikdar. 2013]
[Garnero, Paul, S., Thilikos. 2014]

Extra: Bidimensionality, irrelevant vertices, protrusion decomposition...
Some ideas of the general algorithms

- For every $\mathcal{F}$: time $2^{2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})}} \cdot n^{\mathcal{O}(1)}$.
- $\mathcal{F}$ connected + planar: time $2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$.
- $G$ planar + $\mathcal{F}$ connected: time $2^{\mathcal{O}(	ext{tw})} \cdot n^{\mathcal{O}(1)}$.

We build on the machinery of boundaried graphs and representatives:

[Bodlaender, Fomin, Lokshtanov, Penninkx, Saurabh, Thilikos. 2009]
[Fomin, Lokshtanov, Saurabh, Thilikos. 2010]
[Kim, Langer, Paul, Reidl, Rossmanith, S., Sikdar. 2013]
[Garnero, Paul, S., Thilikos. 2014]
Some ideas of the general algorithms

- For every $\mathcal{F}$: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
- $\mathcal{F}$ connected + planar: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.
- $G$ planar + $\mathcal{F}$ connected: time $2^{O(tw)} \cdot n^{O(1)}$.

We build on the machinery of **boundaried graphs and representatives**:

- $\mathcal{F}$ connected + planar: time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$.

**Extra**: Bidimensionality, irrelevant vertices, protrusion decomposition...

[Bodlaender, Fomin, Lokshtanov, Penninkx, Saurabh, Thilikos. 2009]
[Fomin, Lokshtanov, Saurabh, Thilikos. 2010]
[Kim, Langer, Paul, Reidl, Rossmanith, S., Sikdar. 2013]
[Garnero, Paul, S., Thilikos. 2014]
Algorithm for a general collection $\mathcal{F}$

- We see $G$ as a $t$-boundaried graph.
Algorithm for a general collection $\mathcal{F}$

- We see $G$ as a $t$-boundaried graph.
- folio of $G$: set of all its $\mathcal{F}$-minor-free minors, up to size $O_{\mathcal{F}}(t)$. 
Algorithm for a general collection $\mathcal{F}$

- We see $G$ as a $t$-boundaried graph.
- *folio* of $G$: set of all its $\mathcal{F}$-minor-free minors, up to size $O_\mathcal{F}(t)$.
- We compute, using DP over a tree decomposition of $G$, the following parameter for every folio $C$:

\[ p(G, C) = \min\{|S| : S \subseteq V(G) \land \text{folio}(G - S) = C\} \]
Algorithm for a general collection $\mathcal{F}$

- We see $G$ as a $t$-boundaried graph.
- **folio** of $G$: set of all its $\mathcal{F}$-minor-free minors, up to size $O_\mathcal{F}(t)$.

We compute, using DP over a tree decomposition of $G$, the following parameter for every folio $C$:

$$p(G, C) = \min\{|S| : S \subseteq V(G) \land \text{folio}(G - S) = C\}$$

- For every $t$-boundaried graph $G$, $|\text{folio}(G)| = 2^{O_\mathcal{F}(t \log t)}$. 
Algorithm for a general collection $\mathcal{F}$

- We see $G$ as a $t$-boundaried graph.

- **folio of $G$:** set of all its $\mathcal{F}$-minor-free minors, up to size $O_\mathcal{F}(t)$.

- We compute, using DP over a tree decomposition of $G$, the following parameter for every folio $C$:

$$p(G, C) = \min\{|S| : S \subseteq V(G) \land \text{folio}(G - S) = C\}$$

- For every $t$-boundaried graph $G$, $|\text{folio}(G)| = 2^{O_\mathcal{F}(t \log t)}$.

- The number of distinct folios is $2^{2^{O_\mathcal{F}(t \log t)}}$. 
Algorithm for a general collection $\mathcal{F}$

- We see $G$ as a $t$-boundaried graph.

- **folio of $G$:** set of all its $\mathcal{F}$-minor-free minors, up to size $O_\mathcal{F}(t)$.

- We compute, using DP over a tree decomposition of $G$, the following parameter for every folio $C$:

  $$\text{p}(G, C) = \min\{|S| : S \subseteq V(G) \land \text{folio}(G - S) = C\}$$

- For every $t$-boundaried graph $G$, $|\text{folio}(G)| = 2^{O_\mathcal{F}(t \log t)}$.

- The number of distinct folios is $2^{2^{O_\mathcal{F}(t \log t)}}$.

- This gives an algorithm running in time $2^{2^{O_\mathcal{F}(tw \cdot \log tw)}} \cdot n^{O(1)}$. 


Algorithm for a connected and planar collection $\mathcal{F}$

For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv (\mathcal{F}, t)$ on $t$-boundaried graphs:

$$G_1 \equiv (\mathcal{F}, t) G_2 \iff \forall G' \in B_t, \mathcal{F} \preceq m G' \circlearrowleft G_1 \iff \mathcal{F} \preceq m G' \circlearrowleft G_2.$$ 

$R(\mathcal{F}, t)$: set of minimum-size representatives of $\equiv (\mathcal{F}, t)$.

We compute, using DP over a tree decomposition of $G$, the following parameter for every representative $R$:

$$p(G, R) = \min \{ |S| : S \subseteq V(G) \land \text{rep}_F, t(G - S) = R \}.$$ 

The number of representatives is $|R(\mathcal{F}, t)| = 2O_F(t \cdot \log t)$.

This gives an algorithm running in time $2O(F(tw \cdot \log tw)) \cdot nO(1)$.

[Baste, Noy, S. 2017]
Algorithm for a connected and planar collection $\mathcal{F}$

- For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv(\mathcal{F}, t)$ on $t$-boundaried graphs:
  
  $G_1 \equiv(\mathcal{F}, t) G_2$ if $\forall G' \in \mathcal{B}^t$,  
  $\mathcal{F} \preceq_m G' \oplus G_1 \iff \mathcal{F} \preceq_m G' \oplus G_2$.

$R(\mathcal{F}, t)$: set of minimum-size representatives of $\equiv(\mathcal{F}, t)$.

We compute, using DP over a tree decomposition of $G$, the following parameter for every representative $R$:

$$p(G, R) = \min\{|S| : S \subseteq V(G) \land \text{rep}_F(t, G - S) = R\}$$

The number of representatives is $|R(\mathcal{F}, t)| = 2^{O(|\mathcal{F}| \cdot \log t)}$.

#labeled graphs of size $\leq t$ and $tw \leq h$ is $2^{O(h(t \cdot \log t))}$.

[Baste, Noy, S. 2017] This gives an algorithm running in time $2^{O(F(tw \cdot \log tw))} \cdot n^{O(1)}$. 
Algorithm for a connected and planar collection $\mathcal{F}$

- For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv(\mathcal{F},t)$ on $t$-boundaried graphs:

  $G_1 \equiv(\mathcal{F},t) G_2$ if $\forall G' \in \mathcal{B}^t$, $\mathcal{F} \preceq_m G' \oplus G_1 \iff \mathcal{F} \preceq_m G' \oplus G_2$.

- $\mathcal{R}(\mathcal{F},t)$: set of minimum-size representatives of $\equiv(\mathcal{F},t)$.
Algorithm for a connected and planar collection $\mathcal{F}$

- For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv(\mathcal{F}, t)$ on $t$-boundaried graphs:
  
  $G_1 \equiv(\mathcal{F}, t) G_2$ if $\forall G' \in B^t$, $\mathcal{F} \preceq_m G' \oplus G_1 \iff \mathcal{F} \preceq_m G' \oplus G_2$.

- $\mathcal{R}(\mathcal{F}, t)$: set of minimum-size representatives of $\equiv(\mathcal{F}, t)$.

- We compute, using DP over a tree decomposition of $G$, the following parameter for every representative $R$:
  
  $p(G, R) = \min\{|S| : S \subseteq V(G) \land \text{rep}_{\mathcal{F}, t}(G - S) = R\}$
For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv(\mathcal{F}, t)$ on $t$-boundaried graphs:

$$G_1 \equiv(\mathcal{F}, t) G_2 \quad \text{if } \forall G' \in B^t,$$

$$\mathcal{F} \preceq_m G' \oplus G_1 \iff \mathcal{F} \preceq_m G' \oplus G_2.$$

$\mathcal{R}(\mathcal{F}, t)$: set of minimum-size representatives of $\equiv(\mathcal{F}, t)$.

We compute, using DP over a tree decomposition of $G$, the following parameter for every representative $R$:

$$p(G, R) = \min\{|S| : S \subseteq V(G) \land \text{rep}_{\mathcal{F}, t}(G - S) = R\}$$

The number of representatives is $|\mathcal{R}(\mathcal{F}, t)| = 2^{O_{\mathcal{F}}(t \cdot \log t)}$. 
Algorithm for a connected and planar collection \( \mathcal{F} \)

- For a fixed \( \mathcal{F} \), we define an equivalence relation \( \equiv(\mathcal{F}, t) \) on \( t \)-boundaried graphs:

\[
G_1 \equiv(\mathcal{F}, t) G_2 \quad \text{if } \forall G' \in \mathcal{B}^t, \quad \mathcal{F} \preceq_m G' \oplus G_1 \iff \mathcal{F} \preceq_m G' \oplus G_2.
\]

- \( \mathcal{R}(\mathcal{F}, t) \): set of minimum-size representatives of \( \equiv(\mathcal{F}, t) \).

We compute, using DP over a tree decomposition of \( G \), the following parameter for every representative \( R \):

\[
p(G, R) = \min \{ |S| : S \subseteq V(G) \land \text{rep}_{\mathcal{F}, t}(G - S) = R \}
\]

- The number of representatives is \( |\mathcal{R}(\mathcal{F}, t)| = 2^{O_{\mathcal{F}}(t \cdot \log t)} \).

The number of labeled graphs of size \( \leq t \) and \( \text{tw} \leq h \) is \( 2^{O_{\mathcal{H}}(t \cdot \log t)} \). [Baste, Noy, S. 2017]
Algorithm for a connected and planar collection $\mathcal{F}$

- For a fixed $\mathcal{F}$, we define an equivalence relation $\equiv(\mathcal{F}, t)$ on $t$-boundaried graphs:
  
  $G_1 \equiv(\mathcal{F}, t) G_2$ if $\forall G' \in B^t$, $\mathcal{F} \preceq_m G' \oplus G_1 \iff \mathcal{F} \preceq_m G' \oplus G_2$.

- $\mathcal{R}(\mathcal{F}, t)$: set of minimum-size representatives of $\equiv(\mathcal{F}, t)$.

- We compute, using DP over a tree decomposition of $G$, the following parameter for every representative $R$:
  
  $p(G, R) = \min\{|S| : S \subseteq V(G) \land \text{rep}_{\mathcal{F}, t}(G - S) = R\}$

- The number of representatives is $|\mathcal{R}(\mathcal{F}, t)| = 2^{O_{\mathcal{F}}(t \cdot \log t)}$.
  
  # labeled graphs of size $\leq t$ and $\text{tw} \leq h$ is $2^{O_h(t \cdot \log t)}$. [Baste, Noy, S. 2017]

- This gives an algorithm running in time $2^{O_{\mathcal{F}}(\text{tw} \cdot \log \text{tw})} \cdot n^{O(1)}$. 
Algorithm when the input graph $G$ is planar

- **Idea**: get an improved bound on $|\mathcal{R}(F, t)|$.

We use a sphere-cut decomposition of the input planar graph $G$.

- Nice topological properties: each separator corresponds to a noose.
- The number of representatives is $|\mathcal{R}(F, t)| = 2^{O(F(t))}$.
- Number of planar triangulations on $t$ vertices is $2^{O(t)}$.

This gives an algorithm running in time $2^{O(F(tw))} \cdot n^{O(1)}$.

We can extend this algorithm to input graphs $G$ embedded in arbitrary surfaces by using surface-cut decompositions.

[Seymour, Thomas. 1994] [Dorn, Penninkx, Bodlaender, Fomin. 2010]

[Tutte. 1962] [Ruè, S., Thilikos. 2014]
Algorithm when the input graph $G$ is planar

- **Idea** get an improved bound on $|\mathcal{R}(F, t)|$.

- We use a sphere-cut decomposition of the input planar graph $G$.

  [Seymour, Thomas. 1994] [Dorn, Penninkx, Bodlaender, Fomin. 2010]

- Number of planar triangulations on $t$ vertices is $2^{O(t)}$.

  [Tutte. 1962]

- We can extend this algorithm to input graphs $G$ embedded in arbitrary surfaces by using surface-cut decompositions.

  [Rué, S., Thilikos. 2014]
Algorithm when the input graph $G$ is planar

- **Idea** get an improved bound on $|\mathcal{R}(\mathcal{F}, t)|$.

- We use a sphere-cut decomposition of the input planar graph $G$.
  
  [Seymour, Thomas. 1994]  
  [Dorn, Penninkx, Bodlaender, Fomin. 2010]

- Nice topological properties: each separator corresponds to a noose.

![Sphere-cut decomposition diagram](image-url)
Algorithm when the input graph $G$ is planar

- **Idea** get an improved bound on $|\mathcal{R}(\mathcal{F},t)|$.

- We use a sphere-cut decomposition of the input planar graph $G$.
  
  [Seymour, Thomas. 1994] [Dorn, Penninkx, Bodlaender, Fomin. 2010]

- Nice topological properties: each separator corresponds to a noose.

  ![](image)

- The number of representatives is $|\mathcal{R}(\mathcal{F},t)| = 2^{O(\mathcal{F})}$. Number of planar triangulations on $t$ vertices is $2^{O(t)}$.

  [Tutte. 1962]
Algorithm when the input graph $G$ is planar

- **Idea** get an improved bound on $|\mathcal{R}(\mathcal{F}, t)|$.

- We use a sphere-cut decomposition of the input planar graph $G$.
  
  [Seymour, Thomas. 1994] [Dorn, Penninkx, Bodlaender, Fomin. 2010]

- Nice topological properties: each separator corresponds to a noose.

  ![Diagram](image)

  - The number of representatives is $|\mathcal{R}(\mathcal{F}, t)| = 2^{O_{\mathcal{F}}(t)}$.
    Number of planar triangulations on $t$ vertices is $2^{O(t)}$.  
    [Tutte. 1962]

  - This gives an algorithm running in time $2^{O_{\mathcal{F}}(tw)} \cdot n^{O(1)}$. 

Algorithm when the input graph $G$ is planar

- **Idea** get an improved bound on $|\mathcal{R}(\mathcal{F},t)|$.

- We use a sphere-cut decomposition of the input planar graph $G$.
  
  [Seymour, Thomas. 1994] [Dorn, Penninkx, Bodlaender, Fomin. 2010]

- Nice topological properties: each separator corresponds to a noose.

![Sphere-cut decomposition](image)

- The number of representatives is $|\mathcal{R}(\mathcal{F},t)| = 2^{O_F(t)}$.
  Number of planar triangulations on $t$ vertices is $2^{O(t)}$.  
  [Tutte. 1962]

- This gives an algorithm running in time $2^{O_F(tw)} \cdot n^{O(1)}$.

- We can extend this algorithm to input graphs $G$ embedded in arbitrary surfaces by using surface-cut decompositions.
  [Rué, S., Thilikos. 2014]
What’s next about $\mathcal{F}$-DELETION?

Goal

classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.

Concerning the minor version:

We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).

Missing: When $|\mathcal{F}| \geq 2$ (connected): $2 \Theta(tw) \text{ or } 2 \Theta(tw \cdot \log tw)$?

Consider families $\mathcal{F}$ containing disconnected graphs.

Deletion to genus at most $g$: $2 O(g)(tw \cdot \log tw) \cdot n O(1)$.

[Kociumaka, Pilipczuk. 2017]

Concerning the topological minor version:

Dichotomy for $\{H\}$-TM-Deletion when $H$ connected (+planar).

We do not know if there exists some $\mathcal{F}$ such that $\mathcal{F}$-TM-Deletion cannot be solved in time $2^{o(tw^{2})} \cdot n O(1)$ under the ETH.

Conjecture

For every (connected) family $\mathcal{F}$, the $\mathcal{F}$-TM-Deletion problem is solvable in time $2^{O(tw \cdot \log tw)} \cdot n O(1)$. 

25/26
What’s next about \( \mathcal{F}\text{-DELETION} \)?

- **Goal**: classify the (asymptotically) tight complexity of \( \mathcal{F}\text{-M-DELETION} \) and \( \mathcal{F}\text{-TM-DELETION} \) for every family \( \mathcal{F} \).
What’s next about $\mathcal{F}$-Deletion?

- **Goal** classify the (asymptotically) tight complexity of $\mathcal{F}$-$\mathbf{M}$-Deletion and $\mathcal{F}$-$\mathbf{TM}$-Deletion for every family $\mathcal{F}$.

- Concerning the **minor** version:

  - When $|\mathcal{F}| \geq 2$ (connected):
    - $2^{\Theta(tw)}$ or $2^{\Theta(tw \cdot \log tw)}$?

  - Consider families $\mathcal{F}$ containing disconnected graphs.

  - Deletion to genus at most $g$:
    - $2^{O(g)(tw \cdot \log tw)} \cdot n^{O(1)}$.

- [Kociumaka, Pilipczuk. 2017]

- Concerning the topological minor version:

  - Dichotomy for $\{H\}$-$\mathbf{TM}$-Deletion when $H$ connected (+planar).

  - We do not know if there exists some $\mathcal{F}$ such that $\mathcal{F}$-$\mathbf{TM}$-Deletion cannot be solved in time $2^{o(tw^2)} \cdot n^{O(1)}$ under the ETH.

- Conjecture:

  - For every (connected) family $\mathcal{F}$, the $\mathcal{F}$-$\mathbf{TM}$-Deletion problem is solvable in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$. 
What’s next about $\mathcal{F}$-Deletion?

- **Goal** classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.

- Concerning the **minor** version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
What’s next about $\mathcal{F}$-DELETION?

- **Goal** classify the (asymptotically) tight complexity of $\mathcal{F}$-M-DELETION and $\mathcal{F}$-TM-DELETION for every family $\mathcal{F}$.

- Concerning the **minor** version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
  - **Missing:** When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(tw)}$ or $2^{\Theta(tw \cdot \log tw)}$?

Concerning the **topological minor** version:

- Dichotomy for $\{H\}$-TM-Deletion when $H$ connected (+planar).
- We do not know if there exists some $\mathcal{F}$ such that $\mathcal{F}$-TM-Deletion cannot be solved in time $2^{o(tw^2)} \cdot n^{O(1)}$ under the ETH.

Conjecture

For every (connected) family $\mathcal{F}$, the $\mathcal{F}$-TM-Deletion problem is solvable in time $2^{O(tw \cdot \log tw)} \cdot n^{O(1)}$. 

25/26
What’s next about \( \mathcal{F} \text{-} \text{Deletion} \)?

- **Goal** classify the (asymptotically) tight complexity of \( \mathcal{F} \text{-M-Deletion} \) and \( \mathcal{F} \text{-TM-Deletion} \) for every family \( \mathcal{F} \).

- Concerning the **minor** version:
  - We obtained a tight dichotomy when \( |\mathcal{F}| = 1 \) (connected).
  - **Missing**: When \( |\mathcal{F}| \geq 2 \) (connected): \( 2^{\Theta(tw)} \) or \( 2^{\Theta(tw \cdot \log tw)} \)?
  - Consider families \( \mathcal{F} \) containing disconnected graphs.

- \([Kociumaka, Pilipczuk. 2017]\) Concerning the topological minor version:
  - Dichotomy for \{\( H \)\} \text{-TM-Deletion} when \( H \) connected (+planar).
  - We do not know if there exists some \( \mathcal{F} \) such that \( \mathcal{F} \text{-TM-Deletion} \) cannot be solved in time \( 2^{o(tw^2)} \cdot n^{O(1)} \) under the ETH.

Conjecture

For every (connected) family \( \mathcal{F} \), the \( \mathcal{F} \text{-TM-Deletion} \) problem is solvable in time \( 2^{O(tw \cdot \log tw)} \cdot n^{O(1)} \).
What's next about $\mathcal{F}$-Deletion?

- **Goal** classify the (asymptotically) tight complexity of $\mathcal{F}$-$\text{M-Deletion}$ and $\mathcal{F}$-$\text{TM-Deletion}$ for every family $\mathcal{F}$.

- Concerning the **minor** version:
  - We obtained a tight dichotomy when $|\mathcal{F}| = 1$ (connected).
  - **Missing:** When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(tw)}$ or $2^{\Theta(tw \cdot \log tw)}$?
  - Consider families $\mathcal{F}$ containing disconnected graphs. Deletion to genus at most $g$: $2^{O_g(tw \cdot \log tw)} \cdot n^{O(1)}$. [Kociumaka, Pilipczuk. 2017]
What’s next about $\mathcal{F}$-Deletion?

- **Goal** classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion for every family $\mathcal{F}$.

Concerning the **minor** version:

- We obtained a **tight dichotomy** when $|\mathcal{F}| = 1$ (connected).
- **Missing**: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(tw)}$ or $2^{\Theta(tw \cdot \log tw)}$?

- Consider families $\mathcal{F}$ containing **disconnected graphs**.
  Deletion to **genus at most $g$**: $2^{O_g(tw \cdot \log tw)} \cdot n^{O(1)}$. [Kociumaka, Pilipczuk. 2017]

Concerning the **topological minor** version:
What’s next about $\mathcal{F}$-Deleting?

- **Goal** classify the (asymptotically) tight complexity of $\mathcal{F}$-M-Deleting and $\mathcal{F}$-TM-Deleting for every family $\mathcal{F}$.

- Concerning the **minor** version:
  - We obtained a **tight dichotomy** when $|\mathcal{F}| = 1$ (connected).
  - **Missing**: When $|\mathcal{F}| \geq 2$ (connected): $2^{\Omega(tw)}$ or $2^{\Theta(tw \cdot \log tw)}$?
  - Consider families $\mathcal{F}$ containing **disconnected** graphs.
    - Deletion to genus at most $g$: $2^\Theta(tw \cdot \log tw) \cdot n^\Theta(1)$. [Kociumaka, Pilipczuk. 2017]

- Concerning the **topological minor** version:
  - Dichotomy for $\{H\}$-TM-Deleting when $H$ connected ( + planar).
What’s next about $\mathcal{F}$-$\text{Deletion}$?

- **Goal** classify the (asymptotically) tight complexity of $\mathcal{F}$-$\text{M-Deletion}$ and $\mathcal{F}$-$\text{TM-Deletion}$ for **every** family $\mathcal{F}$.

- Concerning the **minor** version:
  - We obtained a **tight dichotomy** when $|\mathcal{F}| = 1$ (connected).
  - **Missing:** When $|\mathcal{F}| \geq 2$ (connected): $2^{\Theta(tw)}$ or $2^{\Theta(tw \cdot \log tw)}$?
  - Consider families $\mathcal{F}$ containing **disconnected graphs**.
    - Deletion to genus at most $g$: $2^{O_g(tw \cdot \log tw)} \cdot n^{O(1)}$. [Kociumaka, Pilipczuk. 2017]

- Concerning the **topological minor** version:
  - Dichotomy for $\{H\}$-$\text{TM-Deletion}$ when $H$ connected (+planar).
  - We do not know if there exists some $\mathcal{F}$ such that $\mathcal{F}$-$\text{TM-Deletion}$ **cannot** be solved in time $2^{o(tw^2)} \cdot n^{O(1)}$ under the ETH.
What’s next about \( \mathcal{F}\text{-DELETION} \)?

- **Goal** classify the (asymptotically) tight complexity of \( \mathcal{F}\text{-M-DELETION} \) and \( \mathcal{F}\text{-TM-DELETION} \) for every family \( \mathcal{F} \).

- Concerning the **minor** version:
  - We obtained a tight dichotomy when \( |\mathcal{F}| = 1 \) (connected).
  - **Missing:** When \( |\mathcal{F}| \geq 2 \) (connected): \( 2^{\Theta(tw)} \) or \( 2^{\Theta(tw \cdot \log tw)} \)?
  - Consider families \( \mathcal{F} \) containing disconnected graphs.
    Deletion to genus at most \( g \): \( 2^{O_g(tw \cdot \log tw)} \cdot n^{O(1)} \). [Kociumaka, Pilipczuk. 2017]

- Concerning the **topological minor** version:
  - Dichotomy for \( \{H\}\text{-TM-DELETION} \) when \( H \) connected (+planar).
  - We do not know if there exists some \( \mathcal{F} \) such that \( \mathcal{F}\text{-TM-DELETION} \) cannot be solved in time \( 2^{o(tw^2)} \cdot n^{O(1)} \) under the ETH.
  - **Conjecture** For every (connected) family \( \mathcal{F} \), the \( \mathcal{F}\text{-TM-DELETION} \) problem is solvable in time \( 2^{O(tw \cdot \log tw)} \cdot n^{O(1)} \).
Gràcies!