Query Preserving Watermarking Schemes for Locally Treelike Databases

Agnishom Chattopadhyay
Chennai Mathematical Institute, Chennai, India
UMI ReLaX, Indo-French joint research unit

M. Praveen
Chennai Mathematical Institute, Chennai, India
UMI ReLaX, Indo-French joint research unit

Abstract

Watermarking is a way of embedding information in digital documents. Much research has been done on techniques for watermarking relational databases and XML documents, where the process of embedding information shouldn't distort query outputs too much. Recently, techniques have been proposed to watermark some classes of relational structures preserving first-order and monadic second order queries. For relational structures whose Gaifman graphs have bounded degree, watermarking can be done preserving first-order queries.

We extend this line of work and study watermarking schemes for other classes of structures. We prove that for relational structures whose Gaifman graphs belong to a class of graphs that have locally bounded tree-width and is closed under minors, watermarking schemes exist that preserve first-order queries. We use previously known properties of logical formulas and graphs, and build on them with some technical work to make them work in our context. This constitutes a part of the first steps to understand the extent to which techniques from algorithm design and computational learning theory can be adapted for watermarking.

2012 ACM Subject Classification Security and privacy → Information accountability and usage control; Theory of computation → Finite Model Theory; Information systems → Relational database model

Keywords and phrases Locally bounded tree-width, closure under minors, first-order queries, watermarking

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2019.36

Related Version A full version of the paper is available at `https://arxiv.org/abs/1909.11369`.

Funding *M. Praveen*: Partially supported by a grant from the Infosys foundation.

Acknowledgements The authors thank B. Srivathsan and K. Narayan Kumar for feedback on the draft.

1 Introduction

Watermarking of digital content can be used to check intellectual property violations. The idea is to embed some information, such as a binary string, in the digital content in such a way that it is not easily apparent to the end user. If the legitimate owner of the digital content suspects a copy to be stolen, they should be able to retrieve the embedded information, even with limited access to the stolen copy, even if it has been tampered to remove the embedded information. Here there are two opposing goals. One is to be able to embed large amount of information. The other is to ensure that the embedding doesn't distort the content too much.

39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019).
Editors: Arkadev Chattopadhyay and Paul Gastin; Article No. 36; pp. 36:1–36:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 `EmployeeTable` of Ex. 1.

(a) The original `EmployeeTable`.

FirstName	City	Salary
John	Chennai	10,000
Arjun	Coimbatore	20,000
Pooja	Chennai	15,000
Neha	Vellore	30,000
Padma	Coimbatore	20,000

(b) A distorted `EmployeeTable`.

FirstName	City	Salary
John	Chennai	10,001
Arjun	Coimbatore	19,999
Pooja	Chennai	14,999
Neha	Vellore	30,000
Padma	Coimbatore	20,001

There can be many ways to measure how much distortion is acceptable. In [1], embedding is performed by flipping bits in numerical attributes while preserving the mean and variance of all numerical attributes. There are other works that focus on the specific use of the digital content: in [14], the digital content consists of graphs whose vertices represent locations and weighted edges represent distance between locations. It is shown that information can be embedded in such a way that the shortest distance between any two locations is not distorted too much.

We study embedding information in relational databases such that the distortion on query outputs is bounded.

▶ **Example 1.** The table `EmployeeTable` shown in Table 1(a) is an example of a database instance of an organization's record of employees.

Consider the following query parameterized by the variable x.

$\psi(x) \equiv$ `select FirstName,Salary from EmployeeTable where City=`x

If we substitute the variable x with a particular city c, the above query lists the salaries of individuals working in that city. Let $\texttt{total}(c)$ be the sum of all salaries listed by the query $\psi(c)$. We want to hide data in `EmployeeTable` by distorting the `Salary` field. Let $\texttt{total}'(c)$ be the sum of all salaries listed by the query $\psi(c)$ run on the distorted database. We say that the distortion *preserves the query* $\psi(x)$ if there is a constant B such that for any city c, the absolute value of the difference between $\texttt{total}(c)$ and $\texttt{total}'(c)$ is bounded by B.

Assuming that we can distort each employee's salary by at most 1 unit and we wish to maintain the bound B to be 0, our options are the following: increase the salary of `John` by 1 and decrease the salary of `Pooja` by 1 or vice-versa. Similarly for `Arjun` and `Padma`. This gives us four different ways distort the data base. These distortions are designed to preserve only the query $\psi(x)$. If a different query is run on the same distorted databases, the results may vary widely. Suppose the organization distributes the four distorted databases among it's branches. If a stolen copy of the database is found, the organization can run the query $\psi(x)$ on the stolen copy. By observing the salaries of `John`, `Pooja`, `Arjun` and `Padma` and comparing them with the values from the original database, one can narrow down on the branches where the leakage happened. The organization only needs to run the query $\psi(x)$ on the suspected stolen copy, just like any normal consumer of the database. This is important since the entity possessing the stolen copy may not allow full access to its copy of the database. We say a watermarking scheme is *scalable* if for larger databases, there are larger number of ways to distort the database, while still preserving queries of interest.

In [13], meta theorems are proved regarding the existence of watermarking schemes for classes of databases preserving queries written in classes of query languages. The Gaifman graph of a database is a graph whose set of vertices is the set of elements in the universe of the database. There is an edge between two elements if the two elements participate in some tuple in the database. For databases with unrestricted structures, even simple queries can't be preserved; see [13, Theorem 3.6] and [12, Example 3]. Preserving queries written in powerful query languages and handling databases with minimum restrictions on their structure are conflicting goals. For databases whose Gaifman graphs have bounded degree, first-order queries can be preserved [13]. It is also shown that for databases whose Gaifman graphs are similar to trees, MSO queries can be preserved. The similarity of a graph to a tree is measured by tree-width. For example, XML documents are trees and have tree-width 1.

Contributions. We prove that watermarking schemes exist for databases whose Gaifman graphs belong to a class of graphs that have locally bounded tree-width and is closed under minors, preserving unary first-order queries. Classes of graphs with bounded degree are contained in this class. A graph G has locally bounded tree-width if it satisfies the following property: there exists a function f such that for any vertex v and any number r, the sphere of radius r around v induces a subgraph on G whose tree-width is at most $f(r)$.

Why first-order logic? The pivotal Codd's theorem [3] states that first-order logic is expressively equivalent to relational algebra, and relational algebra is the basis of standard relational database query languages.

Why locally bounded treewidth? Classes of graphs with locally bounded treewidth are good starting points to start using techniques from algorithm design and computational learning theory in other areas. Seese [18] proved that first-order properties can be decided in linear time for graphs of bounded degree. Baker [2] showed efficient approximation algorithms for some specific hard problems, when restricted to planar graphs. Eppstein [7] showed that Baker's technique continues to work in a bigger class of graphs: it suffices for the class of graphs to have locally bounded tree-width and additionally, the class should be closed under minors. Frick and Grohe [9] showed that on any class of graphs with locally bounded tree-width, any problem definable in first-order logic can be decided efficiently[1]. For problems definable in first-order logic, the classes of graphs for which efficient algorithms exist was then extended to bigger and bigger classes: excluded minors [8], locally excluded minors [5], bounded expansion, locally bounded expansion [6] and nowhere dense [11]. It is now known that nowhere dense graphs are the biggest class of graphs for which there are efficient algorithms for first-order definable problems [6, 15, 11], provided some complexity theoretic assumption are true. Results related to computational learning theory have been proved in [12] for classes of graphs with locally bounded treewidth. Recently, similar results have been proven for nowhere dense classes of graphs [17].

Why unary queries? Some of the techniques we have used are difficult to extend to non-unary queries. Some technical details about this are discussed in the conclusion.

Related Works. The fundamental definitions of what it means for a watermarking scheme to be scalable and preserving a query was given in [14]. It was shown in [14] that on weighted graphs, scalable watermarking schemes exist preserving shortest distance between vertices. The adversarial model was also introduced in [14], where the person possessing the stolen

[1] Here, efficiency means fixed parameter tractability; see [9] for details.

copy can introduce additional distortion to evade detection. It is shown in [14] that a watermarking scheme for the non-adversarial model can be transformed to work for the adversarial model, under some assumption about the quantity of distortion introduced by the person trying to evade detection, and the amount of knowledge the person possesses. Gross-Amblard [13] adapted these definitions for relational structures of any vocabulary and any query written in Monadic Second Order (MSO) logic, and showed results about classes of structures of bounded degree and tree-width. Gross-Amblard [13] also provided the insight that existence of scalable watermarking schemes preserving queries from a certain language is closely related to learnability of queries in the same language. We make use of this insight in our work. Grohe and Turán [12] proved that MSO-definable families of sets in graphs of bounded tree-width have bounded Vapnik-Chervonenkis (VC) dimension, which has well known connections in computational learnability theory. It is also shown in [12] that on classes of graphs with locally bounded tree-width, first-order definable families of sets have bounded VC dimension.

2 Preliminaries

Relational databases. A signature (or database schema) τ is a finite set of relation symbols $\{R_1, \dots, R_t\}$. We denote by r_i the arity of R_i for every $i \in \{1, \dots, t\}$. A τ-structure $G = (V, R_1^G, \dots R_t^G)$ (or database instance) consists of a set V called the universe, and an interpretation $R_i^G \subseteq V^{r_i}$ for every relation symbol R_i. For a fixed $s \in \mathbb{N}$, a *weighted structure* (G, W) is a finite structure G together with a weight function W, which is a partial function from V^s to $\mathbb{N}$, that maps a s-tuple $\bar{b}$ to its weight $W(\bar{b})$.

First Order and Monadic Second Order Queries. An atomic formula is a formula of the form $x = y$ or $R(x_1, \dots x_r)$, where $x, y, x_1 \dots x_r$ are variables and R is an r-ary relational symbol in τ. First-order (FO) formulas are formulas built from atomic formulas using the usual boolean connectives and existential and universal quantification over the elements of the universe of a structure.

Monadic Second Order (MSO) logic extends first-order logic by allowing existential and universal quantifications over subsets of the universe. Formally, there are two types of variables. Individual variables, which are interpreted by elements of the universe of a structure, and set variables, which are interpreted by subsets of the universe of a structure. In addition to the atomic formulas of first-order logic mentioned in the previous paragraph, MSO has atomic formulas of the form $X(x)$, saying that the element interpreting the individual variable x is in the set interpreting the set variable X. Furthermore, MSO has quantification over both individual and set variables.

The quantifier rank, denoted $\mathrm{qr}(\psi)$ of a formula ψ is the maximum number of nested quantifiers in ψ. A free variable of a formula ψ is a variable x that does not occur in the scope of a quantifier. The set of free variables of a formula ψ is denoted by $\mathrm{free}(\psi)$. A sentence is a formula without free variables. We write $\psi(x_1, \dots, x_r)$ to indicate that $\mathrm{free}(\psi) \subseteq \{x_1, \dots, x_r\}$. We denote the size of ψ by $||\psi||$. We only work with formulas that have free individual variables, but not free set variables. Given a vector $\overline{x} = \langle x_1, \dots, x_s \rangle$ of variables, a formula $\psi(\overline{x})$ and a structure G, we denote by $\psi(G) = \{\overline{a} \in V^s \mid G \models \psi(\overline{a})\}$ the set of tuples of elements from the universe V of G that can be assigned to the variables $\overline{x}$ to satisfy $\psi(\overline{x})$.

Suppose $\psi(\overline{x}, \overline{y})$ is a formula with two distinguished vectors of free variables $\overline{x}$ of length r and $\overline{y}$ of length s. We call $\psi(\overline{x}, \overline{y})$ a s-ary query with r parameters. Given a structure G, we call $\psi(\overline{a}, G) = \{\overline{b} \in V^s \mid G \models \psi(\overline{a}, \overline{b})\}$ the *output of the query* $\psi(\overline{x}, \overline{y})$ *with parameter*

$\overline{a}$. We refer to r (resp. s), the length of $\overline{x}$ (resp. y), as the *input length* (resp. the *output length*) of $\psi(\overline{x}, \overline{y})$. Given a weighted structure (G, W), a parametric query $\psi(\overline{x}, \overline{y})$ and a parameter $\overline{a}$, we extend the weight function W to weights of query outputs by defining $W(\psi(\overline{a}, G)) = \Sigma_{\overline{b} \in \psi(\overline{a}, G)} W(\overline{b})$. For a given structure G and a query $\psi(\overline{x}, \overline{y})$, we define $U = \bigcup_{\overline{a} \in V^r} \psi(\overline{a}, G)$ to be the set of *active tuples*.

Watermarking schemes. Suppose $c, d \in \mathbb{N}$. A weighted structure (G, W') is a *c-local distortion* of another weighted structure (G, W) if for all $\overline{b} \in V^s$, $|W'(b) - W(b)| \leq c$. The weighted structure (G, W') is a *d-global distortion* of (G, W) with respect to a query $\psi(\overline{x}, \overline{y})$ if and only if, for all $\overline{a} \in V^r$, $|W'(\psi(\overline{a}, G)) - W(\psi(\overline{a}, G))| \leq d$.

▶ **Definition 2** ([14, 13]). *Given a class of weighted structures $\mathcal{K}$ and a query $\psi(\overline{x}, \overline{y})$, a* watermarking scheme preserving $\psi(\overline{x}, \overline{y})$ *is a pair of algorithms $\mathcal{M}$ (called the marker) and $\mathcal{D}$ (called the detector) along with a function $f : \mathbb{N} \to \mathbb{N}$ and a constant $d \in \mathbb{N}$ such that:*

- *The marker $\mathcal{M}$ takes as input a weighted structure $(G, W) \in \mathcal{K}$ and a mark μ, which is a bit string of length $f(|U|)$, where U is the set of active tuples. It outputs a weighted structure $(G, W_\mu) \in \mathcal{K}$ such that (G, W_μ) is a 1-local and d-global distortion of (G, W) for the query $\psi(\overline{x}, \overline{y})$.*
- *The detector $\mathcal{D}$ is given (G, W), the original structure as input and has access to an oracle that runs queries of the form $\psi(\overline{x}, \overline{y})$ on (G, W_μ). The output of $\mathcal{D}$ is the hidden mark μ.*

Intuitively, the marker takes a bit string and hides it in the database by distorting weights. The detector detects the hidden mark by observing the weights and comparing it with the original weights. The term $f(|U|)$ denotes the length of the bit string that is hidden in the database by the marker. We call a watermarking scheme *scalable* if the function f grows at least as fast as some fractional power asymptotically. For example, the scheme is scalable if $f(n) = \sqrt{n}$ for all n, but not scalable if $f(n) = \log n$ for all n. We will mention later why scalability is important in situations where adversaries try to erase watermarks. Note that the algorithm $\mathcal{D}$ interacts with the marked database (G, W_μ) only through $\psi(\overline{x}, \overline{y})$ queries. Hence, it is not worthwhile distorting the weights of tuples that are not active.

Continuing Example 1, the query $\psi(x)$ given there can be written in First-order as $(\psi(\langle city \rangle, \langle name, salary \rangle) = \texttt{EmployeeTable}(name, city, salary)$. The set of active tuples is $U = \{\langle \text{John, 10000} \rangle, \langle \text{Arjun, 20000} \rangle, \langle \text{Pooja, 15000} \rangle, \langle \text{Neha, 30000} \rangle, \langle \text{Padma,20000} \rangle\}$. We can increase the salary of `John` by 1 and decrease the salary of `Pooja` by 1 or vice-versa. Similarly for `Arjun` and `Padma`. This gives 4 different distortions that are 1-local and 0-global. The marker algorithm can take a mark, which is a bit string of length 2, so there are 4 possible marks. The marker can assign these 4 marks to the 4 possible distortions. The detector can observe the changes to the salaries by querying the distorted copy and comparing the results with the original database. The detector can compute the hidden mark by accessing the assignment of the 4 marks to the 4 possible distortions given by the marker. For any instance database of this signature, we can pair off an employee of a city with another employee in the same city and use one such pair to encode one bit of a watermark to be hidden. If there are n active tuples, we can encode $\frac{n}{2}$ bits, assuming that there are at least two employees in each city. For this watermarking scheme, the function f is defined as $f(n) = \frac{n}{2}$ and this is a scalable scheme.

Watermarking schemes can also be put in a context where there are adversaries who know that there is some hidden mark and try to prevent the detector algorithm from working properly, by distorting the database further. Instead of the oracle running queries on (G, W_μ), the queries are run on (G, W'_μ), which is a distortion of (G, W_μ). The detector has to still detect the hidden mark μ correctly. Under some assumptions about the quantity of distortion between (G, W_μ) and (G, W'_μ), watermarking schemes that work in non-adversarial models can be transformed to work in adversarial models; we refer the interested readers to [14, 13]. The correctness of such transformations depend on probabilistic arguments, where scalability helps. With bigger watermarks that are hidden to begin with, there is more room to play around with the distortions introduced by the adversaries.

3 Watermarking schemes

In this section, we prove that scalable watermarking schemes exist for some type of structures. First we prove that if the Gaifman graphs belong to a class of graphs with bounded tree-width, then scalable water marking schemes exist preserving unary MSO queries. Then we prove that if the Gaifman graphs belong to a class of graphs that is closed under minors and that has locally bounded tree-width, then scalable water marking schemes exist preserving unary FO queries.

3.1 MSO Queries on Structures with Bounded Tree-width

3.1.1 Trees, Tree Automata and Clique-width

We begin by reviewing some concepts and known results that are needed.

A binary tree is a $\{S_1, S_2, \preceq\}$-structure, where S_1, S_2 and $\preceq$ are binary relation symbols. A tree $\mathcal{T} = (T, S_1^\mathcal{T}, S_2^\mathcal{T}, \preceq^\mathcal{T})$ has a set of nodes T, a left child relation $S_1^\mathcal{T}$ and a right child relation $S_2^\mathcal{T}$. Relation $\preceq^\mathcal{T}$ stands for the transitive closure of $S_1^\mathcal{T} \cup S_2^\mathcal{T}$, the tree-order relation. Given a finite alphabet Σ, let $\tau(\Sigma) = \{S_1, S_2, \preceq\} \cup \{P_a | a \in \Sigma\}$ where for all $a \in \Sigma$, P_a is a unary symbol. A Σ-tree is a structure $\mathcal{T} = (T, S_1^\mathcal{T}, S_2^\mathcal{T}, \preceq^\mathcal{T}, (P_a^\mathcal{T})_{a\in\Sigma})$, where the restriction $(T, S_1^\mathcal{T}, S_2^\mathcal{T}, \preceq^\mathcal{T})$ is a binary tree and for each $v \in T$ there exists exactly one $a \in \Sigma$ such that $v \in P_a^\mathcal{T}$. We denote this unique a by $\sigma^\mathcal{T}(v)$. Intuitively, this represents the labelling of nodes by letters from Σ where $\sigma^\mathcal{T}(v)$ is the label for the node v. We consider trees with a finite number of pebbles placed on nodes. The pebbles are considered to be distinct: pebble 1 on node v_1 and pebble 2 on node v_2 is not the same as pebble 1 on node v_2 and pebble 2 on node v_1. For some $k \geq 1$, let $\Sigma_k = \Sigma \times \{0, 1\}^k$. This extended alphabet denotes the position of the pebbles in the tree. Suppose $\mathcal{T}$ is a Σ-tree and k pebbles are placed on the nodes $\overline{v} = \langle v_1, \dots, v_k \rangle$. Then $\mathcal{T}_{\overline{v}}$ is the Σ_k-tree with the same underlying tree as $\mathcal{T}$ and $\sigma^{\mathcal{T}_{\overline{v}}}(u) = (\sigma^\mathcal{T}(u), \alpha_1, \dots, \alpha_k)$ where $\alpha_i = 1$ if and only if $u = v_i$.

A Σ-tree automaton is a tuple $A = (Q, \delta, F)$ where Q is a set of states and $F \subseteq Q$ are the accepting states. The function $\delta : ((Q \cup \{*\})^2 \times \Sigma) \to Q$ is the transition function, where $*$ is a special symbol not in Q. A run $\rho : T \to Q$ of A on a Σ-tree $\mathcal{T}$ is a function satisfying the following conditions.

- If v is a leaf then $\rho(v) = \delta(*, *, \sigma^\mathcal{T}(v))$.
- If v has two children u_1 and u_2, then $\rho(v) = \delta(\rho(u_1), \rho(u_2), \sigma^\mathcal{T}(v))$.
- If v has only a left child u then $\rho(v) = \delta(\rho(u), *, \sigma^\mathcal{T}(v))$.
- Similarly if v has only a right child.

If v is the root of $\mathcal{T}$, a run ρ of A on $\mathcal{T}$ is an accepting run if $\rho(v) \in F$. A $\Sigma_{(r+s)}$ tree automaton defines a s-ary query with r parameters. We denote by $A(\overline{a}, \mathcal{T}) = \{\overline{b} \in T^s \mid A$ has an accepting run on $\mathcal{T}_{\overline{a}\overline{b}}\}$ the output of the query A on $\mathcal{T}$ with parameter $\overline{a}$. It is well known that MSO queries and tree automata have the same expressive power.

Clique-width. A well-known notion of measuring the similarity of a graph to a tree is its *tree-width*. Many nice properties of trees carry over to classes of structures of bounded tree-width. For our purposes, we use *clique-width*, a related notion. It is well known that a structure of tree-width at most k has clique-width at most 2^k [4].

A k-colored structure is a pair (G, γ) consisting of a structure G and a mapping $\gamma : V \to \{1, \dots, k\}$. A basic k-colored structure is a k-colored structure (G, γ) where $|V| = 1$ and $R^G = \emptyset$ for all R. We let Γ_k be the smallest class of structures that contain all basic k-colored structures and is closed under the following operations:

- *Union*: take two k-colored structures on disjoint universes and form their union.
- $(i \to j)$ *recoloring, for* $1 \le i, j \le k$: take a k-colored structure and recolor all vertices colored i to j.
- $(R, i_1, \dots i_r)$*-connecting, for every* $r \ge 1$*, every* r*-ary relation symbol* R *and every* $1 \le i_1, \dots i_r \le k$: take a k-colored structure (G, γ) and add all tuples $\langle v_1, \dots v_r \rangle \in V^r$ with $\gamma(v_j) = i_j$ for $1 \le j \le r$ to $\mathcal{R}^G$.

The *clique-width* of a structure G, denoted by $\mathrm{cw}(G)$, is the minimum k such that there exists a k-coloring $\gamma : V \to \{1, \dots k\}$ such that $(G, \gamma) \in \Gamma_k$.

For every k-colored structure $(G, \gamma) \in \Gamma_k$ we can define a binary, labeled parse-tree in a straightforward way. The leaves of this tree are the elements of G labeled by their color, and each inner node is labeled by the operation it corresponds to. A parse-tree (also called a clique decomposition) of a structure G of clique-width k is a parse tree of some $(G, \gamma) \in \Gamma_k$. For the next lemma, it is important to note that if $\mathcal{T}$ is a parse-tree for a structure G, then $V \subseteq T$.

▶ **Lemma 3** ([12, Lemma 16])**.** *Let $k \ge 1$. For every MSO formula $\varphi(\overline{x})$ there is a MSO formula $\tilde{\varphi}(\overline{x})$ such that for every structure G of clique-width k and for every parse-tree $\mathcal{T}$ of G we have $\varphi(G) = \tilde{\varphi}(\mathcal{T})$. Furthermore, there are constants c, d (only depending on k and the signature τ) such that $||\tilde{\varphi}|| \le c||\varphi||$ and $\mathrm{qr}(\tilde{\varphi}) \le \mathrm{qr}(\varphi) + d$.*

3.1.2 Watermarking Schemes to Preserve MSO Queries on Structures With Bounded Tree-width

Now we prove that there are scalable watermarking schemes that work for structures from classes with bounded tree-width and preserve a given MSO query. At a high level, the idea is the following: the given MSO query is converted to an equivalent tree automaton. If the number of active elements is large compared to the number of states in the automaton, we can select pairs of elements that can't be distinguished by the automaton. Either both the elements are in the output of the query or none of them are. Hence, distorting one of them by a positive amount and the other one by a negative amount will not contribute to the global distortion.

We begin with the following lemma, which says that if a tree automaton runs on a large tree, we can find large number of pairs of nodes that are "similar" with respect to the automaton. Some of the proofs have been skipped here due to space constraints; they can be found in the full version. A similar result is proved and used in [12] to show that MSO-definable families of sets in graphs of bounded tree-width have bounded Vapnik-Chervonenkis (VC) dimension. The similarity of the following result with that of [12] hints at some possible connections between watermarking schemes and VC dimension.

▶ **Lemma 4.** *Let A be a Σ_{r+1} tree automaton with m states. Let $\mathcal{T}$ be a Σ tree. Suppose $Y \subseteq T$ is a set of nodes of T with at least $2m^m+2$ elements*[2]*. Then, there exists $n = \lfloor \frac{|Y|}{4m^m+4} \rfloor$ pairwise disjoint sets of nodes $V_1, V_2, \ldots, V_n \subseteq T$ and pairs $(b_i, b'_i) \in (V_i \cap Y)^2$ of distinct nodes for all $i \in \{1, \ldots, n\}$ satisfying the following property: for every $\overline{a} = \langle a_1, a_2, \ldots, a_r \rangle \in T^r$ and every $i \in \{1, \ldots, n\}$, if $\{a_1, a_2, \ldots, a_r\} \cap V_i = \emptyset$ then the runs of A on $\mathcal{T}_{\overline{a}b_i}$ and $\mathcal{T}_{\overline{a}b'_i}$ label the tree roots with the same state.*

The following result is proved in [13], but the proof in that paper used a variant of Lemma 4 whose proof has an error. We give a proof with a different constant factor.

▶ **Theorem 5.** *Suppose $\mathcal{K}$ is a class of structures with bounded clique-width. Suppose $\psi(\overline{x}, y)$ is a unary MSO query of input length r, where all the free variables are individual variables. Then, there exists a scalable watermarking scheme preserving $\psi(\overline{x}, y)$ on structures in $\mathcal{K}$.*

Proof. Suppose G is a structure in $\mathcal{K}$, so it has bounded clique-width. From Lemma 3, we get an MSO formula $\tilde{\psi}(\overline{x}, y)$, which can be interpreted on clique decompositions of G to get the same effect as interpreting $\psi(\overline{x}, y)$ on G. We then get an automaton A equivalent to $\tilde{\psi}(\overline{x}, y)$. Let U be the set of active tuples of G for the query $\psi(\overline{x}, y)$. Now we apply Lemma 4, setting $\mathcal{T}$ to be a clique decomposition of G and Y to be the set of active tuples U. We get n pairs $(b_1, b'_1), (b_2, b'_2), \ldots, (b_n, b'_n)$, where n is a constant fraction of $|U|$. Given a weight function W for G and a mark $\mu : \{0,1\}^n$, we define the new weight function W' as follows. We set $(W'(b_i), W'(b'_i)) = (W(b_i)+1, W(b'_i)-1)$ if $\mu(i) = 0$ and $(W'(b_i), W'(b'_i)) = (W(b_i)-1, W(b'_i)+1)$ if $\mu(i) = 1$. For all other elements, W' is same as W. The modified weight function W' has local distortion bounded by 1 by construction. The detector can recover the bits of the mark μ by querying the original and distorted databases and noting the differences in weights assigned to active tuples by W and W'. We will show that it has global distortion bounded by r, the input length of $\psi(\overline{x}, y)$.

Suppose $\overline{a} = \langle a_1, a_2, \ldots, a_r \rangle$ is used as input parameter to the query $\psi(\overline{x}, y)$ on G and (b_i, b'_i) is a pair selected from a set V_i as in Lemma 4. If $\{a_1, \ldots, a_r\} \cap V_i = \emptyset$, then the runs of A on $\mathcal{T}_{\overline{a}b_i}$ and $\mathcal{T}_{\overline{a}b'_i}$ end in the same state. Hence, $b_i \in \psi(\overline{a}, G)$ iff $b'_i \in \psi(\overline{a}, G)$. This means that either both b_i and b'_i are in $\psi(\overline{a}, G)$ or both of them are absent. Hence, the distortion on b_i, b'_i cancel each other, provided $\{a_1, \ldots, a_r\} \cap V_i = \emptyset$. Hence, a pair (b_i, b'_i) may contribute to the global distortion only when $\{a_1, \ldots, a_r\} \cap V_i \neq \emptyset$. Since all the V_i are mutually disjoint and there are at most r elements in $\{a_1, \ldots, a_r\}$, the global distortion is at most r. ◀

Since bounded tree-width implies bounded clique-width, the above result also holds for classes of structures with bounded tree-width.

3.2 FO Queries on Minor Closed Structures with Locally Bounded Tree-width

In this section, we consider structures whose Gaifman graphs belong to a class of graphs that has bounded local tree-width and is closed under minors. We prove that scalable watermarking schemes exist preserving unary first-order queries. We use concepts and techniques from [12] where it is proved that in similar classes of graphs, sets definable by unary first order formulas have bounded VC dimension. It is observed in [12] that this result extends to non-unary formulas. For this extension, [12] uses a generic result from model

[2] A similar result is stated in [13] with $4m$ elements, but there is an error in the proof; see the full version for details.

theory that deals with VC dimension and doesn't use Gaifman graphs. So far, there are no such generic results about watermarking schemes yet. We have not yet found ways to extend our results on watermarking to non-unary queries.

3.2.1 Gaifman's Locality and Locally Bounded Tree-width

First we review some concepts and known results that we use. Given a structure $G = (V, R_1^G, \dots, R_t^G)$, its *Gaifman graph* is the undirected graph (V, E), where $(v_1, v_2) \in E$ if there is a relation R_i in G and a tuple $\overline{v} \in R_i$ such that v_1 and v_2 appear in $\overline{v}$. The distance between two elements, denoted $d(.,.)$, in a structure is defined to be the shortest distance between them in the Gaifman graph. The distance between two tuples of elements $\overline{v_1}, \overline{v_2}$ is $d(\overline{v_1}, \overline{v_2}) = \min\{d(v_1, v_2) \mid v_1 \in \overline{v_1}, v_2 \in \overline{v_2}\}$. Given $v \in V$, $\rho \in \mathbb{N}$, the ρ-sphere $S_\rho(v)$ is the set $\{v' \mid d(v, v') \leq \rho\}$, and for a tuple $\overline{v}$, $S_\rho(\overline{v}) = \bigcup_{v \in \overline{v}} S_\rho(v)$. We define the ρ-neighborhood around a tuple $\overline{v}$ to be the structure $N_\rho(\overline{v})$ induced on G by $S_\rho(\overline{a})$. The equivalence relation $\approx_\rho$ on tuples of elements is defined as $\overline{a} \approx_\rho \overline{b}$ if $N_\rho(\overline{a}) \approx N_\rho(\overline{b})$ (where $\approx$ denotes isomorphism).

A formula ψ is said to be *local* if there is a number $\rho \in \mathbb{N}$ such that for every G and tuples $\overline{v}_1$ and $\overline{v}_2$ of G, $N_\rho(\overline{v_1}) \approx N_\rho(\overline{v_2})$ implies $G \models \psi(v_1) \iff G \models \psi(v_2)$. This value ρ is then called the *locality radius* of ψ. Gaifman's theorem states that every first-order formula is local. We often annotate a formula ψ with its locality rank r and write it as $\psi^{(r)}$ for the sake of explicitness. Furthermore, $d^{>r}(v_1, v_2)$ is a first-order formula enforcing the distance between v_1 and v_2 to be at least $r + 1$.

▶ **Theorem 6** (Gaifman's locality theorem [10]). *Every First Order formula $\varphi(\overline{x})$ is equivalent to a Boolean combination of the following:*
- *local formulas $\psi^{(\rho)}(\overline{x})$ around $\overline{x}$ and*
- *sentences of the form*

$$\chi = \exists x_1, \dots, x_s \left(\bigwedge_{i=1}^{s} \alpha^{(\rho)}(x_i) \wedge \bigwedge_{1 \leq i < j \leq s} d^{>2\rho}(x_i, x_j) \right) .$$

Furthermore,
- *The transformation from φ to such a Boolean combination is effective;*
- *If $\mathrm{qr}(\varphi) = q$ and n is the length of $\overline{x}$, then $\rho \leq 7^q$, $s \leq q + n$.*

The (q, k)*-type of $\overline{v}$ in G*, denoted by $\mathrm{tp}_q^G(\overline{v})$, is the set of all first-order formulas $\varphi(x_1, \dots, x_k)$ of quantifier rank at most q such that $G \models \varphi(\overline{v})$. A (q, k)-type is a maximal consistent set of first-order formulas $\varphi(x_1, \cdots x_k)$ of quantifier rank at most q. Equivalently, a (q, k)-type is the (q, k)-type of some k-tuple $\overline{v}$ in some structure G. For a specific (q, k), there are only finitely many (q, k) types. The number of such types is denoted by $t(q, k)$.

We get the following as a corollary of Gaifman's locality theorem.

▶ **Corollary 7.** *Let $q \in \mathbb{N}$ and $\rho = 7^q$. Let G be a structure and $\overline{a}, \overline{a}' \in V^r$, $\overline{b}, \overline{b}' \in V^s$ such that $\mathrm{tp}_q^G(\overline{a}) = \mathrm{tp}_q^G(\overline{a}')$, $\mathrm{tp}_q^G(\overline{b}) = \mathrm{tp}_q^G(\overline{b}')$, $d(\overline{a}, \overline{b}) \geq 2\rho + 1$ and $d(\overline{a}', \overline{b}') \geq 2\rho + 1$. Then $\mathrm{tp}_q^G(\overline{a}, \overline{b}) = \mathrm{tp}_q^G(\overline{a}', \overline{b}')$.*

Locally Bounded Tree-width. We say that a class of structures $\mathcal{K}$ has *locally-bounded tree-width* if there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that given any $G \in \mathcal{K}$, any $v \in V$ and any $r \in \mathbb{N}$, the tree-width of $N_r(x)$ is at most $f(r)$.

Next we recall some properties of class of graphs closed under minors. An edge contraction is an operation which removes an edge from a graph while simultaneously merging the two vertices it used to connect. A graph H is a minor of a graph G if a graph isomorphic to H can be obtained from G by contracting some edges, deleting some edges and deleting some isolated vertices. A class $\mathcal{K}$ of graphs is said to be closed under minors if for every graph G in $\mathcal{K}$ and every minor H of G, H is also in $\mathcal{K}$.

Suppose a class of graphs $\mathcal{K}$ is closed under minors and has locally bounded tree-width (the class of planar graphs is an example). Let G be a graph in $\mathcal{K}$ and let v be an arbitrary vertex in G. For $i \geq 0$, let L_i be the set of all vertices of G whose shortest distance from v is i. For any i, r, the subgraph induced by $\cup_{j=1}^{r} L_{i+j}$ on G has tree-width that depends only on r. See the full version for a proof of this. This idea has been used to design approximation algorithms for hard problems [2, 7].

3.2.2 Watermarking Schemes to Preserve FO Queries on Minor Closed Classes with Locally Bounded Tree-width

Now we prove that there exist watermarking schemes that preserve unary FO queries on classes of structures that are closed under minors and that have locally bounded tree-width. We use Gaifman's locality theorem on the FO query and consider the constituent local queries. Answer to local queries only depend on local neighborhoods of the structure, which have bounded tree-width. We can run automata on them and proceed as in the previous section. We have to be careful that queries run on overlapping neighborhoods don't interfere with each other.

Let $\mathcal{K}$ be a class of structures whose Gaifman graphs belong to a class of graphs with locally bounded tree-width and that is closed under minors, let G be a structure in $\mathcal{K}$ and let $\psi(\overline{x}, y)$ be a unary first-order query. Let q be the rank of $\psi(\overline{x}, y)$ and let ρ be its locality radius. Suppose $U \subseteq V$ is the set of active elements for the query $\psi(\overline{x}, y)$. Let $c \in U$ be an active element such that the set $U_c = \{b \in U \mid \mathrm{tp}_q^G(b) = \mathrm{tp}_q^G(c)\}$ has the maximum cardinality. Due to our choice of c, we get $|U_c| \geq \frac{|U|}{t(q,r+1)}$ (recall that r is the length of $\overline{x}$ and $t(q, r+1)$ is the possible number of $(q, r+1)$-types). We will show that there is a number l that is a constant fraction of $|U|$ such that we can hide any mark $\mu \in \{0,1\}^{\leq l}$. Given a weight function W for G and a mark $\mu \in \{0,1\}^l$, we select l pairs of elements $(b_1, b_1'), (b_2, b_2'), \ldots, (b_l, b_l')$ from U_c and define the new weight function W_μ as follows: $(W_\mu(b_i), W_\mu(b_i')) = (W(b_i)+1, W(b_i)-1)$ if $\mu(i) = 1$ and $(W_\mu(b_i), W_\mu(b_i')) = (W(b_i)-1, W(b_i)+1)$ if $\mu(i) = 0$. For all other elements, W_μ is same as W. The new weight function is a 1-local distortion of the old one by construction. The difficulty is to ensure that the global distortion is bounded by a constant. We overcome this difficulty by ensuring that b_i and b_i' cannot be distinguished by the query $\psi(\overline{x}, y)$: for almost all $\overline{a} \in V^r$, $b_i \in \psi(\overline{a}, G)$ iff $b_i' \in \psi(\overline{a}, G)$. The following lemma suggests how to select such pairs.

▶ **Lemma 8.** *Suppose $\psi(\overline{x}, y)$ is a query and $\psi_1^{(\rho)}(\overline{x}, y), \psi_2^{(\rho)}(\overline{x}, y), \ldots, \psi_\alpha^{(\rho)}(\overline{x}, y)$ are the local formulas given by Theorem 6 (Gaifman's locality theorem). Suppose G is a structure and $\overline{a} \in V^r, b, b' \in V$ are such that $G \models \psi_i^{(\rho)}(\overline{a}, b)$ iff $G \models \psi_i^{(\rho)}(\overline{a}, b')$ for every $i \in \{1, 2, \ldots, \alpha\}$. Then $b \in \psi(\overline{a}, G)$ iff $b' \in \psi(\overline{a}, G)$.*

Now our goal is to select a large number of pairs (b, b') from U_c such that they cannot be distinguished by any local query $\psi_i^{(\rho)}(\overline{x}, y)$, as assumed in Lemma 8. Let us fix some $k \geq 1$ and apply Lemma 3 to every local query $\psi_i^{(\rho)}(\overline{x}, y)$. We get a MSO formula $\tilde{\psi}_i(\overline{x}, y)$ such that for every structure G' with a parse tree $\mathcal{T}$ of clique-width at most k, $\psi_i^{(\rho)}(G') = \tilde{\psi}_i(\mathcal{T})$.

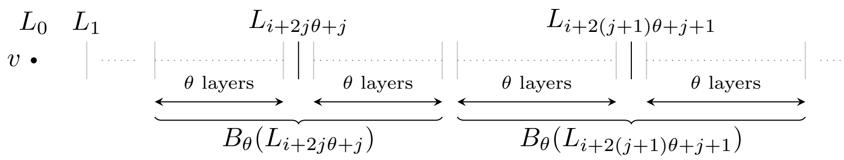

Figure 1 Division of Gaifman's graph of G into Bands and layers.

Our next goal is to identify substructures of G with bounded clique-width. Since we are considering structures of bounded local tree-width, any neighborhood of G of bounded radius has bounded tree-width, hence bounded clique-width.

For the MSO formulas $\tilde{\psi_1}(\overline{x}, y), \tilde{\psi_2}(\overline{x}, y), \ldots, \tilde{\psi_\alpha}(\overline{x}, y)$, let $A_1, A_2, \ldots, A_\alpha$ be the corresponding tree automata. Let A be the tree automaton obtained by applying the usual product construction to $A_1, A_2, \ldots, A_\alpha$ and let m be the number of states in A.

We pick some element $v \in V$ arbitrarily from the universe of G, let $L_0 = \{v\}$, and then define the layer L_i to be the elements of G which are at a distance exactly i from v. This divides the graph into layers $L_0, L_1, L_2, \ldots$. For a layer L_j, define the band $B_{2\rho}(L_j)$ to be the union of the layers $L_{j-2\rho}, L_{j-2\rho+1}, \ldots, L_j, \ldots, L_{j+2\rho-1}, L_{j+2\rho}$. Intuitively, $B_{2\rho}(L_j)$ consists of the layer L_j, 2ρ layers to the left of L_j and 2ρ layers to the right. Let $\theta = (2(r+1)+2)\rho$ and define the band $B_\theta(L_i)$ in an analogous way. For $0 \le i \le 2\theta$, define $\mathcal{L}_i$ to be the set of layers $\{L_i, L_{i+2\theta+1},\ L_{i+4\theta+2}, \ldots\} = \{L_{i+2j\theta+j} \mid j \ge 0\}$. Since there are $2\theta+1$ such sets, it must be the case that there is some $\mathcal{L}_i$ such that $|U_c \cap (\cup \mathcal{L}_i)| \ge \frac{|U_c|}{2\theta+1}$. We denote by $Y_1, Y_2 \ldots$ the layers in this $\cup\mathcal{L}_i$ in increasing order of their distance from L_0. If v is any element in L_j, then $S_{2\rho}(v) \subseteq B_{2\rho}(L_j)$. Notice that by construction, $B_{2\rho}(Y_i) \cap B_{2\rho}(Y_j) = \emptyset = B_\theta(Y_i) \cap B_\theta(Y_j)$ for any $i \neq j$. Refer to Fig. 1 for a visual representation of the bands. The layer L_0 is represented by the single vertex v. The layers $L_{i+2j\theta+j}, L_{i+2(j+1)\theta+j+1}$ are represented by solid vertical lines. Other layers are represented by vertical lines that are grayed out.

In the sequence of layers that we obtained, let $Y'_1, Y'_2, \ldots Y'_\gamma$ be those that contain at least $2m^m+2$ elements from U_c. Let $Y''_1, Y''_2, \ldots, Y''_\delta$ be the layers that contain less than $2m^m+2$ elements from U_c. Let $v''_1, v''_2, \ldots, v''_\delta$ be arbitrarily chosen elements of $Y''_1, Y''_2, \ldots, Y''_\delta$ respectively that are in U_c (we may ignore a particular Y''_i if it does not contain any elements of U_c in it). We will use the set of pairs $M_1 = \{(v''_1, v''_2), \ldots, (v''_{\delta-1}, v''_\delta)\}$ for watermarking.

Next we select watermarking pairs from the layers $Y'_1, Y'_2, \ldots Y'_\gamma$. For each layer Y'_i, let N_i be the substructure induced by the band $B_\theta(Y'_i)$. This is a band of width $2\theta+1$, so its tree-width and hence clique-width (say k) depends only on $2\theta+1$, which in turn depends only on the locality radius ρ and the input length r. Now we can apply Lemma 4 with the tree automaton A and the parse tree $\mathcal{T}$ of N_i of clique width at most k, setting $Y = Y'_i \cap U_c$. We get pairs $(b_{(i,1)}, b'_{(i,1)}), (b_{(i,2)}, b'_{(i,2)}), \ldots, (b_{(i,n)}, b'_{(i,n)})$, where $n = \lfloor \frac{|Y'_i \cap U_c|}{4m^m+4} \rfloor$. We will use the set of pairs $M_2 = \cup_i \cup_j \{(b_{(i,j)}, b'_{(i,j)})\}$ also for watermarking. Again note that all elements in the pairs are in U_c.

▶ **Lemma 9.** *Suppose a watermarking pair $(v''_i, v''_{(i+1)}) \in M_1$ consists of elements from $Y''_i, Y''_{(i+1)}$ respectively. If the tuple $\overline{a} = \langle a_1, \ldots, a_r \rangle$ is such that $\{a_1, \ldots, a_r\} \cap (B_{2\rho}(Y''_i) \cup B_{2\rho}(Y''_{(i+1)})) = \emptyset$, then $v''_i \in \psi(\overline{a}, G)$ iff $v''_{(i+1)} \in \psi(\overline{a}, G)$.*

▶ **Lemma 10.** *Suppose a watermarking pair $(b, b') \in M_2$ was selected from some set V_j (as specified in Lemma 4) of some band N_i. If $\overline{a} = \langle a_1, \ldots, a_r \rangle$ is such that $\{a_1, \ldots, a_r\} \cap V_j = \emptyset$, $b \in \psi(\overline{a}, G)$ iff $b' \in \psi(\overline{a}, G)$.*

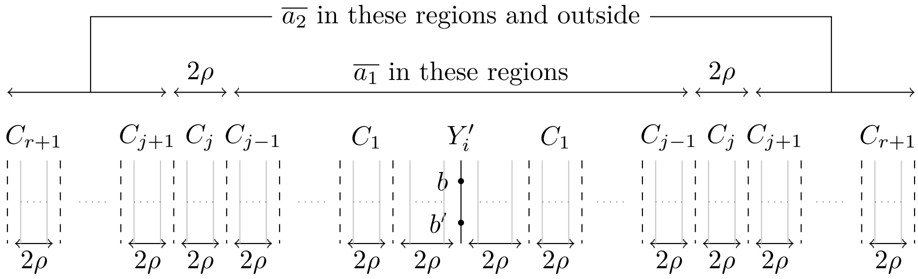

Figure 2 The regions $C_1, \ldots, C_{r+1}$ and $\overline{a_1}, \overline{a_2}$ used in Case III of the proof of Lemma 10.

Proof.

Case I: $\{a_1, \ldots, a_r\} \cap B_{2\rho}(Y'_i) = \emptyset$. In this case, since b, b' are both on the layer Y'_i, we have $S_\rho(\overline{a}) \cap (S_\rho(b) \cup S_\rho(b')) = \emptyset$. Hence we can apply Corollary 7 to infer the result.

Case II: $S_\rho(\overline{a}) \subseteq B_{(2(r+1)+2)\rho}(Y'_i)$. In this case, $S_\rho(\overline{a}bb') \subseteq B_{(2(r+1)+2)\rho}(Y'_i) = B_\theta(Y'_i)$. We selected (b, b') according to Lemma 4, with the tree automaton A running on a parse tree of N_i. Since the tree automaton runs all the automata $A_1, A_2, \ldots, A_\alpha$ simultaneously, we infer that $N_i \models \psi_j^{(\rho)}(\overline{a}, b)$ iff $N_i \models \psi_j^{(\rho)}(\overline{a}, b')$ for every $j \in \{1, 2, \ldots, \alpha\}$. Since $S_\rho(\overline{a}bb') \subseteq B_\theta(Y'_i)$, the substructure induced on N_i by $S_\rho(\overline{a}bb')$ is isomorphic to the substructure induced on G by $S_\rho(\overline{a}bb')$. Since $\psi_j^{(\rho)}(\overline{x}, y)$ is a local formula around $\overline{x}, y$ with locality radius ρ, we infer that $N_i \models \psi_j^{(\rho)}(\overline{a}, b)$ iff $G \models \psi_j^{(\rho)}(\overline{a}, b)$ and $N_i \models \psi_j^{(\rho)}(\overline{a}, b')$ iff $G \models \psi_j^{(\rho)}(\overline{a}, b')$ for every $j \in \{1, 2, \ldots, \alpha\}$. Hence, $G \models \psi_j^{(\rho)}(\overline{a}, b)$ iff $G \models \psi_j^{(\rho)}(\overline{a}, b')$ for every $j \in \{1, 2, \ldots, \alpha\}$. We can now apply Lemma 8 to infer the result.

Case III: $\{a_1, \ldots, a_r\} \cap B_{2\rho}(Y'_i) \neq \emptyset$ and $\{a_1, \ldots, a_r\} \not\subseteq B_{(2(r+1)+2)\rho}(Y'_i)$. In this case, some elements of $\overline{a}$ may be within distance 2ρ from b, b'. Some elements of $\overline{a}$ may be quite far and their ρ neighborhoods may not be included in $B_\theta(Y'_i)$. We divide $B_\theta(Y'_i) \setminus B_{2\rho}(Y'_i)$ into $r+1$ regions $C_1, C_2, \ldots C_{r+1}$. Define $C_1 = B_{4\rho}(Y'_i) \setminus B_{2\rho}(Y'_i), C_2 = B_{6\rho}(Y'_i) \setminus B_{4\rho}(Y'_i)$, etc. Since there are $r+1$ such regions, and only r parameters in $\overline{a}$, there is a region, say C_j that doesn't contain any elements of $\overline{a}$. Let $\overline{a_1}$ be the tuple of elements of $\overline{a}$ that are in $B_{2\rho}(Y'_i) \cup C_1 \cup \cdots \cup C_{j-1}$ and let $\overline{a_2}$ consist of the remaining elements of $\overline{a}$. Note that $S_\rho(\overline{a_1}bb') \cap S_\rho(\overline{a_2}) = \emptyset$ (since the region C_j is of width 2ρ, $\overline{a_1}bb'$ are on the inside of this band and $\overline{a_2}$ are on the outside). Refer to Fig. 2 for a visual presentation of $\overline{a_1}, \overline{a_2}$. The layer Y'_i is represented by a solid vertical line, in which b, b' are highlighted. Other layers are represented by vertical lines that are grayed out. Boundaries of regions are represented by dashed vertical lines. Each region consists of 2ρ layers on the left and 2ρ layers on the right. Since the layer C_j doesn't contain any elements of $\overline{a}$, it acts as a buffer between $S_\rho(\overline{a_1}bb')$ and $S_\rho(\overline{a_2})$.

Let the structure H_1 be an isomorphic copy of N_i (which is the substructure induced by $B_\theta(Y'_i)$). Since H_1 consists of $2\theta+1$ layers, the tree-width and hence clique-width of H_1 depends only on $2\theta+1$. Let H_2 be a disjoint union of at most r spheres of radius at most $2r\rho$, containing an isomorphic copy of $N_\rho(\overline{a_2})$ (details of constructing H_2 are in the full version). The clique-width of H_2 also depends only on r and ρ. Let H be the disjoint union of H_1 and H_2. For the elements $\overline{a_1}bb'$ in N_i, the isomorphism with H_1 will give corresponding elements in H_1; let $h(\overline{a_1}bb')$ be these corresponding elements. Similarly, let $h(\overline{a_2})$ be the elements in H_2 corresponding to $\overline{a_2}$ in $N_\rho(\overline{a_2})$. Let $\mathcal{T}$ be a parse tree of N_i (and so of H_1) and $\mathcal{T}'$ be a parse tree of H_2 of minimum clique-widths, with k being the maximum of these two widths. We obtain a parse tree $\mathcal{T}''$ of H of clique-width at most k by making $\mathcal{T}$ and $\mathcal{T}'$ as subtrees of a new root labeled by the union operation. We selected b, b' according to Lemma 4 with the tree automaton A and parse tree $\mathcal{T}$. We infer that the automaton A labels the roots of $\mathcal{T}_{h(\overline{a_1}b)}$ and $\mathcal{T}_{h(\overline{a_1}b')}$ with the same state. Hence, the automaton A labels the roots of $\mathcal{T}''_{h(\overline{a_1}b\overline{a_2})}$ and $\mathcal{T}''_{h(\overline{a_1}b'\overline{a_2})}$ with the same state (note that $h(\overline{a_1}bb')$ are in $\mathcal{T}$ while $h(\overline{a_2})$ are in $\mathcal{T}'$). Hence, we infer that $H \models \psi_j^{(\rho)}(h(\overline{a}), h(b))$ iff $H \models \psi_j^{(\rho)}(h(\overline{a}), h(b'))$ for every $j \in \{1, 2, \ldots, \alpha\}$. The substructure induced on G by $S_\rho(\overline{a}bb')$ is isomorphic to the substructure induced on H by $S_\rho(h(\overline{a}bb'))$. Since $\psi_j^{(\rho)}(\overline{x}, y)$ is a local formula around $\overline{x}, y$ with locality radius ρ, we infer that $H \models \psi_j^{(\rho)}(h(\overline{a}), h(b))$ iff $G \models \psi_j^{(\rho)}(\overline{a}, b)$ and $H \models \psi_j^{(\rho)}(h(\overline{a}), h(b'))$ iff $G \models \psi_j^{(\rho)}(\overline{a}, b')$. Hence, $G \models \psi_j^{(\rho)}(\overline{a}, b)$ iff $G \models \psi_j^{(\rho)}(\overline{a}, b')$ for every $j \in \{1, 2, \ldots, \alpha\}$. We can now apply Lemma 8 to infer the result. ◀

The technique of considering a small number of layers to get graphs of bounded tree-width was known before [2, 7]. Here, we find ways of using it to bound global distortions and that is the main technical contribution of this paper. Now we state the main result of this sub-section.

▶ **Theorem 11.** *Suppose $\mathcal{K}$ is a class of structures whose Gaifman graphs belong to a class of graphs that is closed under minors and have locally bounded tree-width. Suppose $\psi(\overline{x}, y)$ is a unary first-order query of input length r. Then, there exists a scalable watermarking scheme preserving $\psi(\overline{x}, y)$ on structures in $\mathcal{K}$.*

Proof idea. Every pair in $M_1 \cup M_2$ can hide one bit in the database and $|M_1 \cup M_2|$ is a constant fraction the size of the set of active tuples, as shown in the full version. For a given parameter $\overline{a}$, a pair $(b, b') \in M_1 \cup M_2$ contributes to global distortion only if $\overline{a}$ intersects with some spheres around (b, b'), as proved in Lemma 9 and Lemma 10. The spheres are mutually disjoint, so $\overline{a}$ can intersect with at most r spheres. Hence the global distortion is bounded by r. ◀

4 Conclusion

In [14], there is a transformation of watermarking schemes for non-adversarial models into schemes for adversarial models, under some assumptions. As observed in [13], the same transformation under similar assumptions also work for MSO and FO queries. Hence, our result on FO queries can also use a similar transformation to work on adversarial models.

The difficulty with non-unary queries is that Gaifman graphs don't capture information about active tuples – even if two elements b_1, b_2 appear in the same active tuple, the Gaifman graph may not have an edge between b_1 and b_2. The results on VC dimension use powerful results from model theory [19] or versions of finite Ramsey theorem for hyper graphs [16]. It remains to be seen whether similar results are true for watermarking schemes. It also remains to be seen if the condition on closure under minors can be dropped and watermarking schemes can still be obtained, as shown for VC dimension in [12].

Beginning with graphs of bounded degree, it is now known that for the much bigger class of graphs that are nowhere dense, FO properties can be efficiently decided. It remains to be seen whether results on watermarking schemes can be extended to the class of graphs that are nowhere dense.

We don't know if there are deeper connections between bounded VC dimension and presence of scalable watermarking schemes preserving queries. Some progress is made in [13], where it is shown that unbounded VC dimension doesn't necessarily mean absence of scalable watermarking schemes, but more work is needed in this direction.

References

1 R. Agrawal and J. Kiernan. Watermarking relational databases. In *International Conference on Very Large Databases*, VLDB, 2002.
2 B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41:153–180, 1994.
3 E. F. Codd. Relational completeness of database sublanguages. In *Proceedings of the Sixth Courant Computer Science Symposium on Data Base Systems, Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 65–98, New York, NY, USA, 1972. Prentice-Hall.
4 Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1):77–114, 2000.
5 A. Dawar, M. Grohe, and S. Kreutzer. Locally excluding a minor. In *Proceedings of the 22nd IEEE Symposium on Logic in Computer Science*, LICS, pages 270–279, 2007.
6 Z. Dvořàk, D. Kràl, and R. Thomas. Deciding first-order properties for sparse graphs. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS, pages 133–142, 2010.
7 D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27:275–291, 2000.
8 J. Flum and M. Grohe. Fixed-parameter tractability, definability, and model checking. *SIAM Journal on Computing*, 31:113–145, 2001.
9 M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposable structures. *J. ACM*, 48(6):1184–1206, 2001.
10 H. Gaifman. On Local and Non-Local Properties. In J. Stern, editor, *Proceedings of the Herbrand Symposium*, volume 107 of *Studies in Logic and the Foundations of Mathematics*, pages 105–135. Elsevier, 1982.
11 M. Grohe, S. Kreutzer, and S. Siebertz. Deciding First-Order Properties of Nowhere Dense Graphs. *J. ACM*, 64(3):17:1–17:32, 2017.
12 M. Grohe and G. Turán. Learnability and Definability in Trees and Similar Structures. *Theory of Computing Systems*, 37(1):193–220, January 2004.
13 David Gross-Amblard. Query-preserving Watermarking of Relational Databases and XML Documents. *ACM Trans. Database Syst.*, 36(1):3:1–3:24, 2011.
14 S. Khanna and F. Zane. Watermarking Maps: Hiding Information in Structured Data. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, pages 596–605, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.
15 S. Kreutzer. Algorithmic meta-theorems. In J. Esparza, C. Michaux, and C. Steinhorn, editors, *Finite and Algorithmic Model Theory*, pages 177–270. Cambridge University Press, 2011. chapter 5.
16 M.C. Laskowski. Vapnik-Chervonenkis classes of definable sets. *Journal of the London Mathematical Society*, 45(2):377–384, 1992.
17 M. Pilipczuk, S. Siebertz, and S. Toru'nczyk. On the number of types in sparse graphs. In *Proceedings of LICS*, pages 799–808, 2018.
18 D. Seese. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science*, 6:505–526, 1996.
19 S. Shelah. Stability, the f.c.p. and superstability. *Annals of Mathematical Logic*, 3:271–362, 1971.