The Fine-Grained Complexity of Graph Homomorphism Parameterized by Clique-Width

Robert Ganian
Algorithms and Complexity Group, TU Wien, Austria

Thekla Hamm
Algorithms and Complexity Group, TU Wien, Austria

Viktoriia Korchemna
Algorithms and Complexity Group, TU Wien, Austria

Karolina Okrasa
Faculty of Matematics and Information Science, Warsaw University of Technology, Poland
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland

Kirill Simonov
Algorithms and Complexity Group, TU Wien, Austria

Abstract

The generic homomorphism problem, which asks whether an input graph G admits a homomorphism into a fixed target graph H, has been widely studied in the literature. In this article, we provide a fine-grained complexity classification of the running time of the homomorphism problem with respect to the clique-width of G (denoted cw) for virtually all choices of H under the Strong Exponential Time Hypothesis. In particular, we identify a property of H called the signature number $s(H)$ and show that for each H, the homomorphism problem can be solved in time $\mathcal{O}^*(s(H)^{\mathrm{cw}})$. Crucially, we then show that this algorithm can be used to obtain essentially tight upper bounds. Specifically, we provide a reduction that yields matching lower bounds for each H that is either a projective core or a graph admitting a factorization with additional properties – allowing us to cover all possible target graphs under long-standing conjectures.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases homomorphism, clique-width, fine-grained complexity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.66

Category Track A: Algorithms, Complexity and Games

Funding *Robert Ganian*: Robert Ganian acknowledges support by the Austrian Science Fund (FWF, projects Y1329 and P31336).
Thekla Hamm: Thekla Hamm acknowledges support by the Austrian Science Fund (FWF, projects P31336 and Y1329).
Viktoriia Korchemna: Viktoriia Korchemna acknowledges support by the Austrian Science Fund (FWF, project Y1329).
Karolina Okrasa: Karolina Okrasa acknowledges support by the European Research Council, grant agreement No 714704. Parts of this work were performed while visiting TU Wien, Vienna, Austria.
Kirill Simonov: Kirill Simonov acknowledges support by the Austrian Science Fund (FWF, project P31336).

1 Introduction

A *homomorphism* from a graph G to a graph H is an edge-preserving mapping from the vertices of G to the vertices of H. Homomorphisms are fundamental constructs which have been studied from a wide variety of perspectives [2, 3, 13]. Our focus here will be on the

class of problems which ask whether an input n-vertex graph G admits a homomorphism to a fixed target graph H. This "meta-problem" – which we simply call $\textsc{Hom}(H)$ – captures, among others, the classical c-Coloring problems when H is set to the complete graph on c vertices. Famously, Hell and Nešetřil [15] proved that $\textsc{Hom}(H)$ is polynomial-time solvable if H is bipartite or has a loop, and NP-complete otherwise.

While the aforementioned result provides a basic classification of the complexity of $\textsc{Hom}(H)$, it does not say much in terms of how quickly one can actually solve these problems. Indeed, the usual assumption that $\mathsf{P} \neq \mathsf{NP}$ is not sufficient to obtain tight bounds for the running times of algorithms. While upper bounds can be straightforwardly obtained by designing a suitable algorithm, the corresponding lower bounds usually rely on the Exponential Time Hypothesis (ETH) or the Strong Exponential Time Hypothesis (SETH), which allows for even tighter bounds [18, 19, 23]. It is not difficult to design a brute-force algorithm for the homomorphism problem that runs in time $\mathcal{O}^*(|V(H)|^n)$ for every choice of H, and thanks to the breakthrough result of Cygan et al. we now know that this running time is essentially tight under the Exponential Time Hypothesis (ETH) [6] as long as one considers only the dependency on n and $|V(H)|$.

Still, it is often possible to circumvent this lower bound and obtain significantly better runtime guarantees. One approach to do so is to consider restrictions on the class of targets: if H is a complete graph then $\textsc{Hom}(H)$ can be solved in time $\mathcal{O}^*(2^n)$, and there are also several algorithms that achieve running times of the form $\mathcal{O}^*(\alpha(H)^n)$ where $\alpha(H)$ is some structural parameter of H [12, 29, 30]. The other is to exploit the properties of the input graph G, which are commonly captured by a suitably defined structural parameter. The most commonly used graph parameter in this respect is *treewidth* [28], which informally measures how "tree-like" a graph is.

When considering treewidth, it is once again not difficult to obtain an algorithm that runs in time $\mathcal{O}^*(|V(H)|^{\mathrm{tw}})$, where tw is the treewidth of G; as before, it was much more difficult to show that this is essentially optimal. The first SETH-based tight lower bound in this setting was actually shown for special cases of the related problem of $\textsc{LHom}(H)$, where each vertex in the graph G comes with a list of admissible targets for the homomorphism [9]; this was later lifted to a full classification [24]. A nearly-complete SETH-based lower bound result for $\textsc{Hom}(H)$ itself was only obtained recently by Okrasa and Paweł Rzążewski [25]; in particular, the result covers all targets which are so-called *projective cores*. It is known that almost all graphs are projective cores [16, 25, 31], and it is worth noting that the authors showed that their result can be lifted to all targets under long-standing conjectures on the properties of projective cores [21, 22].

While treewidth is the most prominent structural graph parameter, it is not the most general[1] one that can be used to efficiently solve $\textsc{Hom}(H)$. Indeed, standard dynamic programming techniques can be used to obtain a $\mathcal{O}^*((2^{|V(H)|})^{\mathrm{cw}})$ time algorithm for the problem, where cw stands for *clique-width* [4]: a well-studied graph parameter that is bounded not only on all graph classes of bounded treewidth, but also on well-structured dense classes such as complete graphs. But is this basic algorithm generally optimal (mirroring the situation for treewidth [25]), or can one obtain better runtime dependencies on clique-width?

Contribution. Our aim is to obtain a detailed understanding of the fine-grained complexity of $\textsc{Hom}(H)$ in terms of the clique-width of G and the fixed target H. As a starting point for our investigation, we note that Lampis used the SETH to obtain tight bounds for c-Coloring

[1] There is a hierarchy of graph parameters (see, e.g., [1, Figure 1]), where parameter $\mathbb{A}$ is more general than parameter $\mathbb{B}$ if there are graph classes of bounded $\mathbb{A}$ and unbounded $\mathbb{B}$ but the opposite is not true.

with respect to clique-width [20]. Interestingly, already for this special case, the upper and lower bounds differ from those of the aforementioned simple dynamic programming algorithm: if H is a complete graph, then $\textsc{Hom}(H)$ can be solved in time $\mathcal{O}^*((2^{|V(H)|}-2)^{\mathrm{cw}})$ [20] and this is tight under the SETH. However, as noted by Piecyk and Rzążewski [27], it was not at all obvious how these bounds can be lifted to general choices of H.

In order to achieve our goals we need to improve upon the basic dynamic programming idea to identify a "hopefully correct" base of the exponent for every choice of H. Towards our first result, we identify a structural property of H called the *signature number* (denoted $s(H)$) which, intuitively, captures the number of non-trivial neighborhood classes of vertex subsets in H (the *signature set*). We then obtain a non-trivial dynamic programming algorithm that solves $\textsc{Hom}(H)$ in time where the base of the exponent is precisely the signature number. We note that $s(H)$ is $2^{|V(H)|}-2$ for complete graphs H, and so this result also provides a succinct and broader explanation for the running time of Lampis' algorithm [20].

▶ **Theorem 1.** *Let H be a fixed graph. $\textsc{Hom}(H)$ can be solved in time $\mathcal{O}^*(s(H)^{\mathrm{cw}(G)})$ for each input graph G, assuming an optimal clique-width expression of G is provided as part of the input.*

With this upper bound, we proceed to the main technical contribution of this paper: establishing a corresponding lower bound under the SETH. The main difficulty here is that we need a reduction that is delicate on one hand, since it needs to preserve the clique-width, but is on the other hand also flexible enough to work for many different choices of H; moreover, the reduction has to rely on the signature numbers of these graphs in some way.

To provide an intuitive description of the reduction, let us focus for now on the case where H is a projective core. On a high level, the main building block is an *S-gadget* which, given an arbitrary set S of pairs of vertices in H and two vertices p and q of the input graph G, ensures that every homomorphism f satisfies $(f(p), f(q)) \in S$. After providing a generic construction for such S-gadgets which is clique-width preserving and works for every valid choice of H, we use these to obtain *implication gadgets* and *or gadgets* which restrict how a solution homomorphism can behave on a selected set of vertices in G. The formalization of these gadgets is the main technical hurdle towards the desired result; once that is done, we can lift the idea used in the earlier reduction of Lampis [20] that established clique-width lower bounds for c-Coloring by reducing from Constraint Satisfaction (CSP) to $\textsc{Hom}(H)$. One crucial distinction in our reduction is that we use elements of the signature set (as opposed to color sets) to represent domain values in the CSP instance.

To lift these considerations to cases where H is not a projective core, we unfortunately need to add an extra layer of complexity. Similarly as in the previous treewidth-based lower bound for $\textsc{Hom}(H)$ [25], one can base this step on conjectures of Larose and Tardif [21, 22] that classify all remaining targets as certain graph products with special properties (notably, all of the factors must be "truly projective"). The approach used for treewidth [25] was then to essentially repeat all steps of the proof for projective cores, with the added difficulty that one uses the properties of products instead of dealing directly with projective cores.

While this approach could be used here as well, instead we unify the two cases (H being a projective core, and H being a product) by defining the notion of W-projectivity for some factor W of H. In particular, if H is a projective core then it itself is H-projective, while if H is a product with truly projective factor H_i then it is H_i-projective. As our main result, we obtain an SETH-based lower bound which essentially shows that for each W-projective graph H, $s(W)$ is the optimal base of the clique-width exponent for solving $\textsc{Hom}(H)$:

▶ **Theorem 2.** *Let H be a fixed non-trivial core with prime factorization $H_1 \times \ldots \times H_m$. Assume that H is H_i-projective for some $i \in [m]$. Then there is no algorithm solving* Hom(H) *in time $\mathcal{O}^*((s(H_i) - \varepsilon)^{\mathrm{cw}(G)})$ for any $\varepsilon > 0$, unless the* SETH *fails.*

By also deliberately considering prime factorizations in the algorithm which we provide for Theorem 1, we can obtain an upper bound on the complexity of Hom(H) that matches the lower bound from Theorem 2. For a discussion explicitly relating these complexity bounds in the context of the aforementioned conjectures of Larose and Tardif, we refer to Section 6.

2 Preliminaries

We use standard terminology for graph theory [7]. Let $[i]$ denote the set $\{1, \ldots, i\}$. For a mapping $f : A \to B$ and $A' \subseteq A$, let $f|_{A'}$ denote the restriction of f to A'. We will use the $\mathcal{O}^*(\cdot)$ notation to suppress factors polynomial in the input size.

Homomorphisms and Cores

For two graphs G and H, a *homomorphism* from G to H is a mapping $h : V(G) \to V(H)$, such that for every $uv \in E(G)$ we have $h(u)h(v) \in E(H)$. If there exists a homomorphism from G to H, we denote this fact by $G \to H$, and if h is a homomorphism from G to H, we denote that by $h : G \to H$. If there is no homomorphism from G to H, we write $G \not\to H$. If $G \to H$ and $H \to G$, we say that G and H are *homomorphically equivalent*. In particular, since the composition of homomorphisms is a homomorphism, if G and H are homomorphically equivalent, then for every graph F we have that $F \to G$ if and only if $F \to H$. It is straightforward to verify that homomorphic equivalence is an equivalence relation on the class of all graphs. On the other hand, if $G \not\to H$ and $H \not\to G$ for some graphs G, H, we say that G and H are *incomparable*.

We say that a graph H is a *core* if every homomorphism $h : H \to H$ is an automorphism. Equivalently, H is a core if for every proper induced subgraph H' of H it holds that $H \not\to H'$. We say that a core H' is a *core of* H if H' is an induced subgraph of H and $H \to H'$. Clearly, each core graph is a core of itself. Each graph has a unique (up to isomorphism) core, and the core of H can be equivalently defined as the smallest (with respect to the number of vertices) graph that is homomorphically equivalent with H [16].

A graph H is *ramified* if $N(u) \not\subseteq N(v)$ for every two distinct vertices u, v of H. Observe that each core is ramified; otherwise one could define $f : H \to H$ that is an identity on all vertices of H but u and set $f(u) = v$. This would be a homomorphism to a proper subgraph of H, contradicting the fact that H is a core.

We say that a graph H is *trivial* if its core has at most two vertices.

▶ **Observation 3** ([15])**.** *A graph H is trivial if and only if it is either bipartite or contains a vertex with a loop.*

Proof. It is straightforward to observe that there exist three trivial cores: K_1, K_2, and K_1^*, where by K_1^* we denote the graph that consists of one vertex with a loop.

If H contains a vertex with a loop, then K_1^* is the core of H. If H is bipartite, then the core of H is either K_1 (if H has no edges) or K_2 (since mapping the vertices of one bipartition class to one vertex of K_2, and another bipartition graph to the other, is a homomorphism).

For the other direction, assume that H is a non-bipartite loopless graph. Since it is loopless, K_1^* cannot be its core. Clearly, H has at least one edge, and therefore $H \not\to K_2$ Moreover, H contains an odd cycle C_{2k+1} as a subgraph, hence, $C_{2k+1} \to H$. If now $H \to K_2$, composition of these homomorphism gives that $C_{2k+1} \to K_2$, which is equivalent to stating that C_{2k+1} is 2-colorable, a contradiction. ◀

Observe that trivial cores H correspond precisely to the polynomial cases of the $\textsc{Hom}(H)$ problem. Since our aim is to focus on the NP-hard cases of the problem, from here onward we will assume that the target graph is non-trivial.

Signature Sets

For a vertex v of a graph H, let $N_H(v)$ denote the set of neighbors of v in H. If the graph is clear from the context, we will omit the subscript H and write $N(v)$.

For a non-empty set $T \subseteq V(H)$ we say that $S(T)$ is the *signature set of* T if $S(T) = \bigcap_{t \in T} N(t)$. We say that a non-empty set $S \subseteq V(H)$ is a *signature set*, if there exists T such that $S = S(T)$. We denote by $\mathcal{S}(H)$ the set of all signature sets of H, and we note that $\{V(H), \emptyset\} \cap \mathcal{S}(H) = \emptyset$.

▶ **Observation 4.** *If T is a proper non-empty subset of $V(H)$, and $a \in T$, $b \in S(T)$, then $ab \in E(H)$. Moreover, for non-empty subsets $A, B \subseteq V(H)$, $S(A \cup B) = S(A) \cap S(B)$.*

We note that the operation of taking a signature set is reversible on $\mathcal{S}(H)$:

▶ **Observation 5.** *For every $A \in \mathcal{S}(H)$, $S(S(A)) = A$.*

Proof. By the definition of signature set, $A \times S(A) \subseteq E(H)$, so $A \subseteq S(S(A))$. For the converse direction observe that as $A \in \mathcal{S}(H)$, there exists a non-empty subset T of $V(H)$ such that $A = S(T)$. Pick any $x \in S(S(A))$, then $E(H) \supseteq \{x\} \times S(A) = \{x\} \times S(S(T)) \supseteq \{x\} \times T$. Hence by definition $x \in S(T) = A$. ◀

Let the *signature number* of H, denoted $s(H)$, be defined as $|\mathcal{S}(H)|$. As mentioned in the introduction, the signature number will play a crucial role in our upper and lower bounds.

Observe that, if H is a target and hence non-trivial, for every nonempty $T \subseteq V(H)$ we have that $S(T) \cap T = \emptyset$. From that it is easy to see that $V(H)$ never belongs to $\mathcal{S}(H)$. Since, by definition, $\emptyset \notin \mathcal{S}(H)$, we get the following bounds for $s(H)$.

▶ **Observation 6.** *Let H be a graph with no loops. Then $s(H) \leqslant 2^{|V(H)|} - 2$.*

Notice that since $2^{|V(H)|} - 2$ is the number of all proper non-empty subsets of $V(H)$, the equality in Observation 6 holds if and only if H is a clique.

If $S \in \mathcal{S}(H)$, we call T such that $S(T) = S$ a *witness* of S. Clearly, we can have distinct T_1, T_2 such that $S(T_1) = S(T_2)$, however, notice that in such a case there exists $T = T_1 \cup T_2$ such that $S(T) = S(T_1) = S(T_2)$. Hence, there exists a unique maximal (with respect to inclusion) witness of S, and we denote it by $M(S)$. In fact, it is not difficult to see that $M(S) = \{v \in V(H) \mid S \subseteq N_H(v)\}$; for $S(M(S)) = S$ to hold, it is clearly necessary that $S \subseteq N_H(v)$ for all $v \in M(S)$. On the other hand, as $M(S)$ is maximal all v for which this is true are contained in $M(S)$.

In this way signature sets and their witnesses are in one-to-one correspondence. While not necessary to obtain our algorithmic and lower bounds for $\textsc{Hom}(H)$ parameterized by clique-width, this offers an alternative perspective on the role of signatures in our results.

In fact, the signature number could equivalently be defined as the "maximal witness number" and signature sets could be replaced by maximal witnesses in all our proofs: Let $\mathcal{M}(H) = \{M(S) : S \in \mathcal{S}(H)\}$.

▶ **Observation 7.** $\mathcal{S}(H) = \mathcal{M}(H)$.

Proof. Let T be a fixed non-empty subset of $V(H)$ such that $S(T) \neq \emptyset$. Observe that the definition and the maximality of $M(S(T))$ implies that $S(S(T)) = \bigcap_{t \in S(T)} N(t) = M(S(T))$ Since T and $S(T)$ are non-empty, we get that $\mathcal{S}(H) \supset \mathcal{M}(H)$. For the other direction observe that $M(M(S(T))) = S(T)$. Assume the contrary, then there exists a vertex $v \notin S(T)$ such that $N(v) \supset M(S(T))$. However, $T \subset M(S(T)) \subset N(v)$ meaning that $v \in S(T)$, a contradiction. Thus, $\mathcal{S}(H) = \mathcal{M}(H)$. ◀

We note that if H is a core graph, we can also bound the minimum cardinality of $\mathcal{S}(H)$.

▶ **Observation 8.** *Let H be a core graph, $H \neq K_1$. Then $s(H) \geqslant |V(H)|$.*

Proof. Observe that if H is a core distinct from K_1, then it does not contain isolated vertices. Therefore, for each $v \in V(H)$ we have $N(v) \in \mathcal{S}(H)$. On the other hand, since H is a core, it is ramified. In particular, for every distinct $v, w \in V(H)$ we have $N(v) \neq N(w)$. Hence different vertices give rise to different signature sets. ◀

Clique-Width and Clique-Width Expressions

For a positive integer k, we let a *k-graph* be a graph whose vertices are labeled by $[k]$. For convenience, we consider a graph to be a k-graph with all vertices labeled by 1. We call the k-graph consisting of exactly one vertex v (say, labeled by i) an initial k-graph and denote it by $i(v)$.

The *clique-width* of a graph G is the smallest integer k such that G can be constructed from initial k-graphs by means of iterative application of the following three operations:

1. Disjoint union (denoted by $\oplus$);
2. Relabeling: changing all labels i to j (denoted by $\rho_{i \to j}$);
3. Edge insertion: adding an edge from each vertex labeled by i to each vertex labeled by j ($i \neq j$; denoted by $\eta_{i,j}$).

A construction of a k-graph G using the above operations can be represented by an algebraic term composed of $\oplus$, $p_{i \to j}$ and $\eta_{i,j}$ (where $i \neq j$ and $i, j \in [k]$). Such a term is called a k-expression defining G, and we often view it as a tree with each node labeled with the appropriate operation. Conversely, we call the k-graph that arises from a k-expression its *evaluation*. The *clique-width* of G is the smallest integer k such that G can be defined by a k-expression which we then also call a *clique-width expression* of G.

Many graph classes are known to have constant clique-width; examples include all graph classes of constant treewidth and co-graphs [5]. Moreover a fixed-parameter algorithm is known to compute a k-expression of the input where k is bounded in $f(\mathrm{cw})$ [26].

3 Algorithm

As our first contribution, we obtain an algorithm that will play a crucial role for upper-bounding the fine-grained complexity of $\textsc{Hom}(H)$.

▶ **Theorem 1.** *Let H be a fixed graph. $\textsc{Hom}(H)$ can be solved in time $\mathcal{O}^*(s(H)^{\mathrm{cw}(G)})$ for each input graph G, assuming an optimal clique-width expression of G is provided as part of the input.*

Proof. Assume, w.l.o.g., that G is connected and $|V(G)| > 1$. We will describe a dynamic program that proceeds in a leaf-to-root fashion along the provided k-expression σ of G. For a subexpression $\tau \subseteq \sigma$, we denote the evaluation of τ by G_τ, and by $V_\tau^i \subseteq V(G_\tau)$ the vertex

set that has label i in G_τ. We say that i is a *live label* in τ if there is an edge of G which is incident to V_τ^i and does not appear in G_τ. Denote the set of live labels in τ by L_τ. Since G is connected, $L_\tau \neq \emptyset$ for any proper subexpression τ of σ.

For each subexpression τ of σ, we will compute a set P_τ consisting of functions $p\colon L_\tau \to \mathcal{S}(H)$ where $p \in P_\tau$ if and only if there exists a homomorphism h_p from G_τ to H such that $p(i) \subseteq S(h_p(V_\tau^i))$, $i \in L_\tau$. We will say that $p \in P_\tau$ *describes* the homomorphism h_p in τ or, equivalently, that h_p *witnesses* p in τ. Intuitively, we will use $p(i)$ to preemptively store the images of the neighbors of V_τ^i in the final graph G – that is why we store not only the exact signature, but all signatures that occur as subsets. We remark that storing the "current" images of the neighbors of V_τ^i in G_τ would be sufficient to obtain a conceptually simpler fixed-parameter algorithm parameterized by clique-width, but in that case it is not obvious how one can avoid a quadratic dependency on clique-width in the exponent.

Observe that for any homomorhism $h\colon G_\tau \to H$, images of vertices with live labels should be connected in H with images of their future neighbors. In particular, for any $i \in L_\tau$, $S(h(V_\tau^i)) \neq \emptyset$ and hence $S(h(V_\tau^i)) \in \mathcal{S}(H)$. Therefore h is described in τ by some $p \in P_\tau$. By definition, $L_\sigma = \emptyset$ and hence G is homomorphic to H if and only if P_σ contains the empty mapping, i.e., if $P_\sigma = \{\emptyset\}$ (as opposed to $P_\sigma = \emptyset$). It remains to show how to correctly compute each P_τ. To do so, we distinguish based on the outermost operation of τ:

(a) $\tau = i(v)$ for some $i \in [\mathrm{cw}(G)]$. In this case $L_\tau = \{i\}$, and P_τ contains all functions $p : \{i\} \mapsto \mathcal{S}(H)$ such that $p(i) \subseteq N_H(u)$ for some $u \in V(H)$.

(b) $\tau = \rho_{i\to j}(\tau')$ and $P_{\tau'}$ has already been computed. If $i \notin L_{\tau'}$, we can correctly set $L_\tau = L_{\tau'}$ and $P_\tau = P_{\tau'}$. If $i \in L_{\tau'}$ and $j \notin L_{\tau'}$, then $L_\tau = (L_{\tau'} \setminus \{i\}) \cup \{j\}$ and

$$P_\tau = \left\{ p\colon L_\tau \to \mathcal{S}(H) \;\middle|\; \exists p' \in P_{\tau'} : \quad p(\ell) = \begin{cases} p'(\ell) & \text{if } \ell \neq j \\ p'(i) & \text{if } \ell = j \end{cases} \right\}.$$

Finally, if $\{i, j\} \subseteq L_{\tau'}$, then $L_\tau = L_{\tau'} \setminus \{i\}$ and $P_\tau = \{p'|_{L_\tau} \mid p' \in P_{\tau'} \wedge p'(i) = p'(j)\}$.

For correctness in the last case, let h be a homomorphism from G_τ to H and $S_\ell \in \mathcal{S}(H)$ be such that $S_\ell \subseteq S(h(V_\tau^\ell))$, $\ell \in L_\tau$. Observe that $V_\tau^\ell = V_{\tau'}^\ell$ for $\ell \in L_\tau \setminus \{j\}$ and $V_\tau^j = V_{\tau'}^j \cup V_{\tau'}^i$. In particular, $S_j \subseteq S(h(V_{\tau'}^i))$ and $S_j \subseteq S(h(V_{\tau'}^j))$. By definition of $P_{\tau'}$, there exists $p' \in P_{\tau'}$ such that $p'(\ell) = S_\ell$ for $\ell \in L_\tau \setminus \{j\}$ and $p'(i) = p'(j) = S_j$. The function $p \in P_\tau$, defined by $p = p'|_{L_\tau}$, satisfies $p'(\ell) = S_\ell$, $\ell \in L_\tau$.

On the other hand, fix some $p \in P_\tau$. Let $p' \in P_{\tau'}$ be a function such that p arises from p' in the construction of P_τ. Consider a witness h of p' in τ'. For every $\ell \in L_\tau \setminus \{j\}$ we have $V_\tau^\ell = V_{\tau'}^\ell$ and so $p(\ell) = p'(\ell) \subseteq S(h(V_\tau^\ell))$. Moreover, $p(j) = p'(j) \subseteq S(h(V_{\tau'}^j))$ and $p(j) = p'(i) \subseteq S(h(V_{\tau'}^i))$. By Observation 4, we have $p(j) \subseteq S(h(V_{\tau'}^i)) \cap S(h(V_{\tau'}^j)) = S(h(V_{\tau'}^i \cup V_{\tau'}^j)) = S(h(V_\tau^j))$. Hence p witnesses h in τ.

(c) $\tau = \tau^{(1)} \oplus \tau^{(2)}$ where $P_{\tau^{(1)}}$ and $P_{\tau^{(2)}}$ have already been computed. In this case $L_\tau = L_{\tau^{(1)}} \cup L_{\tau^{(2)}}$. We define

$$P_\tau = \{p = p_1 \cup p_2 \mid p_1 \in P_{\tau^{(1)}} \wedge p_2 \in P_{\tau^{(2)}} \wedge (\forall \ell \in L_{\tau^{(1)}} \cap L_{\tau^{(2)}} : p_1(\ell) = p_2(\ell))\}$$

Intuitively, we construct a homomorphism on the disjoint union of two subgraphs by "gluing together" the homomorphisms on the subgraphs. If the subgraphs share any live labels, after this step they will all be treated equally. For this reason we require the images of the neighbors of such labels to be the same in both subgraphs.

For correctness, let h be a homomorphism from G_τ to H and $S_\ell \in \mathcal{S}(H)$ be such that $S_\ell \subseteq S(h(V^\ell_\tau))$, $\ell \in L_\tau$. Observe that for every $\ell \in L_{\tau^{(1)}} \cap L_{\tau^{(2)}}$, $V^\ell_\tau \supseteq V^\ell_{\tau^{(i)}}$, so $S_\ell \subseteq S(h(V^\ell_{\tau^{(i)}}))$, $i = 1, 2$. By definition, there exists $p_i \in P_{\tau^{(i)}}$ such that $p_i(\ell) = S_\ell$ for $\ell \in L_{\tau^{(i)}}$, $i = 1, 2$. For $p = p_1 \cup p_2$ we have $p(\ell) = S_\ell$, $\ell \in L_\tau$.

For the converse, fix some $p \in P_\tau$. Let $p_1 \in P_{\tau^{(1)}}$, $p_2 \in P_{\tau^{(2)}}$ be functions such that $p = p_1 \cup p_2$. Let h_i be a witness of p_i in $\tau^{(i)}$, $i = 1, 2$. We define $h = h_1 \cup h_2$. Since G_τ doesn't contain edges between $V(G_{\tau^{(1)}})$ and $V(G_{\tau^{(2)}})$, h is a homomorphism from G_τ to H. For all $\ell \in L_{\tau^{(1)}} \setminus L_{\tau^{(2)}}$, we have $p(\ell) = p_1(\ell) \subseteq S(h_1(V^\ell_\tau)) = S(h(V^\ell_\tau))$, similarly for $\ell \in L_{\tau^{(1)}} \setminus L_{\tau^{(2)}}$. For $\ell \in L_{\tau^{(1)}} \cap L_{\tau^{(2)}}$, we have $p(\ell) = p_i(\ell) \subseteq S(h_i(V^\ell_{\tau^{(i)}}))$, $i = 1, 2$, so $p(\ell) \subseteq S(h_1(V^\ell_{\tau^{(1)}})) \cap S(h_2(V^\ell_{\tau^{(2)}})) = S(h(V^\ell_\tau))$. Hence h is a witness of p in τ.

(d) $\tau = \eta_{i,j}(\tau')$ and $P_{\tau'}$ has already been computed. In this case $L_\tau = L_{\tau'} \setminus I$ where $I \subseteq \{i, j\}$ is the set of live labels in τ' that are no longer live labels in τ. We set $P_\tau = \{p : L_\tau \to \mathcal{S}(H) \mid \exists p' \in P_{\tau'} \colon \big(p'(i) \supseteq S(p'(j)) \wedge p|_{L_\tau \setminus \{i,j\}} = p'|_{L_\tau \setminus \{i,j\}} \wedge p(i) \subseteq p'(i) \wedge p(j) \subseteq p'(j)\big)\}$.

Intuitively, we can add the edges between two live labels if and only if there are edges between their images in H. Our restriction on p' is an expression of this condition in terms of images of neighbors and their signatures. Indeed, for correctness, let h be a homomorphism from G_τ to H and $S_\ell \in \mathcal{S}(H)$ be such that $S_\ell \subseteq S(h(V^\ell_\tau))$, $\ell \in L_\tau$. There exists $p' \in P_{\tau'}$ such that $p'(\ell) = S_\ell$ for all $\ell \in L_\tau \setminus \{i, j\}$, $p'(i) = S(h(V^i_\tau))$ and $p'(j) = S(h(V^j_\tau))$. As h is a homomorphism, we have $h(V^i_{\tau'}) \times h(V^j_{\tau'}) \subseteq E(H)$, which means that $S(h(V^j_{\tau'})) \supseteq h(V^i_{\tau'})$, i.e. $p'(j) \supseteq h(V^i_{\tau'})$. Then $S(p'(j)) \subseteq S(h(V^i_{\tau'})) = p'(i)$ and hence p' gives rise to $p \in P_\tau$ such that $p(\ell) = S_\ell$, $\ell \in L_\tau$.

On the other hand, let $p \in P_\tau$ arise from $p' \in P_{\tau'}$. Consider a witness $h \colon G_{\tau'} \to H$ of p' in τ'. To see that h preserves edges between $V^i_{\tau'}$ and $V^j_{\tau'}$, recall that $p'(i) \supseteq S(p'(j))$, so $S(h(V^i_{\tau'})) \supseteq p'(i) \supseteq S(p'(j)) \supseteq S(S(h(V^j_{\tau'}))) \supseteq h(V^j_{\tau'})$. Hence $h(V^i_{\tau'}) \times h(V^j_{\tau'}) \subseteq E(H)$ and h is a homomorphism from G_τ to H. By construction, for every $\ell \in L_\tau$ it holds that $p(\ell) \subseteq p'(\ell) \subseteq S(h(V^\ell_\tau))$, so h witnesses p in τ.

It is easy to verify that $|P_\tau| \leqslant s(H)^{\mathrm{cw}(G)}$ for each subexpression τ of σ. This means that in each step, the computation requires time $\mathcal{O}(\mathrm{cw}(G)s(H)^2 s(H)^{\mathrm{cw}((G))})$. Overall this yields a complexity of $\mathcal{O}(|V(G)|\mathrm{cw}(G)s(H)^2 s(H)^{\mathrm{cw}((G))}) \subseteq \mathcal{O}^*(s(H)^{\mathrm{cw}(G)})$. ◀

4 On Products and Projectivity

While Theorem 1 will serve as the upper bound that will match our target SETH-based lower bounds for $\textsc{Hom}(H)$ for the "most difficult" choices of H, in many cases one can in fact supersede the algorithm's runtime by exploiting well-known properties of target graphs.

As a simple example showcasing this, consider the *wheel* graph W_6 (see Figure 1). Since W_6 is 3-colorable, it holds that $W_6 \to K_3$, and since K_3 is a core and an induced subgraph of W_6, it is the core of W_6. We recall that if H is a core of H', then for every graph G it holds that $G \to H$ if and only if $G \to H'$. Hence, having an instance G of $\textsc{Hom}(W_6)$, we can compute a core of W_6 (since we assume that the target graph is fixed, this can be done in constant time), and use Theorem 1 for $H = K_3$ to decide whether $G \to W_6$ in total running time $\mathcal{O}^*(s(K_3)^{\mathrm{cw}(G)})$. As $s(K_3) < s(W_6)$ (as showcased in Figure 1), this yields a better running time bound than the direct use of Theorem 1. While this example shows that the signature number can decrease by taking an induced subgraph, we remark that it can never increase.

▶ **Observation 9.** *Let H and H' be graphs such that H is an induced subgraph of H'. Then $s(H) \leqslant s(H')$.*

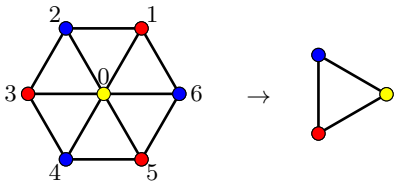

Figure 1 The graphs W_6 (left) and K_3 (right). Colors on the vertices of W_6 indicate the homomorphism $h : W_6 \to K_3$. We note that $\{\{0\}\} \cup \{\{0, i\} \mid i \in [6]\} \subseteq \mathcal{S}(W_6)$. Since K_3 is a clique, we have $s(K_3) = 6 < 7 \leqslant s(W_6)$.

Proof. Given a connected graph Q without loops, one may consider an equivalence relation $\sim_Q$ on the set of proper nonempty subsets of $V(Q)$ defined as follows: $V_1 \sim_Q V_2$ if and only if V_1 and V_2 have the same signature sets in Q. Observe that $s(Q)$ is equal to the number of equivalence classes of $\sim_Q$. Hence to prove the claim, it suffices to show that whenever two sets belong to different equivalence classes of $\sim_H$, they also belong to different equivalence classes of $\sim_{H'}$. For this, consider any two proper non-empty subsets V_1 and V_2 of $V(H)$ such that $V_1 \nsim_H V_2$. Without loss of generality, we assume that there exists $v \in (\bigcap_{t \in V_1} N_H(t)) \setminus (\bigcap_{t \in V_2} N_H(t))$. Then $vt \in E(H) \subseteq E(H')$ for every $t \in V_1$, i.e., v belongs to the signature set of V_1 in H'. On the other hand, $vt_0 \notin E(H)$ for some $t_0 \in V_2$. As H is induced subgraph of H', it means that $vt_0 \notin E(H')$, so v doesn't belong to the signature set of V_2 in H' and hence $V_1 \nsim_{H'} V_2$. ◀

At this point, we may ask whether the procedure of simply computing the unique core H of the fixed target H' and then applying Theorem 1 for H could yield a tight upper bound for $\textsc{Hom}(H')$. Unfortunately, the situation is more complicated than that, and we need to introduce a few important notions in order to capture the problem's fine-grained complexity.

Let the *direct product* $H_1 \times H_2$ of graphs H_1, H_2 be the graph defined as follows:

$$V(H_1 \times H_2) = V(H_1) \times V(H_2),$$
$$E(H_1 \times H_2) = \{(x_1, x_2)(y_1, y_2) : x_i y_i \in E(H_i) \text{ for every } i \in \{1, 2\}\}.$$

We call H_1 and H_2 the *factors* of $H_1 \times H_2$. Clearly, the operation $\times$ is commutative, and since it is also associative, we can naturally extend the definition of direct product to more than two factors, i.e., $H_1 \times H_2 \times \ldots \times H_m = H_1 \times (H_2 \times \ldots \times H_m)$. Note that for every graph H that contains at least one edge, it holds that $H \times K_1^* = H$.

In the remaining part of the paper we will often consider vertices that are tuples. If such a vertex is an argument of some function and in cases where this does not lead to confusion, we omit one pair of brackets; similarly, we omit internal brackets in nested tuples where this does not lead to confusion. Moreover, for any graph H and for an integer ℓ, we denote by H^ℓ the graph $\overbrace{H \times \ldots \times H}^{\ell}$. As an example, instead of writing $((x_1, x_2), y_1) \in ((H_1 \times H_1) \times H_2)$, we write $(x_1, x_2, y_1) \in (H_1^2 \times H_2)$.

If $H = H_1 \times \ldots \times H_m$ for some graphs $H_1, \ldots, H_m$, we say that $H_1 \times \ldots \times H_m$ is a *factorization* of H. A graph H on at least two vertices is *prime* if the fact that $H = H_1 \times H_2$ for some graphs H_1, H_2 implies that $H_1 = K_1^*$ or $H_2 = K_1^*$. If H has a factorization $H_1 \times \ldots \times H_m$ such that for every $i \in [m]$ the graph H_i is prime, we call $H_1 \times \ldots \times H_m$ a *prime factorization* of H.

▶ **Theorem 10** ([8, 14]). *Any connected non-bipartite graph with more than one vertex has a unique prime factorization (into factors with possible loops).*

Consider a graph $H_1 \times \ldots \times H_m$, and let $i \in [m]$. A mapping $\pi_i : V(H_1 \times \ldots \times H_m) \to V(H_i)$ such that $\pi_i(x_1, \ldots, x_\ell) = x_i$ is called the (i-th) *projection* of $H_1 \times \ldots \times H_m$. Clearly, such a mapping is always a homomorphism.

▶ **Observation 11.** *Let $G, H_1, \ldots, H_m$ be graphs. Then $G \to H_1 \times \ldots \times H_m$ if and only if for every $i \in [m]$ we have $G \to H_i$.*

Proof. Let $f : G \to H_1 \times \ldots \times H_m$. Then for every $i \in [m]$ (by $[m]$ we denote the set of integers $\{1, \ldots, m\}$) we have a homomorphism $f_i : G \to H_i$. Conversely, if for every $i \in [m]$ we have $g_i : G \to H_i$, then we can define $g : G \to H_1 \times \ldots \times H_m$ as $g(x) = (g_1(x), \ldots, g_m(x))$. ◀

Crucially, in some cases Observation 11 allows us to improve the bounds given by Theorem 1 even if H is a core, simply by considering all possible factorizations of H.

▶ **Corollary 12.** *Let H be a graph with factorization $H_1 \times \ldots \times H_m$, and let G be an instance graph of Hom(H). Assuming that the clique-width expression σ of G of width $\mathrm{cw}(G)$ is given, the Hom(H) problem can be solved in time $\max_{i \in [m]} \mathcal{O}^*\big(s(H_i)^{\mathrm{cw}(G)}\big)$.*

Proof. Observe that if G is an instance of Hom(H), by Theorem 1 for every $i \in [m]$ we can decide whether $G \to H_i$ in time $\mathcal{O}^*\big(s(H_i)^{\mathrm{cw}(G)}\big)$. Then, if G is a yes-instance of Hom(H_i) for every $i \in [m]$, we return that G is a yes-instance of Hom(H). Otherwise, we return that G is a no-instance of Hom(H). The correctness of this procedure follows from Observation 11. ◀

On the other hand, the notion of signature sets we introduced in the previous section behaves multiplicatively with respect to taking direct product of graphs.

▶ **Observation 13.** *Let $H = H_1 \times H_2$. Then $\mathcal{S}(H) = \mathcal{S}(H_1) \times \mathcal{S}(H_2)$.*

Proof. We prove that $\mathcal{S}(H)$ is of form $\{S(T_1) \times S(T_2) : T_i \subseteq V(H_i), S(T_i) \neq \emptyset \text{ for } i = 1, 2\}$.

Let T_1 and T_2 be some subsets of, respectively, $V(H_1)$ and $V(H_2)$. Clearly,

$$S(T_1) \times S(T_2) = [\bigcap_{t \in T_1} N(t)] \times [\bigcap_{t' \in T_2} N(t')] = \bigcap_{(t,t') \in T_1 \times T_2} N(t, t') = S(T_1 \times T_2). \tag{1}$$

Therefore, if $S(T_1)$ and $S(T_2)$ are non-empty, we get that $S(T_1) \times S(T_2) \in \mathcal{S}(H)$.

To see that $\mathcal{S}(H) \subseteq \mathcal{S}(H_1) \times \mathcal{S}(H_2)$, we show that for every $T \subseteq V(H)$ set $S(T)$ is of the form $S(T_1) \times S(T_2)$ for some T_1, T_2. Define T_1 and T_2 to be minimal sets such that $T \subseteq T_1 \times T_2$. Hence, by (1), $S(T_1) \times S(T_2) = S(T_1 \times T_2) \subseteq S(T)$. On the other hand, for every $(s, s') \in S(T)$ we have $s \in \bigcap_{t \in T_1} N(t)$ and $s' \in \bigcap_{t' \in T_2} N(t')$, so the equality follows. ◀

It follows from Observation 13 that if H is a graph with factorization $H_1 \times \ldots \times H_m$, then $s(H) = s(H_1) \cdot \ldots \cdot s(H_m)$. Therefore if there exist at least two factors H_i, H_j such that $s(H_i), s(H_j) > 1$, Corollary 12 yields a better running time than Theorem 1.

In order to analyze the possible matching lower bounds for our algorithms, in the remaining part of the section, we focus only on connected non-trivial cores H that are provided with their unique prime factorization $H_1 \times \ldots \times H_m$; if H is prime, we technically consider this factorization to be $H \times K_1^*$ (noting that this is not a prime factorization, and that K_1^* is the only non-simple graph in this article). We note that the factors of a core must satisfy some necessary conditions.

▶ **Observation 14** ([25])**.** *Let H be a connected, non-trivial core with factorization $H = H_1 \times \ldots \times H_m$ such that $H_i \neq K_1^*$ for all $i \in [m]$. Then for every $i \in [m]$ the graph H_i is a connected non-trivial core, incomparable with H_j for $j \in [m] - \{i\}$.*

Observation 14 in particular implies that if H is a connected non-trivial graph with factorization $H_1 \times \ldots \times H_m$, then at least one of the factors H_i must be non-trivial, and that K_1 and K_2 never appear as factors of a connected non-trivial graph.

In the remaining part of this section we introduce a few more important definitions, in particular, the well-established notion of *projectivity* for non-trivial graphs H.

We say that a homomorphism $f : H^\ell \to H$, for some $\ell \geqslant 2$, is *idempotent* if for each $x \in V(H)$ it holds that $f(x, \ldots, x) = x$. Graph H is *projective* if for every $\ell \geqslant 2$, every idempotent homomorphism $f : H^\ell \to H$ is a projection. We note that every projective graph on at least three vertices must be connected, ramified, non-bipartite and prime [22].

Here, we introduce a generalization of the projectivity property for non-trivial cores, which turns out to be the central component required to establish the lower bound for our problem. As a first step towards this, we lift the notion of idempotency as follows. Let H be a connected, non-trivial prime core, and let W be either a connected core on at least three vertices incomparable with H, or the graph K_1^*. Let $f : A \to H$ be a homomorphism where $A = H^\ell \times W$; observe that ℓ is uniquely determined by either W being incomparable with the prime core H, or W being K_1^*. We say that f is *H-idempotent* if for each $x \in V(H), y \in V(W)$ it holds that $f(x, \ldots, x, y) = x$.

Now, let us consider a non-trivial core H which admits a prime factorization $H_1 \times \ldots \times H_m$ and let $i \in [m]$. We say that H is *H_i-projective* if H_i is non-trivial and every H_i-idempotent homomorphism $f : H_1 \times \ldots \times H_{i-1} \times H_i^\ell \times H_{i+1} \times \ldots \times H_m \to H_i$ is a projection. In other words, for every homomorphism $f : H_1 \times \ldots H_{i-1} \times H_i^\ell \times H_{i+1} \times \ldots \times H_m \to H_i$ such that for every $x \in V(H_i), y_j \in V(H_j)$ for $j \in [m] - \{i\}$ it holds that $f(y_1, \ldots, y_{i-1}, x, \ldots, x, y_{i+1}, \ldots, y_m) = x$, we must have that there exists $q \in \{i, \ldots, i + \ell - 1\}$ such that $f \equiv \pi_q$. Recall that if H is a non-trivial projective core, then it must be prime, so $H \times K_1^*$ is its only possible factorization. It is straightforward to verify that in a such case H is H-projective.

Since the direct product of graphs is commutative, if $H = H_1 \times \ldots \times H_m$ is H_i-projective for some $i \in [m]$, to simplify the notation we will often assume w.l.o.g. that $i = 1$.

5 Hardness

In this section, we focus on establishing the desired lower bounds, stated below.

▶ **Theorem 2.** *Let H be a fixed non-trivial core with prime factorization $H_1 \times \ldots \times H_m$. Assume that H is H_i-projective for some $i \in [m]$. Then there is no algorithm solving $\textsc{Hom}(H)$ in time $\mathcal{O}^*((s(H_i) - \varepsilon)^{\mathrm{cw}(G)})$ for any $\varepsilon > 0$, unless the* SETH *fails.*

We divide our proof into two main steps. First, we show that in our setting, instead of considering the $\textsc{Hom}(H)$ problem, we may focus on the Homomorphism Extension problem, denoted $\textsc{HomExt}(H)$. For a fixed H, $\textsc{HomExt}(H)$ takes as an instance a pair (G', h'), where $h' : V' \to V(H)$ is a mapping from some $V' \subseteq V(G')$. We ask whether there exists *an extension* of h' to G', i.e., a homomorphism $h : G' \to H$ such that $h|_{V'} \equiv h'$.

The $\textsc{HomExt}(H)$ is clearly a generalization of the $\textsc{Hom}(H)$ problem. However, as the first step of our proof, we show that if H is a fixed non-trivial core, each instance (G', h') of $\textsc{HomExt}(H)$ can be transformed in polynomial time into an instance G of $\textsc{Hom}(H)$, such that $\mathrm{cw}(G')$ and $\mathrm{cw}(G)$ differ only by a constant.

▶ **Theorem 15.** *Let H be a fixed non-trivial core. Given an instance (G', h') of $\textsc{HomExt}(H)$, we can construct an equivalent instance G of $\textsc{Hom}(H)$ such that $\mathrm{cw}(G) \leqslant \mathrm{cw}(G') + |V(H)|$.*

Proof. Let $V' \subseteq V(G')$ be the domain of h'. We construct G by taking a copy $\hat{G}'$ of G' and a copy $\hat{H}$ of H. Then, for every $v \in V'$ we add all the edges with one endpoint in v and another one in $N_{\hat{H}}(h'(v))$.

Observe that if there exists an extension $h : G' \to H$ of h', then h can be also extended to G, by setting $h(v) = v$ for every $v \in V(\hat{H})$. Indeed, let $uv \in E(G)$. If $u, v \in V(\hat{G}')$, then $h(u)h(v) \in E(H)$, by definition of the extension of h' to G'. If $u, v \in V(\hat{H})$, then $h(u)h(v) = uv \in E(H)$. Finally, assume that $u \in V(\hat{G}'), v \in V(\hat{H})$. Note that, by definition of G, this can happen only if $u \in V'$ and v is adjacent to $h'(u)$. Hence, $h(u)h(v) = h'(u)v \in E(H)$.

For the reverse direction, assume that there exists a homomorphism $f : G \to H$. We show that there exists an extension $h : G' \to H$ of h'. Let $\sigma : H \to H$ be a restriction of f to H. Since H is a core, σ is an automorphism of H. We claim that $g = \sigma^{-1} \circ f : G \to H$, restricted to G', is an extension of h'. Clearly, g is a composition of homomorphisms, so also a homomorphism. Therefore, it remains to show that for every $v \in V'$ we have $h'(v) = g(v)$. Since H is a core, and $N_H(h'(v)) \subseteq N_G(v)$, we have that $f(v) = f(h'(v))$. It follows that $g(v) = \sigma^{-1} \circ f(v) = \sigma^{-1} \circ f(h'(v)) = g(h'(v))$. However, recall that for every $u \in V(H)$ we have that $g(u) = u$, so in particular, $g(h'(v)) = h'(v)$.

To see that $\mathrm{cw}(G) \leqslant \mathrm{cw}(G') + |V(H)|$, observe that we added exactly $|V(H)|$ vertices to G'. This means we can modify a clique-width expression σ for G' to obtain a clique-width expression of G as follows. Each added vertex is introduced with a designated label that is distinct from all labels used in σ. Then each subexpression of σ that introduces a vertex of G' can be replaced by an expression that introduces the vertex and inserts all required edges to the added vertices. Finally, one can insert the missing edges between added vertices. ◀

As the second step, we prove the following theorem.

▶ **Theorem 16.** *Let H be a fixed non-trivial core with prime factorization $H_1 \times \ldots \times H_m$. Assume that H is H_i-projective for some $i \in [m]$. Then there is no algorithm solving* HomExt*(H) in time $\mathcal{O}^*((s(H_i) - \varepsilon)^{\mathrm{cw}(G')})$ for any $\varepsilon > 0$, unless the SETH fails.*

Before we proceed to the proof of Theorem 16, we show that it implies Theorem 2.

Theorem 16 → Theorem 2: Let H be a non-trivial core with a prime factorization $H_1 \times \ldots \times H_m$. W.l.o.g. assume that H is H_1-projective. Suppose that Theorem 2 does not hold, i.e., there exists an algorithm A that solves every instance G of Hom(H) in time $\mathcal{O}^*((s(H_1) - \varepsilon)^{\mathrm{cw}(G)})$.

Let (G', h') be an instance of HomExt(H). We use Theorem 15 to transform (G', h') into an equivalent instance G of Hom(H), such that $\mathrm{cw}(G) \leqslant \mathrm{cw}(G') + |V(H)|$. Then, we use A to decide whether $G \to H$ in time

$$\mathcal{O}^*\left((s(H_1) - \varepsilon)^{\mathrm{cw}(G)} = \mathcal{O}^*\left((s(H_1) - \varepsilon)^{\mathrm{cw}(G')} \cdot (s(H_1) - \varepsilon)^{|V(H)|}\right)\right.\!.$$

Since H is a fixed graph, $(s(H_1) - \varepsilon)^{|V(H)|}$ is a constant, and therefore $\mathcal{O}^*\left((s(H_1) - \varepsilon)^{\mathrm{cw}(G)}\right) = \mathcal{O}^*\left((s(H_1) - \varepsilon)^{\mathrm{cw}(G')}\right)$. Since $G \to H$ if and only if (G', h') is a yes-instance of HomExt(H), we get a contradiction with Theorem 16.

We will prove Theorem 16 for $i = 1$, which covers other cases by commutativity of direct products. We begin by constructing certain gadgets that will be used later. Let H be a fixed core with factorization $H_1 \times \ldots \times H_m$. We define $W = H_2 \times \ldots \times H_m$ if $m \geqslant 2$, and $W = K_1^*$ otherwise. Clearly, $H_1 \times W$ is a (not necessarily prime) factorization of H. Moreover, if for some graph G we have a homomorphism $f : G \to H_1 \times \ldots, \times H_m$, for $i \in [m]$ we denote by f_i the homomorphism $\pi_i \circ f : G \to H_i$.

Let S be a set of pairs of vertices of H_1, and let $w, w' \in V(W)$. We say that a tuple (F, h', p, q), such that F is a graph, $h' : V' \to H$ is a mapping with domain $V' \subseteq V(F)$, and $p, q \in V(F)$, is an (S, w, w')-*gadget* if

(S1) for every extension $h : F \to H$ of h', it holds that $(h_1(p), h_1(q)) \in S$,

(S2) for every pair $(s_1, s_2) \in S$ there exists an extension $h : F \to H$ of h' such that $h(p) = (s_1, w)$ and $h(q) = (s_2, w')$.

▶ **Lemma 17.** *Let H be a non-trivial connected core with factorization $H_1 \times W$, let $S \subseteq V(H_1)^2$, and let $w, w' \in V(W)$. Assume that H is H_1-projective. Then there exists an (S, w, w')-gadget.*

Proof. Let $S = \{(s_1^1, s_2^1), \ldots, (s_1^\ell, s_2^\ell)\}$. Define

$$F = H_1^\ell \times W, \quad \text{and} \quad p = (s_1^1, \ldots, s_1^\ell, w), \quad \text{and } q = (s_2^1, \ldots, s_2^\ell, w').$$

Let $V' = \{(x, x, \ldots, x, y) \mid x \in V(H_1), y \in V(W)\}$, and let $h'(x, \ldots, x, y) = (x, y)$. We claim that (F, h', p, q) is an (S, w, w')-gadget.

The condition 1 follows from the fact that H is H_1-projective. Indeed, observe that if $h : F \to H_1 \times W$ is an extension of h', then h_1 must be H_1-idempotent, and hence a projection on one of the ℓ first coordinates. Therefore, we must have $(h_1(p), h_1(q)) \in S$.

For (S2), take any $(s_1^i, s_2^i) \in S$ and let $h : F \to H_1 \times W$, $h(x) = (\pi_i(x), \pi_{\ell+1}(x))$. Clearly, h is an extension of h', and it is easy to verify that $h(p) = (s_1^i, w)$ and $h(q) = (s_2^i, w')$. ◀

We say that $S \subseteq V(H_1)^2$ is *proper*, if for every coordinate there exist two elements is S that differ on that coordinate, i.e., S is not of the form $\{s\} \times U$ nor $U \times \{s\}$ for some $s \in V(H_1)$ and $U \subseteq V(H_1)$. Note that if S is proper and (F, h', p, q) is an (S, w, w')-gadget, then neither p nor q belong to the domain of h'.

For fixed vertices $a, b \in V(H_1)$, let $S_{a,b} = \{(a', b') : a' \neq a, b' \in V(H_1)\} \cup \{(a, b)\}$. We call the $(S_{a,b}, w, w')$-gadget (F, h', p, q) an $((a, b), w, w')$-*implication-gadget*. Intuitively, an $((a, b), w, w')$-implication-gadget works as the implication $a \Rightarrow b$, since in every homomorphism $h : F \to H$ that extends h', if $h_1(p) = a$, then $h_1(q) = b$.

Let $a, b, c \in V(H_1)$, $w \in V(W)$, and let t be an integer. A triple (F, h', R) such that F is a graph, $h' : V' \to H_1 \times W$ is a partial mapping from some $V' \subseteq V(F)$, and R is a subset of $V(F)$ of cardinality t is an *t-or-gadget with domain* $((a, b, c), w)$ if

(O1) for every homomorphism $h : F \to H$ that is an extension of h', and for every $u \in R$ we have that $h_1(u) \in \{a, b, c\}$ and there exists $v \in R$ such that $h_1(v) = a$,

(O2) for every $v \in R$ there exists a homomorphism $h : F \to H$ that is an extension of h', such that $h(v) = (a, w)$ and for every $u \in R - \{v\}$ it holds that $h(u) \in \{(b, w), (c, w)\}$.

▶ **Lemma 18.** *Let H be a non-trivial core with factorization $H_1 \times W$. Assume that H is H_1-projective. Then for every distinct $a, b, c \in V(H_1)$, every $w \in V(W)$ and every t, there exists a t-or-gadget (F, h', R) with domain $((a, b, c), w)$.*

Proof. We consider separately the cases $t = 1$ and $t = 2$. Observe that in case $t = 1$ our gadget needs to be a graph that has a vertex $r \in R$ that is always mapped to (a, w). Hence, we set $F = K_1$, $R = V(F)$, and $h'(v) = (a, w)$ for $v \in V(F)$.

If $t = 2$, let $S = \{(a, b), (b, a), (a, a)\}$, we introduce an independent set $R = \{r_1, r_2\}$ and (S, w, w)-gadget (F, h', r_1, r_2). To see that (F, h', R) satisfies 1, consider any extension $f : F \to H$ of h'. As (F, h', r_1, r_2) is (S, w, w)-gadget, we have $(f_1(r_1), f_1(r_2)) \in \{(a, b), (a, a), (b, a)\}$. For 2, recall that by the property 2 of S-gadget there exist extensions $f^{(1)}$ and $f^{(2)}$ of h' such that $(f_1^{(1)}(r_1), f_1^{(1)}(r_2)) = (a, b)$ and $(f_1^{(2)}(r_1), f_1^{(2)}(r_2)) = (b, a)$.

Assume then that $t \geqslant 2$, and let

$$\begin{aligned} S &= \{a,b,c\}^2 - \{(b,c),(c,b)\}, \\ S_{\text{left}} &= \{(a,a),(a,b),(a,c),(c,a),(c,c)\}, \\ S_{\text{right}} &= \{(a,a),(a,b),(b,a),(b,b),(c,a)\} \end{aligned}$$

be subsets of $V(H_1)^2$. We introduce an independent set $R = \{r_1, \ldots, r_t\}$ of t vertices and one copy of (S_{left}, w, w)-gadget (F_1, h'_1, p_1, q_1) from r_1 to r_2. Then, for $j \in \{2, \ldots, t-2\}$, we introduce an (S, w, w)-gadget (F_j, h'_j, p_j, q_j) from r_j to r_{j+1} (we note that if $t \leqslant 3$, we do not introduce these). Last, we introduce one copy of (S_{right}, w, w)-gadget $(F_{t-1}, h'_{t-1}, p_{t-1}, q_{t-1})$ from r_{t-1} to r_t. We note that sets S, S_{left}, and S_{right} are proper, so the domains of the partial mappings h'_j, $j \in \{1, \ldots, t-1\}$, are pairwise disjoint. In particular, the union $h' = \bigcup_{j=1}^{t} h'_j$ is a well-defined mapping. We define F to be the union of all the graphs from the introduced gadgets and claim that (F, h', R) is a t-or-gadget. We first show that 1 holds. Assume that there exists an extension $f : F \to H$ of h', and $j' \in [t]$ such that $f_1(r_{j'}) \notin \{a, b, c\}$. This implies that there exists $j \in \{j'-1, j'\}$ such that $(f_1(r_j), f_1(r_{j+1})) \notin S'$ for any $S' \in \{S_{\text{left}}, S, S_{\text{right}}\}$. This is a contradiction with (F, h'_j, p_j, q_j) being an (S', w, w)-gadget, as it violates 1.

Now assume that there exists an extension $f : F \to H$ of h' such that for every $j \in [t]$ we have that $f_1(r_j) \in \{b, c\}$. The definition of S_{left} and S_{right}, respectively, implies that $f_1(r_1) = c$ and $f_1(r_t) = b$. Hence, there exists $j \in [t-1]$ such that $f_1(r_j) = c$ and $f_1(r_{j+1}) = b$. However, observe that the pair (c, b) does not belong to set S', for $S' \in \{S_{\text{left}}, S, S_{\text{right}}\}$, and since we introduced an S'-gadget from r_j to r_{j+1}, this leads to a contradiction.

To see that 2 holds as well, fix some $r_j \in R$ and define

$$f'(r_\ell) = \begin{cases} (a, w), & \text{if } \ell = j, \\ (c, w), & \text{if } \ell < j, \\ (b, w), & \text{if } \ell > j, \end{cases}$$

If $j = 1$, then since $(a, b) \in S_{\text{left}}$, $(b, b) \in S$ and $(b, b) \in S_{\text{right}}$, the property 2 asserts that we can construct a homomorphism $f : F \to H$ that extends h' and f'. The same holds also if $j = t$, (since $(c, c) \in S_{\text{left}}$, $(c, c) \in S$ and $(c, a) \in S_{\text{right}}$), and if $1 < j < t$ (since $(c, c) \in S_{\text{left}}$, $(c, c), (c, a), (a, b), (b, b) \in S$ and $(b, b) \in S_{\text{right}}$). ◀

Finally, all that remains is to prove Theorem 16. Our reduction generalizes the construction used by Lampis [20] to reduce an SETH lower-bounded constraint satisfaction problem to k-Coloring. Intuitively speaking, in that construction possible variable assignments are encoded by mapping specified vertices to arbitrary non-trivial subsets of the colors. The straightforward generalization of this approach to our setting would be to map to non-trivial subsets of $V(H)$. However, the structure of H allows only certain configurations of subsets as images for the specified vertices in a solution for $\textsc{Hom}(H)$ – which is precisely where the signature sets come into play.

Let $q, B \geqslant 2$ be integers. We will reduce from the q-CSP-B problem that is defined as follows. An instance of q-CSP-B consists of a set X of variables and a set C of q-constraints. A q-constraint $c \in C$ is a q-tuple of elements from X and a set $P(c)$ of q-tuples of elements from $[B]$ (i.e., $P(c) \subseteq [B]^q$). The q-CSP-B problem asks whether there exists an assignment $\gamma : X \to [B]$, such that each constraint is satisfied, i.e., if $c = ((x_1, \ldots, x_q), P(c)) \in C$, then $(\gamma(x_1), \ldots, \gamma(x_q)) \in P(c)$. Note that we can assume that q-constraints in our q-CSP-B instance may have less than q vertices, as it is always possible to add at most $q-1$ dummy variables to X and add them to constraints that are of smaller size.

We will use the following theorem.

▶ **Theorem 19** ([20]). *For any $B \geqslant 2, \varepsilon > 0$ we have the following: assuming the SETH, there exists q such that n-variable q-CSP-B cannot be solved in time $\mathcal{O}^*((B-\varepsilon)^n)$.*

We have all the tools to perform the final reduction.

Proof of Theorem 16. Recall that it is sufficient to prove the theorem when $H = H_1 \times W$ is non-trivial H_1-projective core ($W = K_1^*$ if $H = H_1$). Fix $\varepsilon > 0$ and set $B = s(H_1)$. As H is H_1-projective, H_1 is non-trivial and hence contains at least three distinct vertices a, b and c. In particular, $B \geqslant 3$ by Observation 8. Since $H = H_1 \times W$ is a non-trivial core, W must have at least one edge ww' (it may happen that $w = w'$). From now on a, b, c, w and w' are fixed. Let q be the smallest number such that q-CSP-B on n variables cannot be solved in time $\mathcal{O}^*((B-\varepsilon)^n)$ assuming the SETH, given by Theorem 19.

Let φ be an instance of q-CSP-B, where $X = \{x_1, \ldots, x_n\}$ is the set of variables and $C = \{c_0, \ldots, c_{m-1}\}$ is the set of constraints. For every $j \in \{0, \ldots, m-1\}$ denote by X_j the set of variables that appear in the constraint c_j. Let $P(c_j) = \{f_1^j, \ldots, f_{p_j}^j\}$ be the set of assignments from X_j to $[B]$ that satisfy the constraint c_j. Let $L = m(n|H_1| + 1)$, and let $\lambda : [B] \to \mathcal{S}(H_1)$ be some fixed bijection.

We construct the instance G_φ of HomExt(H). For each $j \in \{0, \ldots L-1\}$, let $j' = j \mod m$. Let $R_j = \{r_1^j, \ldots, r_{p_{j'}}^j\}$, where each vertex r_k^j corresponds to the assignment $f_k^{j'}$. We introduce the $p_{j'}$-or-gadget (F_j, h'_j, R_j) with domain $((a,b,c), w)$.

For each $x_i \in X_{j'}$, and for each $f_k^{j'} \in P(c_{j'})$ we do the following:

1. Let $y = f_k^{j'}(x_i) \in [B]$. Construct an independent set $V_i^{j,k}$ of $|\lambda(y)|$ vertices and an independent set $U_i^{j,k}$ of $|S(\lambda(y))|$ vertices.
2. For each $d \in \lambda(y)$ select a distinct vertex $z \in V_i^{j,k}$ and add an $((a,d), w, w')$-implication-gadget from r_k^j to z. For each $d \in S(\lambda(y))$ select a distinct vertex $z \in U_i^{j,k}$ and add an $((a,d), w, w')$-implication-gadget from r_k^j to z.
3. Connect all vertices of $U_i^{j,k}$ with all vertices of previously constructed sets $V_i^{\ell,k'}$ for $\ell < j$ and $k' \in [p_\ell]$.

The partial mapping h' is the union of all the partial mappings that are introduced by all the gadgets. This finishes the construction of the instance (G_φ, h') of HomExt(H).

▷ Claim 20. If φ is a yes-instance of q-CSP-B, then there exists a homomorphism $h : G_\varphi \to H$ that extends h'.

Proof of Claim. If φ is a yes-instance of q-CSP-B, then there exists an assignment $\gamma : X \to [B]$ satisfying each constraint. We define $h : G_\varphi \to H$ as follows.

Fix $j \in \{0, \ldots, L-1\}$, and consider the or-gadget (F_j, h'_j, R_j). Recall that the set $P(c_{j'})$ consists of all assignments of variables in $X_{j'}$ that satisfy the constraint $c_{j'}$. Therefore, there exists an assignment $f_k^{j'} \in P(c_{j'})$ such that $\gamma|_{X_{j'}} \equiv f_k^{j'}$. Consider the vertex r_k^j that corresponds to that assignment. By the property 2 of the or-gadget, we know that there exists a H-coloring of F_j that extends h', such that (i) $h_1(r_k^j) = a$ and (ii) for every $r_{k'}^j \in R_j, k' \neq k$ we have that $h_1(r_{k'}^j) \in \{b, c\}$.

Let $x_i \in X_{j'}$ and let $y = f_k^{j'}(x_i) \in [B]$. Since for each $d \in \lambda(y)$ there exists a vertex $z \in V_i^{j,k}$ such that there is an $((a,d), w, w')$-implication-gadget from r_k^j to z, the condition (i) implies that $h_1(V_i^{j,k}) = \lambda(y)$. We color the vertices of $V_i^{j,k}$ in a way that $h(V_i^{j,k}) = \lambda(y) \times \{w\}$.

Also, since for each $d \in S(\lambda(y))$ there exists a vertex $z \in U_i^{j,k}$ such that there is an (a,d)-implication-gadget from r_k^j to z, the condition (i) implies that $h_1(U_i^{j,k}) = S(\lambda(y))$. We color the vertices of $U_i^{j,k}$ in a way that $h(U_i^{j,k}) = S(\lambda(y)) \times \{w'\}$.

Because of (ii), the implication gadgets from $r_{k'}$ to the vertices of $V_i^{j,k'} \cup U_i^{j,k'}$ do not put any constraints on the coloring of the sets $V_i^{j,k'}$ and $U_i^{j,k'}$. Therefore, for each $v \in V_i^{j,k'}$ we set $h(v)$ to be any vertex from $\lambda(y)$. Similarly, for each $u \in U_i^{j,k'}$ we set $h(u)$ to be any vertex from $S(\lambda(y))$. Since $(b,x),(c,x) \in S_{a,b}$ for any $x \in V(H_1)$, property 2 applied to the implication gadgets asserts that since $h(r_{k'}) \in \{b,c\}$, we can always extend this mapping to a homomorphism of the whole gadget to H.

It remains to argue that the edges between the sets $V_i^{j_1,k_1}$ and $U_i^{j_2,k_2}$ are mapped to edges of H, for any $j_1 < j_2$ and k_1, k_2. However, observe that since γ is an extension of some $f_{k_1}^{j_1'} \in S_{j_1'}$ and $f_{k_2}^{j_2'} \in S_{j_2'}$, we must have $f_{k_1}^{j_1'}(x_i) = f_{k_2}^{j_2'}(x_i) = y$. Hence, h maps every $v \in V_i^{j_1,k_1}$ to some element of $\lambda(y) \times \{w\}$, and every $u \in U_i^{j_2,k_2}$ to some element of $S(\lambda(y)) \times \{w'\}$. By Observation 4, and since $ww' \in E(W)$, we get that $h(v)h(u) \in E(H)$. That concludes the proof of the claim. ◁

▷ Claim 21. If there exists a homomorphism $h : G_\varphi \to H$ that extends h', then φ is a yes-instance of q-CSP-B.

Proof of Claim. We will define the assignment $\gamma : X \to [B]$ that makes every constraint from C satisfied.

Fix $j \in \{0,\ldots,L-1\}$, and consider the $p_{j'}$-or-gadget (F_j, h'_j, R_j). By the property 1 of the or-gadget, there exists $k_j \in [p_{j'}]$ such that $h_1(r_{k_j}^j) = a$. Implication gadgets whose p-vertices were identified with $r_{k_j}^j$ assert that $h_1(V_i^{j,k_j}) \in \mathcal{S}(H)$ and $h_1(U_i^{j,k_j})$ is the signature of $h_1(V_i^{j,k_j})$. Then, by Observation 5, $h_1(V_i^{j,k_j})$ is a signature of $h_1(U_i^{j,k_j})$. Denote $h_1(U_i^{j,k_j})$ by T_i^j, then $h_1(V_i^{j,k_j}) = S(T_i^j)$ and $S(S(T_i^j)) = T_i^j$ Let $y_i^j = f_{k_j}^{j'}(x_i)$ be the *candidate assignment* for $x_i \in X_{j'}$ at index j, recall that $y_i^j = \lambda^{-1}(S(T_i^j))$.

Let $i \in [n]$ be fixed and let $j_1, j_2 \in [L], j_1 < j_2$ be such that $x_i \in X_{j_1'} \cap X_{j_2'}$. Observe that in such case $T_i^{j_1} \supseteq T_i^{j_2}$. Indeed, denote $k_1 = k_{j_1}$, $k_2 = k_{j_2}$, then we have (1) $h_1(V_i^{j_1,k_1}) = S(T_i^{j_1})$ and $h_1(U_i^{j_1,k_1}) = T_i^{j_1}$, and (2) $h_1(V_i^{j_2,k_2}) = S(T_i^{j_2})$ and $h_1(U_i^{j_2,k_2}) = T_i^{j_2}$. Recall that each vertex from $U_i^{j_2,k_2}$ is adjacent to each vertex from $V_i^{j_1,k_1}$. Since h_1 is a homomorphism, the same holds for their images: each vertex from $T_i^{j_2}$ is adjacent to each vertex from $S(T_i^{j_1})$. Then $S(T_i^{j_2}) \supseteq S(T_i^{j_1})$, so $T_i^{j_1} = S(S(T_i^{j_1})) \supseteq S(S(T_i^{j_2})) \supseteq T_i^{j_2}$.

We say that the index $j_1 \in \{0,\ldots,L-1\}$ is *problematic* for i if there is $j_2 > j_1$ such that $x_i \in X_{j_1'} \cap X_{j_2'}$ and $T_i^{j_1} \neq T_i^{j_2}$. Since for each variable we have at most $|H_i|$ problematic indices, there are at most $|H_i| \cdot n$ problematic indices for all variables. Since $L = m(|H_i| \cdot n + 1)$, by pigeonhole principle we get that there exists a set $J \subseteq \{0,\ldots,L-1\}$ of m consecutive indices such that none of them is problematic for any i. For every $i \in [n]$, we fix some $j \in J$ such that $x_i \in X_{j'}$ and set $\gamma(x_i) = y_i^j$ (observe that the choice of j does not matter).

We claim that γ is an assignment that satisfies every constraint from φ. Indeed, for any $j' \in [m]$ there exists $j \in J$ such that $j' = j \mod m$. For every $i \in X_{j'}$, we have $\gamma(x_i) = y_i^j = f_{k_j}^{j'}(x_i)$, so γ satisfies the constraint $c_{j'}$. ◁

Finally, it remains to adapt the arguments of Lampis [20] to establish the desired linear clique-width bound.

▷ Claim 22. G_φ can be constructed in time polynomial in $|\varphi|$, and we have $\mathrm{cw}(G_\varphi) \leqslant n + f(\varepsilon,\nu)$ for some function f, where $\nu = |V(H)|$.

Proof of Claim. Observe that any (S,w,w')-gadget constructed as in Lemma 17 for $i = 1$ has at most $|V(H_i)|^{|S|-1} \cdot |V(H)| \leqslant \nu^{|S|}$ vertices. In particular, we can ensure that every implication gadget in G_φ has at most $\nu^{\mathcal{O}(\nu^2)}$ vertices. Moreover, we will assume that all the

or-gadgets of G_φ are constructed as in Lemma 18 and the subgadgets for S, S_{left} and S_{right} contain at most ν^7 vertices. Then for every $j \in \{0, \ldots, L-1\}$, $p_{j'}$-or-gadget (F_j, h'_j, R_j) has at most $(p_{j'} - 1) \cdot \nu^7$ vertices.

For fixed H and $\varepsilon > 0$ we have that $B = s(H_i) \leqslant 2^{|V(H_i)|} - 2$ and q is a constant that only depends on B, ε (that is, on $|V(H_i)|$, ε). Each constraint of the $q - \text{CSP} - B$ instance has at most B^q satisfying assignments. In particular, the number of vertices in each or-gadget is upper-bounded by $B^q \cdot \nu^7$. Therefore, it is not hard to see that the whole construction can be performed in polynomial time, if H is fixed and ε is a constant. For clique-width we use the following labels:

1. n *main* labels, representing the variables of φ.
2. A single *done* label. Its informal meaning is that a vertex that receives this label will not be connected to anything else not yet introduced in the graph.
3. $B^q \cdot \nu^7$ *constraint work* labels.
4. $qB^q \cdot \nu^{\mathcal{O}(\nu^2)}$ *variable-constraint incidence work* labels.

To give a clique-width expression we will describe how to build the graph, following essentially the steps given in the description of the construction by maintaining the following invariant: before starting iteration j, all vertices of the set $W_i^j = \bigcup_{j'<j} \bigcup_{k \in [p_{j'}]} V_i^{j',k}$ have label i, and all other vertices have the *done* label. This invariant is vacuously satisfied before the first iteration, since the graph is empty. Suppose that for some $j \in \{0, \ldots, L-1\}$ the invariant is true. We use the $B^q \cdot \nu^7$ constraint work labels to introduce the vertices of the $p_{j'}$-or-gadget (F_j, h'_j, R_j), giving each vertex a distinct label. We use join operations to construct the internal edges of the or-gadget. Then, for each variable x_i that appears in the current constraint we do the following: we use $B^q \cdot \nu^{\mathcal{O}(\nu^2)}$ of the variable-constraint incidence work labels to introduce for all $k \in [p_{j'}]$ the vertices of $V_i^{j,k}$ and $U_i^{j,k}$ as well as the implication gadgets connecting these to r_k^j. Again we use a distinct label for each vertex, but the number of vertices (including internal vertices of the implication gadgets) is $B^q \cdot \nu^{\mathcal{O}(\nu^2)}$, so we have sufficiently many labels to use distinct labels for each of the q variables of the constraint. We use join operations to add the edges inside all implication gadgets. Then we use join operations to connect $U_i^{j,k}$ to all vertices W_i^j. This is possible, since the invariant states that all the vertices of W_i^j have the same label i. We then rename all the vertices of $U_i^{j,k}$ for all k to the done label, and do the same also for internal vertices of all implication gadgets. We proceed to the next variable of the same constraint and handle it using its own $B^q \cdot \nu^{\mathcal{O}(\nu^2)}$ labels. Once we have handled all variables of the current constraint, we rename all vertices of each $V_i^{j,k}$ to label i for all k. We then rename all vertices of the $p_{j'}$-or-gadget (F_j, h'_j, R_j) gadget to the done label and increase j by 1. It is not hard to see that we have maintained the invariant and constructed all edges induced by the vertices introduced in steps up to j, so repeating this process constructs the graph. ◁

Together the claims imply Theorem 16 in the following way: For an arbitrary instance of q-CSP-B, our construction produces an instance of HomExt(H), and the instances are equivalent by Claim 20 and Claim 21. If one could solve HomExt(H) in $\mathcal{O}^*((s(H_i)-\varepsilon)^{\text{cw}(G)})$ for some $\varepsilon > 0$, one could use our construction to solve q-CSP-B, and by our choice of B and Claim 22 this procedure would have complexity $\mathcal{O}^*((B-\varepsilon)^{n+c})$ for some constant c. By our choice of q according to Theorem 19, this contradicts the SETH. ◀

6 Summary and Concluding Remarks

Extensions and Corollaries. We observe that Corollary 12 can be combined with Theorem 2 to obtain the following statement, which summarizes our results.

▶ **Theorem 23.** *Let H' be a fixed graph with the non-trivial connected core H. Let $H_1 \times \ldots \times H_m$ be the prime factorization of H. Let $i \in [m]$ be such that $s(H_i) = \max_{j\in[m]} s(H_j)$. Let G be an instance of Hom(H').*

1. *Assuming a clique-width expression σ of G of width $\mathrm{cw}(G)$ is given, the Hom(H') problem can be solved in time $\max_{i\in[m]} \mathcal{O}^*(s(H_i)^{\mathrm{cw}(G)})$.*
2. *Assuming that H is H_i-projective, there is no algorithm to solve Hom(H') in time $\max_{i\in[m]} \mathcal{O}^*((s(H_i)-\varepsilon)^{\mathrm{cw}(G)})$ for any $\varepsilon > 0$, unless the SETH fails.*

We note that the restriction to connected targets can be avoided by known properties of homomorphisms to disconnected graphs [25]; on the algorithmic side, one branches over all connected components of H, while for the lower bound one considers the component with maximum signature number.

It is clear that obtaining a full complexity classification with respect to clique-width may require weakening the assumption in the second statement of Theorem 23. We recall that an analogous situation occurs in the work of Okrasa and Rzążewski [25]; as mentioned in the introduction, the authors obtain the SETH-conditioned tight complexity bound for the Hom(H) problem parameterized by treewidth for all targets H, assuming two conjectures of Larose and Tardif [21,22]. The notion of H_i-projectivity allows us to restate these conjectures as one, which is not only sufficient in our setting but is also weaker in the sense of it being implied by the former two conjectures, but not necessarily equivalent to them.

▶ **Conjecture 1.** *Let H be a non-trivial core with prime factorization $H_1 \times \ldots \times H_m$ and let $i \in [m]$. Then H is H_i-projective.*

Using Conjecture 1, we can restate our main result as follows.

▶ **Theorem 24.** *Let H' be a fixed graph with the non-trivial connected core H. Let $H_1 \times \ldots \times H_m$ be the prime factorization of H. Let G be an instance of Hom(H').*

1. *Assuming the clique-width expression σ of G of width $\mathrm{cw}(G)$ is given, the Hom(H') problem can be solved in time $\max_{i\in[m]} \mathcal{O}^*(s(H_i)^{\mathrm{cw}(G)})$.*
2. *Assuming that Conjecture 1 holds, there is no algorithm to solve Hom(H') in time $\max_{i\in[m]} \mathcal{O}^*((s(H_i)-\varepsilon)^{\mathrm{cw}(G)})$ for any $\varepsilon > 0$, unless the SETH fails.*

We also observe that since each non-trivial projective core H is H-projective, in this case we already obtain a tight complexity bound.

▶ **Corollary 25.** *Let H' be a fixed graph with the non-trivial connected projective core H. Let G be an instance of Hom(H').*

1. *Assuming the clique-width expression σ of G of width $\mathrm{cw}(G)$ is given, the Hom(H') problem can be solved in time $\mathcal{O}^*(s(H)^{\mathrm{cw}(G)})$.*
2. *There is no algorithm to solve Hom(H') in time $\mathcal{O}^*((s(H)-\varepsilon)^{\mathrm{cw}(G)})$ for any $\varepsilon > 0$, unless the SETH fails.*

Generalizations and Other Research Directions. We remark that our hardness reduction is via HomExt(H), and in fact our algorithm can also easily be adapted to this setting (by removing all records that do not adhere to the partial mapping from the input graph to H) without an increase in complexity. However, since the dichotomy between P and NP-complete cases of HomExt(H) is more complicated (see [11], studied as the graph-retract problem) there exist target graphs H that are not covered by Theorem 24. On a similar note, let us also point out that setting up the SETH-conditioned tight complexity bounds for clique-width for a more general list problem LHom(H) [10,27] is widely open.

Another direction that is very closely related to our results is to determine similarly tight complexity bounds for the *rank-width* (rw) of the input graph: rank-width [17, 26] is a graph parameter that is known to be asymptotically equivalent to clique-width and is in fact used as an approximation of clique-width that can be computed in fixed-parameter tractable time. Our results together with the known relationship between clique-width and rank-width imply an upper bound of $\mathcal{O}^*(s(H)^{2^{\mathrm{rw}}+1})$ and a SETH lower bound of $(s(H)-\varepsilon)^{\mathrm{rw}}$ on the complexity of $\textsc{Hom}(H)$ for projective H parameterized by the rank-width of the input.

References

1 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Yota Otachi. Grundy distinguishes treewidth from pathwidth. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 14:1–14:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

2 Jan Böker. Graph similarity and homomorphism densities. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 32:1–32:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

3 Andrei A. Bulatov and Amineh Dadsetan. Counting homomorphisms in plain exponential time. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 21:1–21:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

4 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

5 Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.

6 Marek Cygan, Fedor V. Fomin, Alexander Golovnev, Alexander S. Kulikov, Ivan Mihajlin, Jakub Pachocki, and Arkadiusz Socała. Tight lower bounds on graph embedding problems. *J. ACM*, 64(3):18:1–18:22, 2017. doi:10.1145/3051094.

7 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

8 Willibald Dörfler. Primfaktorzerlegung und Automorphismen des Kardinalproduktes von Graphen. *Glasnik Matematički*, 9:15–27, 1974.

9 László Egri, Dániel Marx, and Paweł Rzążewski. Finding list homomorphisms from bounded-treewidth graphs to reflexive graphs: a complete complexity characterization. In *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, pages 27:1–27:15, 2018.

10 Tomás Feder, Pavol Hell, and Jing Huang. Bi-arc graphs and the complexity of list homomorphisms. *J. Graph Theory*, 42(1):61–80, 2003. doi:10.1002/jgt.10073.

11 Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766.

12 Fedor V. Fomin, Pinar Heggernes, and Dieter Kratsch. Exact algorithms for graph homomorphisms. *Theory Comput. Syst.*, 41(2):381–393, 2007. doi:10.1007/s00224-007-2007-x.

13 Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1):1:1–1:24, 2007.

14 Richard H. Hammack, Wilfried Imrich, and Sandi Klavžar. *Handbook of product graphs*. CRC press, 2011.

15 Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *J. Comb. Theory, Ser. B*, 48(1):92–110, 1990. doi:10.1016/0095-8956(90)90132-J.

16 Pavol Hell and Jaroslav Nešetřil. The core of a graph. *Discret. Math.*, 109(1-3):117–126, 1992.

17 Sang il Oum and Paul Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006. doi:10.1016/j.jctb.2005.10.006.

18 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

19 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

20 Michael Lampis. Finer tight bounds for coloring on clique-width. *SIAM J. Discret. Math.*, 34(3):1538–1558, 2020. doi:10.1137/19M1280326.

21 Benoît Larose. Families of strongly projective graphs. *Discuss. Math. Graph Theory*, 22(2):271–292, 2002. doi:10.7151/dmgt.1175.

22 Benoît Larose and Claude Tardif. Strongly rigid graphs and projectivity. *Multiple-Valued Logic*, 7:339–361, 2001.

23 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the exponential time hypothesis. *Bull. EATCS*, 105:41–72, 2011.

24 Karolina Okrasa, Marta Piecyk, and Paweł Rzążewski. Full complexity classification of the list homomorphism problem for bounded-treewidth graphs. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 74:1–74:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.74.

25 Karolina Okrasa and Paweł Rzążewski. Fine-grained complexity of the graph homomorphism problem for bounded-treewidth graphs. *SIAM J. Comput.*, 50(2):487–508, 2021.

26 Sang-il Oum. Approximating rank-width and clique-width quickly. In Dieter Kratsch, editor, *Graph-Theoretic Concepts in Computer Science*, pages 49–58, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

27 Marta Piecyk and Paweł Rzążewski. Fine-grained complexity of the list homomorphism problem: Feedback vertex set and cutwidth. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 56:1–56:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

28 Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

29 Paweł Rzążewski. Exact algorithm for graph homomorphism and locally injective graph homomorphism. *Inf. Process. Lett.*, 114(7):387–391, 2014.

30 Magnus Wahlström. New plain-exponential time classes for graph homomorphism. *Theory Comput. Syst.*, 49(2):273–282, 2011. doi:10.1007/s00224-010-9261-z.

31 Tomasz Łuczak and Jaroslav Nešetřil. Note on projective graphs. *J. Graph Theory*, 47(2):81–86, 2004.