New Limits to Classical and Quantum Instance Compression

Andrew Drucker*

Abstract

Given an instance of a hard decision problem, a limited goal is to *compress* that instance into a smaller, equivalent instance of a second problem. As one example, consider the problem where, given Boolean formulas $\psi^1, \ldots, \psi^t$, we must determine if at least one ψ^j is satisfiable. An OR-*compression scheme* for SAT is a polynomial-time reduction R that maps $(\psi^1, \ldots, \psi^t)$ to a string z, such that z lies in some "target" language L' if and only if $\bigvee_j [\psi^j \in \text{SAT}]$ holds. (Here, L' can be arbitrarily complex.) AND-compression schemes are defined similarly. A compression scheme is *strong* if $|z|$ is polynomially bounded in $n = \max_j |\psi^j|$, independent of t.

Strong compression for SAT seems unlikely. Work of Harnik and Naor (FOCS '06/SICOMP '10) and Bodlaender, Downey, Fellows, and Hermelin (ICALP '08/JCSS '09) showed that the infeasibility of strong OR-compression for SAT would show limits to instance compression for a large number of natural problems. Bodlaender et al. also showed that the infeasibility of strong AND-compression for SAT would have consequences for a different list of problems. Motivated by this, Fortnow and Santhanam (STOC '08/JCSS '11) showed that if SAT is strongly OR-compressible, then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Finding similar evidence against AND-compression was left as an open question.

We provide such evidence: we show that strong AND- *or* OR-compression for SAT would imply *non-uniform, statistical zero-knowledge proofs* for SAT—an even stronger and more unlikely consequence than $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. (By a different argument, we also show such compression would imply the *uniform* collapse $\mathsf{NP} \subseteq \mathsf{coAM}$.) Our methods apply against *probabilistic* compression schemes of sufficient "quality" with respect to the reliability and compression amount (allowing for tradeoff). This greatly strengthens the evidence given by Fortnow and Santhanam against probabilistic OR-compression for SAT. We also give negative results for the analogous task of *quantum instance compression*, in which a polynomial-time quantum reduction must output a quantum state that, in an appropriate sense, "preserves the answer" to the input instance.

The central idea in our proofs is to exploit the information bottleneck in an AND-compression scheme for a language L in order to fool a cheating prover in a proof system for $\overline{L}$. Our key technical tool is a new method to "disguise" information being fed into a compressive mapping; we believe this method may find other applications.

*andy.drucker@gmail.com. This paper appeared in FOCS 2012 and formed part of the author's Ph.D. dissertation (EECS Dept., MIT). Work conducted at MIT was supported an NSF Career grant of Scott Aaronson. Author's current affiliation: School of Mathematics, IAS, Princeton, NJ. Supported by the NSF under agreements Princeton University Prime Award No. CCF-0832797 and Sub-contract No. 00001583. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Contents

1 Introduction

1.1 Instance compression and parametrized problems

Given an instance of a hard decision problem, we may hope to *compress* that instance into a smaller, equivalent instance, either of the same or of a different decision problem. Here we do not ask to be able to recover the original instance from the smaller instance; we only require that the new instance have the same (yes/no) *answer* as the original. Such *instance compression* may be the first step towards obtaining a solution; this has been a central technique in the theory of *fixed-parameter-tractable* algorithms [DF99, GN07]. Strong compression schemes for certain problems would also have important implications for cryptography [HN10]. Finally, compressing an instance of a difficult problem may also be a worthwhile goal in its own right, since it can make the instance easier to store and communicate [HN10].

It is unknown whether one can efficiently, significantly compress an arbitrary instance of a natural NP-complete language like SAT, the set of satisfiable Boolean formulas.[1] A more limited

[1] If we could efficiently reduce instances of some NP-complete problem to shorter instances of the *same* problem, then we could iterate the reduction to solve our problem in polynomial time, implying P = NP. However, even if

goal is to design an efficient reduction that achieves compression on instances that are particularly "simple" in some respect. To explore this idea, one needs a formal model defining "simple" instances; the versatile framework of *parametrized problems* [DF99] is one such model, and has been extensively used to study instance compression. A parametrized problem is a decision problem in which every instance has an associated *parameter value* k, giving some measure of the complexity of a problem instance.[2] As an example, one can parametrize a Boolean formula ψ by the number of distinct variables appearing in ψ.

An ambitious goal for a parametrized problem P is to compress an arbitrary instance x of the decision problem for P into an equivalent instance x' of a second, "target" decision problem, where the output length $|x'|$ is bounded by a *polynomial* in $k = k(x)$. If P has such a reduction running in time $\text{poly}(|x|+k)$, we say P is *strongly compressible*; we say P is *strongly self-compressible* if the target problem of the reduction is P itself. (In the literature of parametrized problems, a strong self-compression reduction is usually referred to as a *polynomial kernelization*. More generally, a *kernelization* is a polynomial-time self-compression reduction whose output size is bounded by *some* function of the parameter k alone.)

1.2 Previous work: results and motivation

Let VAR-SAT denote the Satisfiability problem for Boolean formulas, parametrized by the number of distinct variables in the formula. In their study of instance compression for NP-hard problems, Harnik and Naor [HN10] asked whether VAR-SAT is strongly compressible.[3] They showed that a positive answer would have several significant consequences for cryptography. Notably, they proved that a *deterministic* strong compression reduction for VAR-SAT (with any target problem) would yield a construction of collision-resistant hash functions based on any one-way function—a long-sought goal.

In fact, Harnik and Naor showed that for their applications, it would suffice to achieve strong compression for a simpler parametrized problem, the "OR(SAT) *problem:*" this is the Satisfiability problem for Boolean formulas expressed as disjunctions $\psi = \bigvee_{j=1}^{t} \psi_j$, where the parameter is now defined as the maximum bit-length of any sub-formula ψ_j. Strong compression for VAR-SAT easily implies strong compression for OR(SAT). Harnik and Naor defined a hierarchy of decision problems called the "VC hierarchy," which can be modeled as a class of parametrized problems (see [FS11]). They showed that a strong compression reduction for any of the problems "above" OR(SAT) in this hierarchy would also imply strong compression for OR(SAT); this includes parametrized versions of natural problems like the Clique and Dominating Set problems. While Harnik and Naor's primary motivation was to *find* a strong compression scheme for OR(SAT) to use in their cryptographic applications, their work also provides a basis for showing *negative* results: in view of the reductions in [HN10], any evidence against strong compression for OR(SAT) is also evidence against strong compression for a variety of other parametrized problems.

In subsequent, independent work, Bodlaender, Downey, Fellows, and Hermelin [BDFH09] also studied the compressibility of OR(SAT) and of related problems; these authors' motivations came from the theory of *fixed-parameter tractable (FPT)* algorithms [DF99]. An FPT algorithm for a parametrized problem P is an algorithm that solves an arbitrary instance x, with parameter

---

$\mathsf{P} \neq \mathsf{NP}$, it is still conceivable that SAT might have an efficient compressive reduction to a different target problem—to the Halting problem, say.

[2] See Section 5.1 for details. The parameter k is explicitly given as part of the input to the algorithm.

[3] Strictly speaking, they asked a slightly different question whose equivalence to this one was pointed out in [FS11].

$k = k(x)$, in time $g(k) \cdot \text{poly}(|x| + k)$, for some function $g(\cdot)$. The idea is that even if P is hard in general, an FPT algorithm for P may be practical on instances where the parameter k is small. Now as long as P is decidable, a kernelization reduction for P provides the basis for an FPT algorithm for P: on input x, first compress x, then solve the equivalent, compressed instance. The kernelization approach is one of the most widely-used schemas for developing FPT algorithms.[4] Of course, one hopes to compress by as large an amount as possible, to maximize the efficiency of the resulting FPT algorithm; this motivates the search for *strong* self-compression reductions.

Strong self-compression reductions are known for parametrized versions of many natural NP-complete problems, such as the Vertex Cover problem; see, e.g., the survey [GN07]. However, for many other such parametrized problems, including numerous problems known to admit FPT algorithms (such as OR(SAT)), no strong compression reduction is known, to any target problem. Bodlaender et al. [BDFH09] conjectured that no strong self-compression reduction exists for OR(SAT). They made a similar conjecture for the closely-related "AND(SAT) *problem,*" in which one is given Boolean formulas $\psi_1, \ldots, \psi_t$ and asked to decide whether $\bigwedge_{j=1}^{t}[\psi_j \in \text{SAT}]$ holds—that is, whether every ψ_j is individually satisfiable. As with OR(SAT), we parametrize AND(SAT) by the maximum bit-length of any ψ_j.

Bodlaender et al. showed that these conjectures (sometimes referred to as the "OR-" and "AND-conjectures") would have considerable explanatory power. First, they showed [BDFH09, Theorem 1] that the nonexistence of strong self-compression reductions for OR(SAT) would rule out strong self-compression for a large number of other natural parametrized problems; these belong to a class we call "OR-expressive problems."[5] Under the assumption that AND(SAT) does not have strong self-compression, Bodlaender et al. ruled out strong self-compression reductions for a second substantial list of problems [BDFH09, Theorem 2], belonging to a class we will call "AND-expressive." Despite the apparent similarity of OR(SAT) and AND(SAT), no equivalence between the compression tasks for these two problems is known.

In light of their results, Bodlaender et al. asked for complexity-theoretic evidence against strong self-compression for OR(SAT) and AND(SAT). Fortnow and Santhanam [FS11] provided the first such evidence: they showed that if OR(SAT) has a strong compression reduction (to any target problem), then $\text{NP} \subseteq \text{coNP/poly}$ and the Polynomial Hierarchy collapses to its third level.

The techniques of [BDFH09, FS11] were refined and extended by many researchers to give further evidence against efficient compression for parametrized problems, e.g., in [DLS09, DvM10, BTY11, BJK11a, BJK11b, BJK11c, CFM11, HW12, DM12, Kra12]. (See [DM12] for further discussion and references.) As one notable development that is relevant to our work, Dell and Van Melkebeek [DvM10] combined the techniques of [BDFH09, FS11] with new ideas to provide tight compression-size lower bounds for certain problems that *do* admit polynomial kernelizations. Researchers also used ideas from [BDFH09, FS11] in other areas of complexity, giving new evidence of lower bounds for the length of PCPs [FS11, DvM10] and for the density of NP-hard sets [BH08].

Finding evidence against strong compression for AND(SAT) was left as an open question by these works, however. The limits of *probabilistic* compression schemes for OR(SAT) and for OR-expressive problems (including VAR-SAT) also remained unclear. The results and techniques of [FS11] give evidence only against some restrictive sub-classes of probabilistic compression schemes

[4] In fact, *every* problem with an FPT algorithm is kernelizable [CCDF97]. This does not mean, however, that the most efficient FPT algorithms always arise from the kernelization approach.

[5] See Section 5.2. The class of OR-expressive problems is not identical to the class described in [BDFH09], but it is closely related and contains their class, as well as other classes of problems identified in [HN10, BJK11a].

for OR(SAT): schemes with one-sided error, avoiding false negatives; schemes whose error probability is exponentially small in the length of the entire input; and schemes using $O(\log n)$ random bits, where $n = \max_j |\psi_j|$.

1.3 Our results

1.3.1 Results on classical compression

We complement the results of [FS11] by providing evidence against strong compression for AND(SAT): we prove that such a compression scheme, to any target problem, would also imply $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. In fact, we show that reductions compressing even by a much more modest amount would imply the same conclusion. For concreteness, we state our most "basic" result on compression of AND(SAT) in a self-contained way below.

Theorem 1.1. *Let L be any NP-complete language. Suppose there is a deterministic polynomial-time reduction R that takes an arbitrarily long list of input strings $(x^1, \ldots, x^t)$ and outputs a string z, with*

$$z \in L \iff \bigwedge_{j \in [t]} [x^j \in L] \ .$$

Suppose further that R obeys the output-size bound $|z| \leq \left(\max_{j \leq t} |x^j|\right)^{O(1)}$, with the polynomial bound independent of t. Then, $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

More strongly, we show the following. Suppose there is any second, "target" language L', a pair of polynomially-bounded functions $t(n), t'(n) : \mathbb{N} \to \mathbb{N}$ with $t(n) = \omega(1)$ and $t'(n) + 1 \leq t(n)/2$, and a deterministic polynomial-time reduction $R : \{0,1\}^{t(n) \times n} \to \{0,1\}^{\leq t'(n)}$, such that

$$R(x^1, \ldots, x^{t(n)}) \in L' \iff \bigwedge_{j \in [t(n)]} [x^j \in L] \ .$$

Then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

We prove Theorem 1.1 in Section 3. In later sections, we will strengthen and generalize Theorem 1.1 using related but more powerful proof techniques. However, we feel it is worthwhile to present a proof of this basic result with a minimum of tools and preliminaries.[6]

The techniques we use to generalize Theorem 1.1 will extend naturally (and in a strong fashion) to the *probabilistic* setting with two-sided error, in which we expect the compression reduction to obey some success-probability guarantee on every input. We show (in Theorem 7.4, item 1) that any sufficiently "high-quality" compression scheme for AND(SAT) would imply $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Here, "quality" is defined by a certain relationship between the reliability and the compression amount of the reduction, and allows for tradeoff.

We also show (in Theorem 7.4, item 2, and Theorem 7.5) that beyond a second, somewhat more demanding quality threshold, probabilistic compression reductions either for AND(SAT) *or* for OR(SAT) would imply the existence of *non-uniform, statistical zero-knowledge proofs* for NP

[6] To be precise, in our elementary proof we avoid any overt use of information-theoretic results and concepts; we also avoid the use of the minimax theorem. These tools are central to our stronger and more general approach (which, in particular, is much better suited for analyzing bounded-error reductions), but familiarity with these tools is not necessary to understand Theorem 1.1. We mention that the decision to use or avoid information theory in the proof is essentially independent of the choice to use or avoid the minimax theorem.

languages—a stronger (and even more unlikely) consequence than $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. The more-demanding quality threshold in this second set of results is still rather modest, and allows us to prove the following result as a special case:

Theorem 1.2 (Informal)**.** *Suppose that either of* AND(SAT) *or* OR(SAT) *is strongly compressible, with success probability* $\geq .5 + 1/\operatorname{poly}(n)$ *for an* AND *or* OR *of length-*n *formulas. Then there are non-uniform, statistical zero-knowledge proofs for all languages in* NP*.*

At the other extreme, where we consider compression schemes with more modest compression amounts, but with greater reliability, our techniques yield the following result:

Theorem 1.3 (Informal)**.** *Let* $t(n) : \mathbb{N}^+ \to \mathbb{N}^+$ *be any polynomially bounded function. Suppose there is a compression scheme compressing an* AND *of* $t(n)$ *length-*n *SAT instances into an instance* z *of a second decision problem* L'*, where* $|z| \leq C \cdot t(n) \log t(n)$ *for some* $C > 0$*. If the scheme's error probability on such inputs is bounded by a sufficiently small inverse-polynomial in* n *(depending on* $t(n)$ *and* C*), then there are non-uniform, statistical zero-knowledge proofs for all languages in* NP*. The corresponding result also holds for* OR*-compression.*[7]

Beyond a third and significantly more-demanding threshold of quality, we show in Section 9 that probabilistic compression reductions either for AND(SAT) *or* for OR(SAT) would imply the *uniform* complexity-class collapse $\mathsf{NP} \subseteq \mathsf{coAM}$.[8] The proof of this result uses some of the same information-theoretic techniques used to prove Theorems 1.2 and 1.3, but is substantially different and draws further inspiration from a work of Sivakumar on the so-called "membership comparability" of NP-complete languages [Siv99].

Our results give the first strong evidence of hardness for compression of AND(SAT). They also greatly strengthen the evidence given by Fortnow and Santhanam against *probabilistic* compression for OR(SAT), and provide the first strong evidence against probabilistic compression for the potentially-harder problem VAR-SAT. For *deterministic* (or error-free) compression of OR(SAT), the limits established by our techniques also follow from the techniques of [FS11], which apply given an OR-compression scheme with compression bound of form $|z| \leq O(t(n) \log t(n))$.[9] On the other hand, we provide somewhat stronger complexity-theoretic evidence for these limits to compression.

Using our results on the infeasibility of compression for AND(SAT) and OR(SAT), and building on [HN10, BDFH09, FS11], we give new complexity-theoretic evidence against strong compressibility for a list of interesting parametrized problems with FPT algorithms. (See Theorem 7.7.) This is the first strong evidence against strong compressibility for any of the ten "AND-expressive" problems identified in [BDFH09] (and listed in Section 5.2). For the numerous "OR-expressive" problems identified in [HN10, BDFH09] and other works, this strengthens the negative evidence given by [FS11].

Our methods also extend the known results on limits to compression for parametrized problems that do possess polynomial kernelizations: we can partially extend the results of Dell and Van Melkebeek [DvM10] to the case of probabilistic algorithms with two-sided error. For example, for

[7] In fact, *error-free* OR-compression of this sort for SAT would give non-uniform *perfect* zero-knowledge proofs for NP, and error-free AND-compression for SAT would give non-uniform perfect zero-knowledge proofs for coNP; see Theorem 7.3.

[8] Such a collapse is not known to imply or be implied by the existence of non-uniform statistical zero-knowledge proofs for NP.

[9] This is not explicitly shown in [FS11], but follows from the technique of [FS11, Theorem 3.1]; see also [DvM10, Lemma 3] for a more general result that makes the achievable bounds clear.

$d > 1$ and any $\varepsilon > 0$, Dell and Van Melkebeek proved that if the Satisfiability problem for N-variable d-CNFs has a polynomial-time compression reduction with output-size bound $O(N^{d-\varepsilon})$, then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Their result applies to co-nondeterministic reductions, and to probabilistic reductions without false negatives; we prove (in Theorem 7.11) that the result also holds for probabilistic reductions with two-sided error, as long as the success probability of the reduction is at least $.5+N^{-\beta}$ for some $\beta = \beta(d, \varepsilon) > 0$. Using reductions described in [DvM10], we also obtain quantitatively-sharp limits to probabilistic compression for several other natural NP-complete problems, including the Vertex Cover and Clique problems on graphs and hypergraphs. (However, the limits we establish do not give lower bounds on the cost of *oracle communication protocols*; these protocols are a generalization of compression reductions, studied in [DvM10], to which that work's results do apply. Trying to extend our results to this model seems like an interesting challenge for further study.)

Our results about AND(SAT) and OR(SAT) follow from more general results about arbitrary languages. For any language L, we follow previous authors and consider the "OR(L) *problem,*" in which one is given a collection $x^1, \ldots, x^t$ of strings, and is asked to determine whether at least one of them is a member of L. We show (in Theorem 7.1, item 1) that if a sufficiently "high-quality" probabilistic polynomial-time compression reduction exists for the OR(L) problem, then $L \in \mathsf{NP/poly}$. (As before, "high-quality" is defined by a relation between the reliability of the reduction and the compression amount.) We also show (in Theorem 7.1, item 2) that a polynomial-time compression scheme for OR(L) meeting a more demanding standard of quality implies that L possesses non-uniform statistical zero-knowledge proof systems, and lies in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. (For deterministic compression, the conclusion $L \in \mathsf{coNP/poly}$ was established earlier in [FS11].) Applying these results to $L := \overline{\mathrm{SAT}}$ gives our hardness-of-compression results for AND(SAT); applying the second set of results to $L := \mathrm{SAT}$ gives our improved negative results for OR(SAT).

In unpublished work, Buhrman [Buh] constructed an oracle A such that, for every NP^A-complete language L, the decision problem AND(L) does not have a P^A-computable strong compression reduction. This gave earlier, indirect evidence against efficient strong compression for the AND(SAT) problem—or at least, it indicated that exhibiting such a compression reduction would require novel techniques. Now, inspection of the proofs reveals that our new results on compression for OR(L) are all perfectly relativizing. This allows us to identify many more oracles obeying the property of Buhrman's oracle: namely, we may take any A for which $\mathsf{NP}^A \not\subseteq \mathsf{coNP}^A\mathsf{/poly}$. For example, this holds with probability 1 for a *random oracle* [BG81].[10] Such an oracle can also be obtained through a simple diagonalization argument.

For any Boolean function $f : \{0,1\}^* \to \{0,1\}$, we may generalize the OR(L) decision problem to the problem $f \circ L$, in which one is given a collection of strings $x^1, \ldots, x^t$ and must output $f(L(x^1), \ldots, L(x^t))$. We restrict attention to the most interesting case in which f depends on all variables. So far it would seem that our negative results for instance compression are fairly specific to the case where the outer "combining function" f is either AND or OR. (These are also the only cases known to be directly applicable to the study of natural kernelization tasks.) However, by an idea suggested in [FS11, Section 7], our negative result on compression for AND(SAT), combined with those authors' negative results on compression for OR(SAT), actually allows us to rule out strong compression schemes from $f \circ \mathrm{SAT}$ to a target language $L' \in \mathsf{NP}$, for a quite broad range of functions f: we do so for all non-monotone f, and for all monotone $f = \{f_m : \{0,1\}^m \to \{0,1\}\}$

[10] In [BG81] it is shown that $\mathsf{NP}^A \not\subseteq \mathsf{coNP}^A$ for random A; the technique readily extends to give the stronger claim above.

with reasonably high *block sensitivity* (as defined by Nisan [Nis91]), namely with $bs(f_m) \geq m^{\Omega(1)}$. The quantitative bounds we obtain on the achievable compression amount are somewhat weaker in this result than we are able to show for AND or OR, however. See Theorem 7.6, and note the new requirement that L' be in NP.

For certain (somewhat exceptional) monotone combining functions, we have $bs(f_m) \leq m^{o(1)}$,[11] and the approach of [FS11] does not yield strong results. In Section 9, we address this issue, proving that for functions f with $bs(f_m) \leq m^{o(1)}$ which have a collection of sensitive blocks with certain "nice" properties, strong compression schemes for $f \circ \mathrm{SAT}$ (to *any* target language L') would imply the collapse $\mathsf{NP} \subseteq \mathsf{coAM}$. Our results do not apply to all f, but cover the "natural" examples of which we are aware. As a by-product of this we derive the aforementioned result, that strong compression schemes for either OR(SAT) or AND(SAT) would imply $\mathsf{NP} \subseteq \mathsf{coAM}$. (See Theorem 9.16; this connection is not immediate, since OR and AND have maximal block sensitivity. We believe it *might* be possible to strengthen the implication to yield $\mathsf{NP} \subseteq \mathsf{SZK}$, by making a close analysis of our techniques and using the strong closure properties of SZK proved in [SV03, Sec. 4.2]; we have not proved this, however.) Further discussion of our techniques for these results, and their relation to the techniques of [Siv99], can be found in Section 9.

1.3.2 Results on quantum compression

Up to this point, we have discussed compression reductions in which the input and output are both "classical" bit-strings. However, from the perspective of quantum computing and quantum information [NC00], it is natural to ask about the power of compression reductions that output a *quantum state*. An "n-qubit state" is a quantum superposition over classical n-bit strings; a vast body of research has explored the extent to which information can be succinctly encoded within and retrieved from such quantum states. If quantum computers become a practical reality, quantum instance compression schemes could help to store and transmit hard computational problems; compressing an instance might also be a first step towards its solution by a quantum algorithm.

We propose the following quantum generalization of classical instance compression: a *quantum compression reduction* for a language L is a quantum algorithm that, on input x, outputs a quantum state ρ on some number q of qubits—hopefully with $q \ll |x|$, to achieve significant compression. Our correctness requirement is that there should exist *some* fixed quantum measurement $\mathcal{M}_q$ on q-qubit states for each $q > 0$, such that $\mathcal{M}_q(\rho) = L(x)$ holds with high probability over the inherent randomness in the measurement $\mathcal{M}_q(\rho)$.[12] We do not require that $\mathcal{M}_q$ be an efficiently-performable measurement; this is by analogy to the general version of the classical compression task, in which the target language of the reduction may be arbitrarily complex.

Our results for quantum compression are closely analogous to our results in the classical case. First, we show that for any language L, if a sufficiently "high-quality" quantum polynomial-time compression reduction exists for the OR(L) problem, then L possesses a *non-uniform, 2-message quantum interactive proof system* (with a single prover). Second, we show that a sufficiently higher-quality quantum polynomial-time compression reduction for OR(L) implies that L possesses a *non-uniform quantum statistical zero-knowledge proof system*. Remarkably, the two "quality thresholds" in our quantum results are essentially *the same* as in the corresponding results for the classical

[11] In an earlier version of this paper, such monotone functions were erroneously asserted not to exist, but they do; see Section 9.1.

[12] The precise definition we will use is based on the framework of parametrized problems and is slightly more complex; this is the basic idea, however.

case.[13] It follows that, unless there exist surprisingly powerful quantum proofs of unsatisfiability for Boolean formulas, the limits we establish for probabilistic compression of AND(SAT) and OR(SAT) hold just as strongly for quantum compression.[14]

1.4 Our techniques

In this section we will focus on describing the techniques used to prove Theorems 1.2 and 1.3. As mentioned earlier, we also present a similar, but more "elementary" approach to prove Theorem 1.1. We will give some self-contained intuition about that approach in Section 3. That strategy bears some similarities to work of Fortnow and Santhanam [FS11] on the hardness of compression for OR(SAT). In particular, it shares an *incremental* approach to defining non-uniform advice for a proof system; in each case, the stage-based construction makes progress in correctly classifying more and more strings of a given input length.

1.4.1 The overall approach

We first describe our techniques for the classical case; these form the basis for the quantum case as well. Our first two general results, giving complexity upper bounds on any language L for which $\mathrm{OR}(L)$ has a sufficiently high-quality compression reduction (Theorem 7.1, items 1 and 2), are both based on a single reduction that we describe next. This reduction applies to compression reductions mapping some number $t(n) \leq \mathrm{poly}(n)$ of inputs of length n to an output string z of length $|z| = O(t(n) \log t(n))$.

Fix any language L such that $\mathrm{OR}(L)$ has a possibly-probabilistic compression reduction

$$R(x^1, \ldots, x^t) : \ \{0,1\}^{t \times n} \ \longrightarrow \ \{0,1\}^{\leq t'} \ ,$$

with some target language L', along with parameters t', t satisfying $t' \leq O(t \log t) \leq \mathrm{poly}(n)$.[15] We will use R to derive upper bounds on the complexity of L. (The reader may keep in mind the main intended setting $L = \overline{\mathrm{SAT}}$, which we will use to derive our hardness results for the compression of AND(SAT). No special properties of this language will be used in the argument, however.)

A simple, motivating observation is that if we take a string $y \in L$ and "insert" it into a tuple $\overline{x} = (x^1, \ldots, x^t)$ of elements of $\overline{L}$, replacing some x^j to yield a modified tuple $\overline{x}'$, then the values

$$R(\overline{x}) \ , \ R(\overline{x}')$$

are *different* with high probability—for, by the "OR-respecting" property of R, we will with high probability have $R(\overline{x}) \in \overline{L'}$, $R(\overline{x}') \in L'$. More generally, for any *distribution* $\mathcal{D}$ over t-tuples of inputs from $\overline{L}$, let $\mathcal{D}[y, j]$ denote the distribution obtained by sampling $\overline{x} \sim \mathcal{D}$ and replacing x^j with y; then the two output distributions

$$R(\mathcal{D}) \ , \ R(\mathcal{D}[y, j])$$

[13] We do place a minor additional restriction on quantum compression reductions for $\mathrm{OR}(L)$: we require that the reduction, on input $(x^1, \ldots, x^t)$, outputs a quantum state of size determined by $(\max_j |x^j|)$ and t.

[14] We remark that *3-message* quantum interactive proofs are known to be fully as powerful as quantum interactive proofs in which polynomially many messages are exchanged [Wat03], and that these proof systems are equal in power to PSPACE in the uniform setting [JJUW11]. However, 2-message quantum proof systems seem much weaker, and are not known to contain coNP.

[15] Here we pay exclusive attention to R's behavior on tuples of strings of some equal length n.

are *far apart* in statistical distance. (Of course, the strength of the statistical-distance lower bound we get will depend on the reliability of our compression scheme.)

We want this property to serve as the basis for an interactive proof system by which a computationally powerful Prover can convince a skeptical polynomial-time (but non-uniform) Verifier that a string y lies in L. The idea for our initial, randomized protocol (which we will later derandomize) is that Prover will make his case by demonstrating his ability to *distinguish* between the two R-output distributions described above, when Verifier privately chooses one of the two distributions, samples from it, and sends the sample to Prover.[16] But then to make our proof system meaningful, Verifier also needs to *fool* a cheating Prover in the case $y \notin L$. To do this, we want to choose $\mathcal{D}, j$ in such a way that the distributions $R(\mathcal{D}), R(\mathcal{D}[y,j])$ are as close as possible whenever $y \notin L$.

We may not be able to achieve this for an index j that is poorly-chosen. For instance, $R(\overline{x})$ may always copy the first component x^1 as part of the output string z, so taking $j = 1$ would fail badly. To get around this, we choose our replacement index j *uniformly at random*, aiming in this way to make R "insensitive" to the insertion of y.[17] As R is a compression scheme, it doesn't have room in its output string to replicate its entire input, so there is reason for hope.

This invites us to search for a distribution $\mathcal{D}^*$ over $\left(\overline{L}_n\right)^t$ with the following properties:

(i) For every $y \in \overline{L}_n$, if we select $j \in [t]$ uniformly then the expected statistical distance $\mathbb{E}_j\left[||R(\mathcal{D}^*) - R(\mathcal{D}^*[y,j])||_{\text{stat}}\right]$ is "not too large;"[18]

(ii) $\mathcal{D}^*$ is efficiently sampleable, given non-uniform advice of length $\text{poly}(n)$.

Condition (i) is quite demanding: we need a single distribution $\mathcal{D}^*$ rendering R insensitive to the insertion of *any* string $y \in \overline{L}_n$—a set which may be of exponential size. Condition (ii) is also a strong restriction: $\overline{L}_n$ may be a complicated set, and in general we can only hope to sample from distributions over $\left(\overline{L}_n\right)^t$ in which t-tuples are formed out of a fixed "stockpile" of $\text{poly}(n)$ elements of $\overline{L}_n$, hard-coded into the non-uniform advice.

Remarkably, it turns out that such a distribution $\mathcal{D}^*$ can always be found. In fact, in item (i), we can force the two distributions to be non-neglibly close (with expected statistical distance $\leq 1 - \frac{1}{\text{poly}(n)}$) whenever the output-size bound t' obeyed by R is $O(t \log t)$; the distributions will be much closer when $t' \ll t$. We call our key technical result (Lemma 6.6), guaranteeing the existence of such a $\mathcal{D}^*$, the *"Disguising-Distribution Lemma."*

Assuming this lemma for the moment, we use $\mathcal{D}^*$ as above to reduce any membership claim for L to a distinguishing task for a Prover-Verifier protocol. Given any input y, we've constructed two distributions $\mathfrak{R} = R(\mathcal{D}^*)$ and $\mathfrak{R}' = R(\mathcal{D}^*[y, \mathbf{j}])$ (with $\mathbf{j}$ uniform), where each distribution is sampleable in non-uniform polynomial time. Our analysis guarantees some lower bound $D = D(n)$ on $||\mathfrak{R} - \mathfrak{R}'||_{\text{stat}}$ in the case $y \in L$, and some upper bound $d = d(N)$ on this distance when $y \notin L$. (These parameters depend on the reliability and compression guarantees of R.) If $D(n) - d(n) \geq \frac{1}{\text{poly}(n)}$, we can give non-uniform distinguishing protocols for L, which can converted to public-coin

[16] Interactive proofs based on distinguishing tasks have seen many uses in theoretical computer science, and indeed we will rely upon known protocols of this kind in our work; see Section 4.4.

[17] We emphasize that the "insensitivity" we are looking for is *statistical*; we are not asking that y have small effect on the output of R for most *particular* outcomes to $\overline{x} \sim \mathcal{D}$. This latter goal may not be achievable, e.g., if R outputs the sum of all its input strings x^i taken as vectors over $\mathbb{F}_2^n$.

[18] For our purposes, it actually suffices to bound $||R(\mathcal{D}^*) - R(\mathcal{D}^*[y, \mathbf{j}])||_{\text{stat}}$, where $\mathbf{j}$ is a uniform value sampled "internally" as part of the distribution. In our streamlined proof of Theorem 1.1, we will use this idea. However, our techniques will yield the stronger property in condition (i) above, and this is the course we will follow in proving our general results.

protocols and then non-uniformly derandomized to show that $L \in \mathsf{NP/poly}$. Also, if $D(n)^2 - d(n) \geq \frac{1}{\mathrm{poly}(n)}$ then, using a powerful result due to Sahai and Vadhan [SV03], we can derive a non-uniform, statistical zero-knowledge proof system for L. This also implies $L \in \mathsf{NP/poly} \cap \mathsf{coNP/poly}$.

1.4.2 The Disguising-Distribution Lemma

The Disguising-Distribution Lemma, informally described in Section 1.4.1, is a statement about the behavior of $R(x^1, \ldots, x^t)$ on a specified product subset S^t of inputs ($S = \overline{L}_n$ in our application). This lemma is a "generic" result about the behavior of compressive mappings; it uses no properties of R other than R's output-size bound.[19] In view of its generality and interest, we are hopeful that the lemma will find other applications.

Our proof of this lemma uses two central ideas. First, we interpret the search for the "disguising distribution" $\mathcal{D}^*$ as a two-player game between a "disguising player" (choosing $\mathcal{D}^*$) and an opponent who chooses y; we can then apply simple yet powerful principles of game theory. Second, to build a winning strategy for the disguising player, we will exploit an information bottleneck in R stemming from its compressive property.[20]

To describe the proof, it is helpful to first understand how one may obtain the distribution $\mathcal{D}^*$ if we drop the efficient-sampleability requirement on $\mathcal{D}^*$, and focus on the "disguising" requirement (condition (i)). To build $\mathcal{D}^*$ in this relaxed setting, we will appeal to the *minimax theorem* for two-player, zero-sum games; applied here, it tells us that to guarantee the existence of a $\mathcal{D}^*$ that succeeds in disguising all strings $y \in S$, it is enough to show how to build a $\mathcal{D}^*_Y$ that succeeds in expectation, when y is sampled from some fixed (but arbitrary) distribution Y over S.

Here, a natural idea springs to mind: let $\mathcal{D}^*_Y$ just be a product distribution over t copies of Y! In this case, inserting $y \sim Y$ into $\mathcal{D}^*_Y$ at a random location is equivalent to conditioning on the outcome of a randomly-chosen coordinate of a sample from $\mathcal{D}^*_Y$. The intuition here is that, due to the output-size bound on R, the distribution $R(\mathcal{D}^*_Y)$ shouldn't have enough "degrees of freedom" to be affected much by this conditioning.

We show (in Lemma 6.2) that for any product distribution $\overline{x} \sim (\mathcal{D}_1, \ldots \mathcal{D}_t)$ over t-tuple inputs to R, conditioning on the value of $x^j \sim \mathcal{D}_j$ for a uniformly-chosen index $j \in [t]$ has bounded expected effect on the output distribution $R(\overline{x})$. That is, the *expected statistical distance* between the pre- and post-conditioned distributions is bounded non-negligibly away from 1 (provided that $t' \leq O(t \log t)$). We refer to this important property of R as *"distributional stability."*

Our original proof of the distributional property used an encoding argument and Fano's inequality. Several researchers suggested an alternative proof using Kullback-Leibler divergence and an inequality due to Pinsker. This gives slightly better bounds than our original proof when $t' \leq t$. The author later noticed that, by using an inequality of Vajda's in place of Pinsker's, this approach also allows us to handle values of t' as large as $O(t \log t)$ in a simpler way. Thus we feel that the divergence-based approach is ultimately the most convenient one to work with in general; this is the approach we now use in the main body of the paper.

Colleagues also pointed out that the distributional stability property can also be established using other similar, known results that also follow from divergence-based techniques: a lemma of

[19] Indeed, in our application we have essentially no control on R's behavior when we consider its restriction to inputs from S^t, so a generic result is needed.

[20] This is hardly the first paper in which such a bottleneck plays a crucial and somewhat unexpected role. For example, an interesting and slightly similar application of information-theoretic tools to the study of *metric embeddings* was found recently by Regev [Reg12].

Raz [Raz98], and the "Average Encoding Theorem" of Klauck et al. [KNTSZ07]. The latter was used in [KNTSZ07] to identify a stability property for trace and Hellinger distance metrics, for the inputs to a problem in quantum communication complexity; this was used for a round-elimination argument. Their proof is for inputs drawn from the uniform distribution, but extends readily to general distributions and can be used to derive the kind of lemma we need. We describe these alternative proofs of distributional stability in Appendix A,[21] and we describe our own original, encoding-based approach in Appendix B. We feel that all of these approaches to proving distributional stability are interesting and worth understanding.

Using the distributional-stability property of compressive mappings under product input distributions, we then establish a certain "sparsified variant" of this property (Lemma 6.3), which allows us to replace each $\mathcal{D}_j$ with a small set sampled from $\mathcal{D}_j$;[22] this is an important tool in addressing the efficient-sampleability requirement on our desired $\mathcal{D}^*$. Using this variant, we use the minimax theorem to show (in Lemma 6.4) that there exists a *distribution* $\mathfrak{D}$ over product input-distributions to R—with each product distribution defined over small subsets of S—such that, in expectation, $\mathfrak{D}$ disguises the random insertion of any string $y \in S$ at a uniformly-chosen position j. Finally, in Lemma 6.6 we obtain our desired "disguising distribution" $\mathcal{D}^*$ as a sparsified version of $\mathfrak{D}$, using a result due to Lipton and Young [LY94] and, independently, to Althöfer [Alt94], that guarantees the existence of sparsely-supported, nearly-optimal strategies in 2-player, zero-sum games.

1.4.3 Extension to the quantum case

Our techniques for studying quantum compression are closely analogous to the classical case. The main technical difference is that the output $R(\mathcal{D})$ of our compression reduction, on any input distribution $\mathcal{D}$, is now a (mixed) *quantum state*. In this setting, to carry out an analogue of the argument sketched in Sections 1.4.1 and 1.4.2 and fool a cheating Prover, we need a "disguising distribution" for R that meets a modified version of condition (i) from Section 1.4.1:

(i') For every $y \in \overline{L}_n$, if we select $j \in [t]$ uniformly then, *for any quantum measurement* $\mathcal{M}$, the expected statistical distance $\mathbb{E}_j\left[||\mathcal{M}(R(\mathcal{D}^*)) - \mathcal{M}(R(\mathcal{D}^*[y,j]))||_{\text{stat}}\right]$ is not too large.

A basic measure of distance between quantum states, the *trace distance*, is relevant here: if two states ρ, ρ' are at trace distance $||\rho - \rho'||_{\text{tr}} \leq \delta$, then for any measurement $\mathcal{M}$, the statistical distance $||\mathcal{M}(\rho) - \mathcal{M}(\rho')||_{\text{stat}}$ is at most δ. (In fact, this property *characterizes* the trace distance.) Thus to satisfy condition (i'), it will be enough to construct $\mathcal{D}^*$ so as to upper-bound $\mathbb{E}_j[||R(\mathcal{D}^*) - R(\mathcal{D}^*[y,j])||_{\text{tr}}]$, for uniformly-chosen j. We do this by essentially the same techniques as in the classical case. The one significant difference is that here, we need to establish a "stability property" for trace distance, analogous to the stability property for statistical distance described in Section 1.4.2. This can be obtained using the same basic divergence-based techniques as in the classical case, with the help of suitable tools from quantum information theory.[23]

[21] Russell Impagliazzo suggested the use of Raz's lemma; Salil Vadhan also helped me to understand the connection. Ashwin Nayak and S. Vadhan suggested direct proofs of distributional stability based on divergence and Pinsker's inequality, which we now use as our main approach. Dieter van Melkebeek also suggested the relevance of Pinsker's inequality, and James Lee and Avi Wigderson suggested to find a more direct information-theoretic proof. I thank all of these researchers.

[22] For convenience in the proof, we assume $\mathcal{D}_j = \mathcal{D}_{j'}$ for all j, j'.

[23] Our original approach to proving distributional stability also admits a quantum version, although we do not present it here.

1.5 Organization of the paper

In Section 2, we present the "bare minimum" of preliminaries needed to understand our proof of Theorem 1.1. We present this proof in Section 3.

The rest of the paper is devoted to proving stronger and more general results. In Section 4, we give the additional needed preliminary material for our work, including our definitions of compression reductions. In Section 5, we formally introduce parametrized problems and AND- and OR-expressive problems. In Section 6, we prove the main technical lemmas we use to obtain our results on limits of efficient instance compression (with alternative proofs of the first such lemma appearing in Appendices A and B).

Our results for the classical setting are proved in Sections 7 and 9. Section 7 gives the quantitatively strongest bounds on instance compression we are able to show for OR(SAT) and AND(SAT), under the non-uniform hardness assumption $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$ (or, the weaker assumption that NP does not have non-uniform statistical zero-knowledge proofs). Section 9, on the other hand, proves quantitatively weaker bounds on instance compression for OR(SAT) and AND(SAT) (and for certain other problems), but does so under the *uniform* hardness assumption $\mathsf{NP} \not\subseteq \mathsf{coAM}$.

Our quantum results are proved in Section 8 (along with some needed quantum background). Finally, in Section 10 we present questions for future study.

2 Preliminaries I

Definition 2.1. *The* binary entropy function $H(\alpha) : [0,1] \to [0,1]$ *is defined by*

$$H(\alpha) \ := \ -\alpha \log_2 \alpha - (1-\alpha)\log_2(1-\alpha)$$

on $(0,1)$*, with* $H(0) = H(1) := 0$.

$\binom{n}{k}$ denotes the binomial coefficient $n!k!/(n-k)!$. We will use the following standard, simple bound (see, e.g., [vL99, Chapter 1]) on the number of binary strings of low Hamming weight:

Fact 2.2. *For* $t \in \mathbb{N}$ *and* $\alpha \in (0, .5)$*, we have*

$$\sum_{0 \le \ell \le \alpha t} \binom{t}{\ell} \ \le \ 2^{H(\alpha)t} \ .$$

2.1 Statistical distance and distinguishability

All distributions in this paper will take finitely many values; let $\mathrm{supp}(\mathcal{D})$ be the set of values assumed by $\mathcal{D}$ with nonzero probability, and let $\mathcal{D}(u) := \Pr[\mathcal{D} = u]$.

For a probability distribution $\mathcal{D}$ and $t \ge 1$, we let $\mathcal{D}^{\otimes t}$ denote a t-tuple of outputs sampled independently from $\mathcal{D}$. We let $\mathcal{U}_K$ denote the uniform distribution over a multiset K.

The *statistical distance* of two distributions $\mathcal{D}, \mathcal{D}'$ over a shared universe of outcomes is defined as

$$||\mathcal{D} - \mathcal{D}'||_{\mathrm{stat}} \ := \ \frac{1}{2} \sum_{u \in \mathrm{supp}(\mathcal{D}) \cup \mathrm{supp}(\mathcal{D}')} |\mathcal{D}(u) - \mathcal{D}'(u)| \ .$$

The statistical distance between random variables is defined as the statistical distance between their governing distributions. We will use the following familiar "distinguishability interpretation" of the statistical distance. Suppose a value $b \in \{0,1\}$ is selected uniformly, unknown to us, and a sample $u \in U$ is drawn from $\mathcal{D}$ if $b = 0$, or from $\mathcal{D}'$ if $b = 1$. We observe u, and our goal is to correctly guess b. It is a basic fact that, for *any* $\mathcal{D}, \mathcal{D}'$, our maximum achievable success probability in this "distinguishing" experiment is precisely $\frac{1}{2}(1 + ||\mathcal{D} - \mathcal{D}'||_{\text{stat}})$. Furthermore, the optimal distinguishing algorithm may without loss of generality be a deterministic "maximum-likelihood" rule $ML(b|u)$: guess "$b = 1$" if and only if $\Pr[b = 1|u] \geq 1/2$. Similarly, we define a maximum-likelihood rule $ML(X|Y)$ for guessing any random variable X based on the observed value of any other random variable Y: simply guess the likeliest value of X conditioned on the observation (breaking ties arbitrarily).

The following fact follows from the distinguishability characterization of $||\cdot||_{\text{stat}}$; it is a convenient weakening of that principle.

Fact 2.3. *If X, Y are random variables over some shared domain S, and $\Delta := ||X - Y||_{\text{stat}}$, then there exists a subset $T \subseteq S$ such that*

$$\Pr_{x \sim X}[x \in T] \;\geq\; \Delta \quad \textit{and} \quad \Pr_{y \sim Y}[y \notin T] \;\geq\; \Delta \;.$$

We will also use the following easy facts:

Fact 2.4. *If X, Y are random variables over some shared domain S, and $R(X)$ is any (possibly randomized) function taking inputs from S, then*

$$||R(X) - R(Y)||_{\text{stat}} \;\leq\; ||X - Y||_{\text{stat}} \;.$$

Fact 2.5 ([SV03], Fact 2.3)**.** *Suppose (X_1, X_2, Y_1, Y_2) are distributions on a shared probability space Ω, that X_1 is independent of X_2, and that Y_1 is independent of Y_2. Then,*

$$||(X_1, X_2) - (Y_1, Y_2)||_{\text{stat}} \;\leq\; ||X_1 - Y_1||_{\text{stat}} + ||X_2 - Y_2||_{\text{stat}} \;.$$

3 Proof of Theorem 1.1

This section presents a proof that the "AND-conjecture" of Bodlaender, Downey, Fellows, and Hermelin [BDFH09] holds true unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. As discussed earlier, in this section we aim for a proof that avoids information theory and the minimax theorem. In later sections we will obtain stronger and more general results with these tools.

It will be convenient to consider mappings $R : \{0,1\}^{t \times n} \to \{0,1\}^{\leq t'}$, for fixed n, t, t'. For $A \subseteq \{0,1\}^n$, let $\mathbf{R}_A$ denote the distribution $\mathbf{R}_A := R(\mathcal{U}_A^{\otimes t})$, and for each each $a \in \{0,1\}^n$, define the distribution

$$\mathbf{R}_A[a] := R(\mathcal{U}_A^{\otimes (\mathbf{j}-1)}, a, \mathcal{U}_A^{\otimes (t-\mathbf{j})}) \;,$$

where $\mathbf{j} \sim \mathcal{U}_{[t]}$.

Define the *standout factor*

$$\beta(a, A) \;:=\; ||\mathbf{R}_A[a] - \mathbf{R}_A||_{\text{stat}} \;. \tag{1}$$

The basic idea of our proof of Theorem 1.1 is as follows: we will show that for each $n > 0$, there exists a poly(n)-size collection of poly(n)-size sets $A_i \subseteq L_n$,[24] such that every other element $x \in L_n$ will have standout factor $\beta(x, A_i) < 1 - \Omega(1)$ for at least one A_i. On the other hand, each $x \notin L_n$ will have standout factor 1 against each A_i.[25] Thus, if a polynomial-time Verifier "quizzes" a Prover by randomly sampling, either from $\mathbf{R}_{A_i}$ or from $\mathbf{R}_{A_i}[x]$ on each i, then Prover will be able to reliably guess which distribution was sampled from if and only if $x \notin L$. By known results, this leads to the conclusion $L \in \mathsf{coNP/poly}$.

Toward this end, the next lemma is our main technical tool:

Lemma 3.1. *Let $R : \{0,1\}^{t \times n} \to \{0,1\}^{\leq t'}$ be given. Let $A \subseteq \{0,1\}^n$ be a set of size $M \geq 100t$, and suppose that we select $a^* \sim \mathcal{U}_A$. Then if t is sufficiently large and $t' < 2(t-1)$, we have*

$$\mathbb{E}\left[\beta(a^*, A \setminus a^*)\right] \;\leq\; 1 - 10^{-4} \;. \tag{2}$$

Lemma 3.1 establishes that certain distributions are (at least slightly) "stable" under modification. Related facts, with information-theoretic proofs, appear in [Raz98, KNTSZ07] (see Appendix A), and these can be readily used to obtain our lemma. A distinctive aspect of Lemma 3.1, however, is that it establishes the closeness of the output distribution of R induced by an input to R containing a string a^*, to one from an input distribution to R that does not support a^*. This "apples-to-oranges" comparison is key to our application of Lemma 3.1: we will use it to build small (poly(n)-size) subsets of L_n that serve as helpful non-uniform advice to prevent *exponential*-size chunks of L_n from being accepted by Verifier. In the "minimax-free" proof being presented here, we will do so in an iterative fashion until all of L_n is "covered" by our advice. This is reminiscent of the incremental approach of Fortnow and Santhanam [FS11] to defining their advice, in their proof that the OR-conjecture holds unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

In the more general proofs we give in later sections, Lemmas 6.2 and 6.3 will play a role analogous (but not identical) to that of Lemma 3.1 in the current proof.

Proof of Lemma 3.1. Suppose to the contrary that $\mathbb{E}\left[\beta(a^*, A \setminus a^*)\right] > 1 - 10^{-4}$. Call $a \in A$ "distinctive" if $\beta(a, A \setminus a) \geq .99$; the measure β is bounded by 1, so more than a .99 fraction of $a \in A$ are distinctive.

For each $a \in A$, let $T = T_a$ be the set given by Fact 2.3, with $X := \mathbf{R}_{A \setminus a}[a], Y := \mathbf{R}_{A \setminus a}$; then for all distinctive $a \in A$, we have

$$\Pr_{z \sim \mathbf{R}_{A \setminus a}[a]}[z \in T_a] \;\geq\; .99 \;, \qquad \Pr_{z \sim \mathbf{R}_{A \setminus a}}[z \notin T_a] \;\geq\; .99 \;. \tag{3}$$

Let us index A as $A = \{a^1, \ldots, a^M\}$. Define a random R-input $\mathbf{x} = (x^1, \ldots, x^t) \sim \mathcal{U}_A^{\otimes t}$, and for $i \in [M]$ let $\mathrm{Incl}_i(\mathbf{x})$ be the indicator variable for the event that at least one of the elements x^j is equal to a^i. We also define the indicator variable

$$\mathrm{Corr}_i(\mathbf{x}) \;:=\; [\mathrm{Incl}_i(\mathbf{x}) \Leftrightarrow (R(\mathbf{x}) \in T_{a^i})] \;=\; \neg[\mathrm{Incl}_i(\mathbf{x}) \oplus (R(\mathbf{x}) \in T_{a^i})] \;.$$

The idea is that $R(\mathbf{x}) \in T_{a^i}$ "suggests" that a^i was included among the inputs to R, while $R(\mathbf{x}) \notin T_{a^i}$ suggests the opposite; $\mathrm{Corr}_i(\mathbf{x})$ checks whether the suggestion given is correct.

[24] (here, $L_n = L \cap \{0,1\}^n$)

[25] We note that this amounts to a weakened version of the Disguising-Distribution Lemma of Section 6.

It is easy to see that, if we condition on $[\mathrm{Incl}_i(\mathbf{x}) = 0]$, then $R(\mathbf{x})$ is distributed as $\mathbf{R}_{A\setminus a^i}$. In this case, the conditional probability that $[\mathrm{Corr}_i(\mathbf{x}) = 1]$ holds is at least .99, provided a^i is distinctive.

On the other hand, suppose we condition on $[\mathrm{Incl}_i(\mathbf{x}) = 1]$. Then the conditional probability that a^i appears *twice* among the coordinates of $\mathbf{x}$ is, by basic counting, at most $t/M \leq .01$. (After conditioning on any leftmost occurrence of a^i, there are at most $t-1$ indices which could contain the next occurrence of a^i; and each plays this role with probability at most $1/M$.) Thus under this conditioning, $R(\mathbf{x})$ is .01-close to the distribution $\mathbf{R}_{A\setminus a^i}[a^i]$, so that $[\mathrm{Corr}_i(\mathbf{x}) = 1]$ holds with probability at least $.99 - .01 = .98$ if a^i is distinctive.

It is also the case that $\sum_{i\in[M]} \mathrm{Incl}_i(\mathbf{x}) \geq .95t$ with probability at least .99 (for sufficiently large t), since $t/M \leq .01$. Combining all of our work, we find that for large enough t, with probability at least .5 the following conditions hold:

1. $\sum_{i\in[M]} \mathrm{Incl}_i(\mathbf{x}) \geq .95t$;
2. $\sum_{i\in[M]} [\mathrm{Incl}_i(\mathbf{x}) \wedge \mathrm{Corr}_i(\mathbf{x})] \geq .9t$;
3. $\sum_{i\in[M]} \mathrm{Corr}_i(\mathbf{x}) \geq .9M$.

Say that $\mathbf{x}$ is *good* if all of these conditions hold.

Now fix any R-output $z \in \{0,1\}^{\leq t'}$; we are going to derive an upper bound U on the number of good inputs $\mathbf{x}$ for which $R(\mathbf{x}) = z$. Since every $\mathbf{x}$ maps to a string of length $\leq t'$ under R, it will follow that

$$2^{t'+1} \;\geq\; \frac{.5|A^{\times t}|}{U} \;=\; \frac{.5M^t}{U}\;, \tag{4}$$

which will yield a contradiction to our settings.

First, suppose $z \in T_{a^i}$ for more than $t+.1M$ indices $i \in [M]$. Then for any $\mathbf{x}$ such that $R(\mathbf{x}) = z$, there are more than $.1M$ indices for which $\mathrm{Incl}_i(\mathbf{x}) = 0$ yet $z \in T_{a_i}$. For such i, $\mathrm{Corr}_i(\mathbf{x}) = 0$. Thus $\mathbf{x}$ is not good. So to have *any* good inputs $\mathbf{x}$ map to it under R, z must satisfy

$$|\{i : z \in T_{a^i}\}| \;\leq\; t + .1M\;. \tag{5}$$

Next, suppose $R(\mathbf{x}) = z$ and that $\mathbf{x} = (x^1, \ldots, x^t)$ contains more than $.15t$ components x^j whose value is any element $x^j = a^i \in A$ for which $z \notin T_{a^i}$. If $\mathbf{x}$ is good, then by property 1 of good inputs, among these components we can find a subcollection of more than $.1t$ components x^j whose values are pairwise distinct. For each $a^i = x^j$ in this subcollection, we have $\mathrm{Incl}_i(\mathbf{x}) = 1$ yet $\mathrm{Corr}_i(\mathbf{x}) = 0$. Thus $\sum_{i\in[M]} [\mathrm{Incl}_i(\mathbf{x}) \wedge \mathrm{Corr}_i(\mathbf{x})] < .9t$, so $\mathbf{x}$ is not good—a contradiction. Thus any good $\mathbf{x}$ for which $R(\mathbf{x}) = z$ can contain at most $.15t$ components x^j whose value $x^j = a^i$ satisfies $z \notin T_{a^i}$.

Combining this observation with Eq. (5), there is a set $A' \subseteq A$ (depending on z) of size at most $t + .1M \leq .11M$, such that for any good $\mathbf{x}$ mapping to z under R, at least $.85t$ components x^j satisfy $x^j \in A'$. We can now bound the number of good inputs $\mathbf{x}$ mapping to z under R; any such $\mathbf{x}$ is specifiable by:

- a set of at most $.15t$ "exceptional" indices $j \in [t]$;
- the values of x^j on these exceptional indices;
- the values of x^j on all other indices, which must lie in A'.

The number of such $\mathbf{x}$ is at most

$$\begin{aligned}\sum_{0\le t'\le .15t}\binom{t}{t'}M^{t'}(.11M)^{t-t'} &\le (.11)^{.85t}M^t\cdot\sum_{0\le t'\le .15t}\binom{t}{t'}\\ &\le (.11)^{.85t}M^t\cdot 2^{H(.15)t}\\ &< 4^{-t}M^t\ ,\end{aligned}$$

using Fact 2.2 and a calculation. Thus we may take as our bound $U := 4^{-t}M^t$, so that by Eq. (4),

$$2^{t'+1} \ \ge\ .5\cdot 4^t = 2^{2t-1},$$

which contradicts our assumption that $t' < 2(t-1)$. This proves Lemma 3.1. □

Proof of Theorem 1.1. We will show that the existence of the reduction R for L implies that there exists a two-message, private-coin, *interactive proof system* between a polynomial-time-bounded Verifier and a computationally unbounded Prover to prove that a given string $x \in \{0,1\}^n$ lies in $\overline{L}$. The proof system will be executable using $\mathrm{poly}(n)$ bits of non-uniform advice on length-n inputs; Prover will be able to make Verifier accept with probability 1 if $x \notin L$, and with probability at most $1-\Omega(1)$ if $x \in L$. It then follows from known results on interactive proof systems and non-uniform derandomization [GS86, Adl78] that $\overline{L} \in \mathsf{NP/poly}$ (see Theorem 4.11 and the proof of Theorem 4.15 for details), which gives our desired conclusion.

Using the existence of the reduction R and Lemma 3.1, we will prove the following claim:

Claim 3.2. *There exist multisets $A_1,\ldots,A_{q(n)\le \mathrm{poly}(n)} \subseteq L_n$, each of size bounded by some $s(n) \le \mathrm{poly}(n)$, such that, for all $x \in \{0,1\}^n \setminus \left(\bigcup_{i\in[q(n)]} A_i\right)$:*

1. *If $x \in \overline{L}_n$, then $\beta(x;A_i)=1$ for all $i\in[q(n)]$;*
2. *If $x \in L_n$, there is an $i\in[q(n)]$ for which $\beta(x;A_i)\le 1-10^{-5}$.*

Assuming the truth of Claim 3.2 for the moment, we use it to prove Theorem 1.1. For inputs of length n to our interactive proof system, we let the non-uniform advice be a description of the sets $A_1,\ldots,A_{q(n)}$ given by Claim 3.2, along with the value $t(n)$. The proof system works as follows. On input $x\in\{0,1\}^n$, Verifier first checks if x is in one of the sets A_i. If so, Verifier knows that $x\in L$. Otherwise, Verifier and Prover execute the following procedure in parallel for $i=1,2,\ldots,q(n)$:

- Verifier privately flips an unbiased coin $b_i \sim \mathcal{U}_{\{0,1\}}$;
- Verifier privately samples strings $y^{i,1},\ldots,y^{i,t(n)}\in\{0,1\}^n$ independently from $\mathcal{U}_{A_i}$;
- If $b_i = 0$ then Verifier sets

  $$z \ =\ z(i)\ :=\ R(y^{i,1},\ldots,y^{i,t(n)})\ ;$$

  otherwise ($b_i=1$), Verifier samples $\mathbf{j}=\mathbf{j}(i)\sim\mathcal{U}_{[t(n)]}$ and sets

  $$z\ :=\ R(y^{i,1},\ldots,y^{i,\mathbf{j}-1},x,y^{i,\mathbf{j}+1},\ldots,y^{i,t(n)})\ .$$

- Verifier sends z to Prover.

- Prover makes a guess $\widetilde{b}_i$ for the value of b_i.

Verifier accepts iff $\widetilde{b}_i = b_i$ for all i.

This protocol is clearly polynomial-time executable by Arthur given $t(n)$ and the description of $A_1, \ldots, A_{q(n)}$, and these sets are of polynomial size and polynomial in number. Now let us analyze the behavior of the protocol (assuming $x \notin \bigcup_i A_i$). First, suppose that $x \in \overline{L}_n$. In this case, we have

$$||\mathbf{R}_{A_i}[x] - \mathbf{R}_{A_i}||_{\text{stat}} \;=\; 1$$

for each i, by the first property of our sets A_i. Thus, Prover can guess b_i with perfect confidence for each i, and can cause Verifier to accept with probability 1.

Next, suppose that $x \in L_n$. Then by the second property of our sets, there exists an $i^* \in [q(n)]$ such that

$$\left|\left|\mathbf{R}_{A_{i^*}}[x] - \mathbf{R}_{A_{i^*}}\right|\right|_{\text{stat}} \;\leq\; 1 - 10^{-5} \;.$$

By the distinguishability characterization of statistical distance, and the independence of the trials $i = 1, 2, \ldots, q(n)$, this implies that the probability that Prover guesses b_{i^*} correctly is at most $1 - .5 \cdot 10^{-5}$. Thus Verifier rejects with probability $\Omega(1)$. So our interactive proof has the desired properties. As discussed earlier, this implies $\overline{L} \in \mathsf{NP/poly}$. □

Proof of Claim 3.2. Fixing attention to a single value of n, let $(t, t') = (t(n), t'(n))$. Assume that t is large enough to apply Lemma 3.1. (Note that then t' satisfies the assumptions of that lemma as well.) Let $M := 100t$. We define a sequence of sets $S_1 \supseteq S_2 \supseteq \ldots \supseteq S_{q(n)+1} = \emptyset$, each contained in L_n, and a sequence of sets $A_1, A_2, \ldots, A_{q(n)}$, with all elements of A_i drawn from S_i.

Let $S_1 := L_n$. Inductively, having defined S_i, we define A_i, S_{i+1} as follows. If $|S_i| < M$, we let $A_i := S_i$ and $S_{i+1} := \emptyset$, and set $q(n) := i$, terminating the construction at this stage. Otherwise ($|S_i| \geq M$), we let A_i be a uniformly random size-$(M-1)$ subset of S_i. We let

$$S_{i+1} := \left\{a \in S_i \setminus A_i : \beta(a, A_i) \;>\; 1 - 10^{-5}\right\} \;.$$

The procedure clearly terminates, since $|S_{i+1}| \leq |S_i| - (M-1)$ whenever $S_{i+1} \neq \emptyset$. Let us verify that these A_i satisfy conditions 1-2 of the Claim; we will then argue that $q(n) \leq \text{poly}(n)$ (with high probability over the randomness in the construction).

First, suppose $x \in \overline{L}_n \setminus \left(\bigcup_{i \in [q(n)]} A_i\right)$. Then with attention to Eq. (1), note that R always outputs an element of $\overline{L'}$ when x is one of the inputs to R. On the other hand, when all inputs to R are drawn from some $A_i \subseteq S_i \subseteq L_n$, R outputs an element of L'. Thus these two cases are perfectly distinguishable, and $\beta(x, A_i) = 1$ for each i, as needed.

Next suppose $x \in L_n \setminus \left(\bigcup_{i \in [q(n)]} A_i\right)$. Let $i \in [1, q(n)]$ be the unique index such that $x \in S_i \setminus S_{i+1}$. Then by the definitions, we have $\beta(x, A_i) = ||\mathbf{R}_{A_i}[x] - \mathbf{R}_{A_i}||_{\text{stat}} \leq 1 - 10^{-5}$.

Finally, we argue that $q(n) \leq \text{poly}(n)$ with high probability. Note that when we generate A_i as a uniform set of size $M-1$, we may equivalently generate A_i by first generating a uniform set $\widehat{A}_i \subseteq S_i$ of size M, then selecting a uniform element a^* of $\widehat{A}_i$ to discard to form A_i.

By Lemma 3.1, $\mathbb{E}_{a^*}[\beta(a^*, A_i)] \leq 1 - 10^{-4}$. Then with probability at least .9 over our randomness at this stage, a^* satisfies $\beta(a^*, A_i) \leq 1 - 10^{-5}$. But a^* is distributed as a uniform element of $S_i \setminus A_i$. Thus,

$$\mathbb{E}[|S_{i+1}|] \;\leq\; .1(|S_i| - |A_i|) \;.$$

Thus $q(n) = O(n)$ with high probability. This completes the proof of Claim 3.2. □

4 Preliminaries II

Now we collect facts and definitions that will inform our work in the rest of the paper as we prove more general results.

4.1 Information theory background

Recall from Section 2 that $H(\alpha)$ denotes the binary entropy function on $[0,1]$. For a finitely-supported random variable Z, we let

$$H(Z) \ := \sum_{z \ \in \ \mathrm{supp}(Z)} - \Pr[Z=z] \log_2 \Pr[Z=z]$$

denote the Shannon entropy of Z. Then, for two possibly-dependent random variables Y, Z,

$$H(Z|Y) \ := \ \mathbb{E}_{y \sim Y}[H(Z_{[Y=y]})] \ = \ H((Y,Z)) - H(Y)$$

denotes the *entropy of Z conditional on Y*. ($Z_{[Y=y]}$ denotes Z conditioned on the event $[Y=y]$.)

Fact 4.1. *For all X, Y, $H((X,Y)) \leq H(X) + H(Y)$ and $H(X|Y) \leq H(X)$, with equality holding in each case iff X, Y are independent. Similarly, $H(X|(Y,Z)) \leq H(X|Y)$.*

Definition 4.2 (Mutual information)**.** *The* mutual information *between random variables X, Y is defined as $I(X;Y) := H(X) + H(Y) - H((X,Y))$.*

The next fact follows easily from the definitions.

Fact 4.3. *Mutual information obeys the following properties, for all random variables X, Y, Z:*

1. *$I(X;Y) = I(Y;X)$;*
2. *$I(X;(Y,Z)) = I(X;Y) + I((X,Y);Z) - I(Y;Z)$;*
3. *$I(X;(Y,Z)) \geq I(X;Y)$;*
4. *$I(X;Z) = 0$ if X, Z are independent.*

Lemma 4.4. *If $X^1, \ldots, X^t$ are independent, then*

$$I(Y;(X^1,\ldots,X^t)) \ \geq \ \sum_{j \in [t]} I(Y;X^j) \ .$$

Our proof of this standard claim follows steps in [Nay99a, p. 33].

Proof. We have

$$\begin{aligned} I(Y;(X^1,\ldots,X^t)) \ &= I(Y;X^t) + I((Y,X^t);(X^1,\ldots,X^{t-1})) - \underbrace{I(X^t;(X^1,\ldots,X^{t-1}))}_{=0, \text{ by Fact 4.3, item 4}} \\ &\geq \ I(Y;X^t) + I(Y;(X^1,\ldots,X^{t-1})) \ , \end{aligned}$$

where we used item 2 of Fact 4.3 in the first step, and items 1 and 3 in the second step. Iterating in this way gives the Lemma. □

The next definition is a useful, non-symmetric measure of difference between random variables.

Definition 4.5 (KL divergence)**.** *The (binary)* Kullback-Leibler divergence*, or* KL divergence *between random variables X, Y, is denoted $D_{\mathrm{KL}}(X||Y)$ and defined as*

$$D_{\mathrm{KL}}(X||Y) \ := \sum_{x \in \mathrm{supp}(X)} \Pr[X = x] \cdot \log_2 \left(\frac{\Pr[X = x]}{\Pr[Y = x]} \right) \ .$$

The convention is that for $p \neq 0$, we have $p \log_2(p/0) = +\infty$. So D_{KL} may be infinite. We have the following basic equivalence (see [CT06, Chapter 2]):

Fact 4.6. *Let X, Y be any random variables; let X' be distributed as X and independent of Y. The mutual information and Kullback-Leibler divergence satisfy*

$$I(X;Y) \ = \ D_{\mathrm{KL}}((X,Y)||(X',Y)) \ .$$

Fact 4.6.1 (Chain rule)**.** *For two pairs of random variables (X, Y) and (X, Y), we have*

$$D_{\mathrm{KL}}((X,Y)||(X',Y)) \ = \ D_{\mathrm{KL}}(X||X') + \mathbb{E}_{x \sim X}[D_{\mathrm{KL}}(Y_{[X=x]}||Y'_{[X'=x]})] \ .$$

A proof of the following important result can be found in [CT06] (see Lemma 11.6.1, p. 370).

Theorem 4.7 (Pinsker's inequality, stated for binary KL divergence)**.** *For any random variables Z, Z',*

$$D_{\mathrm{KL}}(Z||Z') \ \geq \ \frac{2}{\ln 2} \cdot ||Z - Z'||^2_{\mathrm{stat}} \ .$$

In particular, $D_{\mathrm{KL}}(Z||Z')$ is always nonnegative. When $||Z - Z'||_{\mathrm{stat}} \approx 1$, the following bound, known as Vajda's inequality (see [FHT03, RW09]), gives better information on the divergence:

Theorem 4.8 (Vajda's inequality, stated for binary KL divergence)**.** *For any random variables Z, Z', let $\Delta := ||Z - Z'||_{\mathrm{stat}}$. Then,*

$$D_{\mathrm{KL}}(Z||Z') \ \geq \ \frac{1}{\ln 2} \left(\ln \left(\frac{1+\Delta}{1-\Delta} \right) - \frac{2\Delta}{1+\Delta} \right) \ \geq \ \frac{1}{\ln 2} \left(\ln \left(\frac{1}{1-\Delta} \right) - 1 \right) .$$

4.2 Basic complexity classes and promise problems

We assume familiarity with the basic complexity classes NP and coNP and the higher levels Σ^p_k, Π^p_k of the Polynomial Hierarchy PH. (For the needed background in complexity theory, see [AB09].) In this paper we define $\mathsf{NP}, \mathsf{coNP}$, etc. as classes of *languages* (not promise problems).

We also assume familiarity with the general model of polynomial-size, non-uniform advice, and with the non-uniform classes $\mathsf{NP/poly}$ and $\mathsf{coNP/poly}$. It is considered unlikely that $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. In particular, this would imply a collapse of the Polynomial Hierarchy:

Theorem 4.9 ([Yap83])**.** *If $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, then $\mathsf{PH} = \Sigma^p_3 = \Pi^p_3$.*

We use $\mathsf{pr\text{-}NP}$, $\mathsf{pr\text{-}coNP}$, etc. to denote the analogous complexity classes for *promise problems*. Recall that the *complement* of a promise problem (Π_Y, Π_N) is the promise problem (Π_N, Π_Y) which swaps the "yes" and "no" cases. Also, for a class C of promise problems, we define the class $\mathsf{coC} = \{(\Pi_Y, \Pi_N) : (\Pi_N, \Pi_Y) \in \mathsf{C}\}$. A *many-to-one* reduction B from the promise problem

$\Pi = (\Pi_Y, \Pi_N)$ to $\Pi' = (\Pi'_Y, \Pi'_N)$ is a mapping satisfying $B(\Pi_Y) \subseteq \Pi'_Y, B(\Pi_N) \subseteq \Pi'_N$. This definition applies as well to the special case where one or both of the promise problems are languages. When we refer to NP-complete languages in this paper, we mean languages complete for NP under deterministic, polynomial-time many-to-one reducibility.

All of the results we prove in this paper about limits of compression for *languages* L and language complexity classes readily extend to the setting of compression for promise problems (under the analogous definitions). However, for notational simplicity we will state our main results for languages, and will only use promise problems and promise classes where doing so helps to streamline our proofs and our result statements.

4.3 Arthur-Merlin protocols

We will make use of the model of (public-coin, two-round) *Arthur-Merlin protocols*. To be precise, these are protocols $\mathcal{P}$, defined by a deterministic polynomial-time predicate $A(x, r, w)$, which operate as follows. On an input x, visible to both a polynomial-time bounded verifier (Arthur) and to a computationally-unbounded prover (Merlin):

1. Arthur generates a uniformly random string r and sends it to Merlin;
2. Merlin sends a response string w to Arthur;
3. Arthur accepts if $A(x, r, w) = 1$, otherwise rejects.

We require that $|r|, |w|$ each be pre-specified lengths $\leq \text{poly}(n)$, where $n = |x|$, and that these lengths be computable in $\text{poly}(n)$ time given 1^n.

We will need to work with promise problems having Arthur-Merlin protocols. Say that such a protocol $\mathcal{P}$ *defines* a promise problem $\Pi = (\Pi_Y, \Pi_N)$ with *completeness* $c(n)$ *and soundness* $s(n)$ if

1. For all $x \in \Pi_Y$, some Merlin strategy causes Arthur to accept with probability $\geq c(n)$;
2. For all $x \in \Pi_N$, all Merlin strategies cause Arthur to accept with probability $\leq s(n)$.

Let $\text{pr-AM}_{c(n),s(n)}$ denote the class of promise problems definable by an Arthur-Merlin protocol with completeness $c(n)$ and soundness $s(n)$; let $\text{pr-AM} := \text{pr-AM}_{1,1/3}$. Then, $\text{pr-coAM} = \{(\Pi_Y, \Pi_N) : (\Pi_N, \Pi_Y) \in \text{pr-AM}\}$.

Theorem 4.10 ([FGM+89])**.** *For any parameters $s(n), c(n) \in (0, 1]$ that are polynomial-time computable*[26] *and satisfy $\frac{1}{\text{poly}(n)} < s(n) < c(n) - \frac{1}{\text{poly}(n)}$, we have $\text{pr-AM}_{c(n),s(n)} = \text{pr-AM}$. If we drop the requirement $s(n) > \frac{1}{\text{poly}(n)}$, but keep the gap requirement, we still have $\text{pr-AM}_{c(n),s(n)} \subseteq \text{pr-AM}$.*

The next, well-known result follows from the non-uniform derandomization technique of Adleman [Adl78]:

Theorem 4.11. $\text{pr-AM} \subseteq \text{pr-NP/poly}$. *Similarly,* $\text{pr-coAM} \subseteq \text{pr-coNP/poly}$.

[26] (say, as rational values represented by their numerator and denominator)

4.4 Statistical zero-knowledge and the SD problem

Next we will define the *statistical zero-knowledge* class SZK. Actually, we will only work with its promise-problem analogue pr-SZK.[27] Informally, these are the promise problems (Π_Y, Π_N) for which a (private-coin) interactive proof of membership in Π_Y can be given, in which the verifier *learns (almost) nothing—except* to become convinced that the input y indeed lies in Π_Y! The "learns nothing" requirement is cashed out by requiring that the verifier be able to *simulate* interactions with the intended prover strategy on any input y, such that if $y \in \Pi_Y$, the resulting distribution is negligibly close in statistical distance to the true distribution generated by their interaction.

Making this definition formal is somewhat delicate. (For details, and for more information on these and related classes, see [SV03].) Fortunately, there is a simple (but non-trivial) alternative characterization of pr-SZK. First, given a Boolean circuit $C = C(r)$ with k output gates, and an ordering on these gates, let $\mathcal{D}_C$ denote the output distribution of C on a uniformly random input r. (This is a random variable over $\{0,1\}^k$.) We use the following problem:

Definition 4.12. *For parameters $0 \leq d \leq D \leq 1$, define the promise problem $\mathrm{SD}^{\geq D}_{\leq d} = (\Pi_Y, \Pi_N)$ as follows:*

$$\Pi_Y := \{\langle C, C' \rangle : ||\mathcal{D}_C - \mathcal{D}_{C'}||_{\mathrm{stat}} \geq D\} ,$$
$$\Pi_N := \{\langle C, C' \rangle : ||\mathcal{D}_C - \mathcal{D}_{C'}||_{\mathrm{stat}} \leq d\} .$$

Define $\mathrm{SD}^{\leq d}_{>D}$ analogously, switching the "yes" and "no" cases. In this definition, both $d = d(n)$ and $D = D(n)$ may be parameters depending on the input length $n = |\langle C, C' \rangle|$.

It is shown in [SV03] that the standard, complicated definition of pr-SZK is equivalent to the following simpler one, which we take as our definition:

Definition 4.13. *Let* pr-SZK *be defined as the class of promise problems for which there is a many-to-one,[28] deterministic polynomial-time reduction from Π to $\mathrm{SD}^{\geq 2/3}_{\leq 1/3}$.*

The constants $2/3, 1/3$ in the above definition are not arbitrary; it is unknown whether we get the same class if we replace them by $.51, .49$. However, we have the following result:

Theorem 4.14 (Follows from [SV03]; described as Theorem 1 in [GV11])**.** *Suppose $0 \leq d = d(n) < D = D(n) \leq 1$ are polynomial-time computable, and satisfy $D^2 > d + \frac{1}{\mathrm{poly}(n)}$. Then, $\mathrm{SD}^{\geq D}_{\leq d} \in$ pr-SZK.*

When we merely have $D - d \geq \frac{1}{\mathrm{poly}(n)}$, the following weaker, standard result holds:

Theorem 4.15. *Suppose $0 \leq d = d(n) < D = D(n) \leq 1$ are polynomial-time computable and satisfy $D > d + \frac{1}{\mathrm{poly}(n)}$. Then, $\mathrm{SD}^{\geq D}_{\leq d} \in$ pr-AM.*

Proof sketch. We describe a *private-coin* two-message protocol, in which the verifier has a source of random bits not viewable by the prover; any such protocol can be efficiently converted into a public-coin one [GS86].

[27] Often the promise class is denoted SZK.

[28] Recall the definition in Section 4.2.

Let $m = m(n) \leq \text{poly}(n)$ be a large value. On input $\langle C, C' \rangle$, Verifier chooses $b_1, \ldots, b_m$ uniformly at random and, for $i \in [m]$, samples

$$z^i \sim \mathcal{D}_C \qquad \text{if } b_i = 0, \qquad z^i \sim \mathcal{D}_{C'} \quad \text{if } b_i = 1,$$

independently for each i. Prover is asked to try to guess the values $b_1, \ldots, b_m$.

If m is chosen appropriately large then, using the distinguishability interpretation of statistical distance (see Section 2.1),

1. If $||\mathcal{D}_C - \mathcal{D}_{C'}||_{\text{stat}} > D$ then Prover can, with high probability, guess at least a $\frac{1}{2}\left(1 + \frac{D-d}{2}\right)$ fraction of the bits b_i correctly;

2. If $||\mathcal{D}_C - \mathcal{D}_{C'}||_{\text{stat}} < d$ then Prover cannot, except with low probability, guess this fraction of the b_is correctly.

Thus, Verifier can use this threshold as an acceptance criterion, so that the protocol has the desired completeness-soundness gap. After converting to a public-coin protocol, we find that $\text{SD}^{\geq D}_{\leq d} \in \text{pr-}\mathsf{AM}_{2/3,1/3} = \text{pr-}\mathsf{AM}$ (using Theorem 4.10). □

We will also use the following important results about pr-SZK:

Theorem 4.16 ([Oka00])**.** pr-SZK *is closed under complement. In particular, if* $\text{SD}^{\geq D}_{\leq d} \in \text{pr-}\mathsf{SZK}$ *then also* $\text{SD}^{\leq d}_{\geq D} \in \text{pr-}\mathsf{SZK}$.

Theorem 4.17. $\text{pr-}\mathsf{SZK} \subseteq \text{pr-}\mathsf{AM} \cap \text{pr-}\mathsf{coAM} \subseteq \text{pr-}\mathsf{NP/poly} \cap \text{pr-}\mathsf{coNP/poly}$.

The containment in pr-coAM is due to Fortnow [For87]; containment in pr-AM was first shown by Aiello and Håstad [AH91].[29] The second containment in Theorem 4.17 uses Theorem 4.11.

Finally, one of our results (Theorem 7.3) will make use of the class pr-PZK of problems having (honest-verifier) *perfect* zero-knowledge proofs. This is a subclass of pr-SZK. We will not define pr-PZK (see, e.g., [SV03]); unfortunately it has no known simple characterization analogous to Definition 4.13 for pr-SZK. We will, however, use the following result:

Theorem 4.18 ([SV03], Proposition 5.7)**.** $\text{SD}^{\geq 1}_{\leq .5} \in \text{pr-}\mathsf{PZK}$.

Next we combine tools described in Sections 4.3 and 4.4, reformulating them slightly.

Theorem 4.19. *Let* $0 \leq d = d(n) < D = D(n) \leq 1$ *be (not necessarily computable) parameters.*

1. *If* $D > d + \frac{1}{\text{poly}(n)}$*, then* $\text{SD}^{\geq D}_{\leq d} \in \text{pr-}\mathsf{NP/poly}$.

2. *If we have the stronger gap* $D^2 > d + \frac{1}{\text{poly}(n)}$*, then* $\text{SD}^{\geq D}_{\leq d}$ *is many-to-one reducible to* $\text{SD}^{\geq 2/3}_{\leq 1/3} \in \text{pr-}\mathsf{SZK}$*, in non-uniform polynomial time. Also,* $\text{SD}^{\geq D}_{\leq d} \in \text{pr-}\mathsf{coNP/poly}$.

[29] These works treat language classes, but the proofs extend without change to the promise-problem setting. Also, these works analyze a so-called "honest-verifier" model of statistical zero-knowledge proofs; these were shown to have the same expressive power as "cheating-verifier" statistical zero-knowledge proofs in [GSV98].

Proof sketch. For item 1, we essentially combine Theorem 4.15 with Theorem 4.11. The only extra ingredient needed is to encode sufficiently accurate approximations of $d(n), D(n)$ into the non-uniform advice for length n, and to use these in defining the private-coin protocol as in the proof of Theorem 4.15. We then convert this non-uniform protocol into an NP/poly one by the same techniques from [GS86] (which shows how to convert private-coin to public-coin protocols), Theorem 4.10 (to get perfect completeness), and Theorem 4.11 (to derandomize).

Similarly, for item 2, we essentially combine Theorems 4.14 and 4.17, except that at each step we need to incorporate approximations of $d(n), D(n)$ as (additional) non-uniform advice. □

4.5 f-compression reductions

Here we define a class of compression reductions for the problems $f \circ L$ introduced in Section 1.3.1, in which one is given $(x^1, \ldots, x^m)$ and must compute $f(L(x^1), \ldots, L(x^m))$. Our main focus will be the case where f is the OR or AND function of its input bits. The problem $f \circ L$ will be formally defined as a parametrized problem in Section 5.1, but it will be useful to have a specialized definition for this problem as well; here we won't explicitly rely on the parametrized-problem framework.

Our next definition is modeled on definitions in [BDFH09, FS11], with some differences. Notably, we will consider reductions where a quantitative compression guarantee is only made when all the input strings x^j are of some equal length n, and the number of input strings x^j is equal to some value $t_1(n)$ determined by n. The error bound will also be a function of n. This specialization is mostly to reduce clutter in our work, and will not lead to loss of generality: we will be *ruling out* the existence of compression reductions (under complexity-theoretic assumptions, and for all $t_1(n)$ that are sufficiently large compared to other parameters), so ruling out even compression algorithms that work only in narrow input-regimes will lead to stronger results.

Definition 4.20 (Probabilistic f-compression reductions)**.** *Let L, L' be two languages, and let $f : \{0,1\}^* \to \{0,1\}$ be a Boolean function. Let $t_1(n), t_2(n) : \mathbb{N}^+ \to \mathbb{N}^+$ and $\xi(n) : \mathbb{N}^+ \to [0,1]$ be given.*

A probabilistic f-compression reduction for L, *with parameters $(t_1(n), t_2(n), \xi(n))$ and target language L', is a randomized mapping $R(x^1, \ldots, x^m)$ outputting a string z, such that for all $(x^1, \ldots, x^{t_1(n)}) \in \{0,1\}^{t_1(n) \times n}$,*

1. $\Pr_R[L'(z) = f\left(L(x^1), \ldots, L(x^{t_1(n)})\right)] \geq 1 - \xi(n)$;
2. $|z| \leq t_2(n)$.

If some reduction R as above is computable in probabilistic polynomial time, we say that L is PPT-f-compressible *with parameters $(t_1(n), t_2(n), \xi(n))$. (This does not require that $(t_1(n), t_2(n), \xi(n))$ themselves be computable.)*

5 Parametrized problems and parametrized compression

A central aim of our work is to better understand the limitations of efficient compressive reductions for a variety of parametrized problems. For this we need to formally define parametrized problems and an appropriate model of probabilistic compression for these problems. However, some readers may be satisfied to understand our work on the limits of efficient AND- and OR-compression (as defined in Section 4.5) for SAT and other NP-complete languages. To prove these results, including

Theorem 1.3 in the Introduction, we will not need the definitions of this section, and readers may choose to skip ahead to Section 6. (We will find it convenient to prove Theorem 1.2 using the definitions below; however, this result can also be derived directly from our Theorem 7.1, item 2 with little trouble.)

5.1 Parametrized problems

We will use the following definition:

Definition 5.1 ([DF99])**.** *A* parametrized problem *is a subset of binary strings of the form* $\langle x, 1^k \rangle$*, for* $x \in \{0,1\}^*$ *and* $k > 0$ *(under some natural binary encoding of such tuples).*

Thus, our convention is that a parametrized problem is just a particular type of decision problem, i.e., a language.[30] However, we will use P to denote a generic parametrized problem, as opposed to an "ordinary" language, denoted L. Sometimes, as in the Introduction, we speak of "parametrized versions" of an ordinary decision problem L. There is no single, canonical way to go from a decision problem to a parametrized problem; often, however, a parametrized problem can be formed from a decision problem L in a natural way. For example, we formally define VAR-SAT, OR(SAT), and AND(SAT) from the Introduction as follows:

Definition 5.2. *Fix some natural encoding of tuples of bit-strings, and some encoding of Boolean formulas as bit-strings. Define*

1. VAR-SAT $:= \{\langle \psi, 1^k \rangle \mid \psi$ *is satisfiable and contains* $\leq k$ *distinct variables*$\}$*;*
2. OR(SAT) $:= \{\langle \psi_1, \ldots, \psi_t, 1^k \rangle \mid$ *at least one* ψ_j *is satisfiable, and each* ψ_j *is of bit-length* $\leq k\}$*;*
3. AND(SAT) $:= \{\langle \psi_1, \ldots, \psi_t, 1^k \rangle \mid$ *every* ψ_j *is satisfiable, and each* ψ_j *is of bit-length* $\leq k\}$*.*

We also generalize items 2 and 3 above:

Definition 5.3. *Let* $L \subseteq \{0,1\}^*$*, and* $f : \{0,1\}^* \to \{0,1\}$*. Define*

1. $\mathrm{OR}(L) := \{\langle (x^1, \ldots, x^t), 1^k \rangle \mid \bigvee_{j=1}^t L(x^i) = 1$ *and* $|x^j| \leq k$ *for each* $j\}$*;*
2. $\mathrm{AND}(L) := \{\langle (x^1, \ldots, x^t), 1^k \rangle \mid \bigwedge_{j=1}^t L(x^i) = 1$ *and* $|x^j| \leq k$ *for each* $j\}$*;*
3. $f \circ L := \{\langle (x^1, \ldots, x^t), 1^k \rangle \mid f(L(x^1), \ldots, L(x^t)) = 1$ *and* $|x^j| \leq k$ *for each* $j\}$*.*

5.2 OR-expressive and AND-expressive parametrized problems

Our compression lower bounds will apply to two classes of parametrized problems. As we will explain, these classes are closely related to classes identified earlier in [HN10, BDFH09, BJK11a, BTY11]; the classes we introduce will help to apply our techniques uniformly to these various earlier classes.

[30] In this definition we are following [FS11]. In [BDFH09] and many other works, parametrized problems are defined as a subset of $\{0,1\}^* \times \mathbb{N}^+$ (the parameter is still presented as part of the input); they refer to the corresponding subset of strings of form $\langle x, 1^k \rangle$ as the "unparametrized version" or "classical version" of the problem.

Definition 5.4 (OR- and AND-expressive problems)**.** *A parametrized problem P is* OR-expressive*, with parameter $S(n) \leq \text{poly}(n)$, if there exists an* NP*-complete language L and a deterministic polynomial-time reduction B acting as follows. Whenever B receives an input of form $\langle (x^1, \ldots, x^t), 1^n \rangle$, for any $t, n \in \mathbb{N}^+$, B outputs a tuple*

$$(\langle y^1, 1^{k_1} \rangle, \ldots, \langle y^s, 1^{k_s} \rangle) \ .$$

We have the following properties:

1. $\langle (x^1, \ldots, x^t), 1^n \rangle \ \in \ \text{OR}(L) \quad \Longleftrightarrow \quad \exists \ i \in [s] : \langle y^i, 1^{k_i} \rangle \ \in P$*;*
2. $s \leq S(n)$ *(in particular, the bound is independent of t);*
3. *For each $i \in [s]$, $|y^i| \leq (t+n)^{O(1)}$ and $k_i \leq n^{O(1)}$.*

Define AND-expressive *problems identically, except we replace condition 1 above by*

1'. $\langle (x^1, \ldots, x^t), 1^n \rangle \ \in \ \text{AND}(L) \quad \Longleftrightarrow \quad \forall \ i \in [s] : \langle y^i, 1^{k_i} \rangle \ \in P$.

The results of [BDFH09] imply that a variety of natural parametrized problems are OR- or AND-expressive:

Theorem 5.5 (Follows from [BDFH09])**.** *1.* OR(SAT) *is* OR*-expressive with $S(n) = 1$. Also, each of the following parametrized problems are* OR*-expressive with $S(n) \leq \text{poly}(n)$:*

- k-Path*,* k-Cycle*,* k-Exact Cycle *and* k-Short Cheap Tour*,*
- k-Graph Minor Order Test *and* k-Bounded Treewidth Subgraph Test*,*
- k-Planar Graph Subgraph Test *and* k-Planar Graph Induced Subgraph Test*,*
- (k, σ)-Short Nondeterministic Turing Machine Computation*,*
- w-Independent Set*,* w-Clique *and* w-Dominating Set*,*

defined in [BDFH09].

2. AND(SAT) *is* AND*-expressive with $S(n) = 1$. Also, each of the following parametrized problems are* AND*-expressive with $S(n) \leq \text{poly}(n)$:*

- k-Cutwidth*,* k-Modified Cutwidth*, and* k-Search Number*,*
- k-Pathwidth*,* k-Treewidth*, and* k-Branchwidth*,*
- k-Gate Matrix Layout *and* k-Front Size*,*
- w*-3*-Coloring *and* w*-3*-Domatic Number*,*

also defined in [BDFH09].

In [BDFH09], the authors define a notion of *compositionality* for parametrized problems. If a parametrized problem P is compositional and NP-complete, then it is OR-expressive, with respect to the NP-complete language $L = P$. Also, if P is NP-complete and $\overline{P}$ is compositional, then P is AND-expressive. These facts follow almost immediately from the definitions. Theorem 5.5 then follows from the compositionality results proved in [BDFH09]. In a number of the problems above we can actually take $S(n) = 1$.

Bodlaender, Jansen, and Kratsch [BJK11a] introduced a notion of *cross-compositionality* of parametrized problems, generalizing compositionality. They showed that the evidence against efficient compression against compositional problems given by [BDFH09, FS11] can be extended to cross-compositional problems. Cross-compositional problems are also OR-expressive, as follows from the definitions [BJK11a, Section 3].[31] As shown in [BJK11a], this class includes interesting parametrized versions of the Clique, Chromatic Number, and Feedback Vertex Set problems.

AND-expressiveness results are fewer in number, although this may be partly due to the fact that, after the results of [FS11] appeared, OR-expressiveness results were preferentially sought. Another example of an AND-expressive problem (not known to be OR-expressive) is presented in [BJK11c].

We also have the following result, derived from the earlier work of [HN10]:

Theorem 5.6 (Follows from [HN10, FS11])**.** *Each of the problems* Clique, Dominating Set,[32] Integer Programming, *described in [HN10] and modeled as parametrized problems in [FS11] (with slightly distinctive, but natural, parametrizations), are* OR*-expressive, with* $S(n) = 1$.

A class of reductions between parametrized problems, called *W-reductions*, is used in these works (see [FS11, Definition 2.10]); OR(SAT) is shown to W-reduce to each of the problems listed in Theorem 5.6. This immediately implies that these problems are OR-expressive with $S(n) = 1$. Also, if an OR-expressive parametrized problem P W-reduces to a second problem Q, then Q is also OR-expressive. This technique was used in [BTY11] to derive additional hardness-of-compression results for problems not easily captured by the compositionality framework; our new results apply to these problems as well.

We remark that the polynomial bounds involved in the reductions of Theorems 5.5 and 5.6 are fairly modest.

5.3 Parametrized compression

We define compression reductions for parametrized problems as follows, following [FS11] (but with some added flexibility in our definitions):

Definition 5.7 (Probabilistic parametrized compression reductions)**.** *Let P be a parametrized problem and L' be a language, and say we are given two functions*

$$c(m,k,w) : (\mathbb{N}^+)^3 \ \rightarrow \ \mathbb{N}^+, \quad \xi(m,k,w) : (\mathbb{N}^+)^3 \ \rightarrow \ [0,1] \ .$$

*Say that a randomized mapping $R : \{0,1\}^* \rightarrow \{0,1\}^*$ is a (c,ξ)-*parametrized compression reduction *for P, with target language L', if for all inputs of form $\langle y, 1^k, 1^w\rangle$, $R(\langle y, 1^k, 1^w\rangle)$ outputs a string z such that:*

1. $\Pr_R[L'(z) = P(\langle y, 1^k\rangle)] \geq 1 - \xi(|y|,k,w)$;
2. $|z| \leq c(|y|,k,w)$.

[31] Strictly speaking, according to their definition, cross-compositional problems are OR-expressive under the minor restriction on the reduction in Definition 5.4 that the input $\langle (x^1,\ldots,x^t), 1^n\rangle$ satisfy $t \leq 2^{n^a}$, for some $a > 0$. This is of no importance to us, since we will always work with the case $t \leq \text{poly}(n)$; we could have required this in Definition 5.4, and could prove the same variety of hardness results.

[32] (these are different parametrized problems than w-Clique and w-Dominating Set in Theorem 5.5 above)

We call c the compression bound *and ξ the* error bound *of the reduction; we call w the* confidence parameter*.*

For a parametrized problem P, if some reduction R as above is computable in probabilistic polynomial time, we say that P is PPT-compressible *with parameters (c, ξ).*

We will not be exploring the full range of possible parameter values in the above definition, but we believe it provides a reasonable framework for future work. (Only a few interesting examples of randomized parametrized compression reductions seem to be known; see [HN10, KW12].) The idea of a confidence parameter w, that one can use to increase the reliability of the compression at the expense of a potentially larger output size, is natural for probabilistic compression and will be useful in our work. (The same basic notion was used earlier in [FS11].)

Next, we define a notion of "strong" compressibility as in the Introduction, preserving flexibility in the error bound:

Definition 5.8. *Say that P is* strongly PPT-compressible *with error bound $\xi(m, k, w)$, if P is PPT-compressible (to some target language L') with error bound ξ and some compression bound c satisfying $c(m, k, 1) \leq k^{O(1)}$, with the polynomial bound independent of m.*

Using the majority-vote technique of [FS11, Proposition 5.1], we have the following easy result:

Lemma 5.9. *Let $a > 0$. Suppose that P is strongly PPT-compressible with error bound satisfying $\xi(m, k, 1) \leq .5 - k^{-O(1)}$. Then, P is also PPT-compressible with compression bound $c'(m, k, w) \leq k^{O(1)} \cdot w$ and error bound $\xi'(m, k, w) \leq 2^{-w}$.*

5.4 Connecting parametrized compression and f-compression

The next lemma shows that to give evidence against efficient compression for "expressive" parametrized problems, it suffices to give evidence against efficient AND- and OR-compression for NP-complete languages. This lemma is modeled on [BDFH09, Lemma 2], but with some slight complications due to the probabilistic setting. For simplicity we only treat strong compression in the result below; our techniques also extend to give evidence against more modest compression amounts for expressive problems. (For more modest compression amounts, the obtainable results are weaker when the parameter $S(n)$ in the definition of expressiveness is fast-growing.)

Lemma 5.10. *Let L be an* NP*-complete language.*

1. *Suppose that the parametrized problem P is* OR*-expressive with respect to L, with parameter $S(n) \leq \text{poly}(n)$. If P is strongly PPT-compressible with error bound $\xi(m, k, 1) \leq .5 - k^{-O(1)}$, then for any polynomially-bounded function $T(n) : \mathbb{N}^+ \to \mathbb{N}^+$, L is PPT-*OR*-compressible with parameters*

$$t_1(n) = T(n), \quad t_2(n) \leq S(n) \cdot n^{O(1)}, \quad \xi'(n) \leq 2^{-n} .$$

2. *Suppose P is* AND*-expressive with respect to L. If P is strongly PPT-compressible with error bound $\xi(m, k, 1) \leq .5 - k^{-O(1)}$, then L is PPT-*AND*-compressible with parameters $(t_1(n), t_2(n), \xi'(n))$ as in item 1.*

Proof of Lemma 5.10. We will prove item 1 above; item 2 is proved similarly. Let R be the PPT compression reduction R for P given by Lemma 5.9. Let L'_0 be the target language of R. Let B be the reduction for P and L as in Definition 5.4.

We define an OR-compression reduction R' for L, with target language $L' := \mathrm{OR}(L'_0)$, as follows. In defining R', we let $t_1(n) := T(n)$. On inputs $x^1, \ldots, x^{T(n)} \in \{0,1\}^{T(n)\times n}$, the reduction first applies B to $\langle (x^1, \ldots, x^{T(n)}), 1^n \rangle$, yielding a tuple $(\langle y^1, 1^{k_1}\rangle, \ldots, \langle y^s, 1^{k_s}\rangle)$. Next, for each $i \in [s]$, R' applies R to the string $\langle y^i, 1^{k_i}, 1^{2n}\rangle$ (here we are selecting the confidence parameter $w := 2n$ for R), yielding an output z^i. Then R' outputs $\langle (z^1, \ldots, z^s), 1^M \rangle$, where $M := \max_i |z^i|$.

R' is clearly polynomial-time computable. Now let us analyze its compression and reliability properties. First, each y^i is of bit-length $|y^i| \leq (T(n)+n)^{O(1)}$, and $k_i \leq n^{O(1)}$, by item 3 of Definition 5.4. Then by the compression guarantee for R, each z^i is of bit-length $\leq n^{O(1)} \cdot w = n^{O(1)}$. Thus for the output-size bound of R' we may take $t_2(n) \leq S(n) \cdot n^{O(1)}$, as needed.

Now we bound the error of R'. Using the correctness property of B (Definition 5.4, item 1), the equivalence

$$\langle (z^1, \ldots, z^s), 1^M \rangle \in \mathrm{OR}(L'_0) \iff \bigvee_{j=1}^{T(n)} [x^j \in L]$$

holds as long as each application of R, namely $R(\langle y^i, 1^{k_i}\rangle)$ for $i \in [s]$, is successful. By a union bound, this occurs with probability $\geq 1 - S(n) \cdot 2^{-2n}$, which is larger than $1 - 2^{-n}$ for sufficiently large n. (For smaller n, R' may solve its input problem directly by brute force.) Thus for the error bound $\xi'(n)$ for R', we may take $\xi'(n) \leq 2^{-n}$. □

6 Technical lemmas

In this section we present our main technical lemmas. Our final goal in this section will be the "Disguising-Distribution Lemma," our key technical tool for our main results.

6.1 Distributional stability

Here we define the notion of "distributional stability" described in Section 1.4.2.

Definition 6.1. *Let U be some finite universe, and let $T, n \geq 1$ be integers. Given a possibly-randomized mapping $F(x^1, \ldots, x^T) : \{0,1\}^{T\times n} \to U$, and a collection $\mathcal{D}_1, \ldots, \mathcal{D}_T$ of mutually independent distributions over $\{0,1\}^n$, for $j \in [T]$ let*

$$\gamma_j \;:=\; \mathop{\mathbb{E}}_{y\sim\mathcal{D}_j}\left[\left|\left|F\left(\mathcal{D}_1, \ldots, \mathcal{D}_{j-1}, y, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_t\right) - F\left(\mathcal{D}_1, \ldots, \mathcal{D}_t\right)\right|\right|_{\mathrm{stat}}\right] .$$

*For $\delta \in [0,1]$, say that F is δ-*distributionally stable (or δ-DS) *with respect to $\mathcal{D}_1, \ldots, \mathcal{D}_T$ if*

$$\frac{1}{T}\sum_{j=1}^{T} \gamma_j \;\leq\; \delta \; .$$

Lemma 6.2. *Let $R(x^1, \ldots, x^t) : \{0,1\}^{t\times n} \to \{0,1\}^{\leq t'}$ be any possibly-randomized mapping, for any $n, t, t' \in \mathbb{N}^+$. R is δ-distributionally stable with respect to any independent input distributions $\mathcal{D}_1, \ldots, \mathcal{D}_t$, where we may take either of the following two bounds:*

1. $\delta := \sqrt{\frac{\ln 2}{2} \cdot \frac{t'+1}{t}}$;

2. $\delta := 1 - 2^{-\frac{t'}{t}-3}$.

Our proof of Lemma 6.2, item 1 essentially follows suggestions by Ashwin Nayak and Salil Vadhan; item 2 is a small modification using Vajda's inequality. When $t'/t = 1 - \Omega(1)$, the bound given in item 1 above is within constant factors of the bound from our original distributional stability lemma, Lemma B.4. On the other hand, when $t'/t = 1 - \alpha \approx 1$, the bound in Lemma 6.2, item 1 is better (i.e., smaller) by a $\Theta\left(\log \frac{1}{\alpha}\right)$ factor. We don't know how to prove a version of item 2 above with the methods of Lemma B.4; this alternative bound is important for our work. In an earlier draft we used a more complicated workaround to prove the results obtainable from item 2.

Proof of Lemma 6.2. Define independent random variables $X^j \sim \mathcal{D}_j$ over $\{0,1\}^n$, for $j \in [t]$. Let $\mathbf{R} := R(X^1, \ldots, X^t)$.

The entropy of $\mathbf{R}$ is at most $\log_2\left(\left|\{0,1\}^{\leq t'}\right|\right) < t'+1$. Thus, the mutual information $I((X^1, \ldots, X^t); \mathbf{R})$ is less than $t'+1$. By the independence of the X^js, Lemma 4.4 gives

$$\sum_{j \in [t]} I(X^j; \mathbf{R}) \;<\; t' + 1 \;. \tag{6}$$

By Fact 4.6,

$$I(X^j; \mathbf{R}) \;= D_{\mathrm{KL}}\left((X^j, \mathbf{R}) \,||\, (Y^j, \mathbf{R})\right) \;, \tag{7}$$

where $Y^j \sim \mathcal{D}_j$ is independent of $\mathbf{R}$. By Theorem 4.7,

$$\begin{aligned} D_{\mathrm{KL}}\left((X^j, \mathbf{R}) \,||\, (Y^j, \mathbf{R})\right) \;&\geq\; \frac{2}{\ln 2} \cdot ||(X^j, \mathbf{R}) - (Y^j, \mathbf{R})||^2_{\mathrm{stat}} \\ &=\; \frac{2}{\ln 2} \cdot \mathbb{E}_{x^j \sim \mathcal{D}_j}\left[\left|\left|R\left(\mathcal{D}_1, \ldots, \mathcal{D}_{j-1}, x^j, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_t\right) - R\left(\mathcal{D}_1, \ldots, \mathcal{D}_t\right)\right|\right|_{\mathrm{stat}}\right]^2 \;, \end{aligned}$$

where the equality follows from the distinguishability interpretation of statistical distance. Using this, we find

$$\begin{aligned} &\left(\frac{1}{t}\sum_{j \in [t]} \mathbb{E}_{x^j \sim \mathcal{D}_j}\left[\left|\left|R\left(\mathcal{D}_1, \ldots, \mathcal{D}_{j-1}, x^j, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_t\right) - R\left(\mathcal{D}_1, \ldots, \mathcal{D}_t\right)\right|\right|_{\mathrm{stat}}\right]\right)^2 \\ \leq\; &\frac{1}{t}\sum_{j \in [t]} \mathbb{E}_{x^j \sim \mathcal{D}_j}\left[\left|\left|R\left(\mathcal{D}_1, \ldots, \mathcal{D}_{j-1}, x^j, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_t\right) - R\left(\mathcal{D}_1, \ldots, \mathcal{D}_t\right)\right|\right|_{\mathrm{stat}}\right]^2 \\ &\text{(by Jensen's inequality)} \\ <\; &\frac{\ln 2}{2} \cdot \frac{t'+1}{t} \;. \end{aligned}$$

Thus, R is $\sqrt{\frac{\ln 2}{2} \cdot \frac{t'+1}{t}}$-distributionally stable with respect to $\mathcal{D}^1, \ldots, \mathcal{D}^t$. This proves item 1 of the Lemma.

For item 2, we apply the alternative bound, Vajda's inequality (Theorem 4.8), to each $j \in [t]$, to find

$$\begin{aligned} D_{\mathrm{KL}}\left((X^j, \mathbf{R}) \,||\, (Y^j, \mathbf{R})\right) &\geq \frac{1}{\ln 2}\left(\ln\left(\frac{1}{1 - ||(X^j, \mathbf{R}) - (Y^j, \mathbf{R})||_{\mathrm{stat}}}\right) - 1\right) \\ &= \frac{1}{\ln 2}\left(\ln\left(\frac{1}{\varepsilon_j}\right) - 1\right) , \end{aligned}$$

where we define

$$\varepsilon_j := 1 - \mathbb{E}_{x^j \sim \mathcal{D}_j}\left[\left|\left|R\left(\mathcal{D}_1, \ldots, \mathcal{D}_{j-1}, x^j, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_t\right) - R\left(\mathcal{D}_1, \ldots, \mathcal{D}_t\right)\right|\right|_{\mathrm{stat}}\right]$$

and note that $\varepsilon_j > 0$. Averaging over $j \in [t]$ and applying Eqs. (6) and (7),

$$\frac{t'+1}{t} \geq \frac{1}{t}\sum_{j\in[t]} \frac{1}{\ln 2}\left(\ln\left(\frac{1}{\varepsilon_j}\right) - 1\right) ,$$

i.e.,

$$\frac{1}{t}\sum_{j\in[t]} \ln\left(\frac{1}{\varepsilon_j}\right) \leq \frac{(\ln 2)(t'+1)}{t} + 1 .$$

The function $f(x) = \ln(1/x)$ has second derivative $x^{-2} > 0$ for $x > 0$, and so Jensen's inequality gives

$$\ln\left(\frac{1}{\frac{1}{t}\sum_{j\in[t]} \varepsilon_j}\right) \leq \frac{(\ln 2)(t'+1)}{t} + 1 .$$

This implies

$$\frac{1}{t}\sum_{j\in[t]} \varepsilon_j \geq \left(e^{\frac{(\ln 2)(t'+1)}{t}+1}\right)^{-1} \geq 2^{-\frac{t'}{t}-3} ,$$

which proves item 2. □

6.2 Sparsified distributional stability

Here we prove a technical lemma showing that if a mapping F is distributionally stable with respect to i.i.d. inputs, then F also obeys a slightly different stability property, in which we replace an input distribution $\mathcal{D}$ with a "sparsified" version of $\mathcal{D}$.

Lemma 6.3. *Let U be a finite set, and let $F(x^1, \ldots, x^T) : \{0,1\}^{T\times n} \to U$ be given. Suppose F is δ-distributionally stable with respect to input distribution $\mathcal{D}^{\otimes T}$, for every distribution $\mathcal{D}$ over $\{0,1\}^n$.*

Fix some distribution $\mathcal{D}$ over $\{0,1\}^n$, and let $x^1, \ldots, x^d$ be independently sampled from $\mathcal{D}$. Let $k^* \sim \mathcal{U}_{[d]}$.

Let $\widehat{\mathcal{D}}$ denote the distribution defined by sampling uniformly from the multiset $\{x^k\}_{k\neq k^*}$. (This distribution is itself a random variable, determined by $x^1, \ldots, x^d$ and by k^*.) Define

$$\beta_j := \mathop{\mathbb{E}}_{k^*, x^1, \ldots, x^d}\left[\left|\left|F\left(\widehat{\mathcal{D}}^{\otimes(j-1)}, x^{k^*}, \widehat{\mathcal{D}}^{\otimes(T-j)}\right) - F\left(\widehat{\mathcal{D}}^{\otimes T}\right)\right|\right|_{\mathrm{stat}}\right] ,$$

where all the $\widehat{\mathcal{D}}$s are mutually independent (for fixed values of $x^1, \ldots, x^d$ and k^*). Then,

$$\frac{1}{T}\sum_{j=1}^{t}\beta_j \;\leq\; \delta + 2T/d \;.$$

Proof. Let $\widetilde{\mathcal{D}}$ denote the distribution, determined by $x^1, \ldots, x^d$, that samples uniformly from the multiset $\{x^k\}_{k\in[d]}$. By an easy calculation, for any values of $x^1, \ldots, x^d$ and k^* we can bound

$$\left|\left|\widetilde{\mathcal{D}} - \widehat{\mathcal{D}}\right|\right|_{\text{stat}} \;\leq\; 1/d \;.$$

It follows that

$$\left|\left|F\left(\widetilde{\mathcal{D}}^{\otimes T}\right) - F\left(\widehat{\mathcal{D}}^{\otimes T}\right)\right|\right|_{\text{stat}} \;\leq\; \left|\left|\widetilde{\mathcal{D}}^{\otimes T} - \widehat{\mathcal{D}}^{\otimes T}\right|\right|_{\text{stat}} \;\leq\; T/d \;,$$

where in the last step we used Fact 2.5 and the fact that for any assignment to $x^1, \ldots, x^d$ and to k^*, the T copies of $\widetilde{\mathcal{D}}$ used are mutually independent, as are the copies of $\widehat{\mathcal{D}}$.

By identical reasoning, for any assignment to $x^1, \ldots, x^d$ and to k^*, and for any index $j \in [T]$ we have

$$\left|\left|F\left(\widetilde{\mathcal{D}}^{\otimes(j-1)}, x^{k^*}, \widetilde{\mathcal{D}}^{\otimes(T-j)}\right) - F\left(\widehat{\mathcal{D}}^{\otimes(j-1)}, x^{k^*}, \widehat{\mathcal{D}}^{\otimes(T-j)}\right)\right|\right|_{\text{stat}} \;\leq\; (T-1)/d \;.$$

Using the triangle inequality for $||\cdot||_{\text{stat}}$, for any values $x^1, \ldots, x^d, k^*$ and any index $j \in [T]$ we always have

$$\begin{aligned}
&\left|\left|F\left(\widehat{\mathcal{D}}^{\otimes(j-1)}, x^{k^*}, \widehat{\mathcal{D}}^{\otimes(T-j)}\right) \;-\; F\left(\widehat{\mathcal{D}}^{\otimes T}\right)\right|\right|_{\text{stat}} \\
&\leq \left|\left|F\left(\widehat{\mathcal{D}}^{\otimes(j-1)}, x^{k^*}, \widehat{\mathcal{D}}^{\otimes(T-j)}\right) - F\left(\widetilde{\mathcal{D}}^{\otimes(j-1)}, x^{k^*}, \widetilde{\mathcal{D}}^{\otimes(T-j)}\right)\right|\right|_{\text{stat}} \\
&\quad + \left|\left|F\left(\widetilde{\mathcal{D}}^{\otimes(j-1)}, x^{k^*}, \widetilde{\mathcal{D}}^{\otimes(T-j)}\right) \;-\; F\left(\widetilde{\mathcal{D}}^{\otimes T}\right)\right|\right|_{\text{stat}} \\
&\quad + \left|\left|F\left(\widetilde{\mathcal{D}}^{\otimes T}\right) - F\left(\widehat{\mathcal{D}}^{\otimes T}\right)\right|\right|_{\text{stat}} \\
&\leq \left|\left|F\left(\widetilde{\mathcal{D}}^{\otimes(j-1)}, x^{k^*}, \widetilde{\mathcal{D}}^{\otimes(T-j)}\right) \;-\; F\left(\widetilde{\mathcal{D}}^{\otimes T}\right)\right|\right|_{\text{stat}} + 2T/d \;. \qquad (8)
\end{aligned}$$

Now suppose we fix any values $x^1, \ldots, x^d$, leaving k^* undetermined. The value k^* is uniform on $[d]$, so that x^{k^*} is distributed exactly according to $\widetilde{\mathcal{D}}$. Under our conditioning, let

$$\gamma_j = \gamma_j\left(\{x^k\}_{k\in[d]}\right) \;:=\; \mathop{\mathbb{E}}_{k^*}\left[\;\left|\left|F\left(\widetilde{\mathcal{D}}^{\otimes(j-1)}, x^{k^*}, \widetilde{\mathcal{D}}^{\otimes(T-j)}\right) \;-\; F\left(\widetilde{\mathcal{D}}^{\otimes T}\right)\right|\right|_{\text{stat}}\;\right] \;.$$

By our original assumption, F is δ-DS with respect to input distribution $\widetilde{\mathcal{D}}^{\otimes T}$. Thus, for any $x^1, \ldots, x^d$ we have

$$\frac{1}{t}\sum_{j=1}^{t}\gamma_j \;\leq\; \delta \;. \qquad (9)$$

Now γ_j is itself a random variable, determined by $x^1, \ldots, x^d$, and from Eq. (8) we have

$$\beta_j \;\leq\; \mathbb{E}[\gamma_j] + 2T/d \;.$$

Using linearity of expectation, we find that

$$\frac{1}{t}\sum_{j=1}^{t}\beta_j \;\leq\; \delta + 2T/d \;.$$

□

6.3 Building disguising distributions

In the next lemmas we show how the distributional stability of a mapping F can be used to obtain a "disguising distribution" for F. In Lemma 6.6 we will apply this to give disguising distributions for any sufficiently compressive mapping R.

Recall that $\mathcal{U}_K$ denotes the uniform distribution over a multiset K.

Lemma 6.4. *Suppose $F(x^1, \ldots, x^T) : \{0,1\}^{T\times n} \to U$ obeys the assumption of Lemma 6.3: namely, F is δ-distributionally stable with respect to input distribution $\mathcal{D}^{\otimes T}$, for every distribution $\mathcal{D}$ over $\{0,1\}^n$.*

Let $S \subseteq \{0,1\}^n$, and fix some value $d > 0$. There exists a distribution $\mathcal{K}$ over size-d multisets $K \subseteq S$, such that for every $y \in S$, the following holds:

$$\mathop{\mathbb{E}}_{K\sim\mathcal{K},j^*\sim\mathcal{U}_{[T]}}\left[\;\left|\left|F\left(\mathcal{U}_K^{\otimes(j^*-1)}, y, \mathcal{U}_K^{\otimes(T-j^*)}\right) \;-\; F\left(\mathcal{U}_K^{\otimes T}\right)\right|\right|_{\text{stat}}\;\right] \;\leq\; \delta + 2T/(d+1) \;.$$

(Here the copies of $\mathcal{U}_K$ are to be mutually independent for fixed K, although the set $K \sim \mathcal{K}$ used is the same for each copy.)

Proof. Consider the following two-player, simultaneous-move, zero-sum game:

- **Player 1:** chooses a size-d multiset $K \subseteq S$.
- **Player 2:** chooses a string $y \in S$.
- **Payoff:** Player 2 receives a payoff equal to

$$\mathop{\mathbb{E}}_{j^*\sim\mathcal{U}_{[T]}}\left[\;\left|\left|F\left(\mathcal{U}_K^{\otimes(j^*-1)}, y, \mathcal{U}_K^{\otimes(T-j^*)}\right) \;-\; F\left(\mathcal{U}_K^{\otimes T}\right)\right|\right|_{\text{stat}}\;\right] \;.$$

  (Note that this payoff is a determinate value, given (K, y).)

Consider any randomized strategy by Player 2, specified by a distribution $y \sim Y$ over S. In response, let $\mathcal{K}_Y$ be the randomized Player-1 strategy that chooses a size-d multiset K of elements sampled independently from Y.

To bound the expected payoff under the strategy-pair $(\mathcal{K}_Y, Y)$, note that we can equivalently generate $(K, y) \sim (\mathcal{K}_Y, Y)$ as follows. First, sample $x^1, \ldots, x^{d+1}$ independently from Y. Sample $k^* \sim \mathcal{U}_{[d+1]}$, set $y := x^{k^*}$, and let

$$K := \{x^1, \ldots, x^{k^*-1}, x^{k^*+1}, \ldots, x^{d+1}\} \;.$$

It is easily verified that $(K, y) \sim (\mathcal{K}_Y, Y)$ as desired. Then Lemma 6.3, applied to our initial distributional-stability assumption on F, informs us that

$$\mathop{\mathbb{E}}_{j^*\sim\mathcal{U}_{[t]},K,y}\left[\;\left|\left|F\left(\mathcal{U}_K^{\otimes(j^*-1)}, y, \mathcal{U}_K^{\otimes(T-j^*)}\right) \;-\; F\left(\mathcal{U}_K^{\otimes T}\right)\right|\right|_{\text{stat}}\;\right] \;\leq\; \delta + 2T/(d+1) \;.$$

Thus Player 2's expected payoff against $\mathcal{K}_Y$ is at most $\delta + 2T/(d+1)$.

As Y was arbitrary, the minimax theorem tells us that there exists a distribution $\mathcal{K}$ over Player-1 moves that forces Player 2's expected payoff under *every* strategy to be at most $\delta + 2T/(d+1)$. The result follows. □

Lemma 6.5. *Let U be a finite set, and let $F(x^1, \ldots, x^T) : \{0,1\}^{T \times n} \to U$ be given. Suppose F is δ-distributionally stable with respect to input distribution $\mathcal{D}^{\otimes T}$, for every distribution $\mathcal{D}$ over $\{0,1\}^n$.*

Let $S \subseteq \{0,1\}^n$, and fix $d > 0$. Given any $\varepsilon > 0$, let $s := \lceil (.5 \ln 2) n/\varepsilon^2 \rceil$. Then there exists a collection $K_1, \ldots, K_s$ of size-d multisets contained in S, such that for every $y \in S$ the following holds:

$$\mathop{\mathbb{E}}_{a \sim \mathcal{U}_{[s]}, j^* \sim \mathcal{U}_{[t]}} \left[\left|\left| F\left(\mathcal{U}_{K_a}^{\otimes (j^*-1)}, y, \mathcal{U}_{K_a}^{\otimes (T-j^*)}\right) \; - \; F\left(\mathcal{U}_{K_a}^{\otimes T}\right) \right|\right|_{\text{stat}} \right] \; \leq \; \delta + 2T/(d+1) + \varepsilon \; .$$

Proof. This is an immediate application (to the game in Lemma 6.4) of a general result due to Lipton and Young [LY94, Theorem 2], showing that all two-player, zero-sum games have sparsely-supported, nearly-optimal player strategies. (Essentially the same result was proved independently by Althöfer [Alt94], and a more general result for many-player, non-zero-sum games was proved later in [LMM03].) The support size required in the Lipton-Young-Althöfer result depends logarithmically on the number of pure strategies available to the player we are opposing; in our case, Player 2 has a choice of $|S| \leq 2^n$ strings y, so we get $s = O(n/\varepsilon^2)$. In their proof technique applied to our setting, the $K_1, \ldots, K_s$ are obtained by sampling independently from the distribution $\mathcal{K}$ given by Lemma 6.4, giving a suitable choice of $K_1, \ldots, K_s$ with nonzero probability. □

Lemma 6.6 (**Disguising-Distribution Lemma**)**.** *Let $R(x^1, \ldots, x^t) : \{0,1\}^{t \times n} \to \{0,1\}^{\leq t'}$ be any possibly-randomized mapping, for $t, t' \in \mathbb{N}^+$. Let $S \subseteq \{0,1\}^n$, and fix $d > 0$. Given any $\varepsilon > 0$, let $s := \lceil (.5 \ln 2) n/\varepsilon^2 \rceil$. Let*

$$\widehat{\delta} := \min \left\{ \sqrt{\frac{\ln 2}{2} \cdot \frac{t'+1}{t}}, \;\; 1 - 2^{-\frac{t'}{t} - 3} \right\} \; .$$

Then there exists a collection $K_1, \ldots, K_s$ of size-d multisets contained in S, such that for every $y \in S$, we have

$$\mathop{\mathbb{E}}_{a \sim \mathcal{U}_{[s]}, j^* \sim \mathcal{U}_{[t]}} \left[\left|\left| R\left(\mathcal{U}_{K_a}^{\otimes (j^*-1)}, \; y, \; \mathcal{U}_{K_a}^{\otimes (t-j^*)}\right) \; - \; R\left(\mathcal{U}_{K_a}^{\otimes t}\right) \right|\right|_{\text{stat}} \right]$$
$$\leq \; \widehat{\delta} + 2t/(d+1) + \varepsilon \; .$$

Proof. This follows immediately from the combination of Lemmas 6.2 and 6.5, applied to $F := R$ (and with $T := t$). □

7 Limits to efficient (classical) compression

In this section, we show that a sufficiently high-quality PPT-OR-compression reduction for any language L implies that $L \in \mathsf{NP/poly}$. We also show that above a higher threshold of quality, such a compression reduction implies that L has non-uniform, statistical zero-knowledge proofs, which in

particular implies $L \in \mathsf{coNP/poly}$ as well. We will then apply these results to give evidence against efficient probabilistic compression for AND(SAT) and OR(SAT), as described in the Introduction, and for other parametrized problems with either of the two "expressiveness" properties described in Section 5.2. We will also present our result on f-compression reductions for more general combining functions f, and our result extending the work of Dell and Van Melkebeek [DvM10] on problems with polynomial kernelizations.

7.1 Complexity upper bounds from OR-compression schemes

Theorem 7.1. *Let L be any language. Suppose $t_1(n), t_2(n) : \mathbb{N}^+ \to \mathbb{N}^+$ are (not necessarily computable) functions. Suppose that there exists a PPT-*OR*-compression reduction $R(x^1, \ldots, x^t) : \{0,1\}^{t_1(n) \times n} \to \{0,1\}^{\leq t_2(n)}$ for L with parameters $t_1(n), t_2(n)$, error bound $\xi(n) < .5$, and some target language L'. Let*

$$\widehat{\delta} := \min \left\{ \sqrt{\frac{\ln 2}{2} \cdot \frac{t_2(n)+1}{t_1(n)}}, \quad 1 - 2^{-\frac{t_2(n)}{t_1(n)} - 3} \right\} .$$

1. *If for some constant $c > 0$ we have*

$$1 - 2\xi(n) - \widehat{\delta} \;\geq\; \frac{1}{n^c} , \tag{10}$$

   then $L \in \mathsf{NP/poly}$.

2. *If for some $c > 0$ we have the (stronger) bound*

$$(1 - 2\xi(n))^2 - \widehat{\delta} \;\geq\; \frac{1}{n^c} , \tag{11}$$

   then there is a many-to-one reduction from L to a promise problem in $\mathsf{pr\text{-}SZK}$. The reduction is computable in non-uniform polynomial time; in particular, this implies $L \in \mathsf{NP/poly} \cap \mathsf{coNP/poly}$.

We remark that, using the technique of [FS11, Proposition 5.1], one can reduce the error bound $\xi(n)$ of an OR-compression scheme, at the cost of increasing the output-length bound $t_2(n)$. (The idea is to perform multiple, independent applications of R to the fixed input tuple $(x^1, \ldots, x^{t_1(n)})$ and to concatenate the results in the output, using a majority-vote rule to define a new target language.) With this amplification, we can in some cases apply Theorem 7.1 where its assumptions do not hold for the original scheme—or, we may obtain the stronger conclusion in item 2 of Theorem 7.1 in cases where only item 1 would apply directly.

Proof of Theorem 7.1. We will use the same basic reduction to prove items 1 and 2. First, with non-uniformity it is easy to handle length-n inputs whenever $L_n = \{0,1\}^n$, so let us assume from this point on that $\overline{L}_n$ is nonempty.

Using R, we define a deterministic, non-uniform polynomial-time reduction $\mathcal{R}$ that, on input $y \in \{0,1\}^n$, builds a description of two circuits C, C'. The aim is that $||\mathcal{D}_C - \mathcal{D}_{C'}||_{\text{stat}}$ should be large if $y \in L$, and small if $y \notin L$. $\mathcal{R}$ works as follows:

- **Non-uniform advice for length n:** a description of the value $t_1(n)$, and the multisets $K_1, \ldots, K_s \subseteq \overline{L}_n$ given by Lemma 6.5 with

$$(t, t') \;:=\; (t_1(n), t_2(n)), \quad S := \overline{L}_n, \quad d := \lceil 8t_1(n) \cdot n^c \rceil\,, \quad \varepsilon := \frac{1}{4n^c}\;.$$

(Here $c > 0$ is as in Eq. (10) or Eq. (11), according to which item of the Theorem we are proving.) Note that d and the value s given by Lemma 6.5 are both $\leq \mathrm{poly}(n)$ under these settings, so our advice is of polynomial length.

- **On input $y \in \{0,1\}^n$:** let $\mathcal{R}$ output descriptions $\langle C, C' \rangle$ of the following two randomized circuits:
  - **Circuit C:** samples $a \sim \mathcal{U}_{[s]}$, then samples

$$\overline{x} \;=\; (x^1, \ldots, x^{t_1(n)}) \;\sim\; \mathcal{U}_{K_a}^{\otimes t_1(n)}\;,$$

and outputs $z := R(\overline{x})$.
  - **Circuit C':** samples values

$$a \sim \mathcal{U}_{[s]}, \quad j^* \sim \mathcal{U}_{[t_1(n)]}\;;$$

then, samples

$$\overline{x} \;\sim\; \left(\mathcal{U}_{K_a}^{\otimes (j^*-1)},\; y,\; \mathcal{U}_{K_a}^{\otimes (t_1(n)-j^*)}\right)\;,$$

and outputs $z := R(\overline{x})$.

Claim 7.2. *The following holds:*

1. *If $y \in L$, then*

$$||\mathcal{D}_C - \mathcal{D}_{C'}||_{\mathrm{stat}} \;\geq\; D(n) \;:=\; 1 - 2\xi(n)\;;$$

2. *If $y \notin L$, then*

$$||\mathcal{D}_C - \mathcal{D}_{C'}||_{\mathrm{stat}} \;\leq\; d(n) \;:=\; \widehat{\delta} + \frac{1}{2n^c}\;. \tag{12}$$

We defer the proof of Claim 7.2, and use it to prove the two items of Theorem 7.1.

For item 1 of Theorem 7.1, if Eq. (10) holds (for sufficiently large n), then $D(n) - d(n) \geq \frac{1}{n^c}$. Now $D(n), d(n)$ were parametrized in terms of $n = |y|$, but the gap $D(n) - d(n)$ is also at least inverse-polynomial in the length $N \leq \mathrm{poly}(n)$ of the output description $\langle C, C' \rangle$. Thus our reduction $\mathcal{R}$ reduces any instance y of the decision problem for L, to an equivalent instance $\mathcal{R}(y) = \langle C, C' \rangle$ of the promise problem $\mathrm{SD}^{\geq D'(N)}_{\leq d'(N)}$, with different parameters $D'(N), d'(N)$ still satisfying the gap condition $D' - d' \geq \frac{1}{\mathrm{poly}(N)}$.

By item 1 of Theorem 4.19, $\mathrm{SD}^{\geq D'}_{\leq d'} \in \mathsf{pr\text{-}NP/poly}$. Let $(A, \{a_N\}_{N>0})$ be an nondeterministic, non-uniform polynomial-time algorithm and advice family solving $\mathrm{SD}^{\geq D'}_{\leq d'}$. Then by applying $(A, \{a_N\})$ to $\mathcal{R}(y)$, we obtain a nondeterministic, non-uniform polynomial-time algorithm for solving L. This shows $L \in \mathsf{NP/poly}$, proving item 1 of the Theorem.

Next, for item 2 of Theorem 7.1, if Eq. (11) holds for sufficiently large n, then $D(n)^2 - d(n) \geq \frac{1}{n^c}$. Arguing as in the previous case, we exhibit a nonuniform polynomial-time reduction from L to

$\mathrm{SD}^{\geq D'}_{\leq d'}$, where this time $D'(N)^2 - d'(N) \geq \frac{1}{\mathrm{poly}(N)}$. This problem lies in pr-SZK, by item 2 of Theorem 4.19. This also yields $L \in \mathsf{NP/poly} \cap \mathsf{coNP/poly}$, and completes the proof of Theorem 7.1. □

Proof of Claim 7.2. **(1.)** First, suppose $y \in L$. We will use the distinguishing interpretation of statistical distance (see Section 2.1) to argue that $||\mathcal{D}_C - \mathcal{D}_{C'}||_{\mathrm{stat}}$ is large. Suppose an unbiased coin $b \sim \mathcal{U}_{\{0,1\}}$ is flipped, unseen by us, and we receive a sample $z \sim \mathcal{D}_C$ if $b = 0$, or $z \sim \mathcal{D}_{C'}$ if $b = 1$. Consider the distinguisher that outputs the guess $\tilde{b} := 0$ if $z \in \overline{L'}$, or $\tilde{b} := 1$ if $z \in L'$.

We lower-bound the success probability $\Pr[\tilde{b} = b]$ as follows. Say we condition on $[b = 0]$, so that $z \sim \mathcal{D}_C$. The distributions $\mathcal{U}_{K_a}$ are supported on $\overline{L}_n$, so in the execution of C we get $\overline{x} \in \left(\overline{L}_n\right)^{t_1(n)}$. Then it follows from the OR-compression property of R for L that $\Pr[z \in \overline{L'}] \geq 1 - \xi(n)$. On the other hand, suppose we condition on $[b = 1]$, so that $z \sim \mathcal{D}_{C'}$. In an execution of C' the input tuple $\overline{x}$ contains $y \in L_n$; thus, by the OR-compression property of R, we have $\Pr[z \in L'] \geq 1 - \xi(n)$. So regardless of the value of b, our distinguisher succeeds with probability $\geq 1 - \xi(n)$. Thus, $1 - \xi(n) \leq \frac{1}{2}(1 + ||\mathcal{D}_C - \mathcal{D}_{C'}||_{\mathrm{stat}})$. This proves item 1.

(2.) Now suppose $y \notin L$; we must upper-bound $||\mathcal{D}_C - \mathcal{D}_{C'}||_{\mathrm{stat}}$. Consider the distinguishing experiment between C and C' as in item 1. If we regard the random variables a and j^* (the latter used only by C') to be part of the joint probability space of both algorithms (noting that a is identically distributed in the two circuits), then revealing the values a, j^* along with z to the distinguisher cannot decrease the distinguisher's maximum achievable success probability. Now conditioned on revealed values a, j^*, the maximum achievable success probability in the modified distinguishing experiment is

$$\frac{1}{2}\left(1 + \left|\left|R\left(\mathcal{U}_{K_a}^{\otimes t_1(n)}\right) - R\left(\mathcal{U}_{K_a}^{\otimes (j^*-1)},\ y,\ \mathcal{U}_{K_a}^{\otimes (t_1(n)-j^*)}\right)\right|\right|_{\mathrm{stat}}\right) ,$$

from which we conclude that

$$||\mathcal{D}_C - \mathcal{D}_{C'}||_{\mathrm{stat}} \ \leq \ \mathop{\mathbb{E}}_{a \sim \mathcal{U}_{[s]}, j^* \sim \mathcal{U}_{[t_1(n)]}} \left[\ \left|\left|R\left(\mathcal{U}_{K_a}^{\otimes t_1(n)}\right) - R\left(\mathcal{U}_{K_a}^{\otimes (j^*-1)},\ y,\ \mathcal{U}_{K_a}^{\otimes (t_1(n)-j^*)}\right)\right|\right|_{\mathrm{stat}}\ \right] . \quad (13)$$

By our choice of $K_1, \ldots, K_s$ and Lemma 6.6, the right-hand side of Eq. (13) is at most

$$\widehat{\delta} + 2t_1(n)/(d+1) + \varepsilon \ < \ \widehat{\delta} + 2 \cdot \frac{1}{4n^c} , \quad (14)$$

by our settings to d, ε. This proves Eq. (12) and completes the proof of Claim 7.2. □

The next result gives a useful consequence of Theorem 7.1 for the case where the compression bound $t_2(n)$ is on the order of $t_1(n) \cdot \log_2(t_1(n))$, and also points out a strengthening of the result's conclusion in the case of error-free compression.

Theorem 7.3. *Let L be any language. Suppose $t_1(n), t_2(n) : \mathbb{N}^+ \to \mathbb{N}^+$ satisfy $t_2(n) \leq C \cdot t_1(n) \log t_1(n)$ and $t_1(n) \leq n^{C'}$, for some $C, C' > 0$. Suppose that R is a PPT-OR-compression reduction $R(x^1, \ldots, x^{t_1(n)}) : \{0,1\}^{t_1(n) \times n} \to \{0,1\}^{\leq t_2(n)}$ for L with parameters $t_1(n), t_2(n)$, error bound $\xi(n) < .5$, and some target language L'.*

1. *If $\xi(n) < n^{-C \cdot C'}/32$, then there is a non-uniform polynomial-time many-to-one reduction from L to a promise problem in* pr-SZK.

2. *Suppose further that R is error-free (i.e., $\xi(n) = 0$). Then, there is a non-uniform polynomial-time many-to-one reduction from L to a promise problem in* pr-PZK.

Proof. **(1.)** We bound the quantity $\widehat{\delta}$ from Theorem 7.1:

$$\begin{aligned}\widehat{\delta} &\leq 1 - 2^{-\frac{t_2(n)}{t_1(n)}-3} \\ &\leq 1 - 2^{-C\log_2(t_1(n))}/8 \\ &\leq 1 - t_1(n)^{-C}/8 \\ &\leq 1 - n^{-C\cdot C'}/8 \ .\end{aligned}$$

If $\xi(n) < n^{-C\cdot C'}/32$, then the left-hand quantity in Eq. (11) is $\geq \frac{1}{\mathrm{poly}(n)}$, and the desired conclusion then follows from Theorem 7.1, item 2.

(2.) Looking into the proof of Theorem 7.1, item 2, we see that it gives a non-uniform polynomial-time many-to-one reduction from L to $\mathrm{SD}^{\geq D'(N)}_{\leq d'(N)}$, where in the current case, using Claim 7.2, we have

$$D'(N) = 1, \quad d'(N) \leq 1 - \frac{1}{\mathrm{poly}(N)} \ .$$

This problem can in turn be uniformly many-to-one reduced to $\mathrm{SD}^{\geq 1}_{\leq .5}$ by mapping a circuit-distribution pair $\langle C, C'\rangle$ to $\langle C^{\otimes T}, (C')^{\otimes T}\rangle$, where $C^{\otimes T}$ is the circuit that outputs T samples drawn independently from C, and where $T \leq \mathrm{poly}(n)$ is chosen suitably large. Finally, $\mathrm{SD}^{\geq 1}_{\leq .5} \in$ pr-PZK by Theorem 4.18. □

7.2 Application to AND- and OR-compression of NP-complete languages

Throughout this section, for parameters $t_1(n), t_2(n)$, we will use the shorthand

$$\widehat{\delta} := \min\left\{\sqrt{\frac{\ln 2}{2}\cdot\frac{t_2(n)+1}{t_1(n)}},\ 1 - 2^{-\frac{t_2(n)}{t_1(n)}-3}\right\}$$

Here is our first main result giving evidence against efficient AND-compression for NP-complete languages:

Theorem 7.4. *Suppose that for some* NP*-complete language L, any target language L', and an error bound $\xi(n) < .5$, L has a PPT-*AND*-compression reduction R with target language L', with parameters $t_1(n), t_2(n) : \mathbb{N}^+ \to \mathbb{N}^+$ and error bound $\xi(n) < .5$.*

1. *If*

$$1 - 2\xi(n) - \widehat{\delta} \geq \frac{1}{\mathrm{poly}(n)} \ , \tag{15}$$

then NP $\subseteq$ coNP/poly *and* PH $= \Sigma^p_3 = \Pi^p_3$.

2. *If we have the bound*

$$(1 - 2\xi(n))^2 - \widehat{\delta} \geq \frac{1}{\mathrm{poly}(n)} \ , \tag{16}$$

then L (and every other language in NP*) is many-to-one reducible in non-uniform polynomial time to a problem in* pr-SZK*, and* NP $\subseteq$ coNP/poly.

3. *The conclusion of item 2 holds if* $t_2(n) \leq C \cdot t_1(n)\log(t_1(n)))$ *and if* $\xi(n)$ *is a sufficiently small inverse-polynomial function of* n *(determined by* t_1 *and the constant* C*).*

Item 3 above establishes the assertion of Theorem 1.3 from the Introduction for the case of AND-compression.

Proof of Theorem 7.4. **(1.)** The reduction R is also a PPT-OR-compression for $\overline{L}$, with target language $\overline{L'}$, and with the same parameters.

If Eq. (15) holds in case 1, we apply item 1 of Theorem 7.1 to $\overline{L}$, concluding that $\overline{L} \in \mathsf{NP/poly}$, i.e., $L \in \mathsf{coNP/poly}$. The consequence for PH is from Theorem 4.9.

(2.) Similarly, if Eq. (16) holds in case 2, we apply item 2 of of Theorem 7.1 to $\overline{L}$, giving a non-uniform many-to-one reduction from $\overline{L}$ to a problem $\Pi = (\Pi_Y, \Pi_N) \in \mathsf{pr\text{-}SZK}$. This is also a reduction from L to (Π_N, Π_Y), which by Theorem 4.14 also lies in $\mathsf{pr\text{-}SZK}$. The extension to other languages in NP follows from the NP-completeness of L.

(3.) In this case we apply Theorem 7.3, item 1 to $\overline{L}$. □

The next theorem gives evidence for the infeasability of efficient OR-compression for NP-complete languages.

Theorem 7.5. *1. Suppose that for some* NP*-complete language* L*, any target language* L'*, and an error bound* $\xi(n) < .5$*,* L *has a PPT-*OR*-compression reduction* R *with target language* L'*, with parameters* $t_1(n), t_2(n) : \mathbb{N}^+ \to \mathbb{N}^+$ *and error bound* $\xi(n) < .5$*. If*

$$(1 - 2\xi(n))^2 - \widehat{\delta} \;\geq\; \frac{1}{\mathrm{poly}(n)}\ , \tag{17}$$

then L *(and every other language in* NP*) is reducible in non-uniform polynomial time to a problem in* $\mathsf{pr\text{-}SZK}$*, and* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$*.*

2. *The conclusion of item 1 holds if* $t_2(n) \leq C \cdot t_1(n)\log(t_1(n)))$ *and if* $\xi(n)$ *is a sufficiently small inverse-polynomial function of* n *(determined by* t_1 *and* C*).*

Item 2 completes the proof of Theorem 1.3 from the Introduction.

Proof of Theorem 7.5. **(1.)** This time, if Eq. (17) holds, we just apply item 2 of Theorem 7.1 to L itself.

(2.) In this case we apply item 1 of Theorem 7.3 to L. □

7.3 On f-compression for combining functions of high block sensitivity

As discussed in Section 1.3.1, our results on AND-compression, combined with ideas of [FS11, Section 7], directly imply some limitations to efficient strong f-compression of SAT or other NP-complete languages, for many other combining functions f. In this section we give the approach suggested by [FS11], that applies to non-monotone functions and functions with high *block sensitivity* (defined below). In Section 9 we describe a new approach that provides evidence against strong f-compression of SAT, for many functions f with low block sensitivity.

Definition 7.5.1 (Sensitive blocks and block sensitivity [Nis91])**.** *Let f_m be a Boolean function on m variables. For $y \in \{0,1\}^m$ and a subset $B \subseteq [m]$ of input variables (called a "block"), we let $y^{(B)}$ denote the string obtained by starting with y and flipping all bits in B. Say that B is* sensitive *for f on input y if $f(y^{(B)}) \neq f(y)$. We say that B is a* minimal *sensitive block (with respect to f, y) if it contains no proper subset which is sensitive.*

Define the block sensitivity *of f_m with respect to input $y \in \{0,1\}^m$, as the maximal size k of any collection $B_1, \ldots, B_k$ of pairwise-disjoint blocks which are each sensitive for f on y. Define the block sensitivity of f_m as $bs(f_m) := \max_y bs(f_m; y)$.*

We also use the classical notion of Boolean certificate complexity.

Definition 7.5.2 (Certificate complexity [VW85])**.** *Let f_m be a Boolean function on m variables. For $y \in \{0,1\}^m$ with $f(y) = b \in \{0,1\}$, and a subset $W \subseteq [m]$ of input variables, we say that W is a* certificate *for y if every y' agreeing with y on the variables in W also satisfies $f(y') = b$.*

Define the certificate complexity *of f_m with respect to y, denoted $C(f_m; y)$, as the minimal size of any certificate W for y. Define the certificate complexity of f_m as $C(f_m) := \max_y C(f_m; y)$.*

Fact 7.5.4 ([Nis91])**.** *For any f, $bs(f) \leq C(f) \leq bs(f)^2$. For monotone f, we have $bs(f) = C(f)$.*

See [BdW02] for more information on these complexity measures. The following class of Boolean functions will be of interest to us:

Definition 7.5.3. *Let $\delta \in (0,1]$. Say that $f : \{0,1\}^* \to \{0,1\}$ is* δ-amenable *if for each input length $m \geq 1$, at least one of the following conditions hold on the restriction f_m of f to inputs of length m:*

1. *f_m is non-monotone;*
2. *$bs(f_m) \geq m^\delta$.*

We prove:

Theorem 7.6. *Suppose $f : \{0,1\}^* \to \{0,1\}$ is δ-amenable for some $\delta > 0$, and suppose that R is a PPT f-compression reduction for an NP-complete language L with a target language $L' \in \mathsf{NP}$, where for some integer $c > 0$, R has the parameters*

$$t_1(n) \;=\; n^{\lceil 2c/\delta \rceil}\,, \qquad t_2(n) \;\leq\; n^c\,, \qquad \xi(n) \;\leq\; .01\,.$$

Then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

As will be clear from the proof, the assumption $L' \in \mathsf{NP}$ is only really needed to handle the case where f is non-monotone but has low sensitivity. If item 2 in the definition of δ-amenability holds for all m, then L' can be arbitrary. Most natural functions are $\Omega(1)$-amenable, including all (non-constant) graph properties and other transitively invariant functions, as well as functions defined by read-once De Morgan formulas. There are, however, monotone Boolean functions depending on all inputs for which $bs(f_m) = O(\log m)$, as described in Section 9.1.

Proof sketch. The f-compression reduction R maps inputs $(x^1, \ldots, x^t) \in \{0,1\}^{t_1(n) \times n}$ to an output z of length at most n^c; for any $(x^1, \ldots, x^t)$, the equality

$$L'(z) \;=\; f(L(x^1), \ldots, L(x^t))$$

holds with probability at least .99 over the randomness in R.

Fix an input length n. We may assume that $L_n, \overline{L}_n$ are both nonempty, otherwise it is trivial to give a small circuit to define L_n. Let $N := t_1(n)$. If f is non-monotone on inputs of size N, say with respect to the first coordinate, then we can non-uniformly fix some $N-1$ strings $x^2, \ldots, x^N \in \{0,1\}^n$ such that for $x \in \{0,1\}^n$, with probability at least .99 over $z = R(x, x^2, \ldots, x^N)$ we have

$$L'(z) \;=\; \neg L(x) \;.$$

This gives a (probabilistic, many-to-one) reduction from the decision problem for $\overline{L}_n$ to the decision problem for $L' \in \mathsf{NP}$. By the non-uniform derandomization technique of Theorem 4.11, it follows that $\overline{L}_n$ has a nondeterministic circuit of size $\leq \mathrm{poly}(n)$.

Otherwise, f is monotone for length-N inputs. Then by the amenability property of f, there is a $y \in \{0,1\}^N$ and a collection $B_1, \ldots, B_K \subseteq [N]$ of disjoint sensitive blocks for f on input y, with $K \geq N^\delta$. Without loss of generality each B_ℓ may be chosen as a *minimal* sensitive block.

Assume first that $f_N(y) = 0$, so that (by the monotonicity of f and the minimality of the sets B_ℓ) we have $y_i = 0$ for each $i \in \bigcup_{\ell \in [K]} B_\ell$. Suppose we are given input strings $(u^1, \ldots, u^K) \in \{0,1\}^{K \times n}$. We will use these to define an N-tuple $(x^1, \ldots, x^N) \in \{0,1\}^{N \times n}$ to feed to R. Let x^+, x^- be any two fixed elements of $L_n, \overline{L}_n$ respectively. For $i \in B_\ell$, let $x^i := u^\ell$. For $i \in [N] \setminus \bigcup_{j \in [K]} B_j$, let us fix the strings $x^i := x^+$ if $y_i = 1$, otherwise $x^i := x^-$.

Observe that if there is an $\ell \in [K]$ for which $u^\ell \in L$, then the vector $(L(x^1), \ldots, L(x^N))$ dominates the vector y^{B_j}, so that $f(L(x^1), \ldots, L(x^N)) = 1$; otherwise, we have $(L(x^1), \ldots, L(x^N)) = y$, and $f(L(x^1), \ldots, L(x^N)) = 0$. The "$f$-preserving" guarantee of R then implies that

$$L'(z) \;=\; \bigvee_{\ell \in [K]} L(u^\ell)$$

with probability at least .99 over $z = R(x^1, x^2, \ldots, x^N)$. Thus for this fixed input length n, we obtain an OR-compression reduction from L to target L' with $t'_1(n) \geq N^\delta \geq n^{2c}$, $t'_2(n) \leq n^c$, and $\xi'(n) \leq .01$. Similarly, if $f(y) = 1$, then we can obtain an AND-compression reduction for strings of input-length n, with parameters as above. In either case, we can apply the techniques of Section 7.2 to get a non-uniform proof system for membership in $\overline{L}_n$. Combining our work for each input length, we find that $\overline{L} \in \mathsf{NP/poly}$. As L is NP-complete, the Theorem is proved. □

Say that a function $f : \{0,1\}^* \to \{0,1\}$ is *δ-anti-amenable* if on each input length m, it is monotone, non-constant, and satisfies $bs(f_m) \leq m^\delta$. Using a simple idea we will show that for such functions, and for any language L, there *is* a non-trivial polynomial-time f-compression reduction for L, if we allow ourselves to use *nondeterminism* in the compression reduction. We establish this fact in the hope that it may prove useful in future work. We use the following definition (a closely related notion is studied in [DvM10]):

Definition 7.6.1 (Nondeterministic f-compression reductions)**.** *Let L, L' be two languages, and let $f : \{0,1\}^* \to \{0,1\}$ be a Boolean function. Let $t_1(n), t_2(n), \ell(n) : \mathbb{N}^+ \to \mathbb{N}^+$ each be $\leq \mathrm{poly}(n)$ and computable in time $\mathrm{poly}(n)$. A* nondeterministic f-compression reduction for L*, with parameters $(t_1(n), t_2(n), \ell(n))$ and target language L', is defined by a polynomial-time (deterministic) mapping $R(x^1, \ldots, x^m, y)$ outputting a string z, such that for all $(x^1, \ldots, x^{t_1(n)}) \in \{0,1\}^{t_1(n) \times n}$,*

1. $f\left(L(x^1), \ldots, L(x^{t_1(n)})\right) = 1 \iff \exists y \in \{0,1\}^{\ell(n)} : z = R(x^1, \ldots, x^{t_1(n)}, y) \in L'$;

2. For all settings to $y \in \{0,1\}^{\ell(n)}$ we have $|z| \leq t_2(n)$.

Theorem 7.6.2. *Suppose $f : \{0,1\}^* \to \{0,1\}$ is δ-anti-amenable for some $\delta < 1$. Then for any integer $C > 0$, there exists a nondeterministic f-compression reduction R for L, where the mapping R has parameters*

$$t_1(n) \;=\; n^C \;, \qquad t_2(n) \;\leq\; O(n^{\delta C+1}) \;, \qquad \ell(n) \;=\; t_1(n) \;.$$

Proof. For $(x^1, \ldots, x^{t_1(n)}, y) \in \{0,1\}^{t_1(n) \times n + \ell(n)}$, let $R(x^1, \ldots, x^{t_1(n)}, y)$ simply output the string

$$z \;:=\; \langle (x^{i_1}, i_1), \ldots, (x^{i_p}, i_p), 1^n \rangle \;,$$

where $1 \leq i_1 < i_2 < \ldots < i_p \leq t_1(n)$ are the first $\lfloor n^{\delta C} \rfloor$ indices $i \in t_1(n)$ for which $y_i = 1$. (If there are fewer than $\lfloor n^{\delta C} \rfloor$ such indices, R simply outputs the entire list as above.) This R is clearly polynomial-time computable, and the output length is of size $O(n^{\delta C+1})$.

Next we define our target language. For $m > 0$ and $S \subseteq [m]$, define $y^S \in \{0,1\}^m$ by $y^S_i := 1$ iff $i \in S$. Say that S is *1-forcing* for f_m if $f_m(y^S) = 1$. (f_m is monotone for each m, so this also implies that $f_m(y^{S'}) = 1$ for any $S' \supseteq S$.) Define

$$L' \;:=\; \left\{ \langle (x^{i_1}, i_1), \ldots, (x^{i_p}, i_p), 1^n \rangle : x^{i_1}, \ldots, x^{i_p} \text{ are each in } L_n \text{ and } \{i_1, \ldots, i_p\} \text{ is 1-forcing for } f_{n^C} \right\} .$$

Now we prove correctness. First suppose $f\left(L(x^1), \ldots, L(x^{t_1(n)})\right) = 0$. Then for any set $S \subseteq [m]$ of indices which are 1-forcing, we must have $L(x_i) = 0$ for some $i \in S$, so the output of R cannot lie in L' for any $y \in \{0,1\}^{n^C}$. On the other hand, suppose $f\left(L(x^1), \ldots, L(x^{t_1(n)})\right) = 1$. Let $S_0 \subseteq [t_1(n)]$ be the set of indices i for which $x^i \in L$. As f_m is monotone, the block sensitivity $bs(f_m)$ is equal to the certificate complexity $C(f_m)$. Thus the latter measure is at most m^δ. So for our setting to $x^1, \ldots, x^{t(n)}$, there exists a subset $S \subseteq S_0$, with $|S| \leq (n^C)^\delta$, which is 1-forcing for f_{n^C}. If $S = \{i_1, \ldots, i_p\}$, we have $\langle (x^{i_1}, i_1), \ldots, (x^{i_p}, i_p), 1^n \rangle \in L'$, and this string can be output by R if we take $y := y^S$. This proves correctness according to our definition. □

7.4 Limits to strong compression for parametrized problems

Next, we use Theorem 7.1 to give evidence against strong compressibility for "expressive" parametrized problems. The result we give below is a simple-to-state, representative example; the quantitative settings studied here are not the only interesting ones our techniques can handle.

Theorem 7.7. *Say that P is* OR*-expressive or* AND*-expressive, e.g., one of the problems listed in Theorems 5.5 and 5.6. Suppose additionally that P is strongly PPT-compressible*[33] *with error bound $\xi(m,k,w)$ satisfying $\xi(m,k,1) \leq .5 - k^{-O(1)}$ (independent of m), i.e., with success probability $\geq .5 + k^{-O(1)}$. Then, every language in* NP *is many-to-one reducible in non-uniform polynomial time to a problem in* pr-SZK *(and* NP $\subseteq$ coNP/poly*).*

Theorem 1.2 from the Introduction follows, by considering the special cases $P = \mathrm{OR}(\mathrm{SAT})$ and $P = \mathrm{AND}(\mathrm{SAT})$.

[33] (as in Definition 5.8)

Proof of Theorem 7.7. Suppose first that P is OR-expressive, with respect to the NP-complete language L and with some parameter $S(n) \leq \text{poly}(n)$. We apply item 1 of Lemma 5.10 to L and the assumed strong compression reduction for P. Using some function $T(n) \leq \text{poly}(n)$ to be determined, and with $w(n) := 1$, we obtain a PPT-OR-compression for L with parameters

$$t_1(n) \;=\; T(n), \quad t_2(n) \;\leq\; S(n) \cdot n^{O(1)}, \quad \xi'(n) \;\leq\; 2^{-n} \;.$$

(Here, the bound on t_2 is independent of the choice of $T(n)$.) We evaluate

$$(1 - 2\xi'(n))^2 - \sqrt{\frac{\ln 2}{2} \cdot \frac{t_2(n)+1}{t_1(n)}} \;\;\geq\;\; (1 - 4 \cdot 2^{-n}) - \sqrt{\frac{\ln 2}{2} \cdot \frac{S(n) \cdot n^{O(1)} + 1}{T(n)}} \;,$$

for some $a > 0$ (using $S(n) \leq \text{poly}(n)$). The expression above can be made greater than .5 for large n by choosing a sufficiently fast-growing $T(n) \leq \text{poly}(n)$. Under such a setting, Eq (17) holds for $(t_1(n), t_2(n), \xi'(n))$. We can then apply the first assertion of Theorem 7.5, item 1 to our PPT-OR-compression for L, which yields the desired conclusion.

The case where P is AND-expressive is handled analogously; in this case we apply Lemma 5.10, item 2 and the first assertion of Theorem 7.4, item 2. □

We can also apply Theorem 7.3 to show that, if any NP-complete language L is PPT-OR-compressible by an error-free reduction with $t_2(n) = O(t_1(n) \log(t_1(n)))$, then NP has non-uniform *perfect* zero-knowledge proofs. From a deterministic AND-compression reduction for L of this type, we get non-uniform perfect zero-knowledge proofs for coNP. (Note that unlike pr-SZK, pr-PZK is not known to be closed under complement.)

7.5 Application to problems with polynomial kernelizations

In this section we prove new limits to efficient compression for the Satisfiability problem on d-CNFs, and for some problems on graphs and hypergraphs, partially extending results of Dell and Van Melkebeek [DvM10] to handle two-sided error. First, we need some background.

Definition 7.8 (Hypergraphs, vertex covers, and cliques)**.** *For any integer $d \geq 2$, a* d-uniform hypergraph*, or* d-hypergraph*, is a set H of size-d subsets of a vertex set $V = [N]$. A* vertex cover *in a d-uniform hypergraph H is a subset of vertices that intersects all hyperedges in H. A subset $V' \subseteq V$ is a* clique *in H if every size-d subset of V' is a member of H.*

Clearly H has a vertex cover of size s exactly if the "complement" hypergraph $\overline{H} := \{e : |e| = d \;\wedge\; e \notin H\}$ contains a clique of size $N - s$.

Definition 7.9. *Define the parametrized problems*

$$d\text{-Vertex Cover} \;:=\; \{\langle (H, s), 1^N \rangle : \; H \text{ is a } d\text{-hypergraph on } [N] \text{ and contains a vertex cover of size } s\} \;,$$

$$d\text{-Clique} \;:=\; \{\langle (H, s), 1^N \rangle : \; H \text{ is a } d\text{-hypergraph on } [N] \text{ and contains a clique of size } s\} \;.^{34}$$

Also define the parametrized d-CNF Satisfiability problem

$$d\text{-SAT}_{par} \;:=\; \{\langle \psi, 1^N \rangle : \; \psi \text{ is a satisfiable } d\text{-CNF on } N \text{ variables}\} \;.$$

[34] This is a different parametrized problem than the two clique-based problems mentioned in Section 5.2.

We will prove new limits on efficient compression for these problems with the help of the following powerful, ingenious reduction of Dell and Van Melkebeek.

Theorem 7.10 ([DvM10], Lemma 2)**.** *Fix $d \geq 2$, and let $T(n) : \mathbb{N}^+ \to \mathbb{N}^+$ be polynomially bounded. There is a deterministic polynomial-time OR-compression reduction*[35] *R^* for $L =$ 3-SAT,*[36] *with target language $L' = d$-Clique. For the first parameter we have $t_1(n) = T(n)$. The d-Clique instance $\langle (H, s), 1^N \rangle$ output by R^* satisfies*

$$N \;=\; O\left(n \cdot \max\left(n, T(n)^{1/d+o(1)}\right)\right) \;.$$

By straightforwardly combining Theorem 7.10 with our Theorem 7.4, we will prove the following theorem:

Theorem 7.11. *Let $d \geq 2, \varepsilon > 0$ be given. There is a $\beta = \beta(d, \varepsilon) > 0$ for which the following holds. Suppose that d-Clique has a polynomial-time compression reduction with output-size bound $O(N^{d-\varepsilon})$ and success probability $.5 + N^{-\beta}$; that is (in the terms of Definition 5.7), suppose that d-Clique is PPT-compressible with parameters c, ξ satisfying*

$$c(M, N, 1) \;\leq\; O(N^{d-\varepsilon}), \quad \xi(M, N, 1) \;\leq .5 - N^{-\beta} \;,$$

with any target language L'.

Then, every language in NP *is many-to-one reducible in non-uniform polynomial time to a problem in* pr-SZK *(and* NP $\subseteq$ coNP/poly*).*

The same result holds if we replace d-Clique with d-Vertex Cover or d-SAT$_{par}$.

Theorem 7.11 gives a version of [DvM10, Theorems 1 and 2] that applies to probabilistic reductions with two-sided error. However, our result does not apply to the more general setting of *oracle communication protocols*, to which those earlier results do apply (for co-nondeterministic protocols, and protocols avoiding false negatives).

Dell and Van Melkebeek use their techniques to show compression lower bounds for several other interesting graph problems (including the Feedback Vertex Set, Bounded-Degree Deletion, and Non-Planar Deletion problems) via reductions from 2-Vertex Cover [DvM10, Section 5.2]. Using our results and the reductions in [DvM10], one can also obtain similarly strong compression lower bounds for these problems for the two-sided error setting.

Proof of Theorem 7.11. We already described a simple reduction (in both directions) between the d-Vertex Cover and d-Clique problems that preserves the parameter N. Also, an instance of d-Vertex Cover on N vertices is efficiently reducible to a d-SAT instance over $O(N)$ variables [DvM10, Lemma 5]. Thus, it suffices to prove the result for d-Clique.

Let R be the compression reduction assumed to exist for d-Clique, with the value $\beta > 0$ to be determined later. Let $C > d$ be a large integer value, also to be determined.

We will define an OR-compression reduction R' for $L =$ 3-SAT and target language L' from our assumption; this will allow us to apply Theorem 7.5. R' works as follows. We let $t_1(n) := n^C$. On input formulas $\psi_1, \ldots, \psi_{n^C}$, each of bit-length n, the reduction first computes $\langle (H, s), 1^N \rangle := R^*(\psi_1, \ldots, \psi_{n^C})$, where R^* is as in Theorem 7.10. Next, R' outputs the value $z := R(\langle (H, s), 1^N \rangle)$.

[35] (as in Definition 4.20)

[36] Here 3-SAT is just the usual language $\{\langle \psi \rangle : \psi \text{ is a satisfiable 3-CNF}\}$.

R' is clearly polynomial-time computable. To analyze R', fix length-n formulas $\psi_1, \ldots, \psi_{n^C}$, and let

$$b \ := \ \bigvee_{j=1}^{n^C} [\psi_j \in \text{3-SAT}] \ .$$

By the OR-compression property of the deterministic mapping R^*, we have

$$[b = 1] \iff \langle (H, s), 1^N \rangle \in d\text{-Clique} \ .$$

Then by the assumed reliability guarantee of R,

$$\begin{aligned} \Pr[L'(z) = b] &\geq .5 + N^{-\beta} \\ &\geq .5 + \left(O\left(n \cdot \max\left(n, n^{C/d+o(1)} \right) \right) \right)^{-\beta} \\ &\geq .5 + n^{-\beta(1+C/d)+o(1)} \ . \end{aligned}$$

Thus the error bound $\xi(n)$ of our reduction R' is at most $.5 - n^{-\beta(1+C/d)+o(1)}$. Also, by the compression guarantee of R, the output z satisfies

$$\begin{aligned} |z| &\leq O(N^{d-\varepsilon}) \\ &\leq O\left(\left(n^{1+C/d+o(1)} \right)^{d-\varepsilon} \right) \\ &\leq O\left(n^{C-1+o(1)} \right) \ , \end{aligned}$$

with the last step valid provided we take $C > d(d+1)/\varepsilon$. Thus as an output-size bound for R', we may take $t_2(n) = O\left(n^{C-1+o(1)}\right)$. We evaluate

$$\begin{aligned} (1 - 2\xi(n))^2 - \sqrt{\frac{\ln 2}{2} \cdot \frac{t_2(n)+1}{t_1(n)}} &\geq 4n^{-2(\beta(1+C/d)-o(1))} - O(n^{-.5+o(1)}) \\ &\geq n^{-\Omega(1)} \ , \end{aligned}$$

provided we take $\beta < .25(1+C/d)^{-1}$. Thus under these settings, Eq. (17) holds. Then Theorem 7.5, item 1 gives the desired conclusion, since $L =$ 3-SAT is NP-complete. □

8 Extension to quantum compression

In this section we will show that our results on OR- and AND-compression have analogues for the model in which the compression scheme is allowed to be a quantum algorithm, outputting a quantum state.

We assume familiarity with the basics of quantum computing and quantum information (for the needed background, consult [NC00]). However, readers without this background should be able to follow the overall structure of the argument if they are willing to regard "qubits," "quantum operations" "quantum algorithms," and "quantum measurements" as certain types of black-box objects, and accept some known facts about them. In particular, a "mixed state on m qubits" is a "quantum superposition" over classical m-bit strings. Let

$$\mathsf{MS}_m$$

denote the collection of m-qubit mixed states. (MS_m can be identified with the set of 2^m-by-2^m, trace-1, positive-semidefinite complex matrices.)

A "quantum operation" is a certain type of mapping $OP : \mathsf{MS}_m \to \mathsf{MS}_{m'}$, for some $m, m' > 0$. (The operations allowed by quantum physics are the *completely positive, trace-preserving (CPTP) maps*; these are a subset of the linear transformations mapping $\mathsf{MS}_m \subset \mathbb{C}^{m \times m}$ into $\mathsf{MS}_{m'} \subset \mathbb{C}^{m' \times m'}$.) We let

$$\mathsf{OP}_{m,m'}$$

denote the valid quantum operations from m-qubit into m'-qubit states.

"Quantum measurements" are measurements performed on quantum states to yield information about these states; in the quantum setting, measurements are inherently probabilistic, and alter the states being measured. See [NC00, Chapter 2] for a formal definition. Quantum states turn out to inherit some of the information-theoretic limitations of their classical counterparts; this fact will be the basis for our results on quantum compression.

8.1 Trace distance and distinguishability of quantum states

The *trace distance* is a metric on mixed quantum states from a shared state space [NC00]; we denote the trace distance between $\rho, \rho' \in \mathsf{MS}_m$ by $||\rho - \rho'||_{\mathrm{tr}} \in [0, 1]$. Formally, treating ρ, ρ' as matrices,

$$||\rho - \rho'||_{\mathrm{tr}} \ := \ \frac{1}{2}\mathrm{Tr}\left[\sqrt{(\rho - \rho')^2}\right] \ .$$

This distance is intimately related to the distinguishability of quantum states. Suppose ρ, ρ' are two known states, and we are sent one or the other, each with equal probability (depending on the outcome of an unbiased coin flip $b \in \{0, 1\}$). We want to guess b, by applying some series of quantum operations and measurements. For any ρ, ρ', it is known [NC00, Theorem 9.1] that our success probability at this task is maximized by using a single binary measurement,[37] depending on ρ, ρ', and that our maximum achievable success probability equals

$$\frac{1}{2}\left(1 + ||\rho - \rho'||_{\mathrm{tr}}\right) \ .$$

A probability distribution over mixed states is again a mixed state. Thus for a distribution $\mathcal{D}$ over a finite universe U and a mapping $R : U \to \mathsf{MS}_m$, $R(\mathcal{D})$ defines a quantum state. We use the following standard claim concerning such states, which follows from the distinguishability characterization of $||\cdot||_{\mathrm{tr}}$:

Claim 8.1. *For any distributions $\mathcal{D}, \mathcal{D}'$ over a shared finite universe U, and any mapping $R : U \to \mathsf{MS}_m$, we have*

$$||R(\mathcal{D}) - R(\mathcal{D}')||_{\mathrm{tr}} \ \leq \ ||\mathcal{D} - \mathcal{D}'||_{\mathrm{stat}} \ .$$

Similarly, for any valid quantum operation $OP \in \mathsf{OP}_{m,m'}$ and states $\rho, \rho' \in \mathsf{MS}_m$, we have

$$||OP(\rho) - OP(\rho')||_{\mathrm{tr}} \ \leq \ ||\rho - \rho'||_{\mathrm{tr}} \ .$$

[37] (i.e., a measurement with two possible outcomes)

8.2 Quantum f-compression

The following notion of quantum compression is modeled on Definition 4.20. The definition is made slightly more complicated by the fact that we no longer have the notion of a "target language" for our reduction; instead, we will require that the answer to our original instance of the decision problem $f \circ L$ be recoverable by some quantum measurement performed on the output state. (This measurement need not be efficiently performable, however.)

Definition 8.2 (Quantum f-compression reductions)**.** *Let L be a language, and let $f : \{0,1\}^* \to \{0,1\}$ be a Boolean function. Let $t_1(n), t_2(n) : \mathbb{N}^+ \to \mathbb{N}^+$ and $\xi(n) : \mathbb{N}^+ \to [0,1]$ be given.*

A quantum f-compression reduction for L, *with parameters $t_1(n), t_2(n), \xi(n)$, is a mapping $R(x^1, \ldots, x^m)$ outputting a mixed state ρ. There must also exist a family of (not necessarily efficiently-performable) binary quantum measurements $\{\mathcal{M}_n\}_{n>0}$ on $t_2(n)$-qubit states. We require the following properties: for all $(x^1, \ldots, x^{t_1(n)}) \in \{0,1\}^{t_1(n) \times n}$,*

1. *The state $\rho = R(x^1, \ldots, x^{t_1(n)})$ is on $t_2(n)$ qubits;*
2. *We have*
$$\Pr\left[\mathcal{M}_n(\rho) \;=\; f\left(L(x^1), \ldots, L(x^{t_1(n)})\right)\right] \;\geq\; 1 - \xi(n)\ .$$

If some reduction R as above is computable in quantum polynomial time, we say that L is QPT-f-compressible *with parameters $(t_1(n), t_2(n), \xi(n))$.*

8.3 Quantum complexity classes

We will be using the class $\mathsf{QIP}[k]$ of languages definable by *k-message, quantum interative proof systems* [Wat03]. Our treatment of these proof systems will be informal, since all the technical properties we need are summarized in theorems from prior work (for details see [Wat03, Wat02]). These are proof systems in which a computationally-unbounded Prover exchanges quantum messages with a quantum polynomial-time Verifier; a total of $k = k(n)$ messages are exchanged. Verifier sends the first message if k is even, or Prover if k is odd, and the parties alternate thereafter. We take $\mathsf{QIP} := \bigcup_{c>0} \mathsf{QIP}[n^c]$.

It was shown in [Wat03, KW00] that for any $3 \leq k(n) \leq \mathrm{poly}(n)$, $\mathsf{PSPACE} \subseteq \mathsf{QIP}[k(n)] = \mathsf{QIP}[3]$; the latter class was recently shown to equal PSPACE [JJUW11]. Importantly for us, however, the class $\mathsf{QIP}[2]$ is not known to contain even coNP. The power of 3-message quantum proof systems is in contrast to the classical (private-coin) interactive-proof classes $\mathsf{IP}[k(n)]$, where for any constant $k \geq 2$, $\mathsf{IP}[k] = \mathsf{IP}[2] = \mathsf{AM}$, and the latter class is believed to be much weaker than $\mathsf{IP}[\mathrm{poly}(n)] = \mathsf{PSPACE}$.

In what follows, we will actually find it more convenient to work with the promise-problem classes $\mathsf{pr\text{-}QIP}[k]$.[38] The results we've summarized carry over to the promise setting as well.

A model of quantum statistical zero-knowledge proofs was proposed by Watrous [Wat02], and used to define the class QSZK of promise problems having polynomial-time proof systems of this type.[39] We will use $\mathsf{pr\text{-}QSZK}$ to denote this class. Watrous showed in [Wat02] that Sahai and Vadhan's "statistical distance characterization" of $\mathsf{pr\text{-}SZK}$, embodied in Definition 4.13, has a

[38] This is to avoid having to define non-uniform versions of these classes, just as we avoided defining non-uniform versions of AM and SZK.

[39] Watrous's original model was of *honest-verifier* quantum statistical zero-knowledge proof systems; he later showed that these proof systems are equivalent in power to "cheating-verifier" ones [Wat09].

quantum analogue. First, we need a promise problem involving trace distance. For a quantum circuit C with an m-qubit output register, let ρ_C denote the output state of C on some fixed input state (say, the all-zeros state). We consider circuits built from a fixed, finite "universal" gate-set (see [NC00, Chapter 4]).

Definition 8.3. *For parameters $0 \leq d \leq D \leq 1$, define the promise problem $\mathrm{TD}^{\geq D}_{\leq d} = (\Pi_Y, \Pi_N)$ as follows:*

$$\Pi_Y := \{\langle C, C' \rangle : ||\rho_C - \rho_{C'}||_{\mathrm{tr}} \geq D\} ,$$
$$\Pi_N := \{\langle C, C' \rangle : ||\rho_C - \rho_{C'}||_{\mathrm{tr}} \leq d\} .$$

In this definition, both $d = d(n)$ and $D = D(n)$ may be parameters depending on the input length $n = |\langle C, C' \rangle|$. (Here, the input description is a classical bit-string.)

Then, appealing to the result of [Wat02], we can use the following definition.

Definition 8.4. *Let* pr-QSZK *be defined as the class of promise problems for which there is a many-to-one (classical, deterministic) polynomial-time reduction from Π to $\mathrm{TD}^{\geq 2/3}_{\leq 1/3}$.*

Theorem 8.5 ([Wat02])**.** pr-QSZK *is closed under complement.*

Theorem 8.6 ([Wat02])**.** $\mathsf{pr\text{-}QSZK} \subseteq \mathsf{pr\text{-}QIP}[2] \cap \mathsf{pr\text{-}coQIP}[2]$.

For upper bounds on the complexity of $\mathrm{TD}^{\geq D(n)}_{\leq d(n)}$, we have the following two results, analogous to Theorems 4.15 and 4.14.

Theorem 8.7 (Follows from [Wat02])**.** *Suppose $0 \leq d = d(n) < D = D(n) \leq 1$ are polynomial-time computable, and satisfy $D > d + \frac{1}{\mathrm{poly}(n)}$. Then, $\mathrm{TD}^{\geq D}_{\leq d} \in \mathsf{pr\text{-}QIP}[2]$.*

If we drop the requirement that d, D be computable, but keep the gap requirement, then $\mathrm{TD}^{\geq D}_{\leq d}$ is many-to-one reducible in non-uniform (classical, deterministic) polynomial time to a problem in pr-QIP[2].

Theorem 8.7 follows from a "distinguishing protocol" analogous to that in Theorem 4.15.[40] Unlike the classical case, there is no known "non-uniform derandomization" result known for QIP[2] (or for other quantum classes). However, we do have a satisfying analogue of Theorem 4.14:

Theorem 8.8 (Follows from [Wat02])**.** *Suppose $0 \leq d = d(n) < D = D(n) \leq 1$ are polynomial-time computable, and satisfy $D^2 > d + \frac{1}{\mathrm{poly}(n)}$. Then, $\mathrm{TD}^{\geq D}_{\leq d} \in \mathsf{pr\text{-}QSZK}$.*

If we drop the requirement that d, D be computable, but keep the gap requirement, then $\mathrm{TD}^{\geq D}_{\leq d}$ is many-to-one reducible in non-uniform (classical, deterministic) polynomial time to a problem in pr-QSZK.

[40] In [Wat02] only the case where D, d are constants is studied, but the result extends easily to when they are functions of n. Also, the second case, where we merely have the gap requirement, is not explicitly analyzed, but follows by a trivial modification of the proof of [Wat02, Theorem 4].

8.4 Quantum distributional stability

We will use a quantum analogue of the distributional stability property:

Definition 8.9. *Let $t, t', n \in \mathbb{N}^+$. Given a mapping $F : \{0,1\}^{t \times n} \to \mathsf{MS}_{t'}$, and a collection $\mathcal{D}_1, \ldots, \mathcal{D}_T$ of mutually independent distributions over $\{0,1\}^n$, for $j \in [t]$ let*

$$\gamma_j \;:=\; \mathop{\mathbb{E}}_{y \sim \mathcal{D}_j} \left[\left|\left| F\left(\mathcal{D}_1, \ldots, \mathcal{D}_{j-1}, y, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_t\right) - F\left(\mathcal{D}_1, \ldots, \mathcal{D}_t\right) \right|\right|_{\mathrm{tr}} \right] .$$

For $\delta \in [0,1]$, say that F is δ-quantumly-distributionally stable (or δ-QDS) *with respect to $\mathcal{D}_1, \ldots, \mathcal{D}_t$ if*

$$\frac{1}{t} \sum_{j=1}^{t} \gamma_j \;\leq\; \delta \; .$$

The next lemma is analogous to Lemma 6.2.

Lemma 8.10. *Let $t, t', n \in \mathbb{N}^+$. Let $R : \{0,1\}^{t \times n} \to \mathsf{MS}_{t'}$ be given.*

Then, R is δ-QDS with respect to any input distributions $\mathcal{D}_1, \ldots, \mathcal{D}_t$, where we may take either of the bounds

1. $\delta := \sqrt{\frac{\ln 2}{2} \cdot \frac{t'}{t}}$;
2. $\delta := 1 - 2^{-\frac{t'}{t} - 2}$.

The slight improvement in the bounds comes from the fact that R outputs *exactly* t' qubits. The proof of Lemma 8.10 is very similar to that of Lemma 6.2, and is described in Appendix C.

8.5 Building quantum disguising distributions

Next we prove quantum analogues of our Disguising-Distribution Lemmas. First, we have the following analogue of Lemma 6.3:

Lemma 8.11. *Let $t, t' \in \mathbb{N}^+$, and let $F(x^1, \ldots, x^T) : \{0,1\}^{T \times n} \to \mathsf{MS}_{t'}$ be given. Suppose F is δ-QDS with respect to input distribution $\mathcal{D}^{\otimes T}$, for every distribution $\mathcal{D}$ over $\{0,1\}^n$.*

Fix some distribution $\mathcal{D}$ over $\{0,1\}^n$, and let $x^1, \ldots, x^d$ be independently sampled from $\mathcal{D}$. Let $k^* \sim \mathcal{U}_{[d]}$. Let $\widehat{\mathcal{D}}$ denote the distribution defined by sampling uniformly from the multiset $\{x^k\}_{k \neq k^*}$. Define

$$\beta_j \;:=\; \mathop{\mathbb{E}}_{k^*, x^1, \ldots, x^d} \left[\; \left|\left| R\left(\widehat{\mathcal{D}}^{\otimes(j-1)}, x^{k^*}, \widehat{\mathcal{D}}^{\otimes(T-j)}\right) \;-\; R\left(\widehat{\mathcal{D}}^{\otimes T}\right) \right|\right|_{\mathrm{tr}} \; \right] ,$$

where all the $\widehat{\mathcal{D}}$s are to be mutually independent (for fixed values of $x^1, \ldots, x^k$ and k^*). Then,

$$\frac{1}{T} \sum_{j=1}^{t} \beta_j \;\leq\; \delta + 2T/d \; .$$

Proof. The proof is identical to that of Lemma 6.3, except that we replace statistical distance with trace distance[41] and appeal to Claim 8.1 to argue that applying R does not increase trace distance between states. □

[41] (where appropriate—the input distributions we manipulate still are to be compared in statistical distance)

After establishing a quantum analogue of Lemma 6.4, we have:

Lemma 8.12. *Suppose F obeys the assumptions of Lemma 8.11. Let $S \subseteq \{0,1\}^n$, and fix $d > 0$. Given any $\varepsilon > 0$, let $s := \lceil (.5 \ln 2) n/\varepsilon^2 \rceil$. Then there exists a collection $K_1, \ldots, K_s$ of size-d multisets contained in S, such that for every $y \in S$ the following holds:*

$$\mathop{\mathbb{E}}_{a \sim \mathcal{U}_{[s]}, j^* \sim \mathcal{U}_{[t]}} \left[\left\| F\left(\mathcal{U}_{K_a}^{\otimes(j^*-1)}, y, \mathcal{U}_{K_a}^{\otimes(T-j^*)}\right) \; - \; F\left(\mathcal{U}_{K_a}^{\otimes T}\right) \right\|_{\mathrm{tr}} \right] \; \leq \; \delta + 2T/(d+1) + \varepsilon \; .$$

The proof is identical to that of Lemma 6.5, but again replacing statistical distance with trace distance. Then, by a proof analogous to that of Lemma 6.6, we obtain:

Lemma 8.13 (Quantum Disguising-Distribution Lemma). *Let $R(x^1, \ldots, x^t) : \{0,1\}^{t \times n} \to \mathsf{MS}_{t'}$ be any possibly-randomized mapping, where $n, t, t' \in \mathbb{N}^+$. Let $S \subseteq \{0,1\}^n$, and fix $d > 0$. Given any $\varepsilon > 0$, let $s := \lceil (.5 \ln 2) n/\varepsilon^2 \rceil$. Let*

$$\widehat{\delta} := \min \left\{ \sqrt{\frac{\ln 2}{2} \cdot \frac{t'}{t}}, \;\; 1 - 2^{-\frac{t'}{t} - 2} \right\} \; .$$

Then there exists a collection $K_1, \ldots, K_s$ of size-d multisets contained in S, such that for every $y \in S$, we have

$$\mathop{\mathbb{E}}_{a \sim \mathcal{U}_{[s]}, j^* \sim \mathcal{U}_{[t]}} \left[\left\| R\left(\mathcal{U}_{K_a}^{\otimes(j^*-1)}, \; y, \; \mathcal{U}_{K_a}^{\otimes(t-j^*)}\right) \; - \; R\left(\mathcal{U}_{K_a}^{\otimes t}\right) \right\|_{\mathrm{tr}} \right] \leq \; \widehat{\delta} + 2t/(d+1) + \varepsilon \; .$$

8.6 Complexity upper bounds from quantum compression schemes

Now we are ready to prove a quantum analogue of Theorem 7.1.

Theorem 8.14. *Let L be any language. Suppose there is a QPT-*OR*-compression reduction $R(x^1, \ldots, x^t) : \{0,1\}^{t_1(n) \times n} \to \mathsf{MS}_{t_2(n)}$ for L with (not necessarily computable) parameters $t_1(n), t_2(n) : \mathbb{N}^+ \to \mathbb{N}^+$, and with error bound $\xi(n) < .5$. Let*

$$\widehat{\delta} := \min \left\{ \sqrt{\frac{\ln 2}{2} \cdot \frac{t'}{t}}, \;\; 1 - 2^{-\frac{t'}{t} - 2} \right\} \; .$$

1. *If for some $c > 0$ we have*

$$(1 - 2\xi(n)) - \widehat{\delta} \; \geq \; \frac{1}{n^c} \; , \tag{18}$$

   then there is a non-uniform (classical, deterministic) polynomial-time many-to-one reduction from L to a problem in pr-QIP[2].

2. *If we have the stronger bound*

$$(1 - 2\xi(n))^2 - \widehat{\delta} \; \geq \; \frac{1}{n^c} \; , \tag{19}$$

   then L has a non-uniform (classical, deterministic) polynomial-time many-to-one reduction to a problem in pr-QSZK.

Proof. The proof is closely analogous to that of Theorem 7.1, except that our non-uniform reduction, on input y, outputs a description $\langle C, C' \rangle$ of a pair of *quantum* circuits. If $y \in L$, then $||\rho_C - \rho_{C'}||_{\text{tr}} \geq D(n) := 1 - 2\xi(n)$; while if $y \notin L$, we have $||\rho_C - \rho_{C'}||_{\text{tr}} \leq d(n) := \hat{\delta} + \frac{1}{2n^c}$. Applying Theorems 8.7 and 8.8 gives us the complexity upper bounds in items 1 and 2. □

Using Theorem 8.14, we can prove quantum versions of Theorems 7.4 and 7.5, giving evidence against efficient quantum OR- and AND-compression for NP-complete languages, under the assumption that such languages are not non-uniformly reducible to problems in pr-QIP[2], or alternatively, in pr-QSZK. A quantum analogue of Theorem 7.3, item 1 can be proved. We can also give an analogue of Theorem 7.7 regarding quantum compression for "expressive" parametrized problems. All of these quantum results treat compression reductions where the output state is of size determined by the various input parameters.

9 On f-compression for combining functions of low block sensitivity

In this section we define a class of Boolean functions called "eligible" functions; these are functions with low block sensitivity and obeying a few extra properties. In Theorem 9.2 we rule out strong f-compression for SAT for eligible combining functions f, under the *uniform* hardness assumption NP $\not\subseteq$ coAM. Note that this is a milder assumption than NP $\not\subseteq$ coNP/poly. As we have already given evidence (in Section 7.3) against strong f-compression for f with high block sensitivity, our results in the present section are complementary. Some functions f are not covered by either approach, but the techniques we have cover the "natural" examples of which we are aware.

In Section 9.6 we also use our result on eligible functions to rule out strong compression for each of OR(SAT) and AND(SAT) assuming NP $\not\subseteq$ coAM. We are able to show this even though OR, AND have maximal block sensitivity and are not themselves eligible. The connection stems from the fact that a monotone, eligible function f has monotone CNF and DNF representations of small clause width. We show that if these representations are efficiently computable and have not too many clauses (two conditions which hold for a suitable choice of f), then any hypothetical OR-compression or AND-compression reduction for SAT could be used to build an f-compression reduction for SAT.

In our analysis of instance compression for eligible combining functions, we will use the information-theoretic techniques behind the "distributional stability" lemma (Lemma 6.2). However, we manage to avoid using the Disguising-Distribution Lemma, whose applications seem to require non-uniform advice. A key additional source of inspiration is a work of Sivakumar [Siv99], who improved upon results of several earlier works [BKS95, Ogi95, AA94] and answered an open question of [BFT97]. Let $c > 0$ be any integer constant. Sivakumar showed the following: Suppose we assume that there is a polynomial algorithm which, given $h(n) := \lceil c \log_2 n \rceil$ SAT instances $\langle \psi^1 \rangle, \ldots, \langle \psi^{h(n)} \rangle$ each of length n, reliably eliminates *at least one* possible value for the vector

$$\left(\chi_{\text{SAT}}(\langle \psi^1 \rangle), \ldots, \chi_{\text{SAT}}(\langle \psi^{h(n)} \rangle) \right) \tag{20}$$

describing the satisfiability status of each ψ^i. (Such an algorithm is an example of a so-called *membership comparison* algorithm for SAT.) Then, there is also a polynomial-time algorithm to satisfy any uniquely-satisfiable Boolean formula (and so, by the Valiant-Vazirani reduction [VV86],

RP = NP). A key ingredient in Sivakumar's work is an algorithm of Ar, Lipton, Rubinfeld, and Sudan [ALRS98] for reconstructing polynomial functions from noisy data—an algorithm which played a major role in the development of the theory of list-decodable codes (see [Sud96, GS99]).

For comparison's sake, let us describe our approach to the study of f-compression reductions for SAT for eligible f; the points of similarity to [Siv99] will be apparent. We first prove[42] that for any eligible f, a strong f-compression reduction for SAT can be used to eliminate not just one value, but *most* possible values, for the characteristic vector in Eq. (20), given $\langle\psi^1\rangle, \ldots, \langle\psi^{h(n)}\rangle$ as above. More precisely, we eliminate all but $2^{\alpha \cdot h(n)}$ candidates for some small $\alpha > 0$.[43] There are two caveats: (i) we assume the *promise* that either none of the ψ^is are satisfiable, or exactly half of them are; (ii) our procedure to eliminate possible values requires a round of interaction with an untrusted Prover. To build such a protocol, we use the idea (which featured prominently in previous sections) of asking Prover to *distinguish* between various pairs of distributions. We use information-theoretic tools to show that most of the pairs of distributions we construct are statistically close, and we make a careful study of eligible functions to show that certain pairs of distributions are far apart. We also crucially use the fact that SZK is closed under complement (Theorem 4.16), to convert "close pairs" into "far pairs" and vice versa.

Next, we show that given any such protocol P to eliminate most possible values for Eq. (20), we can use P to define a one-round interactive protocol to convince a skeptical Verifier that a single formula ψ is unsatisfiable—under the promise that ψ has at most one satisfying assignment. We prove this by an application of the reconstruction algorithm of [ALRS98]. Finally, by an easy application of the "witness isolation" technique of Valiant and Vazirani [VV86], we conclude that NP $\subseteq$ coAM.

A small technical note is in order before we proceed. An input to SAT is a bitstring, interpreted as a description of some Boolean formula. Throughout this section, we will assume that we are working with a sufficiently flexible descriptive system. Namely, we assume that given a length-n description $\langle\psi\rangle \in \{0,1\}^n$ of a Boolean formula ψ, and given any sufficiently large m (say, $m \geq 2n$), one can a produce a "padded" description $\langle\psi\rangle'$ of ψ with bitlength exactly m. Moreover, we assume this padding can be performed in $\text{poly}(m+n)$ computational steps. This mild assumption is made for the sake of compatibility with our definition of f-compression reductions, which (to keep notation simple) always expect multiple input strings $(x^1, \ldots, x^{t_1(n)})$ of a common length n.

In this section, for $t > 0$ and $B \subseteq [t]$, we let $\mathbf{1}_B \in \{0,1\}^t$ denote the characteristic vector of B.

9.1 Eligible functions and their properties

We now define a class of Boolean functions called *eligible* functions. The definition of this class is not especially intuitive, but it identifies a general class of combining functions f for which the techniques of this section can rule out strong f-compression for SAT. First, recall the definition of sensitive blocks and block sensitivity from Section 7.3. Loosely speaking, eligible functions are functions f with low block sensitivity, and for which one can exhibit a large collection of (possibly overlapping) sensitive blocks for the input 0^t which span many indices and are not too "concentrated." We also require that this family of sensitive blocks for f on 0^t is computationally simple to define—although in principle, f itself need not be fully computable.

[42] (the actual order of presentation is different)

[43] Sivakumar in his work also achieves such a strengthened candidate-elimination guarantee, via a "bootstrapping" procedure that we are able to bypass in our application.

Definition 9.1 (Eligible functions)**.** *Let $f : \{0,1\}^t \to \{0,1\}$ be a Boolean function and $w, k, a, u > 0$ be integers. We say that f is* (w, k, a, u)-eligible *if the following conditions hold:*

1. *$bs(f) \leq k$;*
2. *There is a collection $\mathcal{B} = \mathcal{B}_t = (B_1, \ldots, B_a)$ of (nonempty, possibly overlapping, not necessarily distinct) subsets of $[t]$, where each B_ℓ is a minimal sensitive block for input 0^t (thus $|B_\ell| \leq k$, by an observation of Nisan [Nis91]);*
3. *For each $\ell \in [a]$, there is an $i \in B_\ell$ which appears in at most u of the sets $B_1, \ldots, B_a$;*
4. *For $\mathcal{B}$ chosen above, there exists a Boolean circuit $C_{\mathcal{B}} : \{0,1\}^{\lceil \log_2 t \rceil + \lceil \log_2 a \rceil}$ with at most w wires that, given as input a pair $(i, j) \in [t] \times [a]$ (in binary representation), determines if $i \in B_j$.*

We say that a family $f = \{f_t : \{0,1\}^t \to \{0,1\}\}_{t>0}$ is simply eligible *if each f_t is (w, k, a, u)-eligible, for parameters $(w, k, a, u) = (w(t), k(t), a(t), u(t))$ satisfying*

$$w, k, u \;\leq\; t^{o(1)} \qquad \text{and} \qquad t^{1-o(1)} \;\leq\; a \;\leq\; t \;,$$

and if there is a polynomial-time algorithm A which on input 1^t outputs a description of the circuit $C_{\mathcal{B}} = C_{\mathcal{B},t}$ in item 4.

Our main result on eligible functions is the following:

Theorem 9.2. *Suppose that $f = \{f_t : \{0,1\}^t \to \{0,1\}\}_{t>0}$ is an eligible function family as in Definition 9.1. For any integer $K \geq 1$, if there is a PPT f-compression reduction for $L :=$ SAT, with parameters*

$$t_1(n) \;=\; n^{500K} \;, \quad t_2(n) \;\leq\; n^K \;, \quad \xi(n) \;\leq\; .1 \;,$$

and any target language L', then NP $\subseteq$ coAM.

We have not attempted to optimize the parameters in this result; the techniques of this section seem inherently quantitatively weaker than the techniques used in Section 7 to analyze OR- and AND-compression. To reduce clutter in our work, we have made requirements in Definition 9.1 that are a bit stricter than needed. In particular, all parameters required to be $t^{o(1)}$ could instead be t^ε for some sufficiently small $\varepsilon > 0$; Theorem 9.2 would still hold.

Eligible functions need not be monotone. Even for eligible combining functions f which are non-monotone on every input length (to which Theorem 7.6 already applies), we get new information from Theorem 9.2, since the target language L' in the hypothesis is now allowed to be arbitrary and the conclusion is stronger as well.

We pause to give two examples of eligible functions, the second of which will be used in our application to AND- and OR-compression. Both examples have played a role in the study of complexity measures for Boolean functions [Weg91].

Definition 9.3 (Address functions)**.** *For $d > 1$ and input length $t := 2^d + d$, the classical "address" function* $\mathrm{ADDR}_{2^d}(x, y) : \{0,1\}^{2^d+d} \to \{0,1\}$ *defined as follows. Letting $x = (x_0, \ldots, x_{2^d-1})$ and $y = (y_0, \ldots, y_{d-1})$, let $v(y) := \sum_{\ell \in [d]} y_\ell 2^\ell$, and define*

$$\mathrm{IND}_{2^d}(x, y) \;:=\; x_{v(y)} \;.$$

A monotone *variant of this function can also be given [Weg91, Chap. 13] (our definition is slightly different for ease of use). The "monotone address" function* $\mathrm{mADDR}_{2^d}(x,y) : \{0,1\}^{2^d+2d} \to \{0,1\}$ *takes as input a* 2^d*-bit string* $x = (x_0, \ldots, x_{2^d-1})$ *and a* $2d$*-bit string* $y = (y_0, \ldots, y_{2d-1})$*. For each* $y \in \{0,1\}^{2d}$*, first define* $\hat{v}(y) := \sum_{\ell \in [0,d-1]} y_{2\ell+1} \cdot 2^\ell \in [0, 2^d - 1]$*. Letting* $||\cdot||$ *denote Hamming weight, define*

$$\mathrm{mADDR}_{2^d}(x,y) \;:=\; \begin{cases} 0 & \text{if } ||y|| < d; \\ x_{\hat{v}(y)} & \text{if } ||y|| = d \; ; \\ 1 & \text{if } ||y|| > d. \end{cases}$$

We extend ADDR_{2^d} *to all input lengths* $t \geq 6$ *by choosing the largest* d *such that* $2^d + d \leq t$ *and defining* ADDR_t *exactly as* ADDR_{2^d} *(acting on the first* $2^d + d$ *input variables and ignoring any remainder). We do similarly for* mADDR_{2^d}*. We let* ADDR, mADDR : $\{0,1\}^* \to \{0,1\}$ *denote the Boolean functions ranging over all* t *(letting these functions be identically 0 for* $t < 6$*, say).*

The function ADDR_{2^d} can be computed with $d+1$ queries to the input, and therefore its sensitivity and block sensitivity are also at most $d+1$. (In fact, this is exact.) It is easily verified that ADDR is eligible, by a family $\mathcal{B}$ of sensitive blocks containing a set B_r for each $r \in [2^d - 1]$: we write r in its binary expansion $r = \sum_{\ell=0}^{d-1} r_\ell 2^\ell$, and let B_r consist of the y-variables $\{y_\ell : r_\ell = 1\}$ along with the variable x_r. (We describe B_r with reference to the variables it contains rather than their indices, since we have used overlapping index-sets to refer to the x and y variables. Strictly speaking, we need to work with a single numeric indexing of the input variables x, y—beginning with index $i = 1$, not 0—to meet condition 4, but this is easily carried out; we omit the details.)

The monotone function $\mathrm{mADDR}_{2^d}(x,y)$ depends on all its $2^d + 2d$ variables and has query complexity and block sensitivity equal to $2d+1$. It is eligible by nearly the same construction used for ADDR_{2^d}. As we will use this fact in Section 9.6, we now show this explicitly. It is enough to consider input lengths of form $t = 2^d + 2d$. We let $a(t) := 2^d - 1$; we have $(1 - o(1))t \leq a \leq t$. For each $r \in [a]$, we define B_r as the variable-set containing the x-variable x_r along with the y-variables $\{y_{2\ell+1} : r_\ell = 1\} \cup \{y_{2\ell} : r_\ell = 0\}$.

We now argue that this set family $\mathcal{B}_t = (B_r)_{r \in [a]}$ witnesses that mADDR is eligible. The block-sensitivity bound (condition 1) on mADDR has already been observed. For the remaining conditions, first note that each B_r contains exactly d of the y-variables. If we let $(x,y) := \mathbf{1}_{B_r} \in \{0,1\}^t$ be the characteristic vector for B_r, we have $\hat{v}(y) = r$, and as $x_r = 1$ it follows that $\mathrm{mADDR}(x,y) = 1$. On the other hand $\mathrm{mADDR}(0^{2^d}, 0^{2d}) = 0$, so B_r is a sensitive block for the all-zero input. It is also a *minimal* sensitive block by inspection. Thus condition 2 in Definition 9.1 holds. Condition 3 holds with $u(t) = 1$ since each B_r contains a distinct x-variable x_r. For condition 4, note that $x_{r'} \in B_r$ exactly if $r' = r$, and that (for $\ell \in [0, d-1]$) we have $y_{2\ell} \in B_r$ exactly if $r_\ell = 0$, and $y_{2\ell+1} \in B_r$ exactly if $r_\ell = 1$. All of these simple tests can be performed by a circuit of size linear in the bitlength of the binary representations of r, r', ℓ, which is $O(d) = O(\log t)$. (Again, we omit the details of using a single numeric indexing of the x, y variables to meet condition 4.)

Next we describe a class of Boolean functions f, inspired by these previous examples, that in general are neither eligible nor $\Omega(1)$-amenable (as defined in Section 7.3). For general functions f of this class, we do not know how to give strong evidence against f-compression for SAT.

Fix a function $k(n) \geq 10 \log_2 n$ that satisfies $k(n) \leq n^{o(1)}$. For each n, let $k = k(n)$ and let $\mathcal{S}_n = \{S_1, \ldots, S_n\}$ be some family of distinct subsets of $[2k]$, each of Hamming weight k. Recalling

that $\mathbf{1}_{S_\ell}$ denotes the characteristic vector for S_ℓ, we define $f_n : \{0,1\}^{n+k} \to \{0,1\}$ by

$$f_n(x,y) \;:=\; \begin{cases} 0 & \text{if } ||y|| < k; \\ x_\ell & \text{if } y = \mathbf{1}_{S_\ell} \text{ for some } \ell \in n\ ; \\ 1 & \text{if } ||y|| > k. \end{cases}$$

This function depends on all variables, provided the set family $\mathcal{S}_n$ spans all of $[2k]$. Its query complexity and block sensitivity are at most $2k+1 \leq n^{o(1)}$. It will in general fail to have the explicitness property of eligible functions (condition 4 in Definition 9.1), if $\mathcal{S}_n$ is sufficiently complicated.[44]

Moving on from these examples, in the rest of Section 9.1 we prove a lemma giving useful information about the behavior of functions with low block sensitivity. We show that such functions are "stable" with respect to a certain kind of random perturbation to their input. (This is completely distinct from the "distributional stability" of compressive mappings, shown in Lemma 6.2.) We first state the lemma, then briefly describe the setting in which we will use it.

Lemma 9.4. *Let $f : \{0,1\}^t \to \{0,1\}$ be non-constant, with $bs(f) \leq t^{o(1)}$. Let $a \leq t$ and suppose $M \subseteq [a]$ is of size $|M| \geq t^{.99}$, and let $(B'_\ell)_{\ell \in M}$ be a family of nonempty (not necessarily distinct) subsets indexed by M, satisfying $|B'_\ell| \leq bs(f)$. Assume that every $i \in [t]$ is contained in at most $t^{.001}$ of the sets B'_ℓ. Fix any input $z \in \{0,1\}^t$, and consider the following random process:*

1. *Let $(X_\ell)_{\ell \in M}$ be independent, identically distributed 0/1-valued Bernoulli trials, with $\mathbb{E}[X_\ell] \leq t^{.95}/|M|$;*
2. *Let $z' := z \vee \left(\bigvee_{\ell \in M} X_\ell \cdot \mathbf{1}_{B'_\ell}\right)$, with $\vee$ denoting the coordinate-wise OR operation over $\{0,1\}^t$.*

Then for sufficiently large t, we have $\Pr[f(z') \neq f(z)] \leq .01$.

In our application, f will be eligible, and we will define a set $J \subseteq [t]$ consisting of all $i \in [t]$ appearing in more than $t^{.001}$ of the sets B_ℓ from $\mathcal{B}$. We will then define $B'_\ell := B_\ell \setminus J$. For large t, each B'_ℓ will be nonempty (by condition 3 of Definition 9.1). It will be convenient to pass to a subset M of the indices $\ell \in [a]$.

Proof of Lemma 9.4. Consider the equivalent experiment in which we first reveal the sum $X := \sum_{\ell \in M} X_\ell$ and then reveal the indices $\{\ell \in M : X_\ell = 1\}$ sequentially, in a random order.

First, Markov's inequality tells us that with probability at least .999 we have the relation $X \leq 1000\mathbb{E}[X] \leq 1000t^{.95} < t^{.96}$ (for large t). Let us condition on any event $[X = S]$ where $S \leq t^{.96}$. If $S = 0$ then $f(z) = f(z')$, so consider the case $S > 0$. We let $\{\ell_1, \ldots, \ell_S\} := \{\ell \in M : X_\ell = 1\}$; the indices $(\ell_1, \ldots, \ell_S)$ are revealed sequentially to us. Conditioned on $\ell_1, \ldots, \ell_j$, the index ℓ_{j+1} is uniform over all $\ell \in M \setminus \{\ell_1, \ldots, \ell_j\}$.

For each $j \in [0, S]$, let

$$z[j] \;:=\; z \vee \left(\bigvee_{\ell \in \{\ell_1, \ldots, \ell_j\}} X_\ell \cdot \mathbf{1}_{B'_\ell} \right) .$$

Thus $z[0] = z$ and $z[S] = z'$.

[44] Of course, if the set families $\{\mathcal{S}_n\}_{n>0}$ are so complicated that the associated $f = \{f_n\}_{n>0}$ is *undecidable*, then we can unconditionally rule out nontrivial instance compression from $f \circ \mathrm{SAT}$ to any decidable target language L'.

Claim 9.5. *Under our conditioning $[X = S]$ (where $S \le t^{.96}$), fix any $j \in [0, S-1]$; for sufficiently large t, the event $[\ f(z[j+1]) \neq f(z[j])\ ]$ occurs with probability at most $t^{-.98}$.*

Proof. Condition further on any outcomes to $\ell_1, \ldots, \ell_j$, which determine the string $z[j]$. Say that $\ell \in M \setminus \{\ell_1, \ldots, \ell_j\}$ is *pivotal* if $f(z[j] \vee \mathbf{1}_{B'_\ell}) \neq f(z[j])$. Let Q be the number of pivotal indices ℓ under our conditioning; then the conditional probability that $f(z[j+1]) \neq f(z[j])$ is exactly $Q/(|M| - j)$, which is less than $2Q/|M|$ for large t since $|M| > t^{.99}$.

Next, note that there must exist some minimal-size subset $W \subset [t]$ such that the values taken by $z[j]$ on coordinates in W force f to take the value $b = f(z[j])$. Using the definition of certificate complexity and Fact 7.3, we have $|W| \le C(f) \le bs(f)^2 \le t^{o(1)}$. Observe that, if $\ell \in M$ satisfies $W \cap B'_\ell = \emptyset$, then ℓ cannot possibly be pivotal. Using our pairwise-intersection bound on the family $(B'_\ell)_{\ell \in M}$, it follows that $Q \le |W| \cdot t^{.001} \le t^{.001+o(1)}$. Thus, the conditional probability that $f(z[j+1]) \neq f(z[j])$ is less than $2t^{.001+o(1)}/|M| \le t^{-.98}$ (for large t). As $\ell_1, \ldots, \ell_j$ were arbitrary, this proves the Claim.[45] □

By applying this Claim to each $j \in [0, S-1]$ and taking a union bound, it follows that (under our conditioning $[X = S]$), we have

$$\Pr[f(z) \neq f(z')] \ \le \sum_{j \in [0, S-1]} \Pr[z[j+1] \neq z[j]] \ \le \ |S| \cdot t^{-.98} \ \le \ t^{-.02}\ .$$

As S was an arbitrary integer in the range $[1, t^{.96}]$, we can combine our analyses to conclude that, unconditioned, we have (for large t)

$$\Pr[f(z) \neq f(z')] \ \le \ .001 + t^{-.02} \ < \ .01\ .$$

□

We will see eligible functions again in Section 9.5, where we prove Theorem 9.2; first we need to develop some other useful tools.

9.2 A codeword-reconstruction result

In this section we describe the powerful result of Ar et al [ALRS98] that will play a key role in our study of f-compression for eligible f. Our application of the result is strongly influenced by its earlier application in [Siv99].

We will work over certain easy-to-describe finite fields of characteristic 2:

Definition 9.6 (Nice integers and finite-field representations)**.** *Say that an integer m is* nice *if it is of form $m = 2 \cdot 3^w$. Following [Siv99] and earlier works, if m is nice then the polynomial $X^m + X^{m/2} + 1$ is irreducible over $\mathbb{F}_{2^m}[X]$; the finite field $\mathbb{F}_{2^m}$ is isomorphic to $\mathbb{F}_2[X]/(X^m + X^{m/2} + 1)$, and elements of $\mathbb{F}_{2^m}$ can be represented as $\mathbb{F}_2$-polynomials mod $X^m + X^{m/2} + 1$.*

For technical convenience (and in a slight departure from [Siv99]), we will work with a representation of elements of the finite field $\mathbb{F}_{2^m}$ for nice m in which each $u \in \mathbb{F}_{2^m}$ is represented by a string $\mathrm{str}(u) \in \{0,1\}^{2m}$ of Hamming weight exactly m; such a representation is easily obtainable from the polynomial representation just described.

[45] In the proof of Lemma 9.4 we essentially exploited a bound on the *fractional block sensitivity* of f, as defined by Tal [Tal13] building on work of Aaronson [Aar08]; this measure, denoted $fbs(f)$, is known to obey $bs(f) \le fbs(f) \le C(f)$, with all three quantities polynomially related [Tal13, Aar08].

The following result was essentially proved by Sivakumar [Siv99], who derived the technical substance of the result from [ALRS98]. We give the proof for the sake of completeness. As in [Siv99], we do not attempt to give a sharpest-possible statement.

Theorem 9.7. *For each integer $k > 1$ and nice value $m \geq 10 \log_2 k$, there is a mapping*

$$E : \{0,1\}^k \longrightarrow (\mathbb{F}_{2^m})^{2^m} ,$$

with the following properties.

1. *For any family of subsets $\{S_u\}_{u \in \mathbb{F}_{2^m}}$, with each $S_u \subset \mathbb{F}_{2^m}$ satisfying $|S_u| \leq 2^{m/3}$, there are at most $2^{O(m)}$ vectors $y \in \{0,1\}^k$ for which*

$$E_u(y) \in S_u , \qquad \forall u \in \mathbb{F}_{2^m} . \tag{21}$$

   (Here we index the coordinates of $E(y)$ by elements of $\mathbb{F}_{2^m}$, and let $E_u(y) \in \mathbb{F}_{2^m}$ denote the u^{th} coordinate.) Moreoever, there is a polynomial-time algorithm B to compute all y satisfying Eq. (21), given the value m and a collection $\{S_u\}_{u \in \mathbb{F}_{2^m}}$ as above (with $v \in \mathbb{F}_{2^m}$ represented in the input by $\mathrm{str}(v)$).

2. *$\mathrm{str}(E_u(y))$ is computable in time $\leq (k+m)^3$ for sufficiently large k, given m, y, and $\mathrm{str}(u)$.*[46]

Our proof below follows the presentation in [Siv99].[47]

Proof. On input $y = (y_1, \ldots, y_k) \in \{0,1\}^k$, define the polynomial $P_y(X) \in \mathbb{F}_{2^m}[X]$ by

$$P_y(X) := \sum_{i=1}^{k} y_i \cdot X^{i-1} ,$$

with arithmetic over $\mathbb{F}_{2^m}$ (and regarding the bits $\{0,1\}$ as the zero and unit elements of $\mathbb{F}_{2^m}$). Define $E(y)$ coordinatewise by

$$E_u(y) := P_y(u) .$$

The efficient computability property of E, item 2, is clear, since arithmetic can be performed efficiently over $\mathbb{F}_{2^m}$ (see [Pos11] for a discussion of state-of-the-art algorithms, which perform multiplication of two elements of $\mathbb{F}_{2^m}$ in time $O(m \,\mathrm{polylog}(m))$). Now we describe the algorithm B. Suppose we are given m and a collection $\{S_u\}_{u \in \mathbb{F}_{2^m}}$ as in item 1. Let $D := \lfloor 2^{3m/4} \rfloor$. First, we construct a nonzero bivariate polynomial $Q(X,Y) \in \mathbb{F}_{2^m}[X,Y]$, with each monomial of degree between 1 and D in each of X, Y, and which satisfies

$$Q(u,v) = 0 \quad \forall u \in \mathbb{F}_{2^m} , \; v \in S_u . \tag{22}$$

There are $D^2 \geq 2^{3m/2-1}$ coefficients to choose, and our requirements impose at most $\sum_u |S_u| \leq 2^{4m/3}$ linear, *homogeneous* constraints upon these coefficients. As $m \geq 10$ we conclude that some nonzero polynomial Q as above exists, and can be found efficiently using Gaussian elimination over $\mathbb{F}_{2^m}$.

[46] This is a conservative bound and can be improved.

[47] We note that for our application, the result of Theorem 9.7 is overkill; in performing the "recovery procedure" B above, we could allow ourselves one round of interaction with a powerful Merlin. However, this observation seems not to simplify the presentation, so we will pass over it.

Next, we *factor* Q in deterministic polynomial time using known algorithms. Deterministic polynomial-time algorithms are known for the case of polynomially-bounded field size, as used here and in [Siv99]; see the references in [Siv99, ALRS98].

After factoring Q into its irreducible factors, B inspects each to see if it is of the form $(Y - P_y(X))$ for some y (or a scalar multiple of such a polynomial). Finally, B outputs all such candidates y, of which there can be no more than the number of irreducible factors of Q. There are at most $2^{O(m)}$ such factors, by the degree bound on Q.

B is polynomial-time as claimed. To prove correctness, we claim that for every y satisfying Eq. (21), the polynomial $(Y - P_y(X))$ divides $Q(X, Y)$. To see this, note that $Q^*(X) := Q(X, P_y(X))$ is a univariate polynomial of degree at most $D + k \cdot D$, which is less than 2^m (since $m \geq 10 \log_2 k$ by our assumption). But in light of Eq. (22), Q^* vanishes for each $u \in \mathbb{F}_{2^m}$. Thus $Q^*(X)$ must be identically 0. By regarding $Q(X, Y)$ as a univariate polynomial in $(\mathbb{F}_{2^m}[X])[Y]$ and applying the factor theorem, we conclude that $(Y - P_y(X))$ is a factor of $Q(X, Y)$, as claimed, so that B outputs y. This completes the proof. □

9.3 None-versus-one protocols

Next we define one-round interactive protocols to prove that a Boolean formula has *no* satisfying assignment, under the promise that it has at most *one*. By applying the technique of [VV86], we prove that such polynomial-time protocol can be given unless NP ⊆ coAM. This, recall, formed the final step of our argument as sketched in the beginning of Section 9. We note that some results related to our work in this section appear in [CKR95].

Definition 9.8 (None-versus-one protocols)**.** *Let $R = R(n), W = W(n) \leq \text{poly}(n)$ be integer parameters computable in time $\text{poly}(n)$ given n. Let $V(\langle\psi\rangle, r, w)$ be a deterministic algorithm taking a Boolean formula ψ of some description length n, along with $(r, w) \in \{0,1\}^{R+W}$. Say that V is a* none-versus-one protocol (for Boolean formula satisfiability) *if:*

1. *If $\langle\psi\rangle \in \overline{\text{SAT}}$, then with probability $2/3$ over a uniformly chosen $r \in \{0,1\}^R$, there exists a $w \in \{0,1\}^W$ such that $V(\langle\psi\rangle, r, w) = 1$;*
2. *If $\langle\psi\rangle \in \text{SAT}$ and ψ has a unique satisfying assignment, then the probability over uniform r that there is a w with $V(\langle\psi\rangle, r, w) = 1$ is at most $1/3$.*

Lemma 9.9. *If there is a polynomial-time none-versus-one protocol V, then there is a second such protocol V' in which the completeness and soundness parameters $(2/3, 1/3)$ are replaced with $(1 - 2^{-n}, 2^{-n})$ respectively.*

Proof sketch. This follows by a standard amplification argument: we define V' which simulates a sufficiently large number $N = O(n)$ of copies of V in parallel, and outputs the majority vote. □

Lemma 9.10. *Suppose there is a polynomial-time none-versus-one protocol. Then* NP ⊆ coAM.

Proof. Under our assumption, we apply the well-known technique of Valiant and Vazirani [VV86] to give an Arthur-Merlin proof system for $L = \overline{\text{SAT}}$; this will imply coNP ⊆ AM, which yields the Lemma's conclusion.

Valiant and Vazirani give a randomized reduction $A(\langle\psi\rangle, r)$ which, given a formula description $\langle\psi\rangle \in \{0,1\}^n$ and a random string r of length $|r| = t(n)$ and outputs a formula $\psi^{(r)}$, with the following properties:

1. If ψ is unsatisfiable, then $\psi^{(r)}$ is also unsatisfiable;

2. If ψ is unsatisfiable, then with probability $\Omega(1/n)$ over r, the formula $\psi^{(r)}$ has a *unique* satisfying assignment.

Our Arthur-Merlin protocol V^* is as follows. Let $N = O(n)$ be a large multiple of n to be chosen later. Arthur chooses independent random strings $r^1, \ldots, r^N \in \{0,1\}^{t(n)}$ and produces formula descriptions $\langle \psi^{(r^i)} \rangle := A(\langle \psi \rangle, r^i)$ for each $i \in [N]$. For each such i, Arthur simulates the success-amplified none-versus-one protocol V' from Lemma 9.9 on input $\langle \psi^{(r^i)} \rangle$ (with Arthur setting the random string and Merlin choosing the string w in response). Arthur accepts if each invocation of V' outputs 1, otherwise rejects.

The protocol V^* is clearly polynomial-time and requires only a single round of interaction. To prove correctness, first suppose that ψ is unsatisfiable. Then each $\psi^{(r)}$ is also unsatisfiable. By the completeness property of V', with probability $\geq 1 - 2^{-n}$ over Arthur's randomness r' in the invocation of $V'(\langle \psi^{(r^i)} \rangle, \cdot, \cdot)$, there exists a w^i such that $V'(\langle \psi^{(r^i)} \rangle, r', w^i) = 1$. By a union bound, this occurs for each $i \in [N]$ with probability $\geq 1 - N2^{-n} = 1 - o(1)$. Thus if $\langle \psi \rangle \in \overline{\text{SAT}}$ there exists a Merlin strategy for V^* causing Arthur to accept with probability $1 - o(1)$.

Next, suppose ψ is satisfiable. Then by the second property of the Valiant-Vazirani reduction, if we choose $N = O(n)$ sufficiently large then, with probability $\geq .99$, there exists at least one $i \in [N]$ for which $\psi^{(r^i)}$ has a unique satisfying assignment. Then with probability $1 - 2^{-n}$ over r', there is no w for which $V'(\langle \psi^{(r^i)} \rangle, r', w) = 1$. Thus any Merlin strategy causes V^* to accept with probability at most $1 - .99(1 - 2^{-n}) < .02$ for large n. Thus $\overline{\text{SAT}} \in \mathsf{AM}$. This completes the proof. □

9.4 Membership comparability

We will use the notion of *membership comparability* studied in [ABG03, BKS95, AA94, Ogi95, BFT97, Siv99] and other works. The general question studied in these works is: for which languages L, given a collection $x^1, \ldots, x^h$ of instances, can we efficiently eliminate one or more possibilities for the value of the characteristic vector $(L(x^1), \ldots, L(x^h))$? The following definition gives a variant of this property in which the computation is aided by a single round of interaction with a prover, and in which the Hamming weight of $(L(x^1), \ldots, L(x^h))$ obeys a promise.

Definition 9.11 (Promise list-enumeration protocols)**.** *Let L be a language, and let $h = h(n), s = s(n)$ be parameters (each $\leq \text{poly}(n)$ and computable in time $\text{poly}(n)$), with h even-valued for each n. An (h, s)-*promise list-enumeration protocol *is defined by an algorithm P taking inputs $x^1, \ldots, x^h \in \{0,1\}^{h \times n}$ along with strings $(r, w) \in \{0,1\}^{R(n)+W(n)}$, for some additional parameters $R(n), W(n) \leq \text{poly}(n)$ also computable in time $\text{poly}(n)$.*

P either outputs "$\perp$" or outputs a list $(v^1, \ldots, v^s) \in \{0,1\}^{s \times h}$ (possibly with duplicates). We require that for each $x^1, \ldots, x^h$ which satisfy

$$\Big|\Big|(L(x^1), \ldots, L(x^h))\Big|\Big| \in \{0, h/2\}$$

($||\cdot||$ denoting Hamming weight), with probability $\geq 1 - 2^{-n}$ over a uniform r, both of the following conditions hold:

1. *There exists a w such that $P(x^1, \ldots, x^h, r, w)$ does not output "$\perp$";*

2. For each w as in item 1, the list produced by P contains a v^i equal to $(L(x^1), \ldots, L(x^h))$.

We study promise list-enumeration protocols for $L = \text{SAT}$. Note that, on input-formulas $\langle\psi_1\rangle, \ldots, \langle\psi_h\rangle$ obeying the promise, and such that half are satisfiable, it is easy for Prover to help Verifier identify the true characteristic vector, by producing satisfying assignments for the $h/2$ satisfiable formulas. However, there is no obvious way for Prover to help Verifier eliminate candidates in the case where all formulas are unsatisfiable. The next lemma shows that sufficiently strong promise list-enumeration protocols for SAT would imply $\mathsf{NP} \subseteq \mathsf{coAM}$.

Lemma 9.12. *Let $C \geq 20$ be given, and let $h(n)$ be an efficiently computable even parameter in the range $[C \log_2 n, 1.1C \log_2 n]$ for each n. Suppose that $L := \text{SAT}$ has an $(h(n), s(n))$-promise list-enumeration protocol P running in polynomial time, with $s(n) \leq 2^{h(n)/18.5}$. Then there is a polynomial-time none-versus-one protocol for Boolean formula satisfiability (and $\mathsf{NP} \subseteq \mathsf{coAM}$, by Lemma 9.10).*

Proof. We describe how to use the promise list-enumeration protocol P to construct a none-versus-one protocol V. We first describe some setup performed by V prior to interaction with Prover. Given an input Boolean formula $\langle\psi\rangle$ of some description length $n_0 > 1$ and with some number $k_0 \leq n_0$ of variables, we first set $k := n_0$ and regard $\psi = \psi(y)$ as acting on input variables $y = (y_1, \ldots, y_k)$, some of which may not appear in ψ. Let m be a nice value in the range

$$m \in [.5C \log_2 k, 1.5C \log_2 k] .$$

Let $E : \{0,1\}^k \to \mathbb{F}_{2^m}^{2^m}$ be the code given in Theorem 9.7. For $y \in \{0,1\}^k$, $u \in \mathbb{F}_{2^m}$ and $j \in [2m]$, let $E_{u,j}(y)$ denote the j^{th} bit of $\text{str}(E_u(y)) \in \{0,1\}^{2m}$ (recalling the representation $\text{str}(\cdot)$ used in Theorem 9.7).

For each such pair (u, j), define a Boolean formula $\psi^{(u,j)}(y)$ on k variables which accepts $y \in \{0,1\}^k$ exactly if $[\psi(y) = 1] \wedge [E_{u,j}(y) = 1]$ holds. Using the efficiency property in Theorem 9.7, item 2, and padding if necessary, for n_0 sufficiently large compared to C, the formula $\psi^{(u,j)}$ can be implemented by a formula of description length exactly

$$n := n_0^{3.01}$$

(with smaller values of n_0 being handled by brute force). $\langle\psi^{(u,j)}\rangle$ is also constructible in time $\text{poly}(n_0)$. Let

$$(h, s) := (h(n), s(n)) .$$

By our settings and assumptions on $h(\cdot)$ we have

$$2m \leq h \leq 6.02m \tag{23}$$

and

$$s \leq 2^{h/18.5} \leq 2^{m/3} . \tag{24}$$

Finally, let $\langle\phi_{unsat}\rangle$ be an arbitrary unsatisfiable formula of description length n.

With these preparations, our none-versus-one protocol V, on input ψ as above, acts as follows. First, V constructs the desciption $\langle\psi^{(u,j)}\rangle \in \{0,1\}^n$ for each $(u, j) \in \mathbb{F}_{2^m} \times [2m]$. For each $u \in \mathbb{F}_{2^m}$ in parallel, it executes the promise list-enumeration protocol P (assumed to exist in the present Lemma) to the h-tuple of inputs

$$\left(\langle\psi^{(u,1)}\rangle, \langle\psi^{(u,2)}\rangle, \ldots, \langle\psi^{(u,2m)}\rangle, \langle\phi_{unsat}\rangle, \ldots, \langle\phi_{unsat}\rangle, \langle\psi\rangle, \ldots, \langle\psi\rangle\right) , \tag{25}$$

where we "pad" the $2m$-tuple $(\langle\psi^{(u,j)}\rangle)_{j\in[2m]}$ with $(h-2m)/2$ copies each of $\langle\phi_{unsat}\rangle$ and $\langle\psi\rangle$.

If any of these executions of P returns "$\perp$", then V outputs 0. Otherwise, each execution (indexed by $u \in \mathbb{F}_{2^m}$) returns a collection of s strings $Z^{u,1},\ldots,Z^{u,s} \in \{0,1\}^h$. We truncate each of these to their initial $2m$ coordinates, yielding a revised collection $z^{u,1},\ldots,z^{u,s} \in \{0,1\}^{2m}$. Let $S_u \subseteq \mathbb{F}_{2^m}$ be the set $\{z^{u,a}\}_{a\in[s]}$, regarded as elements of $\mathbb{F}_{2^m}$ under the representation $\mathrm{str}(\cdot)$ (and discarding any duplicate elements). Using our guarantee on P and Eq. (24), we have

$$|S_u| \;\leq\; s \;\leq\; 2^{m/3}\;.$$

V then applies the algorithm B from Theorem 9.7 to the inputs $(S_u)_{u\in\mathbb{F}_{2^m}}$, obtaining a (possibly empty) list of strings $y^1,\ldots,y^T \in \{0,1\}^k$. V evaluates $\psi(y^t)$ for each $t \in [T]$ and outputs 0 if $\psi(y^t)=1$ for some t. (If all $\psi(y^t)=0$, or if the list is empty, then V outputs 1.)

The computations performed by V can clearly be carried out in time $\mathrm{poly}(n_0+2^m) \leq \mathrm{poly}(n_0)$ (for each fixed constant C). To prove correctness, first suppose that ψ is unsatisfiable. Each of the 2^m executions of P is applied to a list of $h(n)$ formulas each of description length n, and every formula is unsatisfiable in the present case, so the characteristic vector of each such list is of Hamming weight 0. Then by Definition 9.11, there exists a Merlin strategy that causes "$\perp$" to be output with probability at most 2^{-n} for each fixed execution of P. Then by following this strategy independently on each execution of P, Merlin can cause V to reach and run the simulation of B with probability at least $1-|\mathbb{F}_{2^m}|\cdot 2^{-n} = 1-o(1)$. In such a case V must output 1, since ψ has no satisfying assignments to appear among $y^1,\ldots,y^T$.

Next, suppose that ψ is satisfiable, with a unique satisfying assignment $y^* \in \{0,1\}^k$; this is the remaining case we must consider to satisfy Definition 9.8. For each $u \in \mathbb{F}_{2^m}$, we claim that the $2m$-bit string[48]

$$\overline{b} \;=\; (b_{u,1},\ldots,b_{u,2m}) \;:=\; \Big(\chi_{\mathrm{SAT}}(\psi^{(u,1)}),\ldots,\chi_{\mathrm{SAT}}(\psi^{(u,2m)})\Big)$$

is precisely $\mathrm{str}(E_u(y^*))$. To see this, fix any $j \in [2m]$ for which $E_{u,j}(y^*)=1$. Then by definition of $\psi^{(u,j)}$, we see that it is satisfied by the assignment $y := y^*$, so $b_{u,j}=1$. Conversely, if $b_{u,j}=1$ then we must have $E_{u,j}(y^*)=1$, since y^* is the *unique* satisfying assignment to ψ (and the only candidate satisfying assignment to $\psi^{(u,j)}$). This proves that $(b_{u,1},\ldots,b_{u,h}) = \mathrm{str}(E_u(y^*))$. In particular, it follows from our choice of representation $\mathrm{str}(\cdot)$ that $\overline{b}$ is of Hamming weight exactly m, and from this we easily see that exactly $h/2$ of the formulas in the h-tuple in Eq. (25) are in SAT.

With reference to Definition 9.11, we deduce that, with probability $\geq 1-2^m\cdot 2^{-n} = 1-o(1)$ over the random choices made by V, one of two events must occur (for any choices made by Merlin in the 2^m executions of P): *either* some execution of P returns "$\perp$" (causing V to output 0); *or*, for each $u \in \mathbb{F}_{2^m}$, the list $(z^{u,1},\ldots,z^{u,s}) \in \{0,1\}^{s\times h}$ produced by V contains the vector $(\chi_{\mathrm{SAT}}(\psi^{(u,1)}),\ldots,\chi_{\mathrm{SAT}}(\psi^{(u,h)})) = \mathrm{str}(E_u(y^*))$ occurring as some $z^{u,a}$. In the latter case, we have $E_u(y^*) \in S_u$ for each $u \in \mathbb{F}_{2^m}$. It follows from Theorem 9.7 that, in the $\mathrm{poly}(n)$-sized list $y^1,\ldots,y^T$ produced by V, some y^t must equal y^*. V then determines that $\psi(y^t)=1$ and outputs 0. Thus any Merlin strategy causes V to output 0 with probability $1-o(1)$. We conclude that V is a polynomial-time zero-versus-one protocol, as desired. □

[48] Here and in the rest of Section 9, we use $\chi_{\mathrm{SAT}}(\psi)\in\{0,1\}$ to denote $\chi_{\mathrm{SAT}}(\langle\psi\rangle)$, with the understanding that the satisfiability status of ψ is independent of the particular description $\langle\psi\rangle$ given. Similarly we write $[\psi\in\mathrm{SAT}]$ instead of $[\langle\psi\rangle\in\mathrm{SAT}]$.

9.5 From f-compression to membership-comparison protocols

We are now prepared to prove Theorem 9.2. We will show how, given a PPT f-compression reduction for SAT as in the Theorem statement, one can construct an $(h(n), s(n))$-promise list-enumeration protocol for SAT, for some parameters $h(n)$ and $s(n)$ obeying the assumptions of Lemma 9.12. It will follow from that Lemma that $\mathsf{NP} \subseteq \mathsf{coAM}$.

Our list-enumeration protocol P receives input formula descriptions $(\langle\psi^1\rangle, \ldots, \langle\psi^{h(n)}\rangle)$, each of bitlength n. Our description of the promise list-enumeration protocol P is in two parts. In the first, "setup" part we describe some polynomial-time computations defining various parameters and objects that P will use, and we specify the parameter $h(n)$. In the second, "interaction" part we describe how the interaction with Merlin proceeds.

The setup: Define

$$T \;:=\; n^{1000K} \;.$$

First, P runs the polynomial-time algorithm $A(1^T)$, yielding a circuit $C_{\mathcal{B}}$ defining the set family $\mathcal{B} = (B_1, \ldots, B_a)$ for $a = a(T)$, with each $B_\ell \subset [T]$ of size $k = k(T)$. Next, P explicitly computes $B_1, \ldots, B_a$; this can be done in polynomial time. Recall that $a(t) \geq t^{1-o(1)}$ and $k(t) \leq t^{o(1)}$; thus we can assume that n is large enough that

$$a \;>\; T^{.999} + 2 \;, \quad k \;\leq\; T^{.0001}$$

(we can handle smaller values of n by brute force).

For $i \in [T]$, define $c_i := |\{\ell \in [a] : i \in B_\ell\}|$. Let $J \subseteq [T]$ be the set of all i for which $c_i > T^{.001}$ (a polynomial-time computable set). Using condition 3 in Definition 9.1, we may assume n is large enough that no set B_ℓ is contained entirely within J.

Letting

$$h \;=\; h(n) \;:=\; 2 \cdot \lfloor .5 \log_2(a) \rfloor \;, \qquad s \;=\; s(n) \;:=\; \lfloor 2^{h(n)/18.5} \rfloor \;,$$

we note that h is even and satisfies $h(n) \in [.999 \log_2 T, \log_2 T] = [999K \log_2 n, 1000K \log_2 n]$. Thus to prove the Theorem it will suffice to give an $(h(n), s(n))$-promise list-enumeration protocol for SAT (we will be able to apply Lemma 9.12 with $C := 999K$).

For $v \in \{0,1\}^h$, let $\mathrm{num}(v) := \sum_{h' \in [h]} v_{h'} \cdot 2^{h'-1}$ be v interpreted as an integer in its binary expansion. We have $\mathrm{num}(v) \leq 2^h - 1 < a$. Say that $\ell \in [a]$ is *meaningful* if $\ell = \mathrm{num}(v)$ for some $v \in \{0,1\}^h$ of Hamming weight exactly $h/2$. Let $M \subset [a]$ be the meaningful indices; M can be enumerated in polynomial time. Using a standard binomial estimate we have (for large n)

$$|M| \;=\; \binom{h}{h/2} \;\geq\; \frac{2^h}{\sqrt{2h}} \;>\; T^{.999} \;=\; n^{999K} \;. \tag{26}$$

For each $h' \in [h]$, we let $x^{h'}$ denote the variables appearing in the input formula $\psi^{h'}$, and let $n_i \leq n$ be their number. We assume these variable-sets are disjoint for each h' (after relabeling if necessary). For each $i \in [T]$, define a (fanin-2) Boolean circuit Γ_i which takes input variables

$$(x^1, \ldots, x^h) \;\in\; \{0,1\}^{n_1 + \ldots + n_h} \;,$$

and accepts exactly if the following conditions hold: letting $b_{h'} := \psi^{h'}(x^{h'})$ and $\bar{b} := (b_1, \ldots, b_h)$, we have:

1. $||\overline{b}|| = h/2$;
2. $[i \in B_j]$ holds, for $j := \mathrm{num}(\overline{b})$.

We argue that Γ_i can be implemented efficiently. First, $\overline{b}$ can be computed by a circuit of $O(nh \log_2 n) = O(n \log^2 n)$ gates which simulates each $\psi^{h'}$. Given $\overline{b}$, one may easily check condition 1 with $O(h)$ gates, and check condition 2 with $O(\log n)$ gates to compute j, and $T^{o(1)} \leq n^{o(1)}$ gates to check if $i \in B_j$ (using the circuit $C_{\mathcal{B},T}$ given by Definition 9.1 for the eligible function f). Thus we can implement Γ_i with $O(n \log_2^2 n)$ gates. By applying Cook's reduction, we derive a 3-CNF Γ'_i of $O(n \log_2^2 n)$ clauses, such that Γ'_i is satisfiable exactly if Γ_i is satisfiable.

Each Γ'_i can be given a description of bitlength $O(n \log_2^3 n)$, which for sufficiently large n is less than $\tilde{n} := n^2$. By padding we obtain a description $\langle \Gamma'_i \rangle$ of length exactly $\tilde{n}$. Also, for future use, let ψ_{sat}, ψ_{unsat} be two formulas which are satisfiable and unsatisfiable, respectively, and whose descriptions are of bitlength $\tilde{n}$.

We next define a probabilistic experiment **Expt**, defined with reference to

$$\overline{\psi} \ := \ (\psi^1, \ldots, \psi^h) \ .$$

It will be clear from the description that **Expt** can be performed by a Boolean circuit $C_{\overline{\psi}}^{\mathbf{Expt}}$ (given uniform random input bits) of size $\leq \mathrm{poly}(T, n) = \mathrm{poly}(n)$, and that this circuit can be constructed from $\overline{\psi}$ in $\mathrm{poly}(n)$ time. **Expt** proceeds as follows:

1. For each $i \in J$, set $\phi^i := \Gamma'_i$;
2. Let $d := \left\lceil \log_2 \left(\frac{|M|}{T^{.95}} \right) \right\rceil$;
   // Note that $T^{.95}/(2|M|) \leq 2^{-d} \leq T^{.95}/|M|$.
3. For each $j \in M$, let X_j be an independent Bernoulli trial with $\mathbb{E}[X_j] = 2^{-d}$;
   // $(X_j)_{j \in M}$ can be sampled efficiently and exactly using $|M| \cdot d$ uniform random bits.
4. For each $i \in [T] \setminus J$, let

$$\phi^i \ := \ \begin{cases} \phi_{sat} & \text{if } \exists j \in M : [X_j = 1] \wedge [i \in B_j] \ , \\ \phi_{unsat} & \text{otherwise;} \end{cases}$$

5. Apply the f-compression reduction R to $(\langle \phi^1 \rangle, \ldots, \langle \phi^T \rangle) \in \{0,1\}^{T \times \tilde{n}}$, outputting the resulting string z.
   // Note that $T = n^{1000K} = (n^2)^{500K} = t_1(\tilde{n})$, *so that R is guaranteed to output a string* $z \in \{0,1\}^{\leq t_2(\tilde{n})}$, *where* $t_2(\tilde{n}) \leq (n^2)^K = n^{2K}$.

For any $j \in M$ and $b \in \{0,1\}$, we also define the modified experiment $\mathbf{Expt}'(j;b)$, defined with respect to $\overline{\psi}$, which proceeds exactly as **Expt** *except* that we fix the value $X_j := b$. Let $C_{\overline{\psi}}^{\mathbf{Expt}'(j;b)}$ denote a Boolean circuit of size $\leq \mathrm{poly}(n)$ which performs $\mathbf{Expt}'(j;b)$ upon $\overline{\psi}$ (given uniform random input bits); this circuit can also be efficiently constructed by P. (Our primary interest here is the case where $b = 1$.)

For each $j \in M$, define the string $\sigma^j := \left\langle C_{\overline{\psi}}^{\mathbf{Expt}}, C_{\overline{\psi}}^{\mathbf{Expt}'(j;1)} \right\rangle$. We are interested in the statistical distance between the output distributions of the two circuits described in σ^j on uniformly random inputs. Let $n \leq n' \leq \mathrm{poly}(n)$ be a sufficiently large value such that each σ^j can be given a description of length exactly n' (by padding if necessary; we can and do choose to work with a descriptive system for circuit-pairs that allows such padding).

With reference to Definition 4.12 and Theorems 4.16 and 4.17, recall that the promise problem $\mathrm{SD}^{\leq 1/3}_{\geq 2/3} = (\Pi_Y, \Pi_N)$ is contained in $\mathsf{pr\text{-}SZK} \subseteq \mathsf{pr\text{-}AM}$.[49] Let $V_{\mathrm{SD}}(\sigma, r, w)$ be a polynomial-time verifier algorithm defining an Arthur-Merlin protocol to solve $\mathrm{SD}^{\leq 1/3}_{\geq 2/3}$; we may insist that V_{SD} possesses perfect completeness (although this is not essential) and has soundness $\leq 2^{-n'}$ on input $\sigma \in \Pi_Y \cup \Pi_N$ of length $|\sigma| = n'$. Let $R(n'), W(n') \leq \mathrm{poly}(n')$ be the expected lengths of r, w respectively for σ of length n'.

The interaction: For each $j \in M$, P produces the string $\sigma^j = \left\langle C_{\overline{\psi}}^{\mathbf{Expt}}, C_{\overline{\psi}}^{\mathbf{Expt}'(j;1)} \right\rangle$ as described above. P then runs the protocol V_{SD} in parallel for each σ^j, generating an independent random string $r^j \in \{0,1\}^{R(n)}$ for each $j \in M$. For each such j, Prover is asked to provide $w^j \in \{0,1\}^{W(n)}$. For each $j \in M$, P computes $e_j := V_{\mathrm{SD}}(\sigma^j, r^j, w^j) \in \{0,1\}$.

If there are at least $s = \lfloor 2^{h/18.5} \rfloor$ indices $j \in M$ for which $e_j = 0$, then P outputs "$\perp$". Otherwise P outputs the list consisting of 0^h along with all $v \in \{0,1\}^h$ of Hamming weight $h/2$ for which we have $e_{\mathrm{num}(v)} = 0$. This list contains at most s strings; P pads the output list to size exactly s by duplicating strings if necessary.

Correctness analysis: Now we prove that P is a $(h(n), s(n))$-promise list-enumeration protocol. The bound on the output list size is immediate from our construction; we will argue the other required properties hold.

The next claim is the key to showing that there is always a Prover strategy that forces most of the bits e_j to equal 1. The claim follows from the same ideas used to prove our "distributional stability" lemma (Lemma 6.2). However, direct application of that lemma would be too slack for our application here.

In the following, we use $\mathbf{R}$ to denote the random variable describing the output of $\mathbf{Expt}$. For $b \in \{0,1\}$, let $\mathbf{R}^{(j;b)}$ denote the output of $\mathbf{Expt}'(j;b)$.

Claim 9.13. *For $j \in M$, let*

$$\delta_j \ := \left|\left| \mathbf{R}^{(j;1)} - \mathbf{R} \right|\right|_{\mathrm{stat}} ,$$

and say that $j \in M$ is influential *if $\delta_j > .01$. Let $M^* \subseteq M$ denote the influential indices. Then for sufficiently large n, we have $|M^*| \leq n^{53K}$.*

Proof. The T input formulas to $\mathbf{Expt}$ are each of length $\tilde{n}$. Recall that the f-compression reduction R, on $T = n^{1000K} = (n^2)^{500K} = t_1(\tilde{n})$ inputs of length $\tilde{n}$, is guaranteed to produce an output of length at most $t_2(\tilde{n}) = n^{2K}$. Thus, with $(X_j)_{j \in M}$ as in $\mathbf{Expt}$, we have $I\left((X_j)_{j \in M}; \mathbf{R}\right) \leq n^{2K}$. For each $j \in M$, let Y_j be distributed as X_j but independent of $\mathbf{R}$. By Lemma 4.4 and Fact 4.6,

$$\sum_{j \in M} D_{\mathrm{KL}}\left((X_j, \mathbf{R}) || (Y_j, \mathbf{R})\right) \ \leq \ n^{2K} \ . \tag{27}$$

[49] It is important for our work here that the "Yes" case Π_Y is that in which the two circuits' distributions are *close* in statistical distance.

From the definitions of D_{KL} and of $\mathbf{Expt}'(j;1)$, and using the nonnegativity of D_{KL} and the chain rule for divergence (see Section 4.1), we have

$$D_{\mathrm{KL}}\left((X_j,\mathbf{R})||(Y_j,\mathbf{R})\right) \;\geq\; \Pr[X_j=1]\cdot D_{\mathrm{KL}}\left((1,\mathbf{R}^{(j;1)})||(1,\mathbf{R})\right) \;=\; \Pr[X_j=1]\cdot D_{\mathrm{KL}}\left(\mathbf{R}^{(j;1)}||\mathbf{R}\right) .$$

We also recall that in **Expt**, we have (for each $j\in M$)

$$\Pr[X_j=1] \;\geq\; T^{.95}/(2|M|) \;\geq\; T^{.95}/(2T) \;=\; .5n^{-50K}$$

(for large n). Combining our work gives

$$\sum_{j\in M} D_{\mathrm{KL}}\left(\mathbf{R}^{(j;1)}||\mathbf{R}\right) \;\leq\; 2n^{50K}\cdot n^{2K} \;=\; 2n^{52K} . \tag{28}$$

Using Pinsker's inequality (Theorem 4.7), we note that for each $j\in M^*$, we have $D_{\mathrm{KL}}\left(\mathbf{R}^{(j;1)}||\mathbf{R}\right)\geq (.01)^2 = .0001$. Combining this with Eq. (28), we find that (for large n) $|M^*| < 2\cdot 10^4 n^{52K} < n^{53K}$. □

Now for analysis, let us fix any input formulas $\overline{\psi}=(\psi^1,\ldots,\psi^h)$ to P, each of description length n; for the present part of the analysis we do not need to make any assumption about their satisfiability status. It follows immediately from Claim 9.13 that there are at least $|M|-n^{53K}$ indices $j\in M$ for which the string σ^j produced by P lies in the "yes" set Π_Y of $\mathrm{SD}^{\leq 1/3}_{\geq 2/3}$. As the Arthur-Merlin protocol V_{SD} has perfect completeness, it follows that for any setting to the challenge strings $(r^j)_{j\in M}$, there exist responses $(w^j)_{j\in M}$ such that $e_j = V_{\mathrm{SD}}(\sigma^j,r^j,w^j)=1$ for at least $|M|-n^{53K}$ indices $j\in M$. This is greater than $|M|-s$ since, for sufficiently large n, we have

$$s \;\geq\; 2^{h/18.5}-1 \;>\; 2^{(999/18.5)K\log n}-1 \;=\; n^{54K}-1$$

(where we used our bound $h\geq 999K\log_2 n$). Thus for any $\overline{\psi}$ where n is sufficiently large, there is always a Prover strategy that prevents P from ever outputting "$\perp$".

For the next part of the analysis, again fix the input formulas $\overline{\psi}=(\psi^1,\ldots,\psi^h)$, and let

$$\overline{b}^* \;:=\; \left(\chi_{\mathrm{SAT}}(\psi^1),\ldots,\chi_{\mathrm{SAT}}(\psi^h)\right) . \tag{29}$$

Our remaining task is to show that if $\overline{\psi}$ contains either 0 or $h/2$ satisfiable formulas, then Prover cannot (except with low probability) cause P to output a nonempty list which does *not* contain $\overline{b}^*$. If $||\overline{b}^*||=0$ (all $\psi^{h'}$ are unsatisfiable), then this is immediate, since by construction, any nonempty list output by our protocol P automatically includes 0^h. Thus, for the remainder of the proof we fix $\overline{\psi}=(\psi^1,\ldots,\psi^h)$ and assume

$$||\overline{b}^*|| \;=\; h/2 , \qquad \text{letting} \qquad j^* \;:=\; \mathrm{num}(\overline{b}^*) \;\in\; M .$$

For each $j\in M$, we define

$$B'_j \;:=\; B_j\setminus J ,$$

a nonempty set. (These may not all be distinct.)

Claim 9.14. *1. Let $(\phi^1,\ldots,\phi^T)$ be the (random) formulas generated in* **Expt**, *and for $i\in[T]$ let $u_i := \chi_{\mathrm{SAT}}(\phi^i)$. Let $\mathbf{u}=(u_1,\ldots,u_T)$. For sufficiently large values of n, with probability at least .99 we have $f(\mathbf{u})=f(0^T)$.*

2. *Now let* $(\phi^1, \ldots, \phi^T)$ *be the (random) formulas generated in* $\mathbf{Expt}'(j^*; 1)$. *For* $i \in [T]$ *let* $u'_i := \chi_{\mathrm{SAT}}(\phi^i)$. *Let* $\mathbf{u}' = (u'_1, \ldots, u'_T)$. *For sufficiently large values of* n, *with probability at least* .99 *we have* $f(\mathbf{u}') = \neg f(0^T)$.

Proof. **(1.)** As $||\bar{b}^*|| = h/2$, we see from the definition of the formulas Γ'_i that, for $i \in J$, the formula $\phi^i = \Gamma'_i$ is satisfiable exactly if $i \in B_{j^*}$. For $i \in [T] \setminus J$, with reference to steps 3-4 of **Expt**, we see that $u_i = 1$ exactly if there exists a $j \in M$ for which $X_j = 1$ and $i \in B'_j$. Let $z := \mathbf{1}_{B_{j^*} \cap J} \in \{0,1\}^T$ be the characteristic vector for $B_{j^*} \cap J$. Then our observations yield

$$\mathbf{u} \;=\; z \vee \bigvee_{j \in M} X_j \cdot \mathbf{1}_{B'_j} \; .$$

By an application of Lemma 9.4 (with $\mathbf{u}$ playing the role of z'), we have

$$\Pr[f(\mathbf{u}) \neq f(z)] \;\leq\; .01 \; .$$

Finally, note that $B_{j^*} \cap J$ is a *proper* subset of B_{j^*} (since $B_{j^*} \not\subseteq J$), and B_{j^*} is a *minimal* sensitive block for f with respect to input 0^T (by condition 2 of the eligibility property for f). Thus $f(z) = f(0^T)$, and the Claim follows.

(2.) For $i \in J$, we see with reference to the definition of Γ'_i and the fact $[j^* = \mathrm{num}(\bar{b}^*)]$, that the formula $\phi^i = \Gamma'_i$ is satisfiable (and $u'_i = 1$) exactly if $i \in B_{j^*}$. For $i \in [T] \setminus J$, we have $u'_i = 1$ exactly if there exists a $j \in M$ for which $X_j = 1$ and $i \in B'_j$. Recall that X_{j^*} is fixed to 1 in $\mathbf{Expt}'(j^*; 1)$. Letting $\hat{z} := \mathbf{1}_{B_{j^*}}$, it follows that

$$\mathbf{u}' \;=\; \hat{z} \vee \left(Y_{j^*} \cdot \mathbf{1}_{B'_{j^*}} \vee \bigvee_{j \in M \setminus \{j^*\}} X_j \cdot \mathbf{1}_{B'_j} \right)$$

where we introduce a new variable Y_{j^*} that is independent of and identically distributed to $(X_j)_{j \in M \setminus \{j^*\}}$. ($Y_{j^*}$'s outcome is actually irrelevant to $\mathbf{u}'$, and is introduced only to match the conditions of Lemma 9.4.) Applying Lemma 9.4 to the modified collection of random variables which substitutes Y_{j^*} for X_{j^*}, we infer that with probability $\geq .99$ we have

$$f(\mathbf{u}') \;=\; f(\hat{z}) \; .$$

Now $\hat{z}$ is the characteristic vector of the entire block B_{j^*}, which is sensitive for f on input 0^T, so $f(z) = \neg f(0^T)$. This completes the proof of Claim 9.14. □

Let $\beta := f(0^T) \in \{0,1\}$ (a value which may not be efficiently computable). It follows from item 1 of Claim 9.14 and the f-compression property of R (with $\xi(n) < .1$) that, with probability at least $.99(.9) > .89$ over **Expt** we have

$$L'(\mathbf{R}) \;=\; \beta \; .$$

Similarly, for $j^* = \mathrm{num}(\bar{b}^*)$, then from item 2 of Claim 9.14 we find that probability greater than .89 over $\mathbf{Expt}'(j; 1)$ we have $L'(\mathbf{R}^{(j^*;1)}) = \neg\beta$. It follows that

$$\left|\left| \mathbf{R}^{(j^*;1)} - \mathbf{R} \right|\right|_{\mathrm{stat}} \;>\; 1 - 2 \cdot .11 \;>\; 2/3 \; .$$

Thus, the string $\sigma^{j^*} = \left\langle C_{\overline{\psi}}^{\mathbf{Expt}}, C_{\overline{\psi}}^{\mathbf{Expt}'(j;1)} \right\rangle$ produced by our protocol P is in the "no" set Π_N of the promise problem $\mathrm{SD}^{\leq 1/3}_{\geq 2/3}$. From the soundness property of V_{SD} it follows that, with probability at least $1 - 2^{-n'} \geq 1 - 2^{-n}$ over the random string r^{j^*}, all possible choices of the proof string w^{j^*} yield

$$e_{j^*} \;=\; V_{\mathrm{SD}}(\sigma^{j^*}, r^{j^*}, w^{j^*}) = 0 \;.$$

Notice that P can only output a nonempty list of strings $(v^1, \ldots, v^s) \in \{0,1\}^{s \times h}$ that does *not* contain $\overline{b}^*$ if $e_{\mathrm{num}(\overline{b}^*)} = e_{j^*} = 1$. We conclude that for any Prover strategy in P, the probability this occurs is at most 2^{-n}. We have shown that P is an $(h(n), s(n))$-list-enumeration protocol. This completes the proof of Theorem 9.2.

9.6 Application to AND- and OR-compression

In this section we use Theorem 9.2 to rule out strong compression for OR(SAT) and or AND(SAT). The only further ingredient needed is the following lemma, connecting these compression tasks to the monotone address function mADDR from Definition 9.3—an eligible function.

Lemma 9.15. *Fix integers $C, c \geq 1$. Let $f : \{0,1\}^* \to \{0,1\}$ be the monotone address function* mADDR*.*

1. *Suppose that there is a PPT OR-compression reduction R for $L =$ SAT, with parameters*

$$t_1(n) \;=\; n^C \;, \quad t_2(n) \;\leq\; n^c \;, \quad \xi(n) \;\leq\; .1 \;,$$

and some target language L'.

Then there is a PPT f-compression reduction R' for SAT *with parameters*

$$t_1'(n) \;=\; n^C \;, \quad t_2'(n) \;\leq\; n^{2c} \;, \quad \xi(n) \;\leq\; .1 \;,$$

and target language L'.

2. *The same conclusion from item 1 holds, if there is a PPT AND-compression reduction R for $L =$ SAT, with parameters $(t_1(n), t_2(n), \xi(n))$ and target language L' as in item 1.*

The parameters $t_1'(n), t_2'(n)$ in the statement above are suboptimal for ease of presentation. The idea of the proof of Lemma 9.15, item 1 is to express $f =$ mADDR as a (monotone) DNF, and to transform an input to $f \circ$ SAT to an input to OR(SAT) in a way induced by this DNF representation. Item 2 is similar, but uses a monotone CNF representation of f. All we be will essentially using about f is that the clauses in these monotone representations are short, not too numerous, and efficiently computable.

Proof of Lemma 9.15. **(1.)** Given an input tuple $(\langle \psi^1 \rangle, \ldots, \langle \psi^{n^C} \rangle) \in \{0,1\}^{n^C \times n}$ (here we take $t_1'(n) = n^C$ SAT instances of length n), we first make some definitions and observations that will enable us to define and analyze the behavior of our f-compression reduction R' applied to this input. First, let k be the largest value for which $2^k + 2k \leq n^C$; we can assume $k > 1$, and we have $2^k + 2k \geq .5n^C$. We may safely discard all $\langle \psi^j \rangle$ with $j > 2^k + 2k$ since, by the definition of the

"padded" function $f = \mathrm{mADDR}$, they are irrelevant to our compression task. For ease of reference, we relabel our $2^k + 2k$ relevant formulas as

$$(\phi^0, \ldots, \phi^{2^k-1},\ \pi^0, \ldots, \pi^{2k-1})\ ,$$

and define the (unknown) values $x^* = (x^*_0, \ldots, x^*_{2^k-1}), y^* = (y^*_0, \ldots, y^*_{2k-1})$ by

$$x^*_\ell\ :=\ \chi_{\mathrm{SAT}}(\phi^\ell)\ , \qquad y^*_j\ :=\ \chi_{\mathrm{SAT}}(\pi^j)\ ,$$

so that

$$(f \circ \mathrm{SAT})(\phi^0, \ldots, \phi^{2^k-1},\ \pi^0, \ldots, \pi^{2k-1})\ =\ \mathrm{mADDR}_{2^k}(x^*, y^*)\ .^{50} \tag{30}$$

For use in the reduction, we next describe the DNF representation of the monotone function $\mathrm{mADDR}_{2^k}(x, y)$. For each $w = (w_0, \ldots, w_{k-1}) \in \{0,1\}^k$, define the set $S_w \subseteq [0, 2k-1]$ by

$$S_w\ :=\ \{2\ell + 1 : w_\ell = 1\} \cup \{2\ell : w_\ell = 0\}\ .$$

Also, let $T_1, \ldots, T_{\binom{2k}{k+1}} \subseteq [0, 2k-1]$ be a enumeration of all size-$(k+1)$ subsets of $[0, 2k-1]$, computable in time $\mathrm{poly}(2^k) \le \mathrm{poly}(t)$. Using the notation $v(w) := \sum_{\ell \in [0,k-1]} w_\ell \cdot 2^\ell$, we have the DNF representation

$$\mathrm{mADDR}_{2^k}(x, y)\ =\ \left[\bigvee_{w \in \{0,1\}^k} \left(x_{v(w)} \wedge \bigwedge_{j \in S_w} y_j\right)\right]\ \vee\ \left[\bigvee_{1 \le p \le \binom{2k}{k+1}} \left(\bigwedge_{j \in T_p} y_j\right)\right] \tag{31}$$

(valid for all x, y), whose clauses correspond to the $2^k + \binom{2k}{k+1}$ different minimal inputs to mADDR_{2^k} causing it to output 1. Note that the value $D := 2^k + \binom{2k}{k+1}$ is at most $2^{2k} \le n^{2C} = (n^2)^C = t_1(\tilde{n})$.

With this representation in mind, we define some SAT instances. For each $w \in \{0,1\}^k$, define a Boolean formula $\Pi_w := \phi_{v(w)} \wedge \bigwedge_{j \in S_w} \pi_j$. For $1 \le p \le \binom{2k}{k+1}$, define $\Gamma_p := \bigwedge_{j \in T_p} \pi_j$. Note that

$$[\Pi_w \in \mathrm{SAT}] \iff \left[x^*_{v(w)} \wedge \bigwedge_{j \in S_w} y^*_j = 1\right] \quad \text{and} \quad [\Gamma_p \in \mathrm{SAT}] \iff \left[\bigwedge_{j \in T_p} y^*_j = 1\right]\ ,$$

so that $\mathrm{mADDR}_{2^k}(x^*, y^*) = 1$ exactly if at least one Γ_p or Π_w is satisfiable. Then using Eq. (30),

$$\left(\bigvee_{w \in \{0,1\}^k} \chi_{\mathrm{SAT}}(\Pi_w)\right) \vee \left(\bigvee_{1 \le p \le \binom{2k}{k+1}} \chi_{\mathrm{SAT}}(\Gamma_p)\right)\ =\ (f \circ \mathrm{SAT})(\phi^0, \ldots, \phi^{2^k-1},\ \pi^0, \ldots, \pi^{2k-1})\ . \tag{32}$$

Now by construction, each Π_w and Γ_p is an AND of at most $k + 1 = O(\log_2 n)$ formulas, each of size at most n, so for large n each Π_w and Γ_p can be given a description of size exactly $\tilde{n} := n^2$ (padding as necessary).

The reduction R' acts as follows. It first constructs descriptions $(\langle \Pi_w \rangle)_w, (\langle \Gamma_p \rangle)_p$ as above, combining them into a single list, and pads out this list with $t_1(\tilde{n}) - D \ge 0$ copies of the string

[50] In this proof we use the definition $(f \circ \mathrm{SAT})(\Gamma^1, \ldots, \Gamma^N) := f(\chi_{\mathrm{SAT}}(\Gamma^1), \ldots, \chi_{\mathrm{SAT}}(\Gamma^N))$, a slight abuse of notation since $f \circ \mathrm{SAT}$ is used elsewhere to denote a parametrized problem.

$\langle \psi_{unsat} \rangle \in \{0,1\}^{\tilde{n}}$, describing a fixed unsatisfiable formula ψ_{unsat}. Let $\overline{\Psi} \in \{0,1\}^{t_1(\tilde{n}) \times \tilde{n}}$ denote our resulting list. Our reduction R' then outputs the (possibly randomized) result of applying R to this list:

$$z \;:=\; R\left(\overline{\Psi}\right) \;.$$

The analysis of R' is simple given our work thus far. First, R' is clearly polynomial-time. By the compression guarantee of R applied to the input-tuple $\overline{\Psi}$, we have $|z| \leq t_2(\tilde{n}) \leq (n^2)^c = n^{2c}$, so we may choose $t'_2(\cdot)$ as claimed. For correctness, note that the input $\overline{\Psi}$ to R satisfies $(\mathrm{OR} \circ \mathrm{SAT})(\overline{\Psi}) = 1$ exactly if some Π_w or some Γ_p is satisfiable; by Eq. (32), this occurs exactly if

$$1 \;=\; (f \circ \mathrm{SAT})(\phi^0, \ldots, \phi^{2^k-1},\; \pi^0, \ldots, \pi^{2k-1}) \;=\; (f \circ \mathrm{SAT})(\psi^1, \ldots, \psi^{n^C}) \;.$$

As R is a compression reduction for $OR(\mathrm{SAT})$ with $\xi(n) \leq .1$, we conclude that $[L'(z) = (f \circ \mathrm{SAT})(\psi^1, \ldots, \psi^{n^C})]$ holds with probability at least .9. This proves item 1.

(2.) For this item, we fix notation as before and observe the CNF representation

$$\mathrm{mADDR}_{2^k}(x,y) \;=\; \left[\bigwedge_{w \in \{0,1\}^k} \left(x_{v(w)} \vee \bigvee_{j \in [2k] \setminus S_w} y_j \right) \right] \;\wedge\; \left[\bigwedge_{1 \leq p \leq \binom{2k}{k+1}} \left(\bigvee_{j \in T_p} y_j \right) \right] \;. \tag{33}$$

Using this representation, we reduce from an $f \circ \mathrm{SAT}$ instance to an AND(SAT) instance in perfect analogy with item 1, and again apply the reduction R (which now is an AND-compression reduction for SAT; this time we pad the input-tuple to R with copies of a *satisfiable* formula ψ_{sat}). □

Theorem 9.16. *Let $c \geq 1$ be an integer. Suppose there is either an OR-compression or an AND-compression reduction for* SAT*, with parameters*

$$t_1(n) \;=\; n^{1000c} \;, \quad t_2(n) \;\leq\; n^c \;, \quad \xi(n) \;\leq\; .1 \;,$$

and any target language L'. Then, NP ⊆ coAM*.*

Proof. Let $f := \mathrm{mADDR}$ as in Lemma 9.15. Under either assumption, we can apply one of the two items of that Lemma to obtain an f-compression reduction for SAT, with parameters

$$t'_1(n) \;=\; n^{1000c} \;, \quad t'_2(n) \;\leq\; n^{2c} \;, \quad \xi'(n) \;\leq\; .1 \;,$$

and target language L'. Recall that f is eligible, as shown in Section 9.1. We then apply Theorem 9.2, with $K := 2c$, to conclude NP ⊆ coAM. □

10 Questions for further study

1. Can we extend the limitations we show on efficient compression for AND(SAT) and OR(SAT), to give corresponding lower bounds on the cost of solving these problems in the *oracle communication model* studied by Dell and Van Melkebeek [DvM10]? These authors were able to extend the lower bounds of [FS11] for OR(SAT) to this more general setting. Proceeding by straightforward analogy in our case seems to fail, however.

2. Using our results on the infeasibility of compression for AND(SAT), can we extend the work of [DvM10] to prove new kernel-size lower bounds for interesting problems *with* polynomial kernels, under the assumption $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$?

3. Can we obtain a tighter quantitative understanding of the limits to efficient f-compression of NP-complete languages, where f is a combining function other than OR or AND? The case $f = \bigvee_{i=1}^{m} \left(\bigwedge_{j=1}^{m} x^{i,j} \right)$ is an interesting candidate for study. We have also left open the feasibility of strong f-compression for certain functions f with very low block sensitivity.

4. Can we find other applications for the Disguising-Distribution Lemma?

Acknowledgements

I thank Scott Aaronson, Hans Bodlaender, Holger Dell, Lance Fortnow, Russell Impagliazzo, James Lee, Dieter van Melkebeek, Ashwin Nayak, Karolina Soltys, Salil Vadhan, Thomas Vidick, Avi Wigderson, Ryan Williams, and several reviewers for helpful comments. Thanks especially to Russell, Ashwin, and Salil for allowing me to include their alternative proof suggestions.

References

[AA94] Manindra Agrawal and Vikraman Arvind. Polynomial time truth-table reductions to p-selective sets. In *Structure in Complexity Theory Conference*, pages 24–30, 1994.

[Aar08] Scott Aaronson. Quantum certificate complexity. *J. Comput. Syst. Sci.*, 74(3):313–322, 2008.

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[ABG03] Amihood Amir, Richard Beigel, and William I. Gasarch. Some connections between bounded query classes and non-uniform complexity. *Inf. Comput.*, 186(1):104–139, 2003. Earlier version in Structure in Complexity Theory '90.

[Adl78] Leonard M. Adleman. Two theorems on random polynomial time. In *19th IEEE FOCS*, pages 75–83, 1978.

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. Syst. Sci.*, 42(3):327–345, 1991.

[ALRS98] Sigal Ar, Richard J. Lipton, Ronitt Rubinfeld, and Madhu Sudan. Reconstructing algebraic functions from mixed data. *SIAM J. Comput.*, 28(2):487–510, 1998. Earlier version in FOCS '92.

[Alt94] Ingo Althöfer. On sparse approximations to randomized strategies and convex combinations. *Linear Algebra and its Applications*, 199, Supplement 1(0):339 – 355, 1994.

[BDFH09] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009. Earlier version in ICALP '08.

[BdW02] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002.

[BFT97] Harry Buhrman, Lance Fortnow, and Leen Torenvliet. Six hypotheses in search of a theorem. In *IEEE Conference on Computational Complexity*, pages 2–12, 1997.

[BG81] Charles H. Bennett and John Gill. Relative to a random oracle A, P^A != NP^A != co-NP^A with probability 1. *SIAM J. Comput.*, 10(1):96–113, 1981.

[BH08] Harry Buhrman and John M. Hitchcock. NP-hard sets are exponentially dense unless coNP $\subseteq$ NP/poly. In *23rd IEEE Conference on Computational Complexity*, pages 1–7, 2008.

[BJK11a] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Cross-composition: A new technique for kernelization lower bounds. In *STACS*, pages 165–176, 2011.

[BJK11b] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernel bounds for path and cycle problems. In *IPEC*, pages 145–158, 2011.

[BJK11c] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Preprocessing for treewidth: A combinatorial analysis through kernelization. In *38th ICALP*, pages 437–448, 2011.

[BKS95] Richard Beigel, Martin Kummer, and Frank Stephan. Approximable sets. *Inf. Comput.*, 120(2):304–314, 1995. Earlier version in IEEE Structure in Complexity Theory '94.

[BTY11] Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011. Earlier version in ESA '09.

[Buh] Harry Buhrman. Personal communication.

[CCDF97] Liming Cai, Jianer Chen, Rodney G. Downey, and Michael R. Fellows. Advice classes of parameterized tractability. *Annals of Pure and Applied Logic*, 84(1):119 – 138, 1997.

[CFM11] Yijia Chen, Jörg Flum, and Moritz Müller. Lower bounds for kernelizations and other preprocessing procedures. *Theory Comput. Syst.*, 48(4):803–839, 2011.

[CKR95] Richard Chang, Jim Kadin, and Pankaj Rohatgi. On unique satisfiability and the threshold behavior of randomized reductions. *J. Comput. Syst. Sci.*, 50(3):359–373, 1995.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley-Interscience, 2nd edition, 2006.

[DF99] R. G. Downey and M.R. Fellows. *Parametrized Complexity.* Springer (Monographs in Computer Science), 1st edition, 1999.

[DLS09] Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Incompressibility through colors and IDs. In *36th ICALP*, pages 378–389, 2009.

[DM12] Holger Dell and Dániel Marx. Kernelization of packing problems. In *23rd ACM-SIAM SODA*, pages 68–81, 2012.

[DvM10] Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *42nd ACM STOC*, pages 251–260, 2010.

[FGM+89] Martin Fürer, Oded Goldreich, Yishay Mansour, Michael Sipser, and Stathis Zachos. On completeness and soundness in interactive proof systems. *Advances in Computing Research*, 5:429–442, 1989.

[FHT03] Alexei A. Fedotov, Peter Harremoës, and Flemming Topsøe. Refinements of pinsker's inequality. *IEEE Transactions on Information Theory*, 49(6):1491–1498, 2003.

[For87] Lance Fortnow. The complexity of perfect zero-knowledge (extended abstract). In Alfred V. Aho, editor, *19th ACM STOC*, pages 204–209, 1987.

[FS11] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011. Earlier version in STOC '08.

[GN07] Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *18th ACM STOC*, pages 59–68, 1986.

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of reed-solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999. Earlier version in FOCS '98.

[GSV98] Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *30th ACM STOC*, pages 399–408, 1998.

[GV11] Oded Goldreich and Salil P. Vadhan. On the complexity of computational problems regarding distributions (a survey). *Electronic Colloquium on Computational Complexity (ECCC)*, TR11-004:4, 2011.

[HN10] Danny Harnik and Moni Naor. On the compressibility of NP instances and cryptographic applications. *SIAM J. Comput.*, 39(5):1667–1713, 2010.

[HW12] Danny Hermelin and Xi Wu. Weak compositions and their applications to polynomial lower bounds for kernelization. In *23rd ACM-SIAM SODA*, pages 104–113, 2012.

[JJUW11] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. QIP = PSPACE. *J. ACM*, 58(6):30, 2011. Earlier version in STOC '10.

[KdW04] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. Comput. Syst. Sci.*, 69(3):395–420, 2004. Earlier version in STOC '03.

[KNTSZ07] Hartmut Klauck, Ashwin Nayak, Amnon Ta-Shma, and David Zuckerman. Interaction in quantum communication. *IEEE Transactions on Information Theory*, 53(6):1970–1982, 2007.

[Kra12] Stefan Kratsch. Co-nondeterminism in compositions: a kernelization lower bound for a Ramsey-type problem. In *23rd ACM-SIAM SODA*, pages 114–122, 2012.

[KW00] Alexei Kitaev and John Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In *32nd ACM STOC*, pages 608–617, 2000.

[KW12] Stefan Kratsch and Magnus Wahlström. Compression via matroids: a randomized polynomial kernel for odd cycle transversal. In *23rd ACM-SIAM SODA*, pages 94–103, 2012.

[LMM03] Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *4th ACM Conference on Electronic Commerce*, pages 36–41, 2003.

[LY94] Richard J. Lipton and Neal E. Young. Simple strategies for large zero-sum games with applications to complexity theory. In *26th ACM STOC*, pages 734–740, 1994.

[Nay99a] Ashwin Nayak. *Lower bounds for Quantum Computation and Communication*. PhD thesis, University of California, Berkeley, 1999.

[Nay99b] Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In *40th IEEE FOCS*, pages 369–377, 1999.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[Nis91] Noam Nisan. CREW PRAMs and decision trees. *SIAM J. Comput.*, 20(6):999–1007, 1991. Earlier version in STOC '89.

[Ogi95] Mitsunori Ogihara. Polynomial-time membership comparable sets. *SIAM J. Comput.*, 24(5):1068–1081, 1995.

[Oka00] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *J. Comput. Syst. Sci.*, 60(1):47–108, 2000. Earlier version in STOC '96.

[OP04] Masanori Ohya and Denes Petz. *Quantum Entropy and its Use*. Texts and Monographs in Physics. Springer-Verlag, Heidelberg, 2nd edition, 2004.

[Pos11] Alexey Pospelov. Faster polynomial multiplication via discrete Fourier transforms. In *CSR*, pages 91–104, 2011.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. Earlier version in STOC '95.

[Reg12] Oded Regev. Entropy-based bounds on dimension reduction in l_1. *Israel Journal of Mathematics*, 2012. arXiv:1108.1283.

[RW09] Mark D. Reid and Robert C. Williamson. Generalised pinsker inequalities. In *COLT*, 2009.

[Sha10] Ronen Shaltiel. Derandomized parallel repetition theorems for free games. In *IEEE Conference on Computational Complexity*, pages 28–37, 2010.

[Siv99] D. Sivakumar. On membership comparable sets. *J. Comput. Syst. Sci.*, 59(2):270–280, 1999. Earlier version in CCC '98.

[Sud96] Madhu Sudan. Maximum likelihood decoding of reed solomon codes. In *FOCS*, pages 164–172, 1996.

[SV03] Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003.

[SV08] Pranab Sen and Srinivasan Venkatesh. Lower bounds for predecessor searching in the cell probe model. *J. Comput. Syst. Sci.*, 74(3):364–385, 2008. Earlier version in CCC '03.

[Tal13] Avishay Tal. Properties and applications of boolean function composition. In *ITCS*, pages 441–454, 2013.

[vL99] J. H. van Lint. *Introduction to coding theory*, volume 86 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 3rd edition, 1999.

[VV86] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986. Earlier version in STOC '85.

[VW85] Uzi Vishkin and Avi Wigderson. Trade-offs between depth and width in parallel computation. *SIAM J. Comput.*, 14(2):303–314, 1985. Earlier version in FOCS '83.

[Wat02] John Watrous. Limits on the power of quantum statistical zero-knowledge. In *43rd IEEE FOCS*, pages 459–468, 2002.

[Wat03] John Watrous. PSPACE has constant-round quantum interactive proof systems. *Theor. Comput. Sci.*, 292(3):575–588, 2003. Earlier version in FOCS '99.

[Wat09] John Watrous. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, 39(1):25–58, 2009. Earlier version in STOC '06.

[Weg91] Ingo Wegener. *The Complexity of Boolean Functions*. Wiley Teubner on Applicable Theory in Computer Science. John Wiley and Sons Ltd., 1991.

[Yap83] Chee-Keng Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–300, 1983.

A Alternative proofs of distributional stability

A.1 A proof based on Raz's lemma

R. Impagliazzo and S. Vadhan noted a similarity between distributional stability lemmas and a probabilistic lemma implicit in work of Raz [Raz98]. Vadhan pointed us to the following convenient form, given by Shaltiel in [Sha10, Lemma 3.1]:

Lemma A.1. *There is a $c > 0$ for which the following holds. Let $X^1, \ldots, X^t$ be i.i.d. random variables, and T an event with $\Pr[T] \geq 2^{-\Delta}$. Then for the conditioned variables $X^j_{[T]}$ we have*

$$\mathbb{E}_{j \sim \mathcal{U}_{[t]}}[||X^j_{[T]} - X^j||_{\text{stat}}] \;\leq\; \varepsilon := \sqrt{\frac{c\Delta}{t}} \;.$$

With this lemma we can derive a distributional stability result as follows. Suppose $R : S^t \to \{0,1\}^{\leq t'}$ is given, and consider independent inputs $X^j \sim \mathcal{D}$ to R. We let $\mathbf{R}$ denote the random output value. For any output z of R, Lemma A.1 above implies that

$$\mathbb{E}_{i \sim \mathcal{U}_{[n]}}[||X^j_{[\mathbf{R}=z]} - X^j||_{\text{stat}}] \;\leq\; \sqrt{c \log_2(1/\Pr[\mathbf{R}=z])/t} \;.$$

Taking expectations over $z \sim \mathbf{R}$ and using Jensen's inequality,

$$\begin{aligned}
\mathbb{E}_{z \sim \mathbf{R}, i \sim \mathcal{U}_{[n]}}[||X^j_{[\mathbf{R}=z]} - X^j||_{\text{stat}}] \;&\leq\; \mathbb{E}_z\left[\sqrt{c \log_2(1/\Pr[\mathbf{R}=z])/t}\right] \\
&\leq\; \sqrt{\mathbb{E}_z\left[c \log_2(1/\Pr[\mathbf{R}=z])/t\right]} \\
&= \sqrt{c \cdot H(\mathbf{R})/t} \\
&\text{(by the definition of Shannon entropy)} \\
&< \sqrt{c(t'+1)/t} \;.
\end{aligned}$$

Let $\mathbf{R}' = R(Y^1, \ldots, Y^t)$ denote a sample of $\mathbf{R}$ based on inputs $Y^1, \ldots, Y^t \sim \mathcal{D}^{\otimes t}$ that are independent of $X^1, \ldots, X^t$. Now the crucial observation is that, for each $j \in [t]$, we have the chain of equalities

$$\mathbb{E}_z[||X^j_{[\mathbf{R}=z]} - X^j||_{\text{stat}}] \;=\; ||(X^j, \mathbf{R}) - (X^j, \mathbf{R}')||_{\text{stat}} \;=\; \mathbb{E}_{x^j \sim \mathcal{D}}[||\mathbf{R}_{[X^j = x^j]} - \mathbf{R}||_{\text{stat}}] \;.$$

Each equality follows from the "distinguishability interpretation" of statistical distance. Combining we get

$$\mathbb{E}_{x^j \sim \mathcal{D}}[||\mathbf{R}_{[X^j = x^j]} - \mathbf{R}||_{\text{stat}}] \;<\; \sqrt{c(t'+1)/t}) \;,$$

which is comparable to what we got from the previous approach (up to constant factors; here we have assumed i.i.d. variables X^j, but this is not essential for this approach).

A.2 A proof based on the Average Encoding Theorem

The Average Encoding Theorem of [KNTSZ07] is a tool in quantum information theory, that has the following classical analogue. (See [SV08, Fact 5], where a purely classical proof is given. We restate the result slightly, converting from ℓ^1 distance to statistical distance.)

Theorem A.2. *Let X, M be random variables. Let Π be the distribution governing M. Then for the conditioned distributions $\Pi_{[X=x]}$ we have*

$$\sum_x \Pr[X = x] \cdot ||\Pi_{[X=x]} - \Pi||_{\text{stat}} \ \leq \ \sqrt{\frac{\ln 2 \cdot I(X;M)}{2}} \ .$$

Using Theorem A.2 along with techniques suggested by Nayak and similar to those in the proof of [KNTSZ07, Theorem 5.4],[51] we can derive a distributional-stability result as follows. Again say we are given $R : S^t \to \{0,1\}^{t'}$, and consider independent inputs $X^j \sim \mathcal{D}$ to R, giving an output distribution denoted $\mathbf{R}$. Applying Theorem A.2 to $X := X^j, M := \mathbf{R}$, we have

$$\mathbb{E}_{x^j \sim X^j}\left[||\mathbf{R}_{[X^j=x]} - \mathbf{R}||_{\text{stat}}\right] \ \leq \ \sqrt{\frac{\ln 2 \cdot I(X^j;\mathbf{R})}{2}} \ .$$

Averaging over $j \in [t]$ and applying Jensen's inequality and Lemma 4.4, we obtain

$$\begin{aligned}
\mathbb{E}_{j \sim \mathcal{U}_{[t]}, x^j \sim X^j}\left[||\mathbf{R}_{[X^j=x]} - \mathbf{R}||_{\text{stat}}\right] \ &\leq \ \mathbb{E}_{j \sim \mathcal{U}_{[t]}}\left[\sqrt{\frac{\ln 2 \cdot I(X^j;\mathbf{R})}{2}}\right] \\
&\leq \ \sqrt{\frac{\ln 2 \cdot \mathbb{E}_{j \sim \mathcal{U}_{[t]}}\left[I(X^j;\mathbf{R})\right]}{2}} \\
&\leq \ \sqrt{\frac{\ln 2 \cdot (t'+1)}{2t}} \ .
\end{aligned}$$

B Our original distributional stability lemma

In this section we include our original proof of a distributional stability lemma, based on coding-theoretic ideas. This lemma proves a distributional stability result for mappings $R : \{0,1\}^{t \times n} \to \{0,1\}^{\leq t'}$, where $t' + 2 \leq t$. In an earlier draft, we used more complicated ideas to prove complexity upper bounds from AND-compression reductions where $t' = O(t \log t)$. This latter case can now be handled in the same way as the case $t' \ll t$, using the alternative bound on distributional stability provided by Lemma 6.2, item 2.

First, we provide some further needed background.

B.1 Entropy and the unreliability of compressive encodings

It is a basic principle of information theory that one cannot reliably encode a uniformly-generated t-bit message by an encoding of length $t - 2$ or less. (We can save essentially one bit by using a variable-length output.) Below we state and prove a standard claim that generalizes this fact, giving quantitative bounds on the reliability of compressive encoding methods.

Lemma B.1. *Let $t \in \mathbb{N}^+$, and let U be some finite universe. Say we are given a possibly-randomized "encoding" function*

$$Enc(x, y) : \{0,1\}^t \times \{0,1\}^N \to U \ ,$$

[51] (That proof uses a version of the Average Encoding Theorem that treats Hellinger distance rather than statistical distance.)

depending on a "message" input $x \in \{0,1\}^t$ along with a "public randomness" input $y \in \{0,1\}^N$. (Enc may also have additional internal randomness.) Say we are also given a (possibly-randomized, possibly-unreliable) "decoding" function

$$Dec(x,y) : U \times \{0,1\}^N \to \{0,1\}^t \ ,$$

that also has access to the public randomness y.

Suppose X, Y are two independent random variables over $\{0,1\}^t, \{0,1\}^N$ respectively. For $j \in [t]$, let

$$p_j \ := \ \Pr_{X,Y}[Dec_j(Enc(X,Y),Y) \ = \ X_j] \ ,$$

where $X = (X_1, \ldots, X_t)$, and where Dec_j is the j^{th} output bit of Dec.

Let $p_{avg} := \frac{1}{t}\sum_{j=1}^{t} p_j$. Then, we must have

$$H(p_{avg}) \ \geq \ \frac{1}{t}\left(H(X) - \log_2(|U|)\right) \ .$$

Our proof is closely modeled on the proof of a corresponding, but deeper, quantum result [KdW04, Appendix B].[52] To prove Lemma B.1, we will use another basic information-theoretic fact, Fano's inequality:

Lemma B.2 (Fano)**.** *[CT06, Chapter 2] Suppose Z_{in}, Z_{out} are two random variables: Z_{in} an "input message" over some alphabet Σ, and Z_{out} an "output message" over any domain. Let $\widetilde{Z}_{in}$ be a (possibly-randomized) function of Z_{out}, that attempts to recover the value Z_{in}. Let*

$$p_{err} \ := \ \Pr[\widetilde{Z}_{in} \neq Z_{in}] \ .$$

Here the randomness is over the entire experiment. Then, we have

$$H(p_{err}) \ + \ p_{err} \cdot \log_2\left(|\Sigma| - 1\right) \ \geq \ H(Z_{in}|Z_{out}) \ .$$

We only use the case $|\Sigma| = 2$, so the second term on the left-hand side vanishes.

Proof of Lemma B.1. First we ask whether $p_{avg} \geq .5$. If not, we simply negate all of the decoding functions Dec_j, giving the modified average success probability $p'_{avg} = 1 - p_{avg}$, for which $H(p'_{avg}) = H(p_{avg})$. Next, note that the success probabilities p_j are taken over the randomness both in X and in Y (as well as in Enc, Dec). As Y is independent of X, we may non-uniformly fix some setting to Y that maximizes the sum of the conditional success probabilities. Then the re-modified average success probability satisfies $p''_{avg} \geq p'_{avg} \geq .5$. As $H(\cdot)$ is decreasing on $[.5, 1]$, any lower bound proved for $H(p''_{avg})$ will also lower-bound the $H(p_{avg})$ for the original encoding scheme with public randomness. Thus in the remainder of the proof, we assume $p_{avg} \geq .5$ and that the scheme uses no public randomness: our encoding Enc applies to X alone, and our decoding functions apply to the message $Enc(X)$ alone.

The chain rule for conditional entropy and the subadditivity of entropy imply that

$$H(X|Enc(X)) \ = \ \sum_{j=1}^{t} H(X_j|X_1, \ldots, X_{j-1}, Enc(X)) \ \leq \ \sum_{j=1}^{t} H(X_j|Enc(X)) \ . \tag{34}$$

[52] (or, Appendix A in the arxiv version. This part of [KdW04] is itself a rederivation of a result from [Nay99b]; a similar result and proof appears in Nayak's thesis [Nay99a, Theorem 3.2.8].)

Next, we apply Fano's inequality, with $Z_{in} := X_j$, and $Z_{out} := Enc(X)$. Thus in this analysis we simply view $(X_{j'})_{j' \neq j}$ as additional sources of randomness in the encoding process. We let $\widetilde{Z}_{in} := Dec_j(Enc(X))$. X_j is binary—$\Sigma = \{0, 1\}$—so Fano's inequality gives

$$H(p_j) \;=\; H(1 - p_j) \;\geq\; H(X_j | Enc(X)) \;.$$

Summing over j and using Eq. (34),

$$\begin{aligned} \sum_{j=1}^{t} H(p_j) \;&\geq\; H(X|Enc(X)) \\ &\geq\; H(X) - H(Enc(X)) \;, \end{aligned} \tag{35}$$

again using subadditivity. Now, $Enc(X)$ is a message over U, so $H(Enc(X)) \leq \log_2(|U|)$. Also, the function H is concave on $[0, 1]$. Applying these observations to Eq. (35), and using Jensen's inequality, we have

$$H(p_{avg}) \;\geq\; \frac{1}{t} \sum_{j=1}^{t} H(p_j) \;\geq\; \frac{1}{t} \left(H(X) - \log_2(|U|) \right) \;.$$

□

B.2 Bounds on the inverse entropy function

For $\alpha \in [0, 1]$, we will denote by $H_+^{-1}(\alpha)$ the unique H-preimage of α in the range $[.5, 1]$. Similarly, let $H_-^{-1}(\alpha)$ denote the unique H-preimage of α in the range $[0, .5]$. The following bounds on the inverse entropy function are useful in understanding the bounds provided by our original distributional stability lemma (Lemma B.4, to be presented shortly). These bounds on $H_+^{-1}(\cdot)$ are meant to be simple and illustrative, and are not quite best-possible.

Lemma B.3. *We have the following facts:*

1. *If $m > 0$ is sufficiently large, then $H_+^{-1}(1/m) < 1 - 1/(4m \log_2 m)$.*
2. *$H_+^{-1}(1 - \delta) \leq .5 + \frac{\sqrt{\ln 2}}{2} \sqrt{\delta} + O(\delta^{3/2})$.*

Proof. **(1.)** First consider any value $p \in (0, 1/2)$. We can upper-bound $H(p)$ in the following way:

$$\begin{aligned} H(p) \;&=\; p \log_2(1/p) \;+\; \underbrace{(1-p)}_{\leq 1} \log_2 \underbrace{(1/(1-p))}_{\leq 1+2p} \\ &<\; p \log_2(1/p) \;+\; \underbrace{2p}_{\text{(using } \log(1+c) < c \text{ for } c > 0)} \\ &<\; 3p \log_2(1/p) \;. \end{aligned}$$

From this, one can easily verify that for $m \geq 10^6$, we have

$$H\left(\frac{1}{4m \log_2 m} \right) \;<\; \frac{1}{m} \;.$$

Thus for such m,

$$H_-^{-1}\left(\frac{1}{m}\right) \;>\; \frac{1}{4m\log_2 m}\;,$$

so that

$$H_+^{-1}\left(\frac{1}{m}\right) \;=\; 1 - H_-^{-1}\left(\frac{1}{m}\right) \;<\; 1 - \frac{1}{4m\log_2 m}\;,$$

giving item 1.

(2.) The binary entropy function H is infinitely differentiable on $(0,1)$, with

$$H(.5) = 1, \quad H'(.5) = 0, \quad H''(.5) = -4(\ln 2)^{-1}\;.$$

Thus for $\beta \in [0, .25)$ we have

$$H(.5+\beta) \;\leq\; 1 - 4(\ln 2)^{-1}\beta^2 + O(\beta^3)\;.$$

By considering settings $\beta := \frac{\sqrt{\ln 2}}{2}\sqrt{\delta} \pm O(\delta^{3/2})$ and using that $H(\cdot)$ is decreasing on $(.5,1)$, we verify item 2. □

B.3 The lemma

Lemma B.4. *Let $R(x^1,\ldots,x^t) : \{0,1\}^{t\times n} \to \{0,1\}^{\leq t'}$ be any possibly-randomized mapping, where $t, t' \in \mathbb{N}^+$ satisfy $t'+2 \leq t$.*

Then, R is δ-distributionally stable with respect to any input distributions $\mathcal{D}_1,\ldots,\mathcal{D}_t$, where

$$\delta \;:=\; 2H_+^{-1}\left(1 - \frac{t'+1}{t}\right) - 1\;.$$

We will prove Lemma B.4 by a reduction to an encoding/decoding task that allows us to apply Lemma B.1.

Proof of Lemma B.4. Let $\mathcal{D}_1,\ldots,\mathcal{D}_t$ be independent distributions over $\{0,1\}^n$, and for $j \in [t]$, let γ_j be as in Definition 6.1 for $F := R$.

Consider the following encoding/decoding experiment involving t "Receivers." In the experiment, we will use R as a communication channel to attempt to transmit t bits $b_1,\ldots,b_t$. Receiver $j \in [t]$ will be responsible for attempting to recover the value b_j. Formally:

1. For $j \in [t]$, let $y^j, w^j \sim \mathcal{D}_j$ (here y^j, w^j are independent of each other and of all other $y^{j'}, w^{j'}$);
2. Also, and independently, for $j \in [t]$ let $b_j \sim \mathcal{U}_{\{0,1\}}$;
3. If $b_j = 0$, let $x^j := y^j$. Otherwise, let $x^j := w^j$;
4. Let $z := R(x^1,\ldots,x^t)$ (a possibly-randomized value), and let z be sent to the Receivers. Let $\{y^j\}_{j\in[t]}$ be visible to the receivers as public randomness;
5. Each Receiver j outputs a guess $\tilde{b}_j$ for b_j, based on the values of the two random variables y^j and z). Specifically, Receiver j uses the maximum-likelihood rule $\tilde{b}_j := ML(b|y^j, z)$, described in Section 2.1.

Note that in making the guess $\tilde{b}_j$, Receiver j does *not* inspect the values $y^{j'}$, $j' \neq j$.

We analyze this experiment. First observe that, *conditioned* on a value y^j seen by Receiver j and on the value $b_j \in \{0,1\}$ (which Receiver j does not see in the actual experiment), but leaving the other values $\{y^{j'}\}_{j' \neq j}$ unconditioned, the conditional distribution on z is that $z \sim R(\mathcal{D}_1, \ldots, \mathcal{D}_{j-1}, y^j, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_t)$ if $b_j = 0$, and $z \sim R(\mathcal{D}_1, \ldots, \mathcal{D}_j, \ldots, \mathcal{D}_t)$ if $b_j = 1$.

Also, b_j is unbiased and independent of y^j. Thus, by the distinguishability interpretation of statistical distance (see Section 2.1), Receiver j's success probability in guessing b_j, conditioned exclusively on an observed value y^j, equals

$$\frac{1}{2}\left(1 + \left|\left|R(\mathcal{D}_1, \ldots, \mathcal{D}_{j-1}, y^j, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_t) - R(\mathcal{D}_1, \ldots, \mathcal{D}_j, \ldots, \mathcal{D}_t)\right|\right|_{\text{stat}}\right) .$$

Thus Receiver j's *overall* success probability in the experiment is precisely $\frac{1}{2}(1+\gamma_j)$, where γ_j is as in the definition of distributional stability for R with respect to $\mathcal{D}_1, \ldots, \mathcal{D}_t$.

In our present setup, we can regard

$$z \;=\; R(x^1, \ldots, x^t) \;=:\; Enc(b_1, \ldots, b_t, y^1, \ldots, y^t)$$

as a randomized encoding function of $b_1, \ldots, b_t$, with public randomness $y^1, \ldots, y^t$ and additional private randomness $w^1, \ldots, w^t$. Similarly, we can view

$$(\tilde{b}_1, \ldots, \tilde{b}_t) \;=:\; Dec(z, y^1, \ldots, y^t)$$

as a (deterministic) decoding function. The success probability of our encoding/decoding experiment in successfully decoding b_j is $\frac{1}{2}(1+\gamma_j)$.

Now $H(b_1, \ldots, b_t) = t$, and $H(Enc(b_1, \ldots, b_t, y^1, \ldots, y^t)) \leq \log_2\left(\left|\{0,1\}^{\leq t'}\right|\right) < t'+1$. Applying Lemma B.1, we find that

$$H\left(\frac{1}{2}\left(1 + \frac{1}{t}\sum_{j=1}^{t}\gamma_j\right)\right) \geq \frac{1}{t}\left(t - (t'+1)\right) \;=\; 1 - \frac{t'+1}{t} ,$$

which implies that

$$\frac{1}{t}\sum_{j=1}^{t}\gamma_j \;\leq 2H_+^{-1}\left(1 - \frac{t'+1}{t}\right) - 1 \;= \delta .$$

Thus, R is δ-DS with respect to $\mathcal{D}_1, \ldots, \mathcal{D}_t$. As these distributions were arbitrary, this proves Lemma B.4. □

C Proof of quantum distributional stability

We use various concepts and results of quantum information theory. In particular, we assume familiarity with the notion of bipartite and reduced states. For a bipartite state $\rho_{A,B}$ on subsystems A, B, we let ρ_A (resp. ρ_B) denote the reduced state over A (resp. B). We let $S(\rho) := -\mathrm{Tr}(\rho \log_2 \rho)$ denote the *Von Neumann entropy* of a quantum state (here, identifying ρ with its density matrix). In analogy to the classical case, the entropy of a d-qubit state can be shown to be at least 0 and at

most d. We define the *quantum mutual information* between subsystems A, B of a bipartite state ρ_{AB} as

$$I_q(A;B) \ := \ S(\rho_A) + S(\rho_B) - S(\rho_{AB}) \ .$$

See [NC00], and Chapter 11 in particular, for more background in quantum information.

We will sometimes speak of quantum systems containing a subsystem that is a classical random variable X. By this we mean a state of form

$$\rho_{X,Y} \ = \ \sum_{x \in \mathrm{supp}(X)} \Pr[X = x] \cdot |x\rangle\langle x| \otimes \sigma_x \ , \tag{36}$$

for some collection of quantum states $\{\sigma_x\}$ on a fixed number of qubits (the "Y-subsystem").

Lemma C.1. *[NC00, Theorem 11.8.5, p. 513] For a classical random variable X, and a state of the form in Eq. (36), we have*

$$S(\rho_{X,Y}) \ = \ H(X) + \sum_x \Pr[X = x] S(\sigma_x) \ .$$

In particular, considering the case where Y is an empty register, we have $S(\rho_X) = H(X)$.

We have the following elementary bound on the quantum mutual information between a classical message and its quantum encoding.[53]

Lemma C.2. *For a classical random variable X, and a state of the form in Eq. (36), with the states $\{\sigma_x\}$ on d qubits, we have*

$$I_q(X;Y) \ = \ S(\rho_Y) - \sum_x \Pr[X = x] S(\sigma_x) \ \leq \ d \ .$$

Proof. Using Lemma C.1, we calculate

$$\begin{aligned} I_q(X;Y) \ &= \ S(\rho_X) + S(\rho_Y) - S(\rho_{XY}) \\ &= \ H(X) + S(\rho_Y) - \left[H(X) + \sum_x \Pr[X = x] S(\sigma_x) \right] \\ &= \ S(\rho_Y) - \sum_x \Pr[X = x] S(\sigma_x) \\ &\leq \ d \ , \end{aligned}$$

since ρ_Y consists of d qubits and $S(\sigma_x) \geq 0$ for each x. □

Not all properties of classical entropy and mutual information are inherited by their quantum counterparts.[54] However, we have [Nay99a, p. 33 and Appendix A]:

Fact C.3. *Quantum mutual information obeys the following properties, for all X, Y, Z:*

1. $I_q(X;Y) = I_q(Y;X) \geq 0$;

[53] I thank Scott Aaronson and Thomas Vidick for helping me to understand this fact.

[54] For example, it is not generally true that $S(\rho_{AB}) \leq S(\rho_A)$ by analogy with the fact that $H((X,Y)) \geq H(X)$. Fact 4.1. Note, though, that we don't use this classical fact in proving Lemma 6.2.

2. $I_q(X;(Y,Z)) = I_q(X;Y) + I_q((X,Y);Z) - I_q(Y;Z)$;

3. *(Strong subadditivity)* $I_q(X;(Y,Z)) \geq I_q(X;Y)$;

4. $I_q(X;Z) = 0$ *if the subsystems* X, Z *are independent classical random variables.*

Item 3 is a nontrivial fact in the quantum setting, with multiple equivalent formulations; see [NC00, Chapter 11].

With these facts in hand, the proof of Lemma C.4 below exactly follows that of Lemma 4.4.

Lemma C.4. *If* $X^1, \ldots, X^t$ *are independent classical random variables and* Y *a quantum subsystem, then*

$$I_q(Y;(X^1,\ldots,X^t)) \;\geq\; \sum_{j\in[t]} I_q(Y;X^j) \;.$$

Next, we need quantum analogues of Pinsker's and Vajda's inequalities. For mixed states ρ, σ over the same number of qubits, define the *relative entropy* (a quantum analogue of Kullback-Leibler divergence) as

$$S(\rho||\sigma) \;:=\; \mathrm{Tr}(\rho \log_2(\sigma)) - S(\rho) \;.$$

We also have the following analogue of Fact 4.6 [KNTSZ07, p. 10]:

Fact C.5. $I(A;B) = S(\rho_{AB}||\rho_A \otimes \rho_B)$.

A quantum Pinsker inequality was explicitly proved in [KNTSZ07, Theorem III.1].[55] However, that proof actually demonstrates a more general principle:

Theorem C.6 ([KNTSZ07])**.** *Suppose that for some* $\alpha, \beta \geq 0$*, the (classical) statistical distance and Kullback-Leibler divergence obey the relationship*

$$||X - Y||_{\mathrm{stat}} \;\geq\; \alpha \quad\Longrightarrow\quad D_{\mathrm{KL}}(X||Y) \;\geq\; \beta \;.$$

for every pair of classical distributions X, Y*.*

Then, for any pair ρ, σ *of quantum states,*

$$||\rho - \sigma||_{\mathrm{tr}} \;\geq\; \alpha \quad\Longrightarrow\quad S(\rho||\sigma) \;\geq\; \beta \;.$$

Combining this principle with the classical Pinsker and Vajda inequalities, we obtain:

Corollary C.7 (Quantum Pinsker inequality)**.** *For any states* ρ, σ*,*

$$S(\rho||\sigma) \;\geq\; \frac{2}{\ln 2} \cdot ||\rho - \rho'||_{\mathrm{tr}}^2$$

Corollary C.8 (Quantum Vajda inequality)**.** *For any states* ρ, σ*,*

$$S(\rho||\sigma) \;\geq\; \frac{1}{\ln 2}\left(\ln\left(\frac{1}{1 - ||\rho - \sigma||_{\mathrm{tr}}}\right) - 1\right).$$

[55] An earlier version appears in [OP04].

In the quantum setting we let $\mathbf{R}$ denote the mixed quantum state $R(X^1, \ldots, X^t)$, where $X^j \sim \mathcal{D}_j$. The inequality

$$I_q((X^1, \ldots, X^t); \mathbf{R}) \;\leq\; t'$$

follows from Lemma C.1, since $\mathbf{R} \in \mathsf{MS}_{t'}$. With the assembled tools in hand, the proof of Lemma 8.10 is essentially identical to that of Lemma 6.2. The one difference is that the classical equality

$$||(X^j, \mathbf{R}) - (Y^j, \mathbf{R})||_{\text{stat}} \;=\; \mathbb{E}_{x^j \sim \mathcal{D}_j}\left[\left|\left|R\left(\mathcal{D}_1, \ldots, \mathcal{D}_{j-1}, x^j, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_t\right) - R\left(\mathcal{D}_1, \ldots, \mathcal{D}_t\right)\right|\right|_{\text{stat}}\right]$$

we used there is replaced by the *in*equality

$$||(X^j, \mathbf{R}) - (Y^j \otimes \mathbf{R})||_{\text{tr}} \;\geq\; \mathbb{E}_{x^j \sim \mathcal{D}_j}\left[\left|\left|R\left(\mathcal{D}_1, \ldots, \mathcal{D}_{j-1}, x^j, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_t\right) - R\left(\mathcal{D}_1, \ldots, \mathcal{D}_t\right)\right|\right|_{\text{tr}}\right] \;.$$

This inequality follows by considering the experiment that first measures the X^j register, then performs an optimal distinguishing measurement on $\mathbf{R}$ conditioned on the outcome of the first measurement. Note that this inequality goes in the needed direction.