Is Valiant–Vazirani's Isolation Probability Improvable?

Holger Dell* Valentine Kabanets† Dieter van Melkebeek* Osamu Watanabe‡

*Department of Computer Sciences
University of Wisconsin
Madison, WI 53711, USA
{holger,dieter}@cs.wisc.edu

†School of Computing Science
Simon Fraser University
Burnaby, BC, Canada V5A 1S6
kabanets@cs.sfu.ca

‡Dept. of Math. Comput. Sci.
Tokyo Institute of Technology
Meguro-ku Ookayama, Tokyo 152, Japan
watanabe@is.titech.ac.jp

***Abstract*—The Valiant–Vazirani Isolation Lemma [*TCS*, vol. 47, pp. 85–93, 1986] provides an efficient procedure for *isolating* a satisfying assignment of a given satisfiable circuit: Given a Boolean circuit C on n input variables, the procedure outputs a new circuit C' on the same n input variables such that (i) every satisfying assignment of C' also satisfies C, and (ii) if C is satisfiable, then C' has exactly one satisfying assignment. In particular, if C is unsatisfiable, then (i) implies that C' is unsatisfiable. The Valiant–Vazirani procedure is *randomized*, and when C is satisfiable it produces a uniquely satisfiable circuit C' with probability $\Omega(1/n)$.**

Is it possible to have an efficient *deterministic* witness-isolating procedure? Or, at least, is it possible to improve the success probability of a randomized procedure to a large constant? We argue that the answer is likely *'No'*. More precisely, we prove that there exists a non-uniform randomized polynomial-time witness-isolating procedure with success probability bigger than $2/3$ *if and only if* $\mathsf{NP} \subseteq \mathsf{P/poly}$. Thus, an improved witness-isolating procedure would imply the collapse of the polynomial-time hierarchy. We establish similar results for other variants of witness isolation, such as reductions that remove all but an odd number of satisfying assignments of a satisfiable circuit.

We also consider a blackbox setting of witness isolation that generalizes the setting of the Valiant–Vazirani Isolation Lemma, and give an upper bound of $O(1/n)$ on the success probability for a natural class of randomized witness-isolating procedures.

***Keywords*-Isolation Lemma, Unique Satisfiability, Parity Satisfiability, Derandomization**

I. Introduction

The Isolation Lemma of Valiant and Vazirani [20] (as well as the related Isolation Lemma of Mulmuley et al. [13] and its refinement by Chari et al. [4]) is a basic tool with many important applications in complexity theory. See, e.g., [3, 16, 19] for just a few such applications. The lemma provides an efficient randomized algorithm to "isolate" a single object from a collection of objects satisfying a given efficiently decidable property. More precisely, given a Boolean circuit $C(x_1, \ldots, x_n)$, the algorithm produces a new Boolean circuit $C'(x_1, \ldots, x_n)$ such that (i) every satisfying assignment of C' also satisfies C with probability one (over the internal randomness of the algorithm), and (ii) if C is satisfiable, then, with probability $\Omega(1/n)$, C' has exactly one satisfying assignment. Thus, in case C is satisfiable, the unique satisfying assignment for C' is an "isolated" assignment from among the satisfying assignments for C.

An obvious question is whether efficient *deterministic* isolation is possible. That is, is there a deterministic polynomial-time algorithm that maps an input circuit $C(x_1, \ldots, x_n)$ to an output circuit $C'(x_1, \ldots, x_n)$ such that (i) every satisfying assignment of C' also satisfies C, and (ii) if C is satisfiable, then C' has exactly one satisfying assignment? Another natural question is whether the success probability $\Omega(1/n)$ for randomized isolation can be improved to, say, a large constant probability. We show that the answer to both questions is likely negative.

A. Our results

If $\mathsf{NP} = \mathsf{P}$, then efficient deterministic isolation is trivially possible: Given a circuit C, one can use the standard "search-to-decision" reduction to find in deterministic polynomial time some satisfying assignment w for C, and then construct a circuit C' so that C' accepts the single input w. Naïvely, it seems impossible to produce, efficiently deterministically, a circuit C' with exactly one satisfying assignment that also satisfies C, without actually finding such an assignment efficiently deterministically. In other words, naïvely it seems that *efficient deterministic isolation must be equivalent to* $\mathsf{NP} = \mathsf{P}$.

We show that such an equivalence is actually true in the *non-uniform* setting! We prove that if there is a non-uniform family of polynomial-size circuits that achieve deterministic isolation (in the sense defined above), then every language in NP can be decided by a non-uniform family of polynomial-size circuits, i.e., $\mathsf{NP} \subseteq \mathsf{P/poly}$. Since the standard "search-to-decision" reduction for NP can be run also in the non-uniform setting, we immediately get the other direction: if $\mathsf{NP} \subseteq \mathsf{P/poly}$, then non-uniform efficient deterministic isolation is possible.

Given that deterministic isolation is unlikely, what can we say about the existence of a better randomized isolation algorithm? A natural question is whether one can obtain randomized isolation with success probability better than $\Omega(1/n)$ achieved in [20]. For example, can one obtain (large) constant success probability?

We show that the answer is likely negative. In fact, we extend the result for deterministic isolation and prove

that if there is a (non-uniform) randomized isolation algorithm with success probability greater than $2/3$, then $\mathsf{NP} \subseteq \mathsf{P/poly}$ (and, consequently, the polynomial-time hierarchy collapses). We also consider more restricted and more relaxed notions of witness isolation, such as reductions that remove all but an odd number of satisfying assignments of a satisfiable circuit. For each of these notions, we prove that their existence implies some collapse of NP, namely $\mathsf{NP} = \mathsf{P}$, $\mathsf{NP} \subseteq \mathsf{P/poly}$, $\mathsf{NP} = \mathsf{coNP}$, or $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, and in most cases the collapse is actually equivalent to the existence.

Finally, we consider a natural blackbox setting for isolation that generalizes the setting of [20], and we observe that $O(1/n)$ is an upper bound on the success probability for randomized isolation in this blackbox setting.

B. Our techniques

We now sketch the proof of one of our main results – that efficient randomized isolation with success probability above $2/3$ implies $\mathsf{NP} \subseteq \mathsf{P/poly}$. The proof consists of two steps. Assuming the existence of such a witness-isolating procedure, we show how to

[Step 1] efficiently reduce satisfiability to prUSAT, the promise version of satisfiability on instances with at most one satisfying assignment, and

[Step 2] efficiently solve prUSAT.

Both steps run in $\mathsf{P/poly}$, which results in a $\mathsf{P/poly}$-algorithm for satisfiability and thus for all languages in NP.

Deterministic setting. For reasons of exposition, we first consider the simpler deterministic setting. Suppose there is a deterministic $\mathsf{P/poly}$-algorithm A that achieves isolation. That is, given a circuit $C(x_1, \ldots, x_n)$, A outputs a circuit $C'(x_1, \ldots, x_n)$ on the same number of variables such that (i) every satisfying assignment of C' also satisfies C, and (ii) if C is satisfiable, then C' has exactly one satisfying assignment.

In this setting, Step 1 is trivial as A represents an efficient mapping reduction from satisfiability to prUSAT. For Step 2, we mimic an argument due to Ko [12] and devise a $\mathsf{P/poly}$-algorithm for prUSAT. The two steps combined put satisfiability in $\mathsf{P/poly}$.

Ko [12] proved that if satisfiability has a selector function computable in $\mathsf{P/poly}$, then satisfiability is in $\mathsf{P/poly}$. A selector for satisfiability is a function that takes two input circuits C_1 and C_2, and selects the one that is "most likely" to be satisfiable. More precisely, the function always outputs one of its two inputs, and if exactly one of the two inputs is satisfiable, then it outputs that input. Such a function induces a binary relation R on the set of all inputs, where $R(C_1, C_2)$ holds if and only if the selector outputs C_2 on input (C_1, C_2). The relation R has the following "Ko"-properties:

(K1) If C_1 is satisfiable and $R(C_1, C_2)$, then C_2 is satisfiable.

(K2) If C_1 and C_2 are satisfiable instances of the same length, then $R(C_1, C_2)$ or $R(C_2, C_1)$.

(K3′) R can be decided in polynomial time with oracle access to the selector.

Property (K2) actually holds in a stronger form, but the weaker form is all we need in Ko's argument to deduce that the directed graph induced by R on the set of satisfiable instances of length ℓ has a dominating set D_ℓ of size polynomial in ℓ. Combined with property (K1), this gives us the following criterion for satisfiability on inputs of length ℓ:

$$C \in \mathrm{SAT} \;\Leftrightarrow\; (\exists\, C^* \in D_\ell)\, R(C^*, C)\,. \tag{1}$$

By property (K3′), criterion (1) yields a polynomial-time algorithm for satisfiability when given oracle access to the selector and advice D_ℓ. Thus, we obtain a $\mathsf{P/poly}$-algorithm for satisfiability if satisfiability has a selector computable in P or in $\mathsf{P/poly}$.

Now consider the setting where we have a deterministic isolation algorithm A for circuits. If at least one of C_1 or C_2 is satisfiable and the sets of satisfying assignments are disjoint, the action of A on $C \doteq C_1 \vee C_2$ or on $C \doteq C_2 \vee C_1$ can be viewed as that of a selector: It selects the unique C_i that has a satisfying assignment in common with $A(C)$. As a selector ought to act on the unordered pair $\{C_1, C_2\}$, we actually apply A to $C \doteq \min(C_1, C_2) \vee \max(C_1, C_2)$, where $\min(C_1, C_2)$ denotes the lexicographically smaller of the two circuits C_1 and C_2, and similarly $\max(C_1, C_2)$ the lexicographically larger of the two circuits.

In general, we can define a binary relation R with similar properties as above: $R(C_1, C_2)$ holds if and only if

(a) C_1 and C_2 have a common satisfying assignment, or

(b′) C_1 and $A(C)$ have no common satisfying assignment, where $C \doteq \min(C_1, C_2) \vee \max(C_1, C_2)$.

This relation R satisfies the properties (K1) and (K2). Since these properties were all that was needed to arrive at criterion (1), the criterion still holds. Property (K3′) may no longer hold, but we can guarantee the following instead:

(K3) Whether $R(C_1, C_2)$ holds can be decided in polynomial time with oracle access to A if the set of satisfying assignments of C_1 is given as advice.

Thus, criterion (1) yields a polynomial-time algorithm for satisfiability when given oracle access to A as well as the following advice at input length ℓ: for every $C^* \in D_\ell$, the circuit C^* as well as all its satisfying assignments. In general, the advice may be of superpolynomial length because the circuits C^* may have a superpolynomial number of satisfying assignments. Since Step 1 allows us to reduce the number of satisfying assignments to at most one, we can restrict our attention to the set of all inputs with at most one satisfying assignment. This way, the length of the advice becomes polynomially bounded, and we obtain a $\mathsf{P/poly}$-algorithm for prUSAT whenever A is computable in P or in $\mathsf{P/poly}$.

Randomized setting. Suppose there is an efficient randomized isolation algorithm A with success probability at least p. That is, on input a circuit $C(x_1, \ldots, x_n)$, A outputs a circuit $C'(x_1, \ldots, x_n)$ such that (i) every satisfying assignment of C also satisfies C', and (ii) if C is satisfiable, then, with probability at least p, the circuit C' is a *successful isolation* of C, i.e., C' has a unique satisfying assignment.

For Step 2, we first apply Adleman's argument to transform A into a P/poly-algorithm B that takes a circuit C and outputs a list of circuits C' such that (i) every satisfying assignment of C' also satisfies C, and (ii) if C is satisfiable, then at least a fraction p' of the circuits C' are successful isolations of C, where p' is somewhat smaller than p. We adapt the relation R from the deterministic setting by replacing the condition (b′) by the following:

(b) fewer than a fraction p' of circuits C' on the list $B(C)$ are such that C' and C_1 have a common satisfying assignment, where $C \doteq \min(C_1, C_2) \vee \max(C_1, C_2)$.

Thus we let $R(C_1, C_2)$ hold if and only if (a) or (b) holds. This modified relation R still has property (K1). The main reason is that if C_1 is satisfiable and (b) holds, then $B(C)$ contains at least one successful isolation C' that is not satisfied by any satisfying assignment of C_1 but is satisfiable, and therefore has to be satisfied by a satisfying assignment of C_2.

As for property (K2), suppose that C_1 and C_2 are satisfiable but that neither $R(C_1, C_2)$ nor $R(C_2, C_1)$ holds. By (a), this means that the sets of satisfying assignments of C_1 and C_2 are disjoint. By (b) and inclusion-exclusion, at least a fraction $2p' - 1$ of the circuits C' in $B(C)$ is satisfied by a satisfying assignment of C_1 *as well as* by a satisfying assignment of C_2. Therefore, at least a fraction $2p' - 1$ of the circuits C' have at least two satisfying assignments. This contradicts the success rate p' of B as long as $2p' - 1 > 1 - p'$. Thus, (K2) is guaranteed to hold provided $p' > 2/3$.

Property (K3) also holds for the new R. Since all three properties (K1), (K2), and (K3) hold whenever $p' > 2/3$, and since we can set $p' > 2/3$ when p is a constant exceeding $2/3$, Ko's argument gives us a P/poly-algorithm for prUSAT whenever p is a constant larger than $2/3$. This completes Step 2.

Step 1 is no longer trivial in the randomized setting but we can appeal to an unconditional P/poly reduction that takes a circuit C and outputs a list of circuits C' such that (i) if C is unsatisfiable then every C' is also unsatisfiable, and (ii) if C is satisfiable then at least one C' has a unique satisfying assignment. Such a reduction follows by applying Adleman's argument to the Valiant–Vazirani isolation procedure. On input C, we cycle over all circuits C' on the list and apply the prUSAT-algorithm from Step 2 to each C'. We accept iff our prUSAT-algorithm accepts on at least one circuit C'. Note that for an unsatisfiable C, all circuits C' are also unsatisfiable, and will be rejected by the prUSAT-algorithm. For a satisfiable C, at least one of the circuits C' is uniquely satisfiable, and hence will be accepted by the prUSAT-algorithm. Thus we get a P/poly-algorithm for satisfiability.

C. Related work

Chari, Rohatgi, and Srinivasan [4] consider the problem of minimizing the number of random bits that are used in the isolation lemma. They design an isolation lemma that improves upon the pruning procedure of Mulmuley, Vazirani, and Vazirani [13], and they show that, in the blackbox setting, their improved isolation lemma uses the least possible *number of random bits* while still achieving non-negligible success probability. Our blackbox result shows that it is impossible to increase the *success probability* beyond $O(1/n)$.

The problem of efficient deterministic isolation is related to the problem of multi-valued vs. single-valued NP-computable functions [17], which received considerable attention in the 1990's. In fact, it easily follows from the work of Hemaspaandra et al. [7] that efficient deterministic isolation yields a collapse of the polynomial-time hierarchy. More precisely, [7] implies that efficient deterministic isolation leads to $\mathsf{NP} \subseteq (\mathsf{NP} \cap \mathsf{coNP})/\mathsf{poly}$, which in turn is known to imply the collapse of the polynomial-time hierarchy to the second level. In contrast, we prove that the same assumption implies $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{poly}$. This conclusion is stronger, and, as observed above, is actually equivalent to the existence of efficient non-uniform deterministic isolation.

The problem of efficient deterministic isolation as defined above is different from the problem of *derandomizing* the Valiant–Vazirani Isolation Lemma as studied, e.g., in [11]. In the setting of [11], randomized isolation is defined via the existence of an efficient randomized algorithm that maps an input circuit C to a *list* of circuits $C'_1, \ldots, C'_t$ such that (i) every satisfying assignment of the C'_i also satisfies C, and (ii) if C is satisfiable, then, with high probability, at least one of the C'_i is uniquely satisfiable. This kind of randomized isolation follows from the Valiant–Vazirani Isolation Lemma.

Derandomizing such isolation means designing an efficient deterministic algorithm that produces the list $C'_1, \ldots, C'_t$. One of the results in [11] is that this kind of derandomization is likely to exist since it follows from some plausible circuit complexity assumptions. However, if we want to get a *single* circuit C' that is uniquely satisfiable if C is satisfiable, no better way is known other than to pick one of the circuits on the list at random. But then we end up with a randomized isolation procedure with inverse-polynomial success probability. Thus, while it may be possible to design an efficient deterministic algorithm mapping a given input circuit C to a *list* of circuits $C'_1, \ldots, C'_t$ achieving isolation in the sense of [11], it is unlikely that there is an efficient deterministic isolation mapping C to a *single* circuit C'. Also, by our results, it is unlikely that there is a [11]-

style randomized isolation algorithm mapping a satisfiable circuit C to a list of circuits where more than $2/3$ of the circuits on the list are uniquely satisfiable.

The question whether efficient deterministic isolation exists is also related to the question whether $\mathsf{NP} = \mathsf{UP}$, that is, whether every language in NP can be decided by an NP-machine that has at most one accepting computation path for every input. Clearly, if deterministic polynomial-time isolation is possible, then $\mathsf{NP} = \mathsf{UP}$. However, the converse is not known to be true. It remains an open question whether the assumption $\mathsf{NP} = \mathsf{UP}$ yields any unexpected consequences, e.g., if it implies any collapse of the polynomial-time hierarchy.

For some applications of the isolation lemma, such as Toda's theorem [19], it suffices to efficiently reduce NP to $\oplus\mathsf{P}$, i.e., to map circuits C to circuits C' such that C is satisfiable if and only if C' has an odd number of satisfying assignments. A single application of Valiant–Varizani's isolation lemma gives a randomized such reduction with success probability $\Omega(1/n)$, but in this setting better results are known: Naik et al. [14] achieve success probability arbitrarily close to $1/2$, and Gupta [6] actually reaches $1/2$. All of these reductions have the property that the satisfying assignments of C' also satisfy C. For such reductions, our results imply that the success probability cannot be improved beyond $2/3$ unless $\mathsf{NP} \subseteq \mathsf{P/poly}$.

In general, the pruning property need not hold, and the circuit C' can have more inputs than C. As observed in, e.g., [14, first paragraph of section 3], this freedom allows us to achieve success probability $1 - 1/\exp$ in the setting of $\oplus\mathsf{P}$. The key is the following operation, which efficiently transforms a list $C'_1, \ldots, C'_t$ of circuits into a single circuit C' such that C' has an odd number of satisfying assignments if and only if some C'_i has an odd number of satisfying assignments: (i) modify each circuit C'_i into a circuit C''_i by adding a single new satisfying assignment; (ii) construct a circuit C'' whose number of satisfying assignments is the product of those of the circuits C''_i by defining $C''(x_1, \ldots, x_t) \doteq \wedge_{i=1}^{t} C''_i(x_i)$, where each x_i is of the input size for C''_i; (iii) obtain C' by adding a single new satisfying assignment to C''. Starting from the output $C'_1, \ldots, C'_t$ of polynomially many independent runs of any of the above pruning procedures, we obtain a randomized reduction from NP to $\oplus\mathsf{P}$ with success probability $1 - 1/\exp$. In a similar way, using Adleman's argument, we obtain a deterministic $\mathsf{P/poly}$ reduction from NP to $\oplus\mathsf{P}$, and under the circuit complexity assumption from [11], a deterministic polynomial-time reduction from NP to $\oplus\mathsf{P}$.

D. Organization of the paper

Section II contains basic definitions and notation, the various notions of witness isolation we consider, and lemmas that capture Adleman's argument and Ko's argument in a way that is useful to us. We prove our conditional impossibility results for deterministic and randomized isolation in Section III, and categorize several variants based on which collapse of NP they are equivalent to. In Section IV, we prove our unconditional impossibility result in the blackbox setting. We suggest some directions for further research in Section V. Due to space constraints, some proofs have been omitted in this extended abstract and can be found in the full version [5].

II. Preliminaries

A. Basic definitions and notation

Complexity classes. We use standard definitions and notation for complexity classes such as P, NP, and $\mathsf{P/poly}$ (see, e.g., [1]), which we view as classes of languages over the alphabet $\{0,1\}$, or as classes of Boolean functions on $\{0,1\}^*$. By a slight abuse of notation, we extend the notation P and $\mathsf{P/poly}$ to not necessarily Boolean functions from $\{0,1\}^*$ to $\{0,1\}^*$. Thus, a function $f : \{0,1\}^* \to \{0,1\}^*$ is called P-computable if it is computable by some deterministic polynomial-time algorithm, and f is called $\mathsf{P/poly}$-computable if it is computable by a family of polynomial-size circuits.

Boolean circuits. We let SAT denote the satisfiability problem for deterministic Boolean circuits: Given a deterministic circuit $C(x_1, \ldots, x_n)$ with n variables $x_1, \ldots, x_n$, decide whether it has a satisfying assignment, that is, a binary string $w \in \{0,1\}^n$ with $C(w) = 1$. If C has exactly one satisfying assignment, we say that C is *uniquely satisfiable*.

A *nondeterministic* circuit $C(x_1, \ldots, x_n)$ is a deterministic circuit $D(x_1, \ldots, x_n, y_1, \ldots, y_m)$ with additional "nondeterministic" variables $y_1, \ldots, y_m$. An assignment $w \in \{0,1\}^n$ to the x-variables *satisfies* C if and only if there exists an assignment $w' \in \{0,1\}^m$ to the y-variables such that $D(w, w') = 1$. We say that C is uniquely satisfiable if there exists exactly one such w.

Throughout this paper, we write n for the number of (deterministic) variables of a circuit and ℓ for the length of binary encodings. We assume that the encoding is efficient so that, e.g., for circuits C_1 and C_2 of length ℓ each, the circuit $C_1 \vee C_2$ can be computed in polynomial time and is of length at most $O(\ell)$.

Promise problems. A promise problem is a pair $\Pi = (\mathrm{Yes}, \mathrm{No})$ of disjoint subsets $\mathrm{Yes} \cup \mathrm{No} \subseteq \{0,1\}^*$. For the promise problem of unique satisfiability for deterministic Boolean circuits, prUSAT, the set Yes is the set of all uniquely satisfiable deterministic circuits, and No is the set of all unsatisfiable deterministic circuits.

We say that an algorithm A decides Π if it accepts all $x \in \mathrm{Yes}$, rejects all $x \in \mathrm{No}$, and behaves arbitrarily for all other inputs. In terms of complexity classes, we write $\Pi \in \mathcal{C}$ if there exists a language $L \in \mathcal{C}$ such that $\mathrm{Yes} \subseteq L$ and $\mathrm{No} \subseteq \overline{L}$, where $\overline{L} \doteq \{0,1\}^* \setminus L$ denotes the complement of L.

B. Notions of isolation

We study isolation and several variations that are all motivated by the question whether NP coincides with UP, unambiguous polynomial time. Because of this connection, we use the generic term "disambiguation" to refer to all variants.

$\mathsf{UP} = \mathsf{NP}$ is equivalent to the existence of a polynomial-time verifier $V(C, w)$ for SAT such that each input circuit C has at most one valid witness w with $V(C, w) = 1$. Since the computation of $V(C, .)$ for each fixed C can be modeled as a polynomial-size Boolean circuit C', the $\mathsf{UP} = \mathsf{NP}$ question is equivalent to the existence of a polynomial-time transformation of a deterministic circuit C into a deterministic circuit C' such that (i) if C is unsatisfiable, then C' is unsatisfiable, and (ii) if C is satisfiable, then C' has exactly one satisfying assignment.

More generally, we define a disambiguation *for a class* $\mathcal{C}$ of Boolean circuits as follows, where natural choices for $\mathcal{C}$ are Boolean formulas, deterministic Boolean circuits, and nondeterministic Boolean circuits.

Definition 1 (Disambiguation). A *disambiguation* for a class $\mathcal{C}$ of Boolean circuits is a randomized algorithm that maps a given circuit $C \in \mathcal{C}$ to a circuit $C' \in \mathcal{C}$ such that:

Perfect Soundness: if C is unsatisfiable, then C' is also unsatisfiable (with probability one).

p-Completeness: If C is satisfiable, then with probability at least p the circuit C' has a unique satisfying assignment.

Here $p = p(\ell) \in [0, 1]$ is the *success probability* of the disambiguation, and may depend on the input length ℓ. We typically want an efficient disambiguation; we consider disambiguations computable in P or in $\mathsf{P/poly}$[1]. We call a disambiguation *deterministic* if it does not use any randomness and satisfies the above conditions with $p = 1$. We call a disambiguation *satisfiability-preserving* if C' is satisfiable whenever C is satisfiable.

For general disambiguations, no specific relationship between the satisfying assignments of C and the satisfying assignments of C' is required. In this paper we study notions of disambiguation that additionally impose such restrictions. In decreasing order of restrictiveness we consider witness-isolating disambiguation, or isolation for short, and witness-recoverable disambiguation. We now specify the respective additional conditions as strengthenings of the requirements in Definition 1.

Isolation. An *isolation* is a disambiguation that maps circuits C to circuits C' on the *same* set of variables as C, in such a way that every satisfying assignment of C' also satisfies C, with probability one. Any particular output C' of an isolation is a *successful isolation* of a satisfiable circuit C if C' has a unique satisfying assignment. In a *minimal witness* isolation, we additionally require the unique satisfying assignment of a successful isolation C' to be the lexicographically smallest satisfying assignment of C. The procedures of Valiant and Vazirani [20], Mulmuley, Vazirani, and Vazirani [13], and Chari, Rohatgi, and Srinivasan [4] yield randomized polynomial-time isolations with success probabilities $p = \Omega(1/n)$, $p = \Omega(1/n^2)$, and $p = \Omega(1/n^8)$, respectively.

Witness-recoverable disambiguation. A *witness-recoverable disambiguation* is a disambiguation that maps circuits C to circuits C' on a potentially *different* set of variables. Furthermore, there has to exist a deterministic polynomial-time *witness recovery algorithm* W such that, if C is satisfiable, then with probability at least p the following two conditions hold simultaneously:

- C' has a unique satisfying assignment, say w, and
- given C, C', and w, the algorithm W outputs a satisfying assignment for C.

Every isolation is a witness-recoverable disambiguation: The witness recovery algorithm can just output $W(C, C', w) = w$ since isolation guarantees that any satisfying assignment w of C' also satisfies C. For nondeterministic circuits, the reverse direction also holds, so the notions of isolation and witness-recoverable disambiguation are actually equivalent for nondeterministic circuits. The reverse direction follows because a nondeterministic circuit C'' can guess and verify a satisfying assignment w' for the circuit C' that the witness-recoverable reduction produces, and C'' can further compute $w \doteq W(C, C', w')$ and check that w satisfies C. (See the full version of this paper for more details.)

A witness-recoverable disambiguation for deterministic circuits yields a witness-recoverable disambiguation for nondeterministic circuits – simply apply the former to the deterministic circuit underlying the nondeterministic circuit, and recover the actual input bits. (See the full version of this paper for more details.) Combined with the above argument, a witness-recoverable disambiguation for deterministic circuits yields an isolation for nondeterministic circuits. This motivates the study of isolation for nondeterministic circuits. If we were to require the uniqueness condition *after* recovery rather than before, witness-recoverable disambiguation for deterministic and for nondeterministic circuits would be equivalent to each other, as well as to isolation for nondeterministic circuits.

C. Adleman's argument

We deal with randomness by transforming randomized algorithms into deterministic algorithms with small advice. In the case of decision algorithms, Adleman's argument turns any BPP-machine into a $\mathsf{P/poly}$-algorithm that decides the same language, and it does not really make sense to talk

[1] As explained in Section II-C, in contrast to the standard setting of decision procedures, the combination "randomized $\mathsf{P/poly}$" does make sense in the setting of disambiguation procedures.

about randomized P/poly-algorithms since $\mathsf{BPP}/\mathsf{poly} = \mathsf{P}/\mathsf{poly}$. For transformations such as randomized disambiguations, the notions of randomized P/poly-algorithms and deterministic P/poly-algorithms do seem to be different. Adleman's argument allows us to *list-derandomize* randomized P/poly transformations in the sense of the following lemma.

Lemma 1 (Adleman). *Let A be a randomized P/poly-algorithm that maps strings x to strings y. Let $p_1, p_2 : \mathbb{N} \to [0,1]$ be functions and let $P_1(x,y)$ and $P_2(x,y)$ be properties such that, for all inputs x of length $\ell = |x|$, $P_1\big(x,y\big)$ holds with probability at least $p_1(\ell)$ and $P_2\big(x,y\big)$ holds with probability at least $p_2(\ell)$ over the internal randomness of A.*

Then, for every $\delta > 0$, there exists a deterministic P/poly-algorithm B that, on input x of length ℓ, produces a list $y_1, \ldots, y_t$ such that $P_1(x, y_i)$ holds for at least $p_1'(\ell) \cdot t$ many $i \in [t]$ and $P_2(x, y_i)$ holds for at least $p_2'(\ell) \cdot t$ many $i \in [t]$, where $p_j'(\ell) = 1$ whenever $p_j(\ell) = 1$, and $p_j'(\ell) = p_j(\ell) - \delta\ell^{\delta}$ otherwise.

D. Ko's argument

The following lemma captures the main argument in Ko's proof that the existence of a P-selector for a language L implies $L \in \mathsf{P}/\mathsf{poly}$. The notion of a P-selector is due to Selman [18] and Ko [12] proved the lemma for languages. We formulate it for promise problems so that we can apply it to prUSAT.

Lemma 2 (Ko). *Let $\Pi = (\mathrm{Yes}, \mathrm{No})$ be a promise problem, and let R be a binary relation over $\{0,1\}^*$ satisfying the following properties.*

(K1) *If $x \in \mathrm{Yes}$ and $R(x,y)$, then $y \in \mathrm{Yes}$.*
(K2) *If $x, y \in \mathrm{Yes}$ with $|x| = |y|$, then $R(x,y)$ or $R(y,x)$.*
(K3) *There exists a constant $c > 0$ such that for every $\ell \in \mathbb{N}$ and every $x \in \mathrm{Yes}$ of length ℓ, there is a circuit R_x of size at most $c \cdot \ell^c$ that decides on input $y \in \{0,1\}^\ell$ whether $R(x,y)$ holds.*

If the circuits R_x are deterministic, then there is a P/poly-algorithm for Π.

If the circuits R_x are co-nondeterministic, then there is a $\mathsf{coNP}/\mathsf{poly}$-algorithm for Π.

Proof: We fix the length ℓ of the input and design a polynomial-size circuit that decides instances of length ℓ. We first argue that there is list $a_1, \ldots, a_m \in \mathrm{Yes}$ with $0 \leqslant m \leqslant \ell+1$ such that for all $y \in \mathrm{Yes}$ we have $R(a_i, y)$ for some $i \in [m]$. To see this, assume we already constructed $a_1, \ldots, a_j$ for some $j \geqslant 0$, and let $S_j = \big\{y \in \mathrm{Yes} \cap \{0,1\}^\ell \;\big|\; R(a_i, y)$ does not hold for any $i \in [j]\big\}$. Note that $S_0 \neq \varnothing$. If S_j is empty, we are done and set $m = j$. Otherwise, $S_j \neq \varnothing$ and we define a_{j+1} as follows. Property (K2) implies that, for all $x, y \in S_j$, we have $R(x,y)$ or $R(y,x)$. Thus the average out-degree of the directed graph that R induces on S_j is at least $|S_j|/2$. In particular, there exists an element $a_{j+1} \in S_j$ such that at least half of all $y \in S_j$ satisfy $R(a_{j+1}, y)$. Thus $|S_{j+1}| \leqslant \frac{1}{2}|S_j| \leqslant \frac{1}{2^{j+1}}|S_0|$. Since $|S_0| \leqslant 2^\ell$, this implies that we reach $S_m = \varnothing$ for some $m \leqslant \ell + 1$, and we are done.

Now we devise an algorithm A for $\Pi = (\mathrm{Yes}, \mathrm{No})$ at input length ℓ.

- Given: $y \in \{0,1\}^\ell$.
- Advice: The list $a_1, \ldots, a_m$.
- Accept if and only if $R_{a_i}(y) = 1$ for some $i \in [m]$.

If $y \in \mathrm{No}$, then (K1) guarantees that $R(a,y) = 0$ for any $a \in \mathrm{Yes}$. Hence the circuit outputs $R_a(x) = 0$ and A rejects. On the other hand, if $y \in \mathrm{Yes}$, then the choice of the advice guarantees that some i with $R(a_i, y)$ exists. In this case the circuit R_{a_i} outputs $R_{a_i}(y) = 1$ and A accepts.

If the R_{a_i}'s are deterministic, then A is a P/poly-algorithm. If the R_{a_i}'s are co-nondeterministic, then A can simulate the R_{a_i}'s in $\mathsf{coNP}/\mathsf{poly}$. ∎

III. Isolation is unlikely to exist

In this section we show that efficient witness isolation and several other kinds of disambiguation imply unlikely collapses of complexity classes, namely $\mathsf{NP} = \mathsf{P}$, $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{poly}$, $\mathsf{NP} = \mathsf{coNP}$, or $\mathsf{NP} \subseteq \mathsf{coNP}/\mathsf{poly}$. In fact, in many cases the reverse implication also holds, so we obtain equivalences. Our results can therefore be viewed as taxonomic – they show that the existence of seemingly very restricted isolation procedures, such as deterministic non-uniform minimal witness isolation, is actually equivalent to the existence of more relaxed forms of isolation, such as randomized non-uniform isolation with success probability $p > 2/3$.

We obtain such results for both deterministic and nondeterministic circuits. We first consider deterministic circuits.

A. Uniform disambiguation for deterministic circuits

We argue that polynomial-time minimal witness isolation for the class of deterministic circuits is a very strong notion. In the uniform setting, its existence is equivalent to $\mathsf{NP} = \mathsf{P}$. It is the only form of disambiguation from which we obtain the collapse $\mathsf{NP} = \mathsf{P}$. The argument has a somewhat different flavor than the main collapse result described in the introduction.

Theorem 3. *There is a P-computable minimal witness isolation for deterministic circuits if and only if $\mathsf{NP} = \mathsf{P}$.*

Proof: "⇒". Consider the following polynomial-time algorithm for SAT: Given a deterministic circuit $C(x_1, \ldots, x_n)$, we halt and declare C satisfiable if $C(1, \ldots, 1) = 1$; otherwise, we construct the circuit $\tilde{C}(x_1, \ldots, x_n) \doteq \big(C(x_1, \ldots, x_n) \vee (x_1 = 1 \wedge \cdots \wedge x_n = 1)\big)$. We apply the minimal witness isolation to $\tilde{C}$ and obtain a uniquely satisfiable deterministic circuit C'. If $C'(1, \ldots, 1) = 0$, we declare C satisfiable, and otherwise, we declare C unsatisfiable.

For the correctness, first assume that C is satisfiable. If $C(1,\dots,1) = 1$, the algorithm declares this fact correctly. Otherwise we have $C(1,\dots,1) = 0$ and $\tilde{C}$ has some satisfying assignment other than $(1,\dots,1)$. Since C' isolates the lexicographically smallest satisfying assignment, it does not have $(1,\dots,1)$ as a satisfying assignment. Thus $C'(1,\dots,1) = 0$, and the algorithm correctly declares C satisfiable. On the other hand, if C is unsatisfiable, then $C(1,\dots,1) = 0$ and $C'(1,\dots,1) = 1$, and the algorithm correctly declares C' unsatisfiable.

"$\Leftarrow$". Given a Boolean circuit $C(x_1,\dots,x_n)$ and an assignment $w \in \{0,1\}^n$, we can verify in PH that w is the lexicographically smallest satisfying assignment of C. If $\mathsf{NP} = \mathsf{P}$, we have $\mathsf{PH} = \mathsf{P}$ and this verification can be performed in P. Hence we can efficiently compute a deterministic circuit $C'(x_1,\dots,x_n)$ that outputs 1 if and only if its input is the lexicographically smallest satisfying assignment of C. If C is satisfiable, then the constructed circuit C' is uniquely satisfied by the lexicographically smallest satisfying assignment of C. On the other hand, if C is unsatisfiable, then C' is unsatisfiable. Since C' can be computed from C in polynomial time, this isolation procedure runs in polynomial time. ∎

B. Non-uniform disambiguation for deterministic circuits

Our main result shows that several $\mathsf{P/poly}$-computable notions of disambiguation are equivalent to $\mathsf{NP} \subseteq \mathsf{P/poly}$. To prove the collapse direction, we follow the two-step approach outlined in the introduction. The following lemma implements Step 1, a reduction from SAT to prUSAT.

Lemma 4. *If* $\mathrm{prUSAT} \in \mathsf{P/poly}$ *then* $\mathsf{NP} \subseteq \mathsf{P/poly}$.

The other direction in Lemma 4 trivially holds, but we only need the stated direction.

Proof: Assume M is a $\mathsf{P/poly}$-algorithm for prUSAT. We claim that $\mathrm{SAT} \in \mathsf{P/poly}$. Recall that Valiant–Vazirani gives a randomized isolation procedure A with success probability $p = \Omega\left(\frac{1}{n}\right)$. Adleman's argument (Lemma 1) yields a $\mathsf{P/poly}$-algorithm B that, given a circuit C, produces a list of $t = \mathsf{poly}(n)$ circuits $C'_1,\dots,C'_t$ satisfying the following: (i) if C is unsatisfiable, then each C'_i is unsatisfiable for $i \in [t]$, and (ii) if C is satisfiable then a fraction $\Omega(1/n)$ of the C'_i are successful isolations of C, that is, are uniquely satisfiable.

The following algorithm decides SAT. Given an input circuit C, compute the list $B(C) = (C'_1,\dots,C'_t)$. If $M(C'_i)$ accepts for at least one i, where $i \in [t]$, then accept; otherwise, reject.

The described algorithm is clearly in $\mathsf{P/poly}$. For correctness, if C is unsatisfiable, then by (i) so are all C'_i, and hence M must reject each of them. If C is satisfiable, then by (ii) some C'_i is uniquely satisfiable, and hence M must accept this C'_i. ∎

We are now ready to prove our main result on disambiguations for deterministic circuits in the non-uniform setting.

Theorem 5. *Each of the following statements is equivalent to* $\mathsf{NP} \subseteq \mathsf{P/poly}$.

- (i) *There is a* $\mathsf{P/poly}$*-computable minimal witness isolation for deterministic circuits.*
- (ii) *There is a randomized* $\mathsf{P/poly}$*-computable isolation for deterministic circuits with success probability* $p \geqslant \frac{2}{3} + \frac{1}{\mathsf{poly}(\ell)}$.
- (iii) *There is a randomized* $\mathsf{P/poly}$*-computable satisfiability-preserving isolation for deterministic circuits with success probability* $p \geqslant \frac{1}{\mathsf{poly}(\ell)}$.

Obviously, the statements above are also equivalent to each other. In particular, the implication $(ii) \Rightarrow (i)$ transforms any randomized $\mathsf{P/poly}$-computable isolation with success probability $p = p(\ell)$ into a *deterministic* minimal witness isolation, the strongest notion of disambiguation that we consider. This implication holds for all functions $p : \mathbb{N} \to [0,1]$ for which there exists a constant $\delta > 0$ such that $p(\ell) \geqslant 2/3 + \delta \cdot \ell^{\delta}$ for all $\ell \in \mathbb{N}$.

Proof: The proof that $\mathsf{NP} \subseteq \mathsf{P/poly}$ implies (i) is as in proof of Theorem 3, and the implications $(i) \Rightarrow (ii)$ and $(i) \Rightarrow (iii)$ are immediate with $p = 1$.

$(ii) \Rightarrow (\mathsf{NP} \subseteq \mathsf{P/poly})$. This corresponds to Step 2 as sketched in the introduction. Let $(\mathrm{Yes}, \mathrm{No})$ denote the promise problem prUSAT, i.e., Yes denotes the set of uniquely satisfiable circuits, and No the set of unsatisfiable circuits. Assume that there exists a randomized $\mathsf{P/poly}$ isolation procedure A with success probability $p \geqslant \frac{2}{3} + \frac{1}{\mathsf{poly}(\ell)}$. By Lemma 4, it suffices to show that $\mathrm{prUSAT} \in \mathsf{P/poly}$. We apply Adleman's argument (Lemma 1) to A, where P_1 expresses the soundness property of A, and P_2 its p-completeness. We can set $p_1 = p'_1 = 1$ and $p' \doteq p'_2(\ell) > \frac{2}{3}$ by picking $\delta > 0$ sufficiently small. We obtain a deterministic $\mathsf{P/poly}$-algorithm B that maps any deterministic circuit C to a list of deterministic circuits $C'_1,\dots,C'_t$ with the following properties: (i) every satisfying assignment of every C'_i also satisfies C, and (ii) if C is satisfiable, then at least a p'-fraction of the circuits C'_i have a unique satisfying assignment. We want to apply Ko's argument, Lemma 2, to prove $\mathrm{prUSAT} \in \mathsf{P/poly}$. For this, we construct the following binary relation $R \subseteq \mathrm{Yes} \times (\mathrm{Yes} \cup \mathrm{No})$. For $C_1 \in \mathrm{Yes}$ with the unique satisfying assignment w_1 and for $C_2 \in (\mathrm{Yes} \cup \mathrm{No})$, we set $R(C_1, C_2)$ true if and only if at least one of the following conditions holds:

(a) w_1 satisfies C_2.
(b) w_1 satisfies less than a p'-fraction of the circuits C'_i on the list $B(C)$, where $C \doteq \min(C_1, C_2) \vee \max(C_1, C_2)$.

It remains to verify the three conditions in Lemma 2. For (K1), if $R(C_1, C_2)$, then w_1 satisfies C_2 and hence $C_2 \in \mathrm{Yes}$, or w_1 satisfies less than a p'-fraction of all circuits C'_i in the list $B(C)$. The latter implies that the

list $B(C)$ contains at least one successful isolation C'_i of C that is not satisfied by w_1. Since the unique satisfying assignment of this C'_i is not w_1, it must be a satisfying assignment of C_2. In either case, we have that $C_2 \in \text{Yes}$.

To show (K2), assume for contradiction that there are $C_1, C_2 \in \text{Yes}$ such that neither $R(C_1, C_2)$ nor $R(C_2, C_1)$ holds. Recall that the list $(C'_1, \ldots, C'_t) \doteq B(C)$ depends only on the set $\{C_1, C_2\}$ and not on the order of the inputs. By the assumption, we know that C_1 and C_2 have different unique satisfying assignments w_1 and w_2 that each satisfy at least a p'-fraction of the C'_i. Inclusion-exclusion yields that at least a fraction $2 \cdot p' - 1$ of the circuits C'_i on the list $B(C)$ are satisfied by *both* assignments. Since $2 \cdot p' - 1 > 1/3 > 1 - p'$, this contradicts the fact that B produces a list of circuits, at least p' of which have a unique satisfying assignment. Hence $R(C_1, C_2)$ or $R(C_2, C_1)$ holds.

For (K3), note that, for a fixed $C_1 \in \text{Yes}$, the membership of (C_1, C_2) in R can be decided by a deterministic circuit R_{C_1} that uses C_1, w_1, and $p't$ as advice, and B as a subroutine. The size of the circuit is a fixed polynomial in the length of C_1 and the circuit complexity of B. Thus R satisfies the conditions of Lemma 2, and we get $\text{prUSAT} \in \mathsf{P/poly}$.

$(iii) \Rightarrow (\mathsf{NP} \subseteq \mathsf{P/poly})$. This is analogous to the previous case, with the exception that we slightly modify the definition of R. We start from a randomized $\mathsf{P/poly}$-computable satisfiability-preserving isolation A and transform it into a deterministic algorithm B, again using Adleman's argument, where we can set $p_1 = p'_1 = 1$ and $p' \doteq p'_2(\ell) > 0$ by picking $\delta > 0$ sufficiently small. On input C, the algorithm B outputs a list of circuits C'_i such that: all satisfying assignments of C'_i also satisfy C, and if C is satisfiable, then each C'_i is satisfiable and at least one of the C'_i in the list is uniquely satisfiable. For $C_1 \in \text{Yes}$ with the unique satisfying assignment w_1 and for $C_2 \in (\text{Yes} \cup \text{No})$, we set $R(C_1, C_2)$ true if and only if at least one of the following conditions holds:

(a) w_1 satisfies C_2.
(b) Some circuit C'_i on the list $B(C)$ is not satisfied by w_1, where $C \doteq \min(C_1, C_2) \vee \max(C_1, C_2)$.

To argue (K1), if $R(C_1, C_2)$ holds, then w_1 satisfies C_2 and hence $C_2 \in \text{Yes}$, or some circuit C'_i in the list $B(C)$ is not satisfied by w_1. In the latter case, since B is satisfiability-preserving, this implies that $C_2 \in \text{Yes}$. For (K2), if neither $R(C_1, C_2)$ nor $R(C_2, C_1)$ holds, then C_1 and C_2 have two distinct unique satisfying assignments w_1 and w_2, respectively, and every circuit C'_i is satisfied by *both* assignments w_1 and w_2. This contradicts the fact that B outputs at least one uniquely satisfiable C'_i. The efficiency condition (K3) can be argued just as in the previous case. Thus, by Ko's argument, we have $\text{prUSAT} \in \mathsf{P/poly}$. ∎

Remark 6. We stated Theorem 5 for randomized isolation and randomized satisfiability-preserving isolation, but the proof does not make use of all properties of these notions. For example, the algorithm A only ever gets invoked for inputs C that have exactly one or exactly two satisfying assignments, so we do not need to make any assumptions on A's behavior for other inputs. The soundness and p-completeness conditions on those inputs can also be relaxed. These observation allow us to generalize the theorem as follows. Assume that A is a randomized $\mathsf{P/poly}$-algorithm that maps a deterministic circuit C to a deterministic circuit C' such that the following two conditions hold:

(1) If C has a unique satisfying assignment w, then, with probability at least p_1, the circuit C' is satisfied by w.
(2) If C has exactly two satisfying assignments w_1 and w_2, then, with probability at least p_2, the circuit C' is not satisfied by both assignments w_1 and w_2.

Note that in case (1), C' can have satisfying assignments other than w, and in case (2), C' can be unsatisfiable or have satisfying assignments other than w_1 and w_2. Using Adleman's argument, we obtain from A a list-derandomization B that achieves (1) with p_1 replaced by $p'_1 = p_1 - \epsilon$ (or $p'_1 = 1$ if $p_1 = 1$) and (2) with p_2 replaced by $p'_2 = p_2 - \epsilon$ (or $p'_2 = 1$ if $p_2 = 1$), where $\epsilon \doteq \epsilon(\ell)$ is any function such that $\epsilon(\ell) \geqslant 1/\mathsf{poly}(\ell)$ and the probabilities are interpreted with respect to the uniform distribution over the list $B(C)$.

To adapt the proof of Theorem 5 to this more general setting, we define $R(C_1, C_2)$ for C_1 uniquely satisfied by w_1 and C_2 having at most one satisfying assignment, by the following conditions:

(a) w_1 satisfies C_2, or
(b) w_1 satisfies less than a p'_1-fraction of the C'_i in the list $B(C)$, where $C \doteq \min(C_1, C_2) \vee \max(C_1, C_2)$.

The relation R satisfies (K3) just as in the proof of Theorem 5. We claim that R also satisfies (K1) and (K2) if $p'_1 + \frac{1}{2}p'_2 > 1$. The latter inequality can be achieved whenever $p_1 + \frac{1}{2}p_2 \geqslant 1 + \frac{1}{\mathsf{poly}(\ell)}$. By Ko's argument, the existence of such an algorithm A then implies $\mathsf{NP} \subseteq \mathsf{P/poly}$. We briefly argue (K1) and (K2).

Proof of (K1). Let $R(C_1, C_2)$ hold. If (a) holds, then C_2 is satisfiable. Otherwise (b) holds, and we assume for contradiction that C_2 is unsatisfiable. Then C has the unique witness w_1, in which case (1) guarantees that a fraction at least p'_1 of the C'_i has w_1 as a witness. But this contradicts (b), so C_2 must be satisfiable.

Proof of (K2). Assume neither $R(C_1, C_2)$ nor $R(C_2, C_1)$ holds for some uniquely satisfiable C_1 and C_2. Then C has exactly two witnesses w_1 and w_2, which must be distinct since (a) does not hold. Because (b) does not hold, a fraction at least $2p'_1 - 1$ of the C'_i are satisfied by both assignments. This contradicts (2) since $2p'_1 - 1 > 1 - p'_2$.

This view simultaneously generalizes the cases (ii) and (iii) of the above theorem, and interpolates between them.

In particular, (ii) is captured by $p_1, p_2 \geqslant \frac{2}{3} + \frac{1}{\mathsf{poly}(\ell)}$, and (iii) by $p_1 = 1$ and $p_2 \geqslant 1/\mathsf{poly}(\ell)$.

On the other hand, the Valiant–Vazirani isolation lemma yields $p_1 = 1/2$ and $p_2 = 3/4$. Recall that the Valiant–Vazirani isolation lemma intersects the solution space with a random number of random hyperplanes. Applied to circuits with at most two solutions, it suffices to fix the number of hyperplanes to one. This achieves the above guarantees since any given witness is on the hyperplane with probability $p_1 = 1/2$, and two distinct witnesses are not both on the hyperplane with probability $p_2 = 3/4$.

The remark also applies to randomized P/poly-reductions that map satisfiable circuits C to circuits C' with an odd number of satisfying assignments such that all satisfying assignments of C' also satisfy C. A randomized polynomial-time algorithm that achieves this with success probability $1/2$ was given by Gupta [6]. Since such reductions satisfy (1) and (2) where $p_1 = p_2$ is the success probability, our results rule out the possibility of improving the success probability to $2/3 + 1/\mathsf{poly}(n)$, unless $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{poly}$. In fact, we obtain the following corollary to the proof of Theorem 5.

Corollary 7. *Each of the following statements is equivalent to* $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{poly}$.

- (i) *There is a randomized* P/poly*-computable reduction that maps circuits C to circuits C' such that, if C is satisfiable, then with probability at least $p \geqslant \frac{2}{3} + \frac{1}{\mathsf{poly}(\ell)}$ the circuit C' has an odd number of satisfying assignments, each of which also satisfies C.*
- (ii) *There is a randomized* P/poly*-computable reduction that maps circuits C to circuits C' such that, if C is satisfiable, then C' is satisfiable, and with probability at least $p \geqslant \frac{1}{\mathsf{poly}(\ell)}$ the circuit C' has an odd number of satisfying assignments, each of which also satisfies C.*

C. Uniform disambiguation for nondeterministic circuits

Similar to the case of deterministic circuits, we first show that the existence of polynomial-time minimal witness isolation for the class of nondeterministic circuits is equivalent to $\mathsf{NP} = \mathsf{coNP}$.

Theorem 8. *There is a* P*-computable minimal witness isolation for nondeterministic circuits if and only if* $\mathsf{NP} = \mathsf{coNP}$.

D. Non-uniform disambiguation for nondeterministic circuits

We now develop the analog of our main result (Theorem 5) for nondeterministic instead of deterministic circuits. One motivation is the fact that a witness-recoverable disambiguation for deterministic circuits implies an isolation for nondeterministic circuits.

To prove the collapse direction, we again follow the two-step approach outlined in the introduction. The following lemma corresponds to Step 1.

Lemma 9. *If* $\mathrm{prUSAT} \in \mathsf{coNP}/\mathsf{poly}$ *then* $\mathsf{coNP} \subseteq \mathsf{NP}/\mathsf{poly}$.

The other direction in Lemma 9 trivially holds but we don't need it. Here is the analog of Theorem 5 for nondeterministic circuits.

Theorem 10. *Each of the following statements is equivalent to* $\mathsf{coNP} \subseteq \mathsf{NP}/\mathsf{poly}$.

- *(i) There is a* P/poly*-computable minimal witness isolation for nondeterministic circuits.*
- *(ii) There is a randomized* P/poly*-computable isolation for nondeterministic circuits with success probability $p \geqslant \frac{2}{3} + \frac{1}{\mathsf{poly}(\ell)}$.*
- *(iii) There is a randomized* P/poly*-computable satisfiability-preserving isolation for nondeterministic circuits with success probability $p \geqslant \frac{1}{\mathsf{poly}(\ell)}$.*
- *(iv) There is a randomized* P/poly*-computable witness-recoverable disambiguation for nondeterministic circuits with success probability $p \geqslant \frac{2}{3} + \frac{1}{\mathsf{poly}(\ell)}$.*

Furthermore, a randomized P/poly*-computable witness-recoverable disambiguation for deterministic circuits with success probability $p \geqslant \frac{2}{3} + \frac{1}{\mathsf{poly}(\ell)}$ also implies* $\mathsf{coNP} \subseteq \mathsf{NP}/\mathsf{poly}$.

Remark 11. We pointed out after the proof of Theorem 5 that a relaxed form of disambiguation is sufficient for the proof of cases (ii) and (iii) to go through. The same relaxation, this time for nondeterministic circuits, is possible for the cases (ii) and (iii) of Theorem 10, for the same reasons.

IV. Blackbox isolation

We consider a general situation where some randomized procedure is used to isolate one element in a given unknown set W in some specified family $\mathcal{W}$ of subsets of $\{0,1\}^n$. The randomized procedure can be designed depending on $\mathcal{W}$, but it is not given any information on which $W \in \mathcal{W}$ is chosen. The randomized procedure can check whether a given $w \in \{0,1\}^n$ is chosen or not; in other words, it is specified as a distribution $\mathcal{D}$ over subsets of $\{0,1\}^n$, where each $D \in \mathcal{D}$ is the set of strings that the randomized procedure selects when its random seed is fixed. This leads to the following type of isolation. Below, for a distribution $\mathcal{D}$ and an element D from the support of $\mathcal{D}$, we denote by $D \leftarrow \mathcal{D}$ the fact that D is chosen according to the distribution $\mathcal{D}$.

Definition 2 (Blackbox isolation). For any family $\mathcal{W}$ of nonempty subsets of $\{0,1\}^n$, a *blackbox isolation procedure* is a distribution $\mathcal{D}$ over subsets D of $\{0,1\}^n$. For any $D \in \mathcal{D}$ and any $W \in \mathcal{W}$, we say that D *succeeds on* W if $|D \cap W| = 1$.

The *isolation probability* of $\mathcal{D}$ for $\mathcal{W}$ is defined as $\min_{W \in \mathcal{W}} \Pr_{D \leftarrow \mathcal{D}}[\,|D \cap W| = 1\,]$. While this is regarded as the "worst-case" isolation probability, we may also consider

an average isolation probability. For this, we regard $\mathcal{W}$ as a distribution over subsets of $\{0,1\}^n$. For any distribution $\mathcal{W}$ over subsets of $\{0,1\}^n$ and any blackbox isolation procedure $\mathcal{D}$, the *average isolation probability* of $\mathcal{D}$ for $\mathcal{W}$ is defined as $\mathrm{E}_{W\leftarrow\mathcal{W}}[\,\mathrm{Pr}_{D\leftarrow\mathcal{D}}[\,|D\cap W|=1\,]\,]$. Clearly, the average isolation probability for a distribution $\mathcal{W}$ is an upper bound on the isolation probability for the corresponding subset family $\mathcal{W}$.

We now construct a distribution $\mathcal{W}^*$ for which the average isolation probability of any blackbox isolation $\mathcal{D}$ is $O(1/n)$. In order to do so, we first analyze what happens with the distribution $\mathcal{W}_K$ defined as follows, where K is any integer in the range $1 \leqslant K \leqslant N \doteq 2^n$: We put each $w \in \{0,1\}^n$ into W independently with probability $p_K \doteq K/N$. Roughly, $W \leftarrow \mathcal{W}_K$ has K strings on average. That is, we consider the isolation when we can approximate the target set size well. The Valiant–Vazirani procedure achieves an isolation probability of at least $1/8$ when given an integer k such that $|W| \in [2^k, 2^{k+1}]$, and an isolation probability of at least $1/4$ when given an integer k such that $|W| = 2^k$ (see, e.g., [15, p. 450–451]). We show that one cannot go beyond $(1+o(1))/e$ using any blackbox isolation procedure when $K = o(N)$. More precisely, we obtain the following bound.

Theorem 12. *For any blackbox isolation procedure $\mathcal{D}$, its average isolation probability for $\mathcal{W}_K$ is at most* $(1-\frac{K}{N})^{-1}e^{-1}$.

Proof: Consider any set D with H elements. Then its isolation probability for $\mathcal{W}_K$ is

$$\begin{aligned}\Pr_{W\leftarrow\mathcal{W}_K}[\,|D\cap W|=1\,] &= H\cdot p_K\,(1-p_K)^{H-1}\\ &= \Big(1-\frac{K}{N}\Big)^{-1}\cdot\frac{HK}{N}\cdot\Big(1-\frac{K}{N}\Big)^{H} \qquad (2)\\ &\leqslant \Big(1-\frac{K}{N}\Big)^{-1}\frac{HK}{N}e^{-HK/N} \leqslant \Big(1-\frac{K}{N}\Big)^{-1}e^{-1}\,,\end{aligned}$$

where the last inequality follows since $x\cdot e^{-x}$ has e^{-1} as its maximum value, which is achieved for $x=1$, i.e., for $H = N/K$. Note that the upper bound is the same for any D. Since the average isolation probability of $\mathcal{D}$ is a convex combination of the probabilities that $|D\cap W|=1$, the result follows. ∎

We construct the distribution $\mathcal{W}^*$ as a uniform superposition of the distributions $\mathcal{W}_K$, where K ranges over a well-chosen set $\mathcal{K}$. For K not too close to N, (2) shows that the isolation probability for $\mathcal{W}_K$ of a set D with H elements is maximized for H around N/K, and decreases rapidly when H deviates from N/K. This means that if we pick the values of K in $\mathcal{K}$ such that their ratios remain far from 1, then any set D can only have a significant contribution to the isolation probability for $\mathcal{W}_K$ for a few $K \in \mathcal{K}$, and the overall isolation probability of D for $\mathcal{W}^*$ becomes $O(1/|\mathcal{K}|)$. In particular, for a geometrically increasing set of values $K \in \mathcal{K}$, we obtain the tight upper bound of $\Theta(1/\log N) = \Theta(1/n)$ on the isolation probability of any blackbox isolation for $\mathcal{W}^*$.

The next theorem refers to the following specific distribution $\mathcal{W}^*$: Choose K from $\mathcal{K} \doteq \{1,2,4,\ldots,2^{n-1}\}$ uniformly at random, and then sample W according to the distribution $\mathcal{W}_K$.

Theorem 13. *For any blackbox isolation procedure $\mathcal{D}$, its average isolation probability for $\mathcal{W}^*$ is $O(1/n)$.*

Proof: Since the average isolation probability of $\mathcal{D}$ is a convex combination of the probabilities that $|D\cap W|=1$ for all fixed D, it suffices to upper bound the latter probabilities. Let D be any set with H elements. By (2), we have that

$$\begin{aligned}\Pr_{W\leftarrow\mathcal{W}^*}[\,|D\cap W|=1\,] &= \frac{1}{n}\sum_{K\in\mathcal{K}}\Pr_{W\leftarrow\mathcal{W}_K}[\,|D\cap W|=1\,]\\ &= \frac{1}{n}\cdot\sum_{K\in\mathcal{K}}\Big(1-\frac{K}{N}\Big)^{-1}\cdot\frac{HK}{N}\cdot\Big(1-\frac{K}{N}\Big)^{H}.\end{aligned}$$

To upper bound the right-hand side, we split the sum into the cases $K \le N/H$ and $K > N/H$. Then noting that $K \le 2^{n-1} = N/2$, we have

$$\begin{aligned}&\sum_{K\in\mathcal{K}}\Big(1-\frac{K}{N}\Big)^{-1}\cdot\frac{HK}{N}\cdot\Big(1-\frac{K}{N}\Big)^{H}\\ &\quad\le \sum_{K\in\mathcal{K}}\frac{2HK}{N}\Big(1-\frac{K}{N}\Big)^{H}\\ &\quad\le \sum_{\substack{K\in\mathcal{K}\\K\le N/H}}\frac{2HK}{N}\Big(1-\frac{K}{N}\Big)^{H}\\ &\qquad+\sum_{K>N/H}\frac{2HK}{N}e^{-HK/N}\\ &\quad\le \sum_{\substack{K\in\mathcal{K}\\K\le N/H}}\frac{2HK}{N}\Big(1-\frac{K}{N}\Big)^{H}+O(1)\,, \qquad (3)\end{aligned}$$

where the last line follows from the fact that $\sum_{x\ge1} xe^{-x} = O(1)$. On the other hand, since we have

$$\frac{2HK}{N}\Big(1-\frac{K}{N}\Big)^{H} \le \frac{2HK}{N}\Big(1-\frac{HK}{N}+\frac{1}{2}\Big(\frac{HK}{N}\Big)^2\Big)\,,$$

and

$$\begin{aligned}&\sum_{\substack{K\in\mathcal{K}\\K\le N/H}}\frac{2HK}{N}\left(1-\frac{HK}{N}+\frac{1}{2}\left(\frac{HK}{N}\right)^2\right)\\ &\qquad\le \frac{2H}{N}\cdot\frac{2N}{H}-\frac{2H^2}{N^2}\cdot\frac{4N^2}{3H^2}+\frac{2H^3}{2N^3}\cdot\frac{8N^3}{7H^3}\le 3\,,\end{aligned}$$

we get that (3) is $O(1)$ and obtain the desired bound. ∎

One application of isolation is finding witnesses using *nonadaptive* queries to a satisfiability oracle. The standard search-to-decision reduction constructs a witness bit-by-bit using n adaptive queries to a satisfiability oracle.

If the witness is unique, then the queries can be made in a nonadaptive fashion. The Valiant–Vazirani procedure thus yields a nonadaptive search-to-decision procedure that makes n queries and succeeds with probability $\Omega(1/n)$. By running the procedure $O(n)$ times in parallel, we obtain a nonadaptive search-to-decision procedure that makes $O(n^2)$ queries and succeeds with probability $\Omega(1)$. Ben-David et al. [3] present an alternate procedure with similar behavior. Recently, Kawachi, Rossman, and Watanabe [10] extended our blackbox framework and showed that in that setting every nonadaptive search-to-decision procedure with success probability $\Omega(1)$ has to make $\Omega(n^2)$ queries.

V. Further discussion

Our result that an efficient deterministic isolation procedure would imply $\mathsf{NP} \subseteq \mathsf{P/poly}$ (Theorem 5) can be interpreted as saying that derandomizing the Isolation Lemma (in the strong sense, where the output of the isolation procedure is a *single* circuit) would imply circuit *upper* bounds for NP. This is in contrast to the previous results showing that derandomization would imply circuit *lower* bounds for NEXP [2, 8, 9].

While we have argued that an efficient randomized isolation with success probability $p > 2/3$ is unlikely to exist, it remains an interesting open problem to consider intermediate values of p, namely $\omega(1/n) < p \leqslant 2/3$. Regarding more general mapping reductions from NP to UP, does the assumption $\mathsf{NP} = \mathsf{UP}$ lead to any surprising consequences?

Our results also apply to mapping reductions from NP to $\oplus\mathsf{P}$ that can only remove witnesses. In this setting the open range for the success probability is $1/2 < p \leqslant 2/3$. In contrast, general mapping reductions from NP to $\oplus\mathsf{P}$ can have success probabilities arbitrarily close to 1, and are therefore strictly more powerful unless $\mathsf{NP} \subseteq \mathsf{P/poly}$.

Acknowledgements. We would like to thank Leslie Valiant for his insightful comments on the results presented in the paper.

H. Dell partially supported by the Alexander von Humboldt Foundation and by NSF grant 1017597. Most of V. Kabanets' research was done during a visit to Tokyo Institute of Technology in the Summer of 2011. D. van Melkebeek partially supported by NSF grant 1017597.

References

[1] S. Arora and B. Barak, *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009.

[2] V. Arvind and P. Mukhopadhyay, "Derandomizing the isolation lemma and lower bounds for circuit size," in *APPROX-RANDOM*, Springer, 2008, pp. 276–289.

[3] S. Ben-David, B. Chor, O. Goldreich, and M. Luby, "On the theory of average-case complexity," *JCSS*, vol. 44, no. 2, pp. 193–219, 1992.

[4] S. Chari, P. Rohatgi, and A. Srinivasan, "Randomness-optimal unique element isolation with applications to perfect matching and related problems," *SICOMP*, vol. 24, no. 5, pp. 1036–1050, 1995.

[5] H. Dell, V. Kabanets, D. van Melkebeek, and O. Watanabe, "Is Valiant–Vazirani's isolation probability improvable?," ECCC, TR11-151 Rev. 1, 2012.

[6] S. Gupta, "Isolating an odd number of elements and applications in complexity theory," *Theory of Computing Systems*, vol. 31, pp. 27–40, 1998.

[7] L. A. Hemaspaandra, A. V. Naik, M. Ogihara, and A. L. Selman, "Computing solutions uniquely collapses the polynomial hierarchy," *SICOMP*, vol. 25, no. 4, pp. 697–708, 1996.

[8] R. Impagliazzo, V. Kabanets, and A. Wigderson, "In search of an easy witness: Exponential time vs. probabilistic polynomial time," *JCSS*, vol. 65, no. 4, pp. 672–694, 2002.

[9] V. Kabanets and R. Impagliazzo, "Derandomizing polynomial identity tests means proving circuit lower bounds," *C. Compl.*, vol. 13, no. 1–2, pp. 1–46, 2004.

[10] A. Kawachi, B. Rossman, and O. Watanabe, "Query complexity and error tolerance of witness finding algorithms," ECCC, TR12-002, 2012.

[11] A. R. Klivans and D. van Melkebeek, "Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses," *SICOMP*, vol. 31, no. 5, pp. 1501–1526, 2002.

[12] K.-I. Ko, "On self-reducibility and weak P-selectivity," *JCSS*, vol. 26, pp. 209–211, 1983.

[13] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani, "Matching is as easy as matrix inversion," *Combinatorica*, vol. 7, no. 1, pp. 105–113, 1987.

[14] A. V. Naik, K. W. Regan, and D. Sivakumar, "On quasilinear time complexity theory," *TCS*, vol. 148, no. 2, pp. 325–349, 1995.

[15] C. H. Papadimitriou, *Computational Complexity*. Addison-Wesley, 1994.

[16] K. Reinhardt and E. Allender, "Making nondeterminism unambiguous," *SICOMP*, vol. 29, no. 4, pp. 1118–1131, 2000.

[17] A. L. Selman, "A taxonomy of complexity classes of functions," *JCSS*, vol. 48, pp. 357–381, 1994.

[18] ——, "P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP," *Math. Sys. Th.*, vol. 13, pp. 55–65, 1979.

[19] S. Toda, "PP is as hard as the polynomial-time hierarchy," *SICOMP*, vol. 20, no. 5, pp. 865–877, 1991.

[20] L. G. Valiant and V. V. Vazirani, "NP is as easy as detecting unique solutions," *TCS*, vol. 47, pp. 85–93, 1986.