Electronic Colloquium on Computational Complexity, Report No. 328 (2026)

The Ultimate Karp–Lipton Collapse

Edward A. Hirsch* Ilya Volkovich†

November 4, 2026

Abstract

Karp and Lipton [KL80] proved that if $\mathbf{NP} \subseteq \mathbf{P/poly}$, then the polynomial hierarchy collapses. A long line of work pushed the collapse to smaller classes. In particular, Chakaravarthy and Roy obtained two such collapses, to (apparently) incomparable classes $\mathbf{O}_2^{\mathbf{p}}$ [CR06] and $\mathbf{P}^{\mathbf{prMA}}$ [CR11]. Hirsch and Volkovich [HV25] united them by a collapse to the input-oblivious version $\mathbf{P}^{\mathbf{prOMA}}$ of $\mathbf{P}^{\mathbf{prMA}}$, which is contained in both; Hirsch, Volkovich and Williams [HVW26] obtained a collapse to a slightly smaller class. Currently, the three deepest known collapses, pairwise incomparable to the best of our current knowledge, are to $\mathbf{P}^{\mathbf{prOMA}}$ and its versions; to $\mathbf{ZPP}^{\mathbf{ONP}}$, the input-oblivious version of the collapse to $\mathbf{ZPP}^{\mathbf{NP}}$ of Köbler and Watanabe [KW98], which builds on [BCG+96]; and to $\mathbf{OMIN} = \mathbf{OL}_2^{\mathbf{p}}$ [HV25, HVW26], the input-oblivious version of the class $\mathbf{L}_2^{\mathbf{p}}$ of Korten and Pitassi [KP24]. The last collapse is proved in this paper. The class **OMIN** has fixed-polynomial circuit lower bounds [HVW26]; no smaller collapse class with such lower bounds is known.

We unite the three collapses in the same spirit, by a collapse to a class contained in all of them. Derandomization results for input-oblivious classes in [HV25, HVW26] in fact use a two-stage model: first the computation uses one oracle (and ignores the input), then another oracle. This model is inspired by the untrusted advice classes of Aaronson and Drucker (including $\mathbf{YP}^*$) [AD14] and by the learning algorithm of [BCG+96] as used in [KW98]. We use three-stage *generator–tester–decoder* computations. In this model, a polynomial-time generator with an oracle computes an advice string from 1^n, a tester certifies it, and a polynomial-time decoder uses it on the input. Let **OM** be the class of languages decided in this model by a generator with an oracle for a restricted version of **prOMA**, a **co-ONP** tester, and a **P** decoder. This class is contained in the three collapse classes above, $\mathbf{OM} \subseteq \mathbf{OMIN} \cap \mathbf{P}^{\mathbf{prOMA}} \cap \mathbf{ZPP}^{\mathbf{ONP}}$. We argue that it is a natural class (for example, closed under polynomial-time Turing reductions). Each of the three containments comes from executing the generator in a different way: with its oracle; with its oracle replaced by a fixed language in $\mathbf{L}_2^{\mathbf{p}}$; or with the oracle answers sampled at random, after which the tester checks the advice. We show that if $\mathbf{NP} \subseteq \mathbf{P/poly}$, then $\mathbf{PH} = \mathbf{OM}$. The collapse also holds with a unique certified advice string.

At the other end of the picture, all these input-oblivious collapse classes lie in $\mathbf{P}^{\mathbf{prSBP}}$, and so do even the non-input-oblivious versions $\mathbf{P}^{\mathbf{prMA}}$, $\mathbf{L}_2^{\mathbf{p}}$ and $\mathbf{S}_2^{\mathbf{p}}$ of $\mathbf{P}^{\mathbf{prOMA}}$, $\mathbf{OL}_2^{\mathbf{p}}$ and $\mathbf{O}_2^{\mathbf{p}}$, as well as $\mathbf{P}^{\mathbf{prS}_2^{\mathbf{p}}}$: we prove $\mathbf{P}^{\mathbf{prS}_2^{\mathbf{p}}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$. This strengthens $\mathbf{S}_2^{\mathbf{p}} \subseteq \mathbf{P}^{\mathbf{prAM}}$ of Chakaravarthy and Roy [CR11] and our earlier containment $\mathbf{L}_2^{\mathbf{p}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$ [HV25]. The proof unites the Goldreich–Wigderson-type [GW99] procedure of [CR11] with the approximate-counting technique of [HV25]. Its input-oblivious counterpart concerns $\mathbf{O}_2^{\mathbf{p}}$, the first collapse class of Chakaravarthy and Roy: every language in $\mathbf{O}_2^{\mathbf{p}}$ has polynomial-size circuits constructible in $\mathbf{FP}^{\mathbf{prSBP}}$.

*Department of Computer Science, Ariel University, Israel. Email: edwardh@ariel.ac.il. This research was conducted with the support of the State of Israel, the Ministry of Immigrant Absorption, and the Center for the Absorption of Scientists.

†Computer Science Department, Boston College, Chestnut Hill, MA. Email: ilya.volkovich@bc.edu.

ISSN 1433-8092

Contents

1 Introduction **3**
- 1.1 Karp–Lipton collapses . . . 3
- 1.2 Generator–tester–decoder computations . . . 4
- 1.3 Our results . . . 5
- 1.4 Techniques . . . 7
- 1.5 Related work . . . 8
- 1.6 Organization . . . 9

2 Preliminaries **9**
- 2.1 Conventions . . . 9
- 2.2 Promise problems and oracle access . . . 10
- 2.3 Complexity classes . . . 10
- 2.4 Generator–tester–decoder computations . . . 13

3 Linear orders and the collapse to OMIN **14**
- 3.1 Closure under polynomial-time Turing reductions . . . 14
- 3.2 Tally languages make linear orders oblivious . . . 15
- 3.3 Replacing Merlin–Arthur oracles by oblivious linear orders . . . 16
- 3.4 The collapse to **OMIN** . . . 17

4 The ultimate collapse **18**
- 4.1 Strict oblivious Merlin–Arthur oracles . . . 18
- 4.2 The class **OM** . . . 19
- 4.3 Three ways to run the generator . . . 20
- 4.4 The collapse . . . 20

5 A Goldreich–Wigderson lemma with approximate counting, and $\mathbf{P}^{\mathbf{prS}_2^{\mathbf{P}}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$ **21**

6 The oblivious setting of symmetric alternation **22**

7 Open problems **24**

References **25**

A Other routes to the collapses **28**
- A.1 Through approximate counting . . . 28
- A.2 Without promise problems . . . 29
- A.3 The collapse to **OM** without **OMIN** . . . 29
- A.4 Relativization . . . 30

B Further consequences **30**
- B.1 Derandomizing approximate counting . . . 30
- B.2 Function classes, exponential time, and other classes inside $\mathbf{S}_2^{\mathbf{P}}$ . . . 30
- B.3 Minimum finding . . . 31

1 Introduction

1.1 Karp–Lipton collapses

Karp and Lipton [KL80] proved that if $\mathbf{NP} \subseteq \mathbf{P/poly}$, then the polynomial hierarchy collapses to its second level (the second level is credited in [KL80] to Sipser). Together with Kannan's theorem [Kan82] that $\mathbf{PH}$ contains, for every k, a language without circuits of size n^k, such a collapse yields *fixed-polynomial circuit lower bounds* by a *win–win argument*. Namely, if $\mathbf{NP} \not\subseteq \mathbf{P/poly}$, the lower bound holds already for $\mathbf{NP}$; otherwise $\mathbf{PH}$ collapses to the class in question, which then inherits Kannan's lower bound. A long line of work pushed the collapse to smaller classes: $\mathbf{ZPP^{NP}}$ [BCG+96, KW98]; the symmetric alternation class $\mathbf{S_2^P}$ [Can96, RS98], by an argument of Sengupta (see [Cai07]); its input-oblivious version $\mathbf{O_2^P}$ [CR06]; and $\mathbf{P^{prMA}}$ [CR11]. Here $\mathbf{prMA}$ is the class of promise problems with Merlin–Arthur protocols; the letter $\mathbf{O}$ in the class name stands for *input-obliviousness*: $\mathbf{S_2^P}$ certificates (or Merlin's proofs in $\mathbf{prMA}$) may depend on the length of the input but not on the input itself, which defines the class $\mathbf{O_2^P}$ (or $\mathbf{prOMA}$, respectively).

While for a class containing $\mathbf{NP}$, a collapse immediately gives fixed-polynomial lower bounds, input-oblivious classes lie in $\mathbf{P/poly}$ by definition and therefore do not contain $\mathbf{NP}$ unless $\mathbf{NP} \subseteq \mathbf{P/poly}$; for them, the collapse and the lower bound are two separate problems. For $\mathbf{O_2^P}$, a lower bound was found nearly two decades after the collapse, by Gajulapalli, Li and Volkovich [GLV24], using the range avoidance problem [KKMP21, Kor22, CHR24, Li24]. In [KP24], Korten and Pitassi introduced the class $\mathbf{L_2^P}$: an input is accepted when the minimum of a polynomial-time decidable linear order, which depends on the input, starts with 1. In the same paper, they also proved fixed-polynomial lower bounds for $\mathbf{L_2^P}$ and asked for a Karp–Lipton collapse. Subsequently, Hirsch and Volkovich [HV25] showed the collapse by proving $\mathbf{P^{prMA}} \subseteq \mathbf{L_2^P} \subseteq \mathbf{P^{prSBP}}$ and, in addition, introduced $\mathbf{OL_2^P}$ as the input-oblivious version of $\mathbf{L_2^P}$. Hirsch, Volkovich and Williams [HVW26] provided a different, yet equivalent, characterization of $\mathbf{OL_2^P}$ by showing that it equals the class $\mathbf{OMIN}$, in which the linear order itself depends only on the input length. They also obtained fixed-polynomial lower bounds for this class.

Uniting collapses. The two collapses of Chakaravarthy and Roy, to $\mathbf{O_2^P}$ [CR06] and to $\mathbf{P^{prMA}}$ [CR11], are incomparable as far as we know: $\mathbf{O_2^P}$ is input-oblivious, while $\mathbf{P^{prMA}}$ contains $\mathbf{NP}$. Hirsch and Volkovich [HV25] united them by deepening the collapse to $\mathbf{P^{prOMA}}$, the input-oblivious version of $\mathbf{P^{prMA}}$. This class is contained in both: $\mathbf{P^{prOMA}} \subseteq \mathbf{P^{prMA}}$ trivially, and $\mathbf{P^{prOMA}} \subseteq \mathbf{O_2^P}$ [HV25]. They also observed that input-oblivious collapses are currently the deepest ones. Hirsch, Volkovich and Williams proved that $\mathbf{NP} \subseteq \mathbf{P/poly}$ implies the collapse $\mathbf{PH} = \mathbf{P^{OprMA}}$ to a slightly smaller class [HVW26, Proposition 6.9], and their proof in fact gives a stricter promise class $\mathbf{sOprMA}$ in which Merlin's oblivious proof dominates every other proof on *every* input and *every* random string (see Definition 2.4 for a formal definition).

Three deepest collapses. Three collapse classes are now the deepest known. As far as we know, they are pairwise incomparable.

- $\mathbf{P^{prOMA}}$ introduced in [HV25], together with its slightly smaller versions $\mathbf{P^{OprMA}}$ [HVW26] and $\mathbf{P^{sOprMA}}$ (introduced in this paper).
- $\mathbf{ZPP^{ONP}}$, the input-oblivious version of the collapse to $\mathbf{ZPP^{NP}}$ of Köbler and Watanabe [KW98], which builds on [BCG+96]. It holds because $\mathbf{NP} \subseteq \mathbf{P/poly}$ implies $\mathbf{NP} = \mathbf{ONP}$ [FSW09].

- $\mathbf{OMIN} = \mathbf{OL_2^P}$ [HV25, HVW26]. We prove that $\mathbf{NP} \subseteq \mathbf{P/poly}$ implies $\mathbf{PH} = \mathbf{OMIN}$ (Theorem 2). Since $\mathbf{OMIN} \not\subseteq \mathbf{SIZE}[n^k]$ [HVW26], no smaller collapse class with fixed-polynomial lower bounds is known. After the lower bound of Gajulapalli, Li and Volkovich [GLV24], the class $\mathbf{O_2^P} \supseteq \mathbf{OL_2^P}$ was for a while the smallest natural input-oblivious class known to have both properties.

In the spirit of [HV25], we unite these three collapses: we prove a collapse to a class contained in all three of them (Theorem 1).

A word of caution is due. A Karp–Lipton class inside all three classes above is easy to obtain, even one of the generator–decoder kind described below. If $\mathcal{C}$ is a Karp–Lipton class closed under polynomial-time Turing reductions, then so is its *tally (unary) closure* $\mathbf{P}^{\mathbf{TALLY}\cap\mathcal{C}} \subseteq \mathcal{C}$ (Theorem 3.10), in which the advice for length n is computed from 1^n with an oracle from $\mathcal{C}$ (Proposition 2.11(b)). For $\mathcal{C} = \mathbf{P}^{\mathbf{sOprMA}} \cap \mathbf{ZPP}^{\mathbf{ONP}}$, the tally closure lies in $\mathbf{P}^{\mathbf{sOprMA}}$ and $\mathbf{ZPP}^{\mathbf{ONP}}$ because it is contained in $\mathcal{C}$, and in $\mathbf{OMIN} = \mathbf{P}^{\mathbf{TALLY}\cap\mathbf{L_2^P}}$ (Theorem 3.3) because $\mathcal{C} \subseteq \mathbf{P}^{\mathbf{prMA}} \subseteq \mathbf{L_2^P}$ [HV25]. Such a class, however, is obtained by a closure construction rather than from a natural model of computation. Our class comes from such a model, described below.

1.2 Generator–tester–decoder computations

Our model has three components. They work as follows on inputs of length n.

- A *generator* receives only 1^n (i.e., the input length instead of the input itself) and computes an advice string a. It is a polynomial-time transducer with access to an oracle. If the oracle is a promise problem, the access is *"loose"*, and the generator may therefore produce several different *legal outputs*.
- A *tester* receives $(1^n, a)$. It must accept every legal output of the generator, and it may accept only advice strings that work for the decoder.
- A *decoder* receives (x, a) and decides whether $x \in L$.

We write $\mathbf{OGTD}[\mathcal{A}, \mathcal{B}, \mathcal{C}]$ ("oblivious generator–tester–decoder") for the class of languages decided this way, with a generator from $\mathcal{A}$, a tester from $\mathcal{B}$, and a decoder from $\mathcal{C}$ (Definition 2.10). We write $*$ when no tester is required, **OUGTD** when the tester must accept exactly one advice string for each n, and we abbreviate $\mathbf{OGTD}[\mathbf{FP}^{\mathcal{X}}, *, \mathbf{P}]$ to $\mathbf{OGTD}[\mathcal{X}]$. Two-stage classes of this kind have been studied in fact since Karp and Lipton's paper: their class $\mathbf{P}/\mathrm{poly}$ is $\mathbf{OGTD}[\mathrm{poly}, *, \mathbf{P}]$, where the generator may be any polynomially bounded advice function. Köbler and Thierauf [KT94] bounded the complexity of computing correct advice. For a language A, $\mathbf{OGTD}[A]$ is the class of languages with $\mathbf{P}^A$-uniform polynomial-size circuits [Kan82, SW14], and it equals $\mathbf{P}^{\mathbf{TALLY}\cap\mathbf{P}^A}$ (Proposition 2.11). See Section 1.5 for more on this line of work.

The three-stage model is inspired by three predecessors.

- *Untrusted advice.* The class $\mathbf{YP}^*$ of Aaronson and Drucker [AD14] is $\mathbf{OGTD}[\mathrm{poly}, \mathbf{P}, \mathbf{P}]$. There, the advice is tested in polynomial time without access to the input.
- *Learning circuits.* In the algorithm of Bshouty, Cleve, Gavaldà, Kannan, and Tamon [BCG+96], as used by Köbler and Watanabe [KW98], a $\mathbf{ZPP}^{\mathbf{NP}}$ learner generates circuits. Equivalence queries test them, and for SAT these tests can be answered with an **NP** oracle.

- *Derandomization of oblivious classes.* The model underlies the classes $\mathbf{P}^{\mathbf{OL_2^P},\mathbf{prONP}}$ and $\mathbf{P}^{\mathbf{OL_2^P},\mathbf{ONP}}$ used in [HV25, HVW26], although it was not stated explicitly there. In the proof of $\mathbf{P}^{\mathbf{OprMA}} \subseteq \mathbf{P}^{\mathbf{OL_2^P},\mathbf{ONP}}$ [HVW26, Proposition 6.9], the $\mathbf{OL_2^P}$ oracle generates the minimum m of an oblivious linear order that encodes a pseudorandom generator, and the rest of the computation decodes. A Merlin–Arthur query x becomes the **ONP** query "some element is smaller than m, or Merlin has a proof for x that is accepted on the pseudorandom strings encoded by m". This query does not certify that m is minimal: it accepts every other candidate, and this disjunct makes the query input-oblivious; the minimum, together with Merlin's oblivious proof, is a common witness. On the minimum, the query tests Merlin's acceptance condition.

The main class of our paper defined along these lines is[1]

$$\mathbf{OM} = \mathbf{OGTD}[\mathbf{FP}^{\mathbf{sOprMA}}, \mathbf{co\text{-}ONP}, \mathbf{P}],$$

with the unambiguous-advice subclass $\mathbf{OUM} = \mathbf{OUGTD}[\mathbf{FP}^{\mathbf{sOprMA}}, \mathbf{co\text{-}ONP}, \mathbf{P}]$.

1.3 Our results

Theorem 1 (the ultimate Karp–Lipton collapse; Theorems 4.5 and 4.6)**.**

(1) $\mathbf{OUM} \subseteq \mathbf{OM} \subseteq \mathbf{OMIN} \cap \mathbf{P}^{\mathbf{sOprMA}} \cap \mathbf{ZPP}^{\mathbf{ONP}}$.

(2) *If* $\mathbf{NP} \subseteq \mathbf{P/poly}$, *then* $\mathbf{PH} = \mathbf{OM} = \mathbf{OUM}$.

Each of the three containments in (1) comes from executing the generator in a different way.

- Running the generator with its oracle gives $\mathbf{P}^{\mathbf{sOprMA}}$.
- Replacing the **sOprMA** oracle by a fixed language in $\mathbf{L_2^P}$ makes the advice canonical, and the argument about tally oracles then gives **OMIN**.
- Sampling the generator with an **ONP** oracle and checking the result with the tester gives $\mathbf{ZPP}^{\mathbf{ONP}}$. Sampling works because Merlin's proof dominates pointwise.

These containments are shown in Figure 1, which also shows the upper bound $\mathbf{OMIN} \subseteq \mathbf{SOUP}^{\mathbf{ONP}[1]}$ of [HVW26], where **SOUP** is their same-witness symmetric oblivious unambiguous class (see Definition 2.5); thus $\mathbf{OM} \subseteq \mathbf{SOUP}^{\mathbf{ONP}[1]}$ as well. Fixed-polynomial lower bounds for **OM** remain open.

The collapse to **OMIN** is the main ingredient of the proof of Theorem 1, and it is of independent interest.

Theorem 2 (Corollary 3.11)**.** *If* $\mathbf{NP} \subseteq \mathbf{P/poly}$, *then* $\mathbf{PH} = \mathbf{OMIN} = \mathbf{OL_2^P}$.

As $\mathbf{OL_2^P} \subseteq \mathbf{P/poly}$, Theorem 2 also implies that $\mathbf{NP} \subseteq \mathbf{P/poly}$ iff $\mathbf{NP} \subseteq \mathbf{OL_2^P}$ iff $\mathbf{PH} = \mathbf{OL_2^P}$ (Corollary 3.13). Unconditionally, it is open whether $\mathbf{ONP} \subseteq \mathbf{OL_2^P}$.

We also improve the upper bound for symmetric alternation, strengthening our earlier upper bound for $\mathbf{L_2^P}$ [HV25], and then discuss what it means in the input-oblivious setting (see Figure 2).

Theorem 3 (Theorem 5.4)**.** $\mathbf{P}^{\mathbf{prS_2^P}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$. *In particular,* $\mathbf{S_2^P} \subseteq \mathbf{P}^{\mathbf{prSBP}}$.

[1] The class name **OM** is the common prefix of **OMA** and **OMIN**: in **OM** the generator uses a (strict) oblivious Merlin–Arthur oracle, and this class lies inside **OMIN**.

Here **prSBP** is the promise version of the class **SBP** of Böhler, Glaßer and Meister [BGM06], which lies between **MA** and **AM**. A **prSBP** oracle is equivalent to an oracle for approximate counting (see, e.g., [HV25]). Moreover, $\mathbf{prSBP} \subseteq \mathbf{prAM}$ and $\mathbf{SBP} \subseteq \mathbf{BPP_{path}} \subseteq \mathbf{PP}$ [BGM06, HHT97]. Theorem 3 thus strengthens both $\mathbf{S_2^P} \subseteq \mathbf{P^{prS_2^P}} \subseteq \mathbf{P^{prAM}}$ of Chakaravarthy and Roy [CR11] and the earlier containment $\mathbf{L_2^P} \subseteq \mathbf{P^{prSBP}}$ [HV25]. In particular, $\mathbf{P^{prSBP}}$ contains all the input-oblivious collapse classes discussed above. The three deepest ones even lie in $\mathbf{L_2^P} \subseteq \mathbf{P^{prSBP}}$ [HV25]; for $\mathbf{ZPP^{ONP}}$ this follows from $\mathbf{ZPP^{ONP}} \subseteq \mathbf{BPP^{ONP}} \subseteq \mathbf{P^{OL_2^P,ONP}} \subseteq \mathbf{L_2^P}$ [HVW26]. Together with $\mathbf{P^{prMA}} \subseteq \mathbf{L_2^P} \subseteq \mathbf{P^{prSBP}}$ [HV25], Theorem 3 shows that $\mathbf{P^{prSBP}}$ contains the non-input-oblivious versions $\mathbf{P^{prMA}}$, $\mathbf{L_2^P}$ and $\mathbf{S_2^P}$ of $\mathbf{P^{prOMA}}$, $\mathbf{OL_2^P}$ and $\mathbf{O_2^P}$, and even $\mathbf{P^{prS_2^P}}$. We do not know whether it contains $\mathbf{ZPP^{NP}}$ (Section 7). Theorem 3 applies to every class inside $\mathbf{S_2^P}$, for instance to the classes **PTW**, **PCW** and **PMA** of unambiguous $\mathbf{\Sigma_2^P}$ problems defined by Gilboa, Goldberg, Koutsoupias and Nisan [GGKN25], who placed them in $\mathbf{S_2^P}$. Since they also showed $\mathbf{L_2^P} \subseteq \mathbf{PTW}$, already the resulting containment $\mathbf{PTW} \subseteq \mathbf{P^{prSBP}}$ strengthens $\mathbf{L_2^P} \subseteq \mathbf{P^{prSBP}}$ [HV25]. Its proof unites the techniques of [CR11] and [HV25]. Chakaravarthy and Roy gave a version of the Goldreich–Wigderson lemma [GW99] for succinctly represented matrices that uses a **prAM** oracle. We obtain one with approximate counting: given a circuit representing a Boolean matrix, a polynomial-time algorithm with a **prSBP** oracle finds a small set of rows that together beat every column (i.e., a *row-side* collectively irrefutable certificate), or a small set of columns that together beat every row (i.e., a *column-side* collectively irrefutable certificate) (Theorem 5.2). As a special case, the same lemma gives another minimum-finding algorithm, and thus another proof of $\mathbf{L_2^P} \subseteq \mathbf{P^{prSBP}}$ [HV25] (Appendix B.3).

In the generator–decoder language, the chain $\mathbf{P^{prMA}} \subseteq \mathbf{L_2^P} \subseteq \mathbf{S_2^P} \subseteq \mathbf{P^{prSBP}}$ has an input-oblivious counterpart (the second line in Figure 2):

Theorem 4 (the oblivious setting; Theorems 3.3, 3.5 and 6.3)**.**

(1) $\mathbf{OGTD[L_2^P]} = \mathbf{OL_2^P}$ $(= \mathbf{OMIN})$.

(2) $\mathbf{OGTD[prMA]} \subseteq \mathbf{OL_2^P}$.

(3) $\mathbf{O_2^P} \subseteq \mathbf{OGTD[prSBP]}$*; that is, every language in* $\mathbf{O_2^P}$ *has polynomial-size circuits constructible in* $\mathbf{FP^{prSBP}}$.

Thus

$$\begin{array}{ccccccc} \mathbf{P^{prMA}} & \subseteq & \mathbf{L_2^P} & \subseteq & \mathbf{S_2^P} & \subseteq & \mathbf{P^{prSBP}} \\ \cup\mathrm{I} & & \cup\mathrm{I} & & \cup\mathrm{I} & & \cup\mathrm{I} \\ \mathbf{OGTD[prMA]} & \subseteq & \mathbf{OL_2^P} & \subseteq & \mathbf{O_2^P} & \subseteq & \mathbf{OGTD[prSBP]} \end{array}$$

(see Figure 2), and Figure 1 places **OM** below the three deepest collapse classes. Part (3) concerns $\mathbf{O_2^P}$, one of the two collapse classes united in [HV25]: its languages have polynomial-size circuits that can be constructed with approximate counting.

Several byproducts.

- *Derandomizing approximate counting* (Corollary B.1). If approximate counting can be done in $\mathbf{FP^{NP}}$, then $\mathbf{P^{prS_2^P}} = \mathbf{P^{NP}}$ and $\mathbf{O_2^P} \subseteq \mathbf{OGTD[NP]}$. Moreover, for every k, some language with polynomial-size circuits constructible in $\mathbf{FP^{NP}}$ has no circuits of size n^k. Previously, to the best of our knowledge the collapse $\mathbf{S_2^P} = \mathbf{P^{NP}}$ was known under a hardness assumption implied by the one used to derandomize **AM** [SU06].

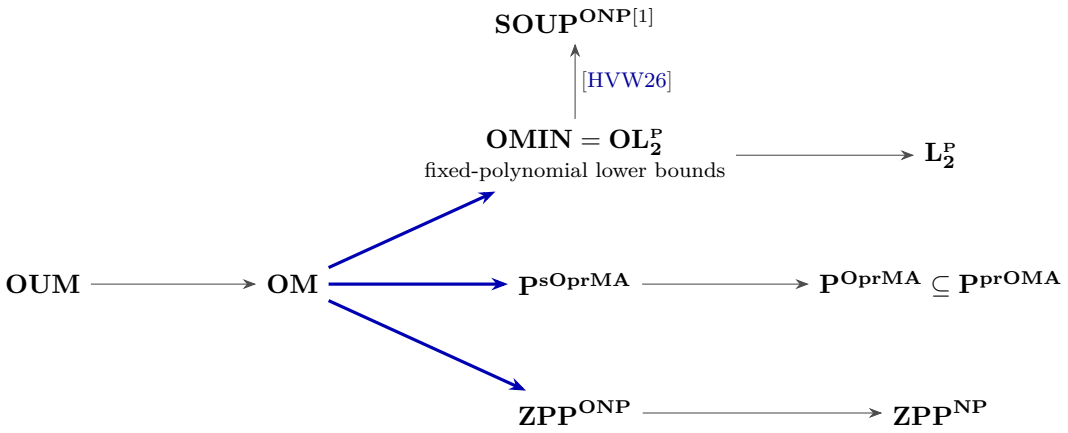


Figure 1: The ultimate collapse (Theorem 1): If $\mathbf{NP} \subseteq \mathbf{P/poly}$, all these classes coincide with **PH**. Thick arrows are the new unconditional containments to the three previously known classes, two of them known collapse classes; a collapse to **OMIN** is proved in this paper.

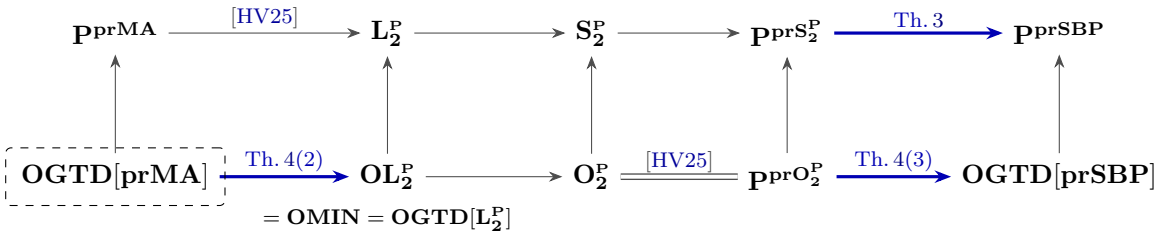


Figure 2: All these classes coincide with **PH** unless $\mathbf{NP} \not\subseteq \mathbf{P/poly}$. Fixed-polynomial lower bounds are known for all of them [CR11, KP24, GLV24, HVW26] except for **OGTD**[**prMA**] (dashed box).

- *Direct proofs for order-based* $\mathbf{L_2^P}$. Korten and Pitassi [KP24] defined $\mathbf{L_2^P}$ via polynomial-time decidable linear orders and stated that this definition is equivalent to the one by reductions to the linear ordering principle [KP24, Theorems 5 and 6]. To keep the paper self-contained, we use only the order-based definition, which is also the one used in [HVW26], and do not rely on this equivalence anywhere (Remark 2.8). In particular, we prove closure under polynomial-time Turing reductions for the order-based definition directly (Lemma 3.1), and we derive from it the pseudorandom generators that we use (Lemma 3.4). Both definitions give subclasses of $\mathbf{S_2^P}$ (Remark 2.8), and hence of $\mathbf{P^{prSBP}}$ by Theorem 3.

- *Function classes and exponential time* (Appendix B.2). $\mathbf{FP^{prS_2^P}} \subseteq \mathbf{FP^{prSBP}}$ and $\mathbf{S_2^E} \subseteq \mathbf{E^{prSBP}}$.

1.4 Techniques

Tally languages and canonical circuits (Theorem 2). If $\mathbf{NP} \subseteq \mathbf{P/poly}$, then every language $L \in \mathbf{PH}$ is polynomial-time reducible to a tally (unary) language $T_L \in \mathbf{PH}$, which encodes the lexicographically first circuits for L (Lemma 3.9). Hence a Karp–Lipton collapse to $\mathbf{P}^{\mathcal{C}}$ transfers to $\mathbf{OGTD}[\mathcal{C}]$ (Theorem 3.10). Tally languages also make linear orders oblivious: if $T \in \mathbf{L_2^P}$ is tally, its order for 1^n depends only on n, so $T \in \mathbf{OMIN}$, and $\mathbf{P}^T \subseteq \mathbf{OMIN}$ since $\mathbf{P^{OMIN}} = \mathbf{OMIN}$ [HVW26] (Lemma 3.2). Finally, a **prMA** oracle can be replaced by an **NP** oracle together

with a pseudorandom generator computed with an $\mathbf{OL_2^P}$ oracle [Kor22, KP24, HV25] (Lemma 3.4). After this replacement the oracle is a language in $\mathbf{L_2^P}$, so the advice becomes canonical and the tally argument applies (Theorem 3.5). We work with the order-based definition of $\mathbf{L_2^P}$ and prove its closure under polynomial-time Turing reductions directly (Lemma 3.1).

Certified minima (Theorem 1). The minimum μ_n of an oblivious linear order is certified by the **co-ONP** test "no element is smaller". Indeed, μ_n itself is a common refutation of all other candidates; this is also behind $\mathbf{OMIN} \subseteq \mathbf{SOUP^{ONP[1]}}$ [HVW26]. If $\mathbf{NP} \subseteq \mathbf{P/poly}$, every language in **PH** is in **OMIN** by Theorem 2. The bits of μ_n then form a tally language in $\mathbf{PH} = \mathbf{P^{sOprMA}}$. So an $\mathbf{FP^{sOprMA}}$ generator computes μ_n, a **co-ONP** tester certifies it, and the **OMIN** decoder decides. This puts **PH** into **OUM**. The three unconditional containments of **OM** come from simulating the generator: directly, by derandomization inside $\mathbf{L_2^P}$, and by random sampling with an **ONP** oracle. The last one uses that, in **sOprMA**, the set of pairs (x, r) accepted with *some* proof is an **ONP** language.

Uniting the techniques of [CR11] and [HV25] (Theorem 3). Let $V(y, z)$ be a Boolean matrix given by a circuit. A row y beats a column z if $V(y, z) = 1$, and z beats y if $V(y, z) = 0$. A set of rows is collectively irrefutable if every column is beaten by one of its rows, and similarly for columns. Goldreich and Wigderson [GW99] observed that every matrix has a small collectively irrefutable set of rows or of columns; if the matrix has an irrefutable row, the set must consist of rows. Chakaravarthy and Roy [CR11] found such sets in $\mathbf{FP^{prAM}}$ by a prefix search guided by the *best* extension of a prefix, an existential quantity. The minimum-finding algorithm of [HV25] follows approximate *averages* instead, which a **prSBP** oracle provides. An above-average row, however, may beat few columns. We therefore decide in each round whether to select a row or a column, by comparing approximate counts of 0- and 1-entries of the remaining submatrix. On the denser side, an above-average choice beats a quarter of the remaining lines.

Collective certificates (Theorem 4(3)). For **ONP** we use the *domination game*. Its rows are candidate witnesses w, its columns are pairs (x, w'), and w loses only if w' certifies x while w does not. For $\mathbf{O_2^P}$ with verifier $V(x, y, z)$ and oblivious certificates y_n, z_n, we use the *consistency game*. Its rows are pairs (y, z) and its columns are triples (x, z', y'), and (y, z) wins if $V(x, y, z') = 1$ or $V(x, y', z) = 0$. A collectively irrefutable set of pairs yields yes-certificates that collectively beat every no-certificate on every yes-instance. Relative to them, the complement of L becomes an **ONP**-type language, and a second application of the lemma finds its witnesses.

1.5 Related work

Advice of bounded complexity. Karp and Lipton [KL80] defined advice classes $\mathcal{C}/\mathcal{F}$ for an arbitrary class $\mathcal{F}$ of advice functions. Ko and Schöning [KS85] bounded the complexity of advice for sparse sets in the polynomial hierarchy. Kämper's nonuniform proof systems [Käm91] delimit the complexity of *proof sets*, i.e., of sets of correct advice; in our terms, these are the sets accepted by a tester. Köbler and Thierauf [KT94] systematically restricted the complexity of *computing* correct advice. They also wrote $\mathcal{C}//\mathcal{F}$ for advice that depends on the input itself, which is why we avoid this notation. Gavaldà [Gav95] bounded the complexity of advice functions for sets in $\mathbf{P/poly}$. Bshouty, Cleve, Gavaldà, Kannan and Tamon [BCG+96] showed that circuits can be learned exactly with equivalence queries and an **NP** oracle in randomized expected polynomial time. As observed by Köbler and Watanabe [KW98], correct advice for a language $A \in \mathbf{P/poly}$

can then be computed in $\mathbf{FZPP}^{\mathbf{NP}^A}$ as a multi-valued function: different successful runs may output different advice strings. Fortnow, Impagliazzo, Kabanets and Umans [FIKU08] obtained this learning algorithm and $\mathbf{S}_2^{\mathbf{P}} \subseteq \mathbf{ZPP}^{\mathbf{NP}}$ from a $\mathbf{ZPP}^{\mathbf{NP}}$ algorithm for succinct zero-sum games, and showed that approximating the value of such a game within an additive error is complete for promise-$\mathbf{S}_2^{\mathbf{P}}$; by Theorem 3, such approximations can be computed in $\mathbf{FP}^{\mathbf{prSBP}}$ (Appendix B.2). Circuits constructible in a class $\mathcal{C}$, i.e., $\mathcal{C}$-uniform circuits, appear in Kannan's work [Kan82]; Santhanam and Williams [SW14] proved that $\mathbf{NP}$ does not have $\mathbf{P}_{\|}^{\mathbf{NP}}$-uniform circuits of size n^k. Marshall, Gyurik and Dunjko [MGD24] consider a related notion of "bounded advice", in which the advice must be computable within a given complexity class.

Untrusted and verifiable advice. Aaronson [Aar07] introduced $\mathbf{YP} = \mathbf{ONP} \cap \mathbf{co\text{-}ONP}$, and Aaronson and Drucker [AD14] introduced $\mathbf{YP}^*$, in which the advice is tested without the input. Input-oblivious proof systems [FSW09, GM15] can be viewed as restricting $\mathbf{P/poly}$ to verifiable advice. The same-witness class $\mathbf{SOUP}$ of [HVW26] is an unambiguous version of $\mathbf{YP}^*$.

Other recent work. Chen, Hirahara and Ren [CHR24] proved that $\mathbf{S}_2^{\mathbf{E}}$ with one bit of advice requires circuits of size $2^n/n$, and Li [Li24] removed the advice bit (see also [CHLR26]). Ren and Williams [RW26] give near-maximum circuit lower bounds for exponential-time classes with Merlin–Arthur queries. Ghentiyala, Li and Stephens-Davidowitz [GLS26] showed that decision problems reducible to range avoidance lie in $\mathbf{AM} \cap \mathbf{co\text{-}AM}$. Gajulapalli, Ghentiyala, Li and Saraogi [GGLS26] placed the linear ordering principle in $\mathbf{UEOPL}^{\mathbf{NP}}$. Gilboa, Goldberg, Koutsoupias and Nisan [GGKN25] defined syntactic classes of unambiguous $\mathbf{\Sigma}_2^{\mathbf{P}}$ problems and placed them in $\mathbf{S}_2^{\mathbf{P}}$. Chen, McKay, Murray and Williams [CMMW19] proved that fixed-polynomial lower bounds for $\mathbf{P}^{\mathbf{NP}}$ are equivalent to a weak Karp–Lipton theorem. Shaltiel and Umans [SU06] and Aydinlioğlu, Gutfreund, Hitchcock and Kawachi [AGHK11] related the derandomization of approximate counting to hardness. On access to promise oracles, see [ESY84, CHV92, GS88, BF99, CR11, HV25, MPR26]; the term *"loose access"* is from [HS15] (while the first source using this concept was apparently [CHV92]).

1.6 Organization

Section 2 gives definitions, including generator–tester–decoder computations. Section 3 treats order-based $\mathbf{L}_2^{\mathbf{P}}$ and proves the collapse to $\mathbf{OMIN}$ (Theorem 2) and parts (1) and (2) of Theorem 4. Section 4 proves the ultimate collapse (Theorem 1). Section 5 proves the Goldreich–Wigderson lemma with approximate counting and Theorem 3, and Section 6 applies it in the oblivious setting and proves Theorem 4(3). Section 7 lists open problems. Appendix A gives alternative routes to the collapses and discusses relativization, and Appendix B derives several byproducts described at the end of Section 1.3 above.

2 Preliminaries

2.1 Conventions

All polynomials are nondecreasing and at least 1. The letter n denotes the length of the input. In definitions based on polynomial-time predicates, every certificate, proof, random string and advice string for inputs of length n ranges over strings of one fixed length, a polynomial in n given by the definition ($p(n)$, or $a(n)$, $b(n)$, ..., when several lengths occur). Quantifiers such as $\exists y$, $\forall z$ and $\forall r$, and probabilities $\Pr_r$ (always over the uniform distribution), range over strings of exactly this

length. In particular, an input-oblivious certificate y_n has the same length as every other certificate for inputs of length n. Machines run in time polynomial in the length of their input, which is n for a tally input 1^n; oracle machines are deterministic unless stated otherwise.

For a polynomial p and $n \in \mathbb{N}$, strings of length $p(n)$ are identified with tuples of strings of prescribed lengths whenever these lengths depend only on n. Pairs are encoded in a length-regular way: the length of (x, w) determines $|x|$ and $|w|$. A language T is *tally* if $T \subseteq \{1\}^*$; **TALLY** denotes the class of tally languages. We use the Cantor pairing $\pi(n, i) = \binom{n+i+1}{2} + i$, a bijection $\mathbb{N}^2 \to \mathbb{N}$ such that π and π^{-1} are computable in time polynomial in the unary lengths and $\pi(n, i) \leq (n + i + 1)^2$. We fix an encoding of circuits with n inputs and at most s gates by strings of length exactly $\ell(n, s) = O(s \log(n + s))$. Whether a string encodes such a circuit, and the value of the circuit on a given input, are decidable in polynomial time. For a circuit C, $\#C$ denotes the number of its satisfying assignments.

2.2 Promise problems and oracle access

A *promise problem* [ESY84] is a pair $\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ of disjoint sets of strings; a language L is identified with $(L, \overline{L})$. A language O is *consistent* with Π if $\Pi_{\mathrm{YES}} \subseteq O$ and $\Pi_{\mathrm{NO}} \cap O = \emptyset$.

Oracle access to a promise problem Π is *loose*: queries in $\Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$ are answered correctly, every other query receives an arbitrary answer, and the computation must be correct for all possible answers. These answers need not be consistent: the same query may receive different answers when it is asked again, and, for a nondeterministic machine, on different computation paths. This mode of access was introduced by Cai, Hemachandra and Vyskoč [CHV92, Definition 5.8] as *arbitrary failure access*, and it is the mode used by Chakaravarthy and Roy [CR11]; the term *loose* is from [HS15]. For deterministic polynomial-time oracle machines, loose access is equivalent to access through an arbitrary language consistent with Π, as in Buhrman and Fortnow [BF99]. Indeed, such a machine can store the answers it has received and never repeat a query, and then every sequence of answers is given by some consistent language. We use the latter form throughout, and we assume that deterministic oracle machines never repeat a query.

Definition 2.1. Let Π be a promise problem. A language L is in $\mathbf{P}^{\Pi}$ if there is a deterministic polynomial-time oracle machine M such that M^O decides L for every language O consistent with Π. A search problem (relation) R is in $\mathbf{FP}^{\Pi}$ if there is a deterministic polynomial-time oracle machine M such that, for every O consistent with Π and every input x, $M^O(x)$ is a solution for x. For a class $\mathcal{C}$ of promise problems, $\mathbf{P}^{\mathcal{C}} = \bigcup_{\Pi \in \mathcal{C}} \mathbf{P}^{\Pi}$, and similarly for $\mathbf{FP}^{\mathcal{C}}$.

For $\Pi_1, \ldots, \Pi_r$, let $\Pi_1 \oplus \cdots \oplus \Pi_r$ be the promise problem with yes-instances $\langle i, q \rangle$ for $q \in (\Pi_i)_{\mathrm{YES}}$ and no-instances $\langle i, q \rangle$ for $q \in (\Pi_i)_{\mathrm{NO}}$, where i is written with a fixed number of bits. A language consistent with it yields, by restriction, languages consistent with each Π_i.

2.3 Complexity classes

Definition 2.2 ($\mathbf{prS_2^P}$, $\mathbf{S_2^P}$ [Can96, RS98]; $\mathbf{prO_2^P}$, $\mathbf{O_2^P}$ [CR06])**.** A promise problem Π is in $\mathbf{prS_2^P}$ if there are polynomials a, b and a polynomial-time predicate V such that for every x,

$$x \in \Pi_{\mathrm{YES}} \Rightarrow \exists y \in \{0,1\}^{a(|x|)}\ \forall z \in \{0,1\}^{b(|x|)}\ \ V(x, y, z) = 1,$$
$$x \in \Pi_{\mathrm{NO}} \Rightarrow \exists z \in \{0,1\}^{b(|x|)}\ \forall y \in \{0,1\}^{a(|x|)}\ \ V(x, y, z) = 0.$$

Such y and z are called *irrefutable certificates*. $\Pi \in \mathbf{prO_2^P}$ if, moreover, for every n there are y_n and z_n that are irrefutable for all $x \in \Pi_{\mathrm{YES}} \cap \{0,1\}^n$ and all $x \in \Pi_{\mathrm{NO}} \cap \{0,1\}^n$, respectively. $\mathbf{S_2^P}$ and $\mathbf{O_2^P}$ are the corresponding classes of languages.

Definition 2.3 (**ONP** [FSW09, CR06])**.** A language L is in **ONP** if there are a polynomial p, a polynomial-time predicate V, and strings $w_n \in \{0,1\}^{p(n)}$ such that, for every $x \in \{0,1\}^n$, $x \in L$ implies $V(x, w_n) = 1$, and $x \notin L$ implies $V(x, w) = 0$ for all w. We call w_n an *oblivious witness*.

Definition 2.4 (**prMA**, **prOMA** [FSW09, HV25]; **OprMA** [HVW26]; **sOprMA**)**.** $\Pi \in$ **prMA** if there are a polynomial s and a polynomial-time predicate V such that for every x, with y, r ranging over $\{0,1\}^{s(|x|)}$,

$$x \in \Pi_{\mathrm{YES}} \Rightarrow \exists y\ \forall r \quad V(x,y,r) = 1,$$
$$x \in \Pi_{\mathrm{NO}} \Rightarrow \forall y \quad \Pr_r[V(x,y,r) = 1] < 1/4.$$

For the input-oblivious versions, there are, in addition, strings $y_n \in \{0,1\}^{s(n)}$, the *oblivious proofs*, and we consider the following conditions on $x \in \{0,1\}^n$, with y, r ranging over $\{0,1\}^{s(n)}$:

$$x \in \Pi_{\mathrm{YES}} \ \Rightarrow\ \forall r \quad V(x, y_n, r) = 1, \tag{OMA1}$$
$$x \in \Pi_{\mathrm{NO}} \ \Rightarrow\ \forall y \quad \Pr_r[V(x,y,r) = 1] < 1/4, \tag{OMA2}$$
$$x \notin \Pi_{\mathrm{NO}} \ \Rightarrow\ \forall y\ \forall r \quad (V(x,y,r) = 1 \Rightarrow V(x, y_n, r) = 1), \tag{OMA3}$$
$$\forall y\ \forall r \quad (V(x,y,r) = 1 \Rightarrow V(x, y_n, r) = 1). \tag{OMA4}$$

$\Pi \in$ **prOMA** if (OMA1) and (OMA2) hold for every n and every $x \in \{0,1\}^n$. $\Pi \in$ **OprMA** if, in addition, (OMA3) holds. $\Pi \in$ **sOprMA** (strict **OprMA**) if (OMA1), (OMA2) and (OMA4) hold.

Perfect completeness and soundness $1/4$ are without loss of generality for **prMA** [ZF87]. Since (OMA4) implies (OMA3), **sOprMA** $\subseteq$ **OprMA** $\subseteq$ **prOMA** $\subseteq$ **prMA**. Condition (OMA3) says that the oblivious proof y_n dominates every other proof on every input outside Π_{NO} and every random string; (OMA4) requires this on every input. Every **ONP** language L is a (total) problem in **sOprMA**: take $V(x,y,r) = V'(x,y)$ for an **ONP** verifier V' of L.

Definition 2.5 (**SOUP** [HVW26])**.** A language L is in $\mathbf{SOUP}^A$ if there are a polynomial p, polynomial-time oracle machines Y, Z, and strings $w_n \in \{0,1\}^{p(n)}$ such that, for every $x \in \{0,1\}^n$: $Y^A(x,u) = Z^A(x,u) = 0$ for every $u \neq w_n$; if $x \in L$, then $Y^A(x, w_n) = 1$ and $Z^A(x,u) = 0$ for all u; and if $x \notin L$, then $Z^A(x, w_n) = 1$ and $Y^A(x,u) = 0$ for all u. In $\mathbf{SOUP}^{\mathcal{C}[1]}$, the machines make one query to an oracle from $\mathcal{C}$.

Approximate counting. The promise problem SSE (set size estimation) has yes-instances (C, m) with $\#C \geq m$ and no-instances (C, m) with $\#C \leq m/2$, where C is a circuit and $m \geq 1$ is an integer given in binary. It is complete for the class **prSBP** of Böhler, Glaßer and Meister [BGM06] (see [HV25, Lemma 3.2]), and we write $\mathbf{P}^{\mathbf{prSBP}} = \mathbf{P}^{\mathrm{SSE}}$ and $\mathbf{FP}^{\mathbf{prSBP}} = \mathbf{FP}^{\mathrm{SSE}}$. Since **prSBP** $\subseteq$ **prAM** [BGM06], $\mathbf{P}^{\mathbf{prSBP}} \subseteq \mathbf{P}^{\mathbf{prAM}}$. Note also that SAT $\in \mathbf{P}^{\mathrm{SSE}}$, via the query $(C, 1)$.

Lemma 2.6 ([HV25, Lemma 3.4])**.** *Given a circuit C and a rational $\varepsilon \in (0,1]$, one can compute an integer t with $\#C \leq t \leq (1+\varepsilon)\#C$ in time polynomial in $|C|$ and $1/\varepsilon$, with any oracle consistent with* SSE*. In particular, $t = 0$ if and only if $\#C = 0$.*

Linear orders. Throughout this paper, $\mathbf{L}_2^{\mathbf{P}}$ is the class defined by polynomial-time decidable linear orders.

Definition 2.7 ($\mathbf{L_2^P}$ [KP24])**.** A language L is in $\mathbf{L_2^P}$ if there are a polynomial p and a polynomial-time predicate V such that, for every x, the relation $u \prec_x w \Leftrightarrow V(x,u,w) = 1$ is a strict linear order on $\{0,1\}^{p(|x|)}$ whose minimum starts with 1 if $x \in L$ and with 0 if $x \notin L$.

Remark 2.8. Korten and Pitassi [KP24] also consider the class of languages that are polynomial-time (or $\mathbf{P^{NP}}$-) reducible to the total search problem LOP. Given a circuit E on pairs of strings, LOP asks for the minimum of the order defined by E or for a violation of the axioms of strict linear orders. They state that the two definitions are equivalent [KP24, Theorems 5 and 6]. To keep the paper self-contained, we use only Definition 2.7 and do not rely on this equivalence anywhere. In particular, we prove closure under polynomial-time Turing reductions for Definition 2.7 directly (Lemma 3.1) rather than deriving it from the reduction-based definition, and we derive the pseudorandom generators of (F3) from it (Lemma 3.4). Both definitions give subclasses of $\mathbf{S_2^P}$, and hence of $\mathbf{P^{prSBP}}$ by Theorem 3. For Definition 2.7, this is immediate. For reductions to LOP, note that LOP has an $\mathbf{FS_2^P}$ algorithm that determines a fixed solution for each instance [KP24, Lemma 5]. The bits of this solution form a language in $\mathbf{S_2^P}$, and a reduction that works for every solution of LOP works for this one. Hence these classes lie in $\mathbf{P^{NP,S_2^P}} = \mathbf{P^{S_2^P}} = \mathbf{S_2^P}$ (see [CR11] for the last equality).

Definition 2.9 ($\mathbf{OL_2^P}$ [HV25]; **OMIN** [HVW26])**.** $L \in \mathbf{OL_2^P}$ if $L \in \mathbf{L_2^P}$ via some V, p such that for every n there are strings y_n, z_n with $\min \prec_x = 1y_n$ for all $x \in L \cap \{0,1\}^n$ and $\min \prec_x = 0z_n$ for all $x \in \{0,1\}^n \setminus L$. $L \in \mathbf{OMIN}$ if there are a polynomial p, a predicate $U(1^n, u, w)$ computable in time polynomial in n such that $u <_n w \Leftrightarrow U(1^n,u,w) = 1$ is a strict linear order on $\{0,1\}^{p(n)}$ for every n, and a polynomial-time decoder D such that $L(x) = D(x, \mu_{|x|})$ for every x, where $\mu_n = \min <_n$.

We use the following known facts.

(F1) $\mathbf{NP} \subseteq \mathbf{L_2^P}$ [KP24], with an explicit linear order for every **NP** verifier [HVW26, Lemma 2.11].

(F2) $\mathbf{OL_2^P} = \mathbf{OMIN}$ [HVW26, Theorem 3.4], and $\mathbf{OL_2^P} \not\subseteq \mathbf{SIZE}[n^k]$ for every k [HVW26, Theorem 3.3]. Clearly $\mathbf{OL_2^P} \subseteq \mathbf{L_2^P}$. Moreover, $\mathbf{OMIN} \subseteq \mathbf{O_2^P}$. If $L \in \mathbf{OMIN}$ via $(p, <_n, D)$, the verifier $V(x,y,z) = D(x, \min_{<_n}\{y,z\})$ has the irrefutable certificates $y_n = z_n = \mu_n$.

(F3) (pseudorandom generators from oblivious orders) There are a language $B \in \mathbf{OL_2^P}$ and a deterministic polynomial-time oracle algorithm that, given 1^N, outputs with oracle B a list $g_1, \ldots, g_m \in \{0,1\}^N$ with $m = \mathrm{poly}(N)$ such that every N-input circuit C of size at most N satisfies $\left|\Pr_i[C(g_i) = 1] - \Pr_{y \in \{0,1\}^N}[C(y) = 1]\right| \leq 1/N$. This was observed in [HV25, Proposition 5.5] and [HVW26, Lemma 6.2], whose proofs pass through the reduction-based definition of $\mathbf{L_2^P}$ (Remark 2.8). Lemma 3.4 derives (F3) from Definition 2.7 alone.

(F4) $\mathbf{P^{prO_2^P}} = \mathbf{O_2^P}$ [HV25, Theorem 4.3], $\mathbf{P^{prOMA}} \subseteq \mathbf{O_2^P}$ [HV25, Corollary 4.4], and, if $\mathbf{NP} \subseteq \mathbf{P/poly}$, then $\mathbf{PH} = \mathbf{P^{prOMA}}$ [HV25, Proposition 4.1]. Moreover, $\mathbf{O_2^P} \not\subseteq \mathbf{SIZE}[n^k]$ for every k [GLV24].

All Karp–Lipton collapse classes considered in this paper, as well as $\mathbf{P^{prSBP}}$ and $\mathbf{P^{prAM}}$, are contained in **PH**. In particular, $\mathbf{OMIN} \subseteq \mathbf{\Sigma_2^P} \cap \mathbf{\Pi_2^P}$: for $x \in \{0,1\}^n$, $x \in L$ iff $\exists m\,[\forall m'\ \neg(m' <_n m) \wedge D(x,m) = 1]$, and similarly for $x \notin L$. Moreover, $\mathbf{P^{prAM}} \subseteq \mathbf{P^{\Sigma_2^P}}$. Indeed, a promise problem with an Arthur–Merlin protocol with perfect completeness [FGM+89] is consistent with the $\mathbf{\Pi_2^P}$ language of inputs x such that, for every random string, Merlin has an accepting reply.

2.4 Generator–tester–decoder computations

Definition 2.10 (OGTD). Let $\mathcal{A}$ be a class of generators, $\mathcal{B}$ a class of languages (testers), and $\mathcal{C}$ a class of languages (decoders). The generators we use are deterministic polynomial-time oracle transducers G with an oracle Π from a fixed class $\mathcal{X}$ of promise problems; this class of generators is denoted $\mathbf{FP}^{\mathcal{X}}$. Thus $\mathcal{A}$ is a class of machines together with their oracles (in our uses, $\mathbf{FP}^{\mathcal{X}}$ or the nonuniform class poly defined below), whereas $\mathcal{B}$ and $\mathcal{C}$ are classes of languages (in our uses, **co-ONP**, **P** or $*$, and **P**, respectively). On input 1^n the output of G has a fixed length $p(n)$ for a polynomial p. Its set of *legal outputs* is

$$\mathrm{Out}_G(n) = \{G^O(1^n) : O \text{ is a language consistent with } \Pi\}.$$

A language L belongs to $\mathbf{OGTD}[\mathcal{A}, \mathcal{B}, \mathcal{C}]$ if there are a generator $G \in \mathcal{A}$, a tester $T \in \mathcal{B}$ (a language of pairs $(1^n, a)$), and a decoder $D \in \mathcal{C}$ (a language of pairs (x, a)) with the following properties. Informally, for inputs of length n, the generator computes an advice string from 1^n alone; every advice string that the generator can output passes the test, which does not see the input either; and every advice string that passes the test lets the decoder decide L correctly on all inputs of length n. Formally, let

$$\mathrm{Acc}_T(n) = \{a : (1^n, a) \in T\} \quad \text{and} \quad \mathrm{Good}_D(n) = \{a \in \{0,1\}^{p(n)} : D(x, a) = L(x) \text{ for all } x \in \{0,1\}^n\}$$

be the sets of *accepted* and of *good* advice strings, where we identify languages with their characteristic functions. Then we require, for every n,

$$\mathrm{Out}_G(n) \subseteq \mathrm{Acc}_T(n) \subseteq \mathrm{Good}_D(n). \tag{1}$$

L belongs to $\mathbf{OUGTD}[\mathcal{A}, \mathcal{B}, \mathcal{C}]$ if, moreover, $\mathrm{Acc}_T(n)$ has exactly one element for every n. We write $*$ for the class of all languages. In $\mathbf{OGTD}[\mathcal{A}, *, \mathcal{C}]$ one can take $\mathrm{Acc}_T(n) = \mathrm{Good}_D(n)$, so only $\mathrm{Out}_G(n) \subseteq \mathrm{Good}_D(n)$ is required. Finally, we abbreviate

$$\mathbf{OGTD}[\mathcal{X}] = \mathbf{OGTD}[\mathbf{FP}^{\mathcal{X}}, *, \mathbf{P}];$$

here the argument $\mathcal{X}$ is the class of oracles of the generator. We also use the nonuniform generator class poly of all polynomially length bounded functions $n \mapsto a_n$, whose only legal output on 1^n is a_n.

For example, $\mathbf{P/poly} = \mathbf{OGTD}[\mathrm{poly}, *, \mathbf{P}]$, and $\mathbf{YP}^* = \mathbf{OGTD}[\mathrm{poly}, \mathbf{P}, \mathbf{P}]$ by the definition of Aaronson and Drucker [AD14, Definition 18]: there the tester checks the advice without access to the input. A language L is in $\mathbf{OGTD}[\Pi]$ if and only if L has polynomial-size circuits that can be constructed from 1^n in $\mathbf{FP}^{\Pi}$. Indeed, $D(\cdot, a)$ can be written as a circuit, and conversely a circuit can be evaluated. Since a generator is a deterministic polynomial-time machine, it makes no difference whether its access to Π is loose or through consistent languages. Our main class is

$$\mathbf{OM} = \mathbf{OGTD}[\mathbf{FP}^{\mathbf{sOprMA}}, \mathbf{co\text{-}ONP}, \mathbf{P}], \qquad \mathbf{OUM} = \mathbf{OUGTD}[\mathbf{FP}^{\mathbf{sOprMA}}, \mathbf{co\text{-}ONP}, \mathbf{P}].$$

Proposition 2.11. *Let $\Pi, \Pi', \Pi_1, \ldots, \Pi_r$ be promise problems.*

(a) $\mathbf{OGTD}[\Pi] \subseteq \mathbf{P}^{\Pi} \cap \mathbf{P/poly}$*, and* $\mathbf{OGTD}[\Pi]$ *is closed under complement.*

(b) *For a language A, we have* $\mathbf{OGTD}[A] = \mathbf{P}^{\mathbf{TALLY} \cap \mathbf{P}^A}$.

(c) *If $L \in \mathbf{P}^{L_1, \ldots, L_r}$ and $L_i \in \mathbf{OGTD}[\Pi_i]$ for all i, then $L \in \mathbf{OGTD}[\Pi_1 \oplus \cdots \oplus \Pi_r]$. In particular,* $\mathbf{P}^{\mathbf{OGTD}[\Pi]} = \mathbf{OGTD}[\Pi]$.

(d) *Suppose that there is a deterministic polynomial-time oracle machine M such that for every language O consistent with Π, the language decided by M^O is consistent with Π'. Then* $\mathbf{OGTD}[\Pi'] \subseteq \mathbf{OGTD}[\Pi]$, $\mathbf{P}^{\Pi'} \subseteq \mathbf{P}^{\Pi}$, *and* $\mathbf{FP}^{\Pi'} \subseteq \mathbf{FP}^{\Pi}$.

Part (b) is folklore (see, e.g., [KT94]), but we include its accurate proof for the sake of self-containedness.

Proof. (a) Compute the advice and run D; for $\mathbf{P}/\mathbf{poly}$, fix any consistent O and hardwire $G^O(1^n)$. Negating D gives closure under complement.

(b) "$\subseteq$": Let $L \in \mathbf{OGTD}[A]$ via (G, D), with $|G^A(1^n)| = p(n)$. Let $T = \{1^{\pi(n,i)} : 1 \le i \le p(n),\ \text{bit } i \text{ of } G^A(1^n) \text{ is } 1\}$. On input 1^N, a polynomial-time machine computes $(n, i) = \pi^{-1}(N)$, checks $1 \le i \le p(n)$, and runs $G^A(1^n)$; since $n \le N$, $T \in \mathbf{P}^A$. On input x of length n, a machine with oracle T queries $1^{\pi(n,i)}$ for $i \le p(n)$, which have polynomial lengths, and runs D. Hence $L \in \mathbf{P}^T$. "$\supseteq$": Let $T \in \mathbf{TALLY} \cap \mathbf{P}^A$ and $L \in \mathbf{P}^T$ via a machine N with running time q. Let $G^A(1^n)$ be the string $T(1^0)T(1^1)\cdots T(1^{q(n)})$, computed with the $\mathbf{P}^A$ machine for T, and let $D(x, s)$ simulate N on x, answering a query w by the $(|w|+1)$-st bit of s if $w = 1^{|w|}$, and by 0 otherwise.

(c) Let M decide L with oracles $L_1, \ldots, L_r$ in time $t(n)$, and let (G_i, D_i) witness $L_i \in \mathbf{OGTD}[\Pi_i]$. Given an oracle O consistent with $\Pi_1 \oplus \cdots \oplus \Pi_r$, its restrictions O_i are consistent with Π_i. The new advice for length n is the tuple of all $a_{i,m} = G_i^{O_i}(1^m)$ for $i \le r$ and $m \le t(n)$. The new decoder simulates M and answers a query q to L_i by $D_i(q, a_{i,|q|})$. For $\mathbf{P}^{\mathbf{OGTD}[\Pi]} = \mathbf{OGTD}[\Pi]$, take $r = 1$; the containment "$\supseteq$" is trivial.

(d) Replace every query of the generator for Π' by a simulation of M^O; the simulated answers form a language consistent with Π'. The same works for $\mathbf{P}^{\Pi'}$ and $\mathbf{FP}^{\Pi'}$. □

By part (d), a **prSBP**-complete problem can replace any promise problem in **prSBP**; thus $\mathbf{OGTD}[\mathrm{SSE}] = \bigcup_{\Pi \in \mathbf{prSBP}} \mathbf{OGTD}[\Pi]$, which we denote $\mathbf{OGTD}[\mathbf{prSBP}]$.

3 Linear orders and the collapse to OMIN

3.1 Closure under polynomial-time Turing reductions

The following lemma is the non-oblivious analogue of $\mathbf{P}^{\mathbf{OMIN}} = \mathbf{OMIN}$ [HVW26, Theorem 4.1]. For a reduction-based definition of $\mathbf{L}_2^{\mathbf{P}}$ it would be immediate; for Definition 2.7 we prove it for the sake of self-containedness.

Lemma 3.1. $\mathbf{L}_2^{\mathbf{P}}$ *is closed under polynomial-time Turing reductions: if* $A_1, \ldots, A_r \in \mathbf{L}_2^{\mathbf{P}}$, *then* $\mathbf{P}^{A_1,\ldots,A_r} \subseteq \mathbf{L}_2^{\mathbf{P}}$.

Proof. Reduction to one oracle. Let $A = \{\langle i, q\rangle : 1 \le i \le r,\ q \in A_i\}$. Then $A \in \mathbf{L}_2^{\mathbf{P}}$. On input $\langle i, q\rangle$ use the order of A_i on q, padded to a common length, and on all other inputs use the lexicographic order, whose minimum $0\cdots0$ starts with 0. Since $\mathbf{P}^{A_1,\ldots,A_r} \subseteq \mathbf{P}^A$, it suffices to treat a single oracle $A \in \mathbf{L}_2^{\mathbf{P}}$, given by V_A and a polynomial p_A.

Setup. Let $L \in \mathbf{P}^A$ via a deterministic oracle machine M whose running time is bounded by a polynomial t. Fix an input x of length n, and put $K = t(n)$ and $P = p_A(t(n))$. For a query q with $|q| \le t(n)$, let $\sqsubset_q$ be the strict linear order on $\{0,1\}^P$ that compares the first $p_A(|q|)$ bits by $\prec_q$ and, in case of equality, the remaining bits lexicographically. Its minimum starts with $A(q)$. Let $\sqsubset_\perp$ be the lexicographic order on $\{0,1\}^P$. For $c = (c_1, \ldots, c_K) \in (\{0,1\}^P)^K$, run M on x, answering the j-th query by the first bit of c_j. Let $q_j(c)$ be the j-th query of this run, or $\perp$ if there are fewer than j queries, and let $\mathrm{out}(c)$ be the output. Then $q_j(c)$ depends only on $c_1, \ldots, c_{j-1}$.

The order. For $c \neq c'$, let j be the least index with $c_j \neq c'_j$. Then $q_j(c) = q_j(c') =: q$, and we define the order $\sqsubset$ on tuples by

$$c \sqsubset c' \quad \iff \quad c_j \sqsubset_q c'_j.$$

This relation is irreflexive, and any two distinct elements are comparable. For transitivity, let $c \sqsubset c' \sqsubset c''$, with first differences at j (for c, c') and at j' (for c', c''). If $j < j'$, then c and c'' first differ at j, with $c''_j = c'_j$. If $j > j'$, then they first differ at j', with $c_{j'} = c'_{j'}$. If $j = j'$, then $c_j \sqsubset_q c'_j \sqsubset_q c''_j$, so they first differ at j. In each case the order used at that index is the same, because it is determined by the common prefix. Hence $c \sqsubset c''$, and $\sqsubset$ is a strict linear order.

The minimum. Define c^* inductively by $c^*_j = \min_{\sqsubset_{q_j(c^*)}}$; this is well defined because $q_j(c^*)$ depends only on $c^*_1, \ldots, c^*_{j-1}$. Then c^* is the minimum of $\sqsubset$. Indeed, if $c \neq c^*$ first differs from c^* at j, then $c^*_j \sqsubset_q c_j$ for $q = q_j(c) = q_j(c^*)$. By induction on j, the first bit of c^*_j equals $A(q_j(c^*))$ whenever $q_j(c^*) \neq \perp$. So the run determined by c^* is the true run of M^A on x, and $\mathrm{out}(c^*) = L(x)$.

The answer bit. On pairs $(b, c) \in \{0,1\} \times (\{0,1\}^P)^K$, written as strings starting with b, define

$$(b, c) \sqsubset' (b', c') \quad \iff \quad c \sqsubset c', \quad \text{or} \quad c = c' \quad \text{and} \quad b = \mathrm{out}(c) \neq b'.$$

This is a strict linear order, the pullback of the lexicographic product of $\sqsubset$ and the order $0 < 1$ under the injection $(b, c) \mapsto (c, b \oplus \mathrm{out}(c))$. Its minimum is $(L(x), c^*)$, which starts with $L(x)$. It can be decided in time polynomial in n: find the first differing index, simulate M to obtain the query there, and evaluate V_A. Hence $L \in \mathbf{L}^{\mathbf{P}}_{\mathbf{2}}$, with strings of length $1 + t(n)\, p_A(t(n))$. □

3.2 Tally languages make linear orders oblivious

Lemma 3.2. *If T is a tally language in $\mathbf{L}^{\mathbf{P}}_{\mathbf{2}}$, then $\mathbf{P}^T \subseteq \mathbf{OMIN}$.*

Proof. Let verifier V and polynomial p define $T \in \mathbf{L}^{\mathbf{P}}_{\mathbf{2}}$. The only input of length n that may be in a tally language is 1^n, so $T \in \mathbf{OMIN}$ via the orders $\prec_{1^n}$, which depend only on n, and the decoder that outputs the first bit of the minimum on input 1^n and 0 on all other inputs. Hence $\mathbf{P}^T \subseteq \mathbf{P}^{\mathbf{OMIN}} = \mathbf{OMIN}$ [HVW26, Theorem 4.1]. □

Theorem 3.3 (part (1) of Theorem 4)**.** $(\mathbf{OL}^{\mathbf{P}}_{\mathbf{2}} =)\ \mathbf{OMIN} = \mathbf{OGTD}[\mathbf{L}^{\mathbf{P}}_{\mathbf{2}}] = \mathbf{P}^{\mathbf{TALLY} \cap \mathbf{L}^{\mathbf{P}}_{\mathbf{2}}}$.

Proof. By Proposition 2.11(b) and Lemma 3.1, $\mathbf{OGTD}[\mathbf{L}^{\mathbf{P}}_{\mathbf{2}}] = \bigcup_{A \in \mathbf{L}^{\mathbf{P}}_{\mathbf{2}}} \mathbf{P}^{\mathbf{TALLY} \cap \mathbf{P}^A} = \mathbf{P}^{\mathbf{TALLY} \cap \mathbf{L}^{\mathbf{P}}_{\mathbf{2}}}$, and $\mathbf{P}^{\mathbf{TALLY} \cap \mathbf{L}^{\mathbf{P}}_{\mathbf{2}}} \subseteq \mathbf{OMIN}$ by Lemma 3.2. It remains to show $\mathbf{OMIN} \subseteq \mathbf{OGTD}[\mathbf{L}^{\mathbf{P}}_{\mathbf{2}}]$. Let $L \in \mathbf{OMIN}$ be defined by a polynomial p, decoder D, and order $<_n$ with oblivious minima μ_n, and let

$$T = \{1^{\pi(n,i)} : 1 \leq i \leq p(n),\ \text{bit } i \text{ of } \mu_n \text{ is } 1\}.$$

Clearly $L \in \mathbf{OGTD}[T]$: the advice for length n is μ_n, computed with the queries $1^{\pi(n,i)}$, and the decoder is D. We show $T \in \mathbf{L}^{\mathbf{P}}_{\mathbf{2}}$. Let y be an input of length N. If $y = 1^N$ and $\pi^{-1}(N) = (n, i)$ with $1 \leq i \leq p(n)$, the elements are pairs (b, u) with $b \in \{0,1\}$ and $u \in \{0,1\}^{p(N)}$. Note that $p(n) \leq p(N)$, since $n \leq N$. Pairs are compared first by u, comparing the first $p(n)$ bits by $<_n$ and the remaining bits lexicographically. If $u = u'$, then (b, u) precedes (b', u) iff $b = u[i] \neq b'$. This is a strict linear order, and its minimum is $(\mu_n[i], \mu_n 0 \cdots 0)$, which starts with $T(y)$. For all other inputs y, use the lexicographic order on $\{0,1\}^{1+p(N)}$, whose minimum starts with $0 = T(y)$. The order is decidable in time polynomial in N. Finally, $\mathbf{OL}^{\mathbf{P}}_{\mathbf{2}} = \mathbf{OMIN}$ by (F2). □

For the sake of self-containedness, we provide a direct proof of (F3) from Definition 2.7 (without the equivalence discussed in Remark 2.8).

Lemma 3.4 ((F3) from order-based $\mathbf{L_2^P}$)**.** *Fact (F3) holds for* $\mathbf{L_2^P}$ *as in Definition 2.7.*

Proof. Korten [Kor22] gives a deterministic polynomial-time algorithm that computes from 1^N a circuit C_N with more outputs than inputs, and that converts every string outside the range of C_N into a list $g_1, \dots, g_m$ as in (F3) (see [HV25, Proposition 5.5]). By Korten's stretch amplification [Kor22], we may assume that C_N has exactly twice as many outputs as inputs: a string outside the range of the amplified circuit is converted into a string outside the range of C_N in polynomial time with an **NP** oracle. For every circuit C with twice as many outputs as inputs, Korten and Pitassi [KP24, Theorem 1] (see [HVW26, Section 3]) give a strict linear order $<_C$, decidable in polynomial time, and a polynomial-time algorithm that converts $\min <_C$ into a string outside the range of C. Let A be the language of pairs (C, i) such that bit i of $\min <_C$ is 1. Then $A \in \mathbf{L_2^P}$: on input (C, i) with $1 \le i \le m$, where $\{0,1\}^m$ is the domain of $<_C$, compare the first m bits of two strings by $<_C$ after moving the first bit to position i, and the remaining bits lexicographically; on all other inputs use the lexicographic order. Hence the list for 1^N is computed in deterministic polynomial time with the oracles A and SAT, and the tally language $T = \{1^{\pi(N,j)} : \text{bit } j \text{ of the list for } 1^N \text{ is } 1\}$ is in $\mathbf{P}^{A,\mathrm{SAT}}$. By (F1) and Lemma 3.1, $T \in \mathbf{L_2^P}$, and by Lemma 3.2, $T \in \mathbf{P}^T \subseteq \mathbf{OMIN} = \mathbf{OL_2^P}$. Thus $B = T$ satisfies (F3). □

3.3 Replacing Merlin–Arthur oracles by oblivious linear orders

Theorem 3.5 is the input-oblivious counterpart of $\mathbf{P^{prMA}} \subseteq \mathbf{L_2^P}$ [HV25]; it does not follow from the stronger containment $\mathbf{P^{prMA}} \subseteq \mathbf{P^{OL_2^P,NP}}$ [HV25, Corollary 5.6] (see also [HVW26, Proposition 6.9]), because $\mathbf{NP} \subseteq \mathbf{OL_2^P}$ is equivalent to $\mathbf{NP} \subseteq \mathbf{P/poly}$ (Corollary 3.13), but it follows easily from the proof of the latter and Theorem 3.3.

Theorem 3.5 (part (2) of Theorem 4)**.** $\mathbf{OGTD[prMA]} \subseteq \mathbf{OL_2^P}$.

Proof. Let polynomial s and verifier V define $\Pi \in \mathbf{prMA}$, with perfect completeness. Let $B \in \mathbf{OL_2^P}$ be the language from (F3). We describe a deterministic polynomial-time oracle machine M with oracle $B \oplus \mathrm{SAT}$ such that the language O' decided by $M^{B \oplus \mathrm{SAT}}$ is consistent with Π. On a query q of length ℓ, let $S(\ell)$ be a polynomial bound on the size of the circuits $r \mapsto V(q, w, r)$ for $|q| = \ell$ and $|w| = s(\ell)$, and put $N = \max(S(\ell), s(\ell), 4)$. The machine M computes the list $g_1, \dots, g_m \in \{0,1\}^N$ of (F3) with oracle B. It lets g_i' be the $s(\ell)$-bit prefix of g_i, and accepts iff

$$\exists w \in \{0,1\}^{s(\ell)} \quad \forall i \le m \quad V(q, w, g_i') = 1,$$

which it decides with one SAT query. If $q \in \Pi_{\mathrm{YES}}$, the witness w from perfect completeness passes all tests, so $q \in O'$. If $q \in \Pi_{\mathrm{NO}}$, then for every w, the circuit that computes $V(q, w, \cdot)$ on the first $s(\ell)$ of its N inputs has size at most N. Hence $\Pr_i[V(q, w, g_i') = 1] < 1/4 + 1/N < 1$, so some test fails and $q \notin O'$.

By Proposition 2.11(d), applied to the language $B \oplus \mathrm{SAT}$ (which is consistent only with itself), $\mathbf{OGTD}[\Pi] \subseteq \mathbf{OGTD}[B \oplus \mathrm{SAT}]$. Now $B \oplus \mathrm{SAT} \in \mathbf{L_2^P}$ by (F1), (F2), and Lemma 3.1. Hence $\mathbf{OGTD}[\Pi] \subseteq \mathbf{OGTD}[\mathbf{L_2^P}] = \mathbf{OL_2^P}$ by Theorem 3.3. □

Remark 3.6 (input-oblivious oracles). The class $\mathbf{OGTD}[\mathcal{X}]$ is monotone in $\mathcal{X}$; in particular, $\mathbf{OGTD[prOMA]} \subseteq \mathbf{OGTD[prMA]}$. For generators that make non-adaptive queries the two classes coincide. Indeed, if the i-th query $q_i(n)$ is computable from 1^n, the generator can ask instead the tally instance $1^{\pi(n,i)}$, which is a yes-instance iff $q_i(n) \in \Pi_{\mathrm{YES}}$ and a no-instance iff $q_i(n) \in \Pi_{\mathrm{NO}}$; and a promise problem with at most one instance of each length is in **prOMA** as soon as it is in **prMA**. We do not know whether $\mathbf{OGTD[prOMA]} = \mathbf{OGTD[prMA]}$ in general. If $\mathbf{NP} \subseteq \mathbf{P/poly}$, both classes equal **PH** (Corollary 3.11, and Theorem 3.10 with (F4)).

Remark 3.7. The same machine shows $\mathbf{P}^{\mathbf{prMA}} \subseteq \mathbf{P}^{B\oplus\mathrm{SAT}} \subseteq \mathbf{L}_2^{\mathbf{P}}$ for Definition 2.7; this is the argument of [HV25, Theorem 5.4 and Corollary 5.6], with Lemma 3.1 supplying the closure step. The point of Theorem 3.5 is that after the replacement, the oracle is a language. Therefore the advice computed from 1^n is unique, and the tally argument of Theorem 3.3 applies.

3.4 The collapse to OMIN

Lemma 3.8 (folklore, cf. [KL80])**.** *If* $\mathbf{NP} \subseteq \mathbf{P/poly}$*, then* $\mathbf{PH} \subseteq \mathbf{P/poly}$*.*

Proof. We show $\mathbf{\Sigma_k^P} \subseteq \mathbf{P/poly}$ by induction on k; the case $k = 1$ is the hypothesis. Let $L \in \mathbf{\Sigma_{k+1}^P}$, so L is accepted by a nondeterministic oracle machine N with an oracle $A \in \mathbf{\Sigma_k^P}$ and running time bounded by a polynomial t. By induction, $A \cap \{0,1\}^m$ has circuits A_m of size at most $a(m)$, for a polynomial a. Let L' be the set of pairs (x, z) such that z encodes circuits $E_0, \ldots, E_{t(|x|)}$, with E_m of size at most $a(m)$, and N has an accepting computation on x in which every query w is answered by $E_{|w|}(w)$. Then $L' \in \mathbf{NP}$, and $x \in L$ iff $(x, z_{|x|}) \in L'$, where z_n encodes $A_0, \ldots, A_{t(n)}$. The length of z_n is polynomial and depends only on n. Since $L' \in \mathbf{NP} \subseteq \mathbf{P/poly}$, hardwiring z_n into the circuits for L' gives polynomial-size circuits for L. □

Lemma 3.9 (canonical circuits)**.** *If* $\mathbf{NP} \subseteq \mathbf{P/poly}$*, then for every* $L \in \mathbf{PH}$ *there is a tally language* $T_L \in \mathbf{PH}$ *such that* $L \in \mathbf{P}^{T_L}$*. More precisely, there are a polynomial* ℓ *and a polynomial-time oracle machine that, on input* 1^n*, queries* $1^{\pi(n,1)}, \ldots, 1^{\pi(n,\ell(n))}$ *and outputs the lexicographically first circuit of a fixed polynomial size computing* L *on* $\{0,1\}^n$*.*

Proof. Let $L \in \mathbf{\Sigma_k^P}$, $k \geq 1$. By Lemma 3.8, there is a constant c such that $L \cap \{0,1\}^n$ is computed by a circuit with n inputs and at most $s(n) = n^c + c$ gates, for every n. Put $\ell(n) = \ell(n, s(n))$. Let C_n be the lexicographically first string of length $\ell(n)$ that encodes a circuit computing L on $\{0,1\}^n$, and define

$$T_L = \{1^{\pi(n,i)} : \ n \geq 0,\ 1 \leq i \leq \ell(n),\ \text{the } i\text{-th bit of } C_n \text{ is } 1\}.$$

On input $x \in \{0,1\}^n$, query $1^{\pi(n,i)}$ for $i = 1, \ldots, \ell(n)$, which yields C_n, and output $C_n(x)$; so $L \in \mathbf{P}^{T_L}$. To show $T_L \in \mathbf{PH}$, let Q be the set of pairs $(1^n, \alpha)$ such that some $w \in \{0,1\}^{\ell(n)}$ with prefix α encodes a circuit C_w with $\forall x \in \{0,1\}^n\ C_w(x) = L(x)$. The predicate $C_w(x) = L(x)$ is in $\mathbf{P}^{\mathbf{\Sigma_k^P}}$, so $Q \in \mathbf{\Sigma_{k+2}^P}$. To decide 1^N, compute $(n, i) = \pi^{-1}(N)$ and reject unless $1 \leq i \leq \ell(n)$. Otherwise compute C_n bit by bit with $\ell(n)$ queries to Q: if α is the prefix found so far, the next bit is 0 iff $(1^n, \alpha 0) \in Q$. Accept iff the i-th bit of C_n is 1. This takes time polynomial in N, so $T_L \in \mathbf{P}^{\mathbf{\Sigma_{k+2}^P}} \subseteq \mathbf{PH}$. □

Theorem 3.10 (transfer principle)**.** *Let* $\mathcal{C}$ *be a class of promise problems (possibly languages) such that* $\mathbf{NP} \subseteq \mathbf{P/poly}$ *implies* $\mathbf{PH} \subseteq \mathbf{P}^{\mathcal{C}}$*. Then* $\mathbf{NP} \subseteq \mathbf{P/poly}$ *implies* $\mathbf{PH} \subseteq \mathbf{OGTD}[\mathcal{C}]$*.*

Proof. Assume $\mathbf{NP} \subseteq \mathbf{P/poly}$ and let $L \in \mathbf{PH}$. By Lemma 3.9, $L \in \mathbf{P}^{T_L}$ for a tally language $T_L \in \mathbf{PH}$. By hypothesis, $T_L \in \mathbf{P}^{\Pi}$ for some $\Pi \in \mathcal{C}$. A tally language in $\mathbf{P}^{\Pi}$ is in $\mathbf{OGTD}[\Pi]$: the advice for length N is the bit $T_L(1^N)$, computed by the $\mathbf{P}^{\Pi}$ machine for T_L, which is correct with every consistent oracle. The decoder outputs this bit on 1^N and 0 on all other inputs. Hence $L \in \mathbf{P}^{T_L} \subseteq \mathbf{P}^{\mathbf{OGTD}[\Pi]} = \mathbf{OGTD}[\Pi]$ by Proposition 2.11(c). □

Corollary 3.11 (Theorem 2)**.** *If* $\mathbf{NP} \subseteq \mathbf{P/poly}$*, then*

$$\mathbf{PH} = \mathbf{OGTD}[\mathbf{prMA}] = \mathbf{OL_2^P} = \mathbf{OMIN}.$$

Consequently, every class $\mathcal{D}$ *with* $\mathbf{OGTD}[\mathbf{prMA}] \subseteq \mathcal{D} \subseteq \mathbf{PH}$ *equals* $\mathbf{PH}$*. This includes all classes of Figure 2.*

Proof. By [CR11], $\mathbf{NP} \subseteq \mathbf{P/poly}$ implies $\mathbf{PH} = \mathbf{P}^{\mathbf{prMA}}$, so $\mathbf{PH} \subseteq \mathbf{OGTD[prMA]}$ by Theorem 3.10. By Theorem 3.5, $\mathbf{OGTD[prMA]} \subseteq \mathbf{OL_2^P}$, and $\mathbf{OL_2^P} = \mathbf{OMIN} \subseteq \mathbf{PH}$. □

Remark 3.12. Appendix A gives two other routes to the collapse to **OMIN**: through Theorem 3 instead of the collapse of [CR11], and through the collapse to $\mathbf{ZPP}^{\mathbf{NP}}$ [KW98], which avoids promise problems. It also discusses relativization.

Corollary 3.13. *The following are equivalent:* (i) $\mathbf{NP} \subseteq \mathbf{P/poly}$*;* (ii) $\mathbf{NP} \subseteq \mathbf{OL_2^P}$*;* (iii) $\mathbf{PH} = \mathbf{OL_2^P}$*;* (iv) $\mathbf{PH} \subseteq \mathbf{OGTD[prMA]}$*;* (v) $\mathbf{P}^{\mathbf{prSBP}} \subseteq \mathbf{P/poly}$.

Proof. (i)⇒(iv) and (i)⇒(iii) by Corollary 3.11; (iv)⇒(iii) since $\mathbf{OGTD[prMA]} \subseteq \mathbf{OL_2^P} \subseteq \mathbf{PH}$; (iii)⇒(ii) is trivial; and (ii)⇒(i) since $\mathbf{OL_2^P} \subseteq \mathbf{P/poly}$. Finally, (i) implies $\mathbf{P}^{\mathbf{prSBP}} \subseteq \mathbf{PH} = \mathbf{OL_2^P} \subseteq \mathbf{P/poly}$, and (v) implies (i) since $\mathbf{NP} \subseteq \mathbf{P}^{\mathbf{prSBP}}$. □

Corollary 3.14 (minimality)**.** $\mathbf{OL_2^P}$ *has a Karp–Lipton collapse, and* $\mathbf{OL_2^P} \not\subseteq \mathbf{SIZE}[n^k]$ *for every* k*. It is contained in* $\mathbf{O_2^P} \cap \mathbf{L_2^P}$*, and no smaller class with both properties is known. Section 4 gives a collapse to the smaller class* **OM**.

Proof. By Corollary 3.11, (F2), (F4), and Proposition 2.11(a). □

4 The ultimate collapse

4.1 Strict oblivious Merlin–Arthur oracles

The pointwise domination condition (OMA4) lets us sample an **sOprMA** oracle with an **ONP** oracle. The following lemma is stated for transducers, because we apply it to generators.

Lemma 4.1 (sampling **sOprMA** oracles)**.** *For every* $\Pi \in \mathbf{sOprMA}$ *there is a language* $K_\Pi \in \mathbf{ONP}$ *with the following property. For every deterministic polynomial-time oracle transducer* M *there is a randomized polynomial-time oracle transducer* M' *such that, for every input* z*, with probability at least* $3/4$*,* $M'^{K_\Pi}(z) = M^O(z)$ *for some language* O *consistent with* Π*. In particular,* $\mathbf{P}^{\mathbf{sOprMA}} \subseteq \mathbf{BPP}^{\mathbf{ONP}}$.

Proof. Let V and (y_n) witness $\Pi \in \mathbf{sOprMA}$, and let $K_\Pi = \{(x, r) : \exists y\ V(x, y, r) = 1\}$. By (OMA4), $(x, r) \in K_\Pi$ iff $V(x, y_{|x|}, r) = 1$, so $y_{|x|}$ is an oblivious witness and $K_\Pi \in \mathbf{ONP}$. If $x \in \Pi_{\mathrm{YES}}$, then $(x, r) \in K_\Pi$ for every r. If $x \in \Pi_{\mathrm{NO}}$, then $\Pr_r[(x, r) \in K_\Pi] = \Pr_r[V(x, y_{|x|}, r) = 1] < 1/4$.

Let $Q \geq 1$ bound the number of queries of M. The transducer M' simulates M. At the first occurrence of a query x, it chooses independent random strings $r_1, \ldots, r_k$, where $4^{-k} \leq 1/(4Q)$, and answers YES iff $(x, r_j) \in K_\Pi$ for all j. It stores the answer and reuses it for later occurrences of x. Queries in Π_{YES} are always answered YES, and each query in Π_{NO} is answered YES with probability less than $1/(4Q)$. By the union bound, with probability at least $3/4$ all queries in $\Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$ are answered correctly. The stored answers then extend to a language O consistent with Π, and $M'^{K_\Pi}(z) = M^O(z)$. □

The following lemma is implicit in the proof of [HVW26, Proposition 6.9], which establishes the collapse to $\mathbf{P}^{\mathbf{OprMA}}$.

Lemma 4.2 (collapse to $\mathbf{P}^{\mathbf{sOprMA}}$)**.** *If* $\mathbf{NP} \subseteq \mathbf{P/poly}$*, then every promise problem in* **prMA** *belongs to* **sOprMA**. *Consequently,* $\mathbf{PH} = \mathbf{P}^{\mathbf{sOprMA}}$.

Proof. Let $\Pi \in \mathbf{prMA}$ via $V(x, y, r)$ with perfect completeness, soundness $1/4$, and proofs of length $p(n)$. Let $t = p(n) + 2$, and for $R = (r_1, \ldots, r_t)$ let $C_{x,R}(y) = \bigwedge_{i \le t} V(x, y, r_i)$. If $x \in \Pi_{\mathrm{NO}}$, then

$$\Pr_R[\exists y \; C_{x,R}(y) = 1] \le 2^{p(n)} 4^{-t} < 1/4.$$

Since $\mathbf{NP} \subseteq \mathbf{P/poly}$, search-to-decision gives polynomial-size circuits E_n that output a satisfying assignment of every satisfiable circuit of the size of $C_{x,R}$, for $x \in \{0,1\}^n$. The new verifier $V'(x, E, R)$ rejects unless E encodes a circuit of the prescribed size; otherwise it computes $y = E(C_{x,R})$ and accepts iff $C_{x,R}(y) = 1$. The oblivious proof is E_n.

(OMA1) If $x \in \Pi_{\mathrm{YES}}$, then $C_{x,R}$ is satisfiable for every R, so $V'(x, E_n, R) = 1$.

(OMA2) If $x \in \Pi_{\mathrm{NO}}$, then $V'(x, E, R) = 1$ implies that $C_{x,R}$ is satisfiable, which happens with probability less than $1/4$.

(OMA4) For every x, E and R, if $V'(x, E, R) = 1$, then $C_{x,R}$ is satisfiable, so $V'(x, E_n, R) = 1$.

Finally, $\mathbf{PH} = \mathbf{P}^{\mathbf{prMA}}$ [CR11], $\mathbf{P}^{\mathbf{prMA}} \subseteq \mathbf{P}^{\mathbf{sOprMA}}$ under the hypothesis, and $\mathbf{P}^{\mathbf{sOprMA}} \subseteq \mathbf{P}^{\mathbf{prMA}} \subseteq \mathbf{PH}$. □

Observation 4.3 (collapse to $\mathbf{ZPP}^{\mathbf{ONP}}$)**.** *If* $\mathbf{NP} \subseteq \mathbf{P/poly}$*, then* $\mathbf{NP} = \mathbf{ONP}$ *and* $\mathbf{PH} = \mathbf{ZPP}^{\mathbf{ONP}}$*.*

Proof. Let $L \in \mathbf{NP}$ via a verifier $U(x, w)$. By the hypothesis and search-to-decision, there are polynomial-size circuits S_n that output a witness for every $x \in L \cap \{0,1\}^n$. The verifier $U'(x, S) = U(x, S(x))$ has the oblivious witness S_n and never accepts $x \notin L$. Hence $\mathbf{NP} = \mathbf{ONP}$, which is [FSW09, Proposition 5], and $\mathbf{PH} = \mathbf{ZPP}^{\mathbf{NP}} = \mathbf{ZPP}^{\mathbf{ONP}}$ [KW98]. □

Lemma 4.2, Observation 4.3 and Theorem 2 are the three deepest collapses discussed in the introduction.

4.2 The class OM

Recall that $\mathbf{OM} = \mathbf{OGTD}[\mathbf{FP}^{\mathbf{sOprMA}}, \mathbf{co\text{-}ONP}, \mathbf{P}]$ and $\mathbf{OUM} = \mathbf{OUGTD}[\mathbf{FP}^{\mathbf{sOprMA}}, \mathbf{co\text{-}ONP}, \mathbf{P}]$. Like **OMIN**, these classes are "robust", i.e., closed under complement and polynomial-time Turing reductions.

Proposition 4.4. $\mathbf{OM}$ *and* $\mathbf{OUM}$ *are closed under complement and under polynomial-time Turing reductions:* $\mathbf{P}^{\mathbf{OM}} = \mathbf{OM}$ *and* $\mathbf{P}^{\mathbf{OUM}} = \mathbf{OUM}$*.*

Proof. For complement, negate the decoder. Let $L \in \mathbf{P}^{L_1, \ldots, L_r}$ via a machine M running in time $t(n)$, where $L_i \in \mathbf{OM}$ via (G_i, T_i, D_i) with oracles $\Pi_i \in \mathbf{sOprMA}$. The problem $\Pi_1 \oplus \cdots \oplus \Pi_r$ is in $\mathbf{sOprMA}$: on input $\langle i, x \rangle$ run the verifier of Π_i, and use the tuple of the oblivious proofs as the oblivious proof; (OMA4) holds componentwise. The new generator outputs the tuple of all $G_i(1^m)$ for $i \le r$ and $m \le t(n)$, computed in one run with this oracle. Each component of a legal output is a legal output of G_i. The new tester accepts a tuple iff every component $a_{i,m}$ satisfies $(1^m, a_{i,m}) \in T_i$. Its complement is in $\mathbf{ONP}$, with the tuple of the oblivious witnesses of the complements of the T_i at all lengths $m \le t(n)$ as the oblivious witness. The new decoder simulates M and answers a query q to L_i by $D_i(q, a_{i,|q|})$. If every T_i accepts exactly one string per length, so does the new tester. □

4.3 Three ways to run the generator

Theorem 4.5. $\mathbf{OUM} \subseteq \mathbf{OM} \subseteq \mathbf{OMIN} \cap \mathbf{P}^{\mathbf{sOprMA}} \cap \mathbf{ZPP}^{\mathbf{ONP}}$.

Proof. Let $L \in \mathbf{OM}$ via a generator G with oracle $\Pi \in \mathbf{sOprMA}$, a tester $T \in \mathbf{co\text{-}ONP}$, and a decoder $D \in \mathbf{P}$. Let $\overline{T} \in \mathbf{ONP}$ be the complement of T; pairs not of the form $(1^n, a)$ with $|a| = p(n)$ belong to $\overline{T}$ and are accepted without a witness.

$\mathbf{P}^{\mathbf{sOprMA}}$*: run the generator.* On input $x \in \{0,1\}^n$, compute $a = G^O(1^n)$ and output $D(x, a)$. Since $a \in \mathrm{Out}_G(n)$ is good advice, this is correct for every O consistent with Π.

OMIN*: derandomize the generator.* Dropping the tester, $\mathbf{OM} \subseteq \mathbf{OGTD}[\mathbf{sOprMA}] \subseteq \mathbf{OGTD}[\mathbf{prMA}]$, because $\mathbf{sOprMA} \subseteq \mathbf{prMA}$. By Theorem 3.5, $\mathbf{OGTD}[\mathbf{prMA}] \subseteq \mathbf{OL}_2^{\mathbf{P}} = \mathbf{OMIN}$.

$\mathbf{ZPP}^{\mathbf{ONP}}$*: sample the generator and test.* On input $x \in \{0,1\}^n$, run the randomized simulation of G on 1^n from Lemma 4.1 with the oracle $K_\Pi \in \mathbf{ONP}$, obtaining a string a. Then query $(1^n, a) \in \overline{T}$. If the answer is yes, output "?". Otherwise T accepts a, so a is good advice, and the algorithm outputs $D(x, a) = L(x)$. With probability at least $3/4$, the string a is in $\mathrm{Out}_G(n)$, which T accepts, so the output is not "?". The two **ONP** oracles are combined into one **ONP** language by tagging inputs and bundling oblivious witnesses. Hence $L \in \mathbf{ZPP}^{\mathbf{ONP}}$. □

4.4 The collapse

Theorem 4.6. *If* $\mathbf{NP} \subseteq \mathbf{P/poly}$*, then* $\mathbf{PH} = \mathbf{OUM} = \mathbf{OM}$.

Proof. By Theorem 4.5, $\mathbf{OUM} \subseteq \mathbf{OM} \subseteq \mathbf{OMIN} \subseteq \mathbf{PH}$. Conversely, assume $\mathbf{NP} \subseteq \mathbf{P/poly}$ and let $L \in \mathbf{PH}$. By Theorem 2, $L \in \mathbf{OMIN}$ via an oblivious order $<_n$ on $\{0,1\}^{p(n)}$, minima μ_n, and a decoder D.

- *Tester.* Let $T = \{(1^n, a) : a \in \{0,1\}^{p(n)}$ and no u satisfies $u <_n a\}$. Every $a \neq \mu_n$ satisfies $\mu_n <_n a$, so the complement of T is in **ONP** with the oblivious witness μ_n. Thus $T \in \mathbf{co\text{-}ONP}$, and T accepts exactly μ_n at length n.
- *Decoder.* D itself: $D(x, \mu_n) = L(x)$ for every $x \in \{0,1\}^n$.
- *Generator.* The tally language $M = \{1^{\pi(n,i)} : 1 \leq i \leq p(n),$ bit i of μ_n is $1\}$ is in $\mathbf{L}_2^{\mathbf{P}}$ (see the proof of Theorem 3.3), hence in **PH**. By Lemma 4.2, $M \in \mathbf{P}^\Pi$ for some $\Pi \in \mathbf{sOprMA}$. The generator computes μ_n bit by bit with this $\mathbf{P}^\Pi$ machine. Every consistent oracle gives the correct bits, so $\mathrm{Out}_G(n) = \{\mu_n\}$.

Hence (1) holds with $\mathrm{Acc}_T(n) = \{\mu_n\}$, and $L \in \mathbf{OUM}$. □

Remark 4.7.

1. The proof combines two of the three collapses: **OMIN** supplies a canonical advice string with a **co-ONP** certificate, and $\mathbf{P}^{\mathbf{sOprMA}}$ supplies the generator. The certificate is the one behind $\mathbf{OMIN} \subseteq \mathbf{SOUP}^{\mathbf{ONP}[1]}$ [HVW26, Theorem 5.2], which, combined with Theorem 4.5, also gives $\mathbf{OM} \subseteq \mathbf{SOUP}^{\mathbf{ONP}[1]}$.
2. The collapse to **OM** can also be proved without **OMIN**, along the lines of the proof of the collapse to $\mathbf{ZPP}^{\mathbf{NP}}$ [BCG+96, KW98] (Appendix A.3).
3. By Theorem 3.10, the tally closure $\mathbf{P}^{\mathbf{TALLY} \cap \mathcal{C}}$ of every Karp–Lipton class $\mathcal{C}$ that is closed under polynomial-time Turing reductions is again a Karp–Lipton class. So some Karp–Lipton class inside $\mathbf{OMIN} \cap \mathbf{P}^{\mathbf{sOprMA}} \cap \mathbf{ZPP}^{\mathbf{ONP}}$ is easy to obtain: the tally closure of $\mathbf{P}^{\mathbf{sOprMA}} \cap \mathbf{ZPP}^{\mathbf{ONP}}$, which lies in **OMIN** by Theorem 3.3, because $\mathbf{P}^{\mathbf{sOprMA}} \subseteq \mathbf{P}^{\mathbf{prMA}} \subseteq \mathbf{L}_2^{\mathbf{P}}$ [HV25].

What distinguishes **OM** is that it is defined by a natural model, and that each of the three containments comes from a different way of running its generator. The zero-error simulation uses the tester: we do not know whether $\mathbf{OGTD}[\mathbf{sOprMA}] \subseteq \mathbf{ZPP}^{\mathbf{ONP}}$.

4. Fixed-polynomial lower bounds for **OM** remain open (see discussion in Section 7).

5 A Goldreich–Wigderson lemma with approximate counting, and $\mathbf{P}^{\mathbf{prS}_2^{\mathbf{P}}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$

Throughout this section, the oracle is a fixed language O consistent with SSE, and "computable" means computable in polynomial time with oracle O. All statements hold for every such O.

Matrices [CR11, Def. 4.6]. Let $V : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}$ be a circuit. A row y *beats* a column z if $V(y,z) = 1$, and z beats y if $V(y,z) = 0$. A row is *irrefutable* if it beats every column, and similarly for columns. A set R of rows is a *row-side collectively irrefutable certificate* if every column is beaten by some $y \in R$. A set T of columns is a *column-side* one if every row is beaten by some $z \in T$.

Lemma 5.1 (above-average rows)**.** *Let a nonempty set $Y \subseteq \{0,1\}^a$, a set $Z \subseteq \{0,1\}^b$, and a predicate V be given by circuits, and let $\varepsilon \in (0,1]$ be rational. For $y \in Y$, let* $\deg(y) = |\{z \in Z : V(y,z) = 1\}|$*. One can compute, in time polynomial in the circuit sizes and $1/\varepsilon$, a row $y \in Y$ with*

$$\deg(y) \;\geq\; \frac{1}{(1+\varepsilon)^{2a}} \cdot \frac{1}{|Y|} \sum_{y' \in Y} \deg(y').$$

If Z is nonempty, the same holds for columns $z \in Z$, with $\deg(z) = |\{y \in Y : V(y,z) = 0\}|$ *and $2b$ in place of $2a$.*

Proof. This is the procedure of [HV25, Lemma 3.12], with the average order rank replaced by the average degree and "smaller" replaced by "larger". We fix the bits of y one at a time, maintaining a prefix β such that $Y_\beta = \{y \in Y : y \text{ starts with } \beta\}$ is nonempty; for a set H of rows, let $\mathrm{av}(H)$ denote the average degree of its elements. In each step, let $H_0 = Y_{\beta 0}$ and $H_1 = Y_{\beta 1}$.

- If one of H_0, H_1 is empty (detected exactly by Lemma 2.6), append the bit c, for which $H_c \neq \varnothing$, to β.
- Otherwise, for $c \in \{0,1\}$ compute by Lemma 2.6 with accuracy ε approximations t_A^c of $|H_c|$ and t_B^c of $|\{(y,z) \in H_c \times Z : V(y,z) = 1\}|$, and let $e_c = t_B^c / t_A^c$. Extend β by the bit c with the larger e_c.

Each estimate satisfies $\mathrm{av}(H_c)/(1+\varepsilon) \leq e_c \leq (1+\varepsilon)\mathrm{av}(H_c)$. Since $\mathrm{av}(Y_\beta)$ is a weighted mean of $\mathrm{av}(H_0)$ and $\mathrm{av}(H_1)$, it is at most their maximum. If the chosen half has the larger average, its average is at least $\mathrm{av}(Y_\beta)$. Otherwise its average is at least $e_c/(1+\varepsilon) \geq e_{1-c}/(1+\varepsilon) \geq \mathrm{av}(H_{1-c})/(1+\varepsilon)^2 \geq \mathrm{av}(Y_\beta)/(1+\varepsilon)^2$. So each step loses a factor of at most $(1+\varepsilon)^2$, and after a steps Y_β consists of a single element y with the claimed degree. □

Theorem 5.2 (Goldreich–Wigderson lemma with approximate counting)**.** *Given any circuit $V : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}$, one can compute a row-side or a column-side collectively irrefutable certificate with at most $3(a+b)+1$ elements.*

Proof. Let $d = \max(a, b, 1)$. Maintain lists R and T, initially empty, and the sets

$$Y = \{y : \forall z \in T\ V(y, z) = 1\}, \qquad Z = \{z : \forall y \in R\ V(y, z) = 0\}$$

of rows and columns not yet beaten. Both are given by conjunctions of copies of V. Repeat the following steps.

1. If $Z = \emptyset$, output R. If $Y = \emptyset$, output T. Lemma 2.6 tests emptiness exactly.
2. Let $N_c = |\{(y, z) \in Y \times Z : V(y, z) = c\}|$ for $c \in \{0, 1\}$. Compute estimates t_0, t_1 with accuracy 1, and let $c = 1$ if $t_1 \geq t_0$, and $c = 0$ otherwise. Then $N_{1-c} \leq t_{1-c} \leq t_c \leq 2N_c$, so $N_c \geq |Y||Z|/3$.
3. If $c = 1$, apply Lemma 5.1 with $\varepsilon = 1/(8d)$, so that $(1 + \varepsilon)^{2a} \leq e^{1/4} < 4/3$. It gives $y \in Y$ that beats more than $|Z|/4$ columns of Z, and we append y to R. If $c = 0$, we symmetrically append to T a column of Z that beats more than $|Y|/4$ rows of Y.

Each round multiplies $|Y| \cdot |Z|$ by less than $3/4$. This product starts at 2^{a+b} and is at least 1 while the procedure runs, so there are at most $3(a + b) + 1$ rounds. The output is a certificate by the definitions of Y and Z. □

Corollary 5.3. *If V has an irrefutable row, the procedure of Theorem 5.2 outputs a row-side collectively irrefutable certificate R with $|R| \leq 3(a + b) + 1$, for every oracle consistent with* SSE*. Symmetrically for columns.*

Proof. An irrefutable row is never beaten by a column, so it stays in Y, and the procedure never outputs T. Since the number of rounds is bounded, it outputs R. □

Theorem 5.4 (Theorem 3)**.** *For every $\Pi \in \mathbf{prS_2^P}$ there is a deterministic polynomial-time oracle machine G such that, for every language O consistent with* SSE*, the language decided by G^O is consistent with Π. Consequently, $\mathbf{P^{prS_2^P}} \subseteq \mathbf{P^{prSBP}}$, $\mathbf{FP^{prS_2^P}} \subseteq \mathbf{FP^{prSBP}}$ and $\mathbf{OGTD[prS_2^P]} \subseteq \mathbf{OGTD[prSBP]}$. In particular, $\mathbf{S_2^P} \subseteq \mathbf{P^{prSBP}}$.*

Proof. Let $\Pi \in \mathbf{prS_2^P}$ via V. On a query q, the machine G runs the procedure of Theorem 5.2 on $V_q(y, z) = V(q, y, z)$ and answers YES iff the output is row-side. If $q \in \Pi_{\mathrm{YES}}$, V_q has an irrefutable row, so the answer is YES by Corollary 5.3; symmetrically, if $q \in \Pi_{\mathrm{NO}}$, the answer is NO. The remaining claims follow from Proposition 2.11(d). □

6 The oblivious setting of symmetric alternation

We now prove part (3) of Theorem 4. The basic tool is a list of witnesses that collectively certify all yes-instances of a given length.

Lemma 6.1 (collective oblivious witnesses)**.** *Let $W(x, w)$ be a predicate computable in time polynomial in $|x|$, where $|w| = p(|x|)$ for a polynomial p, and possibly depending on an auxiliary string α that is part of the input of the procedure below. Let n be such that some w^* satisfies*

$$\forall x \in \{0, 1\}^n : \quad (\exists w\ W(x, w) = 1) \ \Rightarrow\ W(x, w^*) = 1. \tag{2}$$

Then, given 1^n (and α), one can compute in $\mathbf{FP}^{\mathrm{SSE}}$ a list $w_1, \ldots, w_k$ with $k \leq 3(2p(n) + n) + 1$ such that

$$\forall x \in \{0, 1\}^n : \quad (\exists w\ W(x, w) = 1) \ \Rightarrow\ \exists j \leq k\ W(x, w_j) = 1,$$

for every oracle consistent with SSE*.*

Proof. Consider the *domination game* with rows $w \in \{0,1\}^{p(n)}$ and columns $(x, w') \in \{0,1\}^n \times \{0,1\}^{p(n)}$, where row w beats column (x, w') iff $W(x, w) = 1$ or $W(x, w') = 0$. It is given by a circuit of size polynomial in n (and $|\alpha|$). By (2), w^* is an irrefutable row. By Corollary 5.3, we obtain a list $R = \{w_1, \ldots, w_k\}$ such that every column is beaten by some w_j. If $W(x, w') = 1$ for some w', then the column (x, w') is beaten by some w_j, which means $W(x, w_j) = 1$. □

Theorem 6.2. $\mathbf{ONP} \subseteq \mathbf{OGTD}[\mathbf{prSBP}]$.

Proof. Let $L \in \mathbf{ONP}$ via a verifier $V(x, w)$ with oblivious witnesses w_n. Then $W = V$ satisfies (2) with $w^* = w_n$: if $V(x, w) = 1$ for some w, then $x \in L$, so $V(x, w_n) = 1$. The advice for length n is the list of Lemma 6.1, padded to a fixed polynomial length by repeating w_1, and the decoder accepts x iff $V(x, w_j) = 1$ for some j. If $x \in L$, some w_j works; if $x \notin L$, no witness works. □

Theorem 6.3 (part (3) of Theorem 4)**.** $\mathbf{O}_2^{\mathbf{P}} \subseteq \mathbf{OGTD}[\mathbf{prSBP}]$.

Proof. Let $L \in \mathbf{O}_2^{\mathbf{P}}$ via a verifier $V(x, y, z)$ with $|y| = |z| = p(|x|)$ and oblivious irrefutable certificates y_n, z_n:

$$\forall x \in L \cap \{0,1\}^n \ \forall z \ V(x, y_n, z) = 1, \qquad \forall x \in \{0,1\}^n \setminus L \ \forall y \ V(x, y, z_n) = 0.$$

Fix n and an oracle consistent with SSE. The advice is computed in two stages.

Stage 1: the consistency game. Its rows are pairs $(y, z) \in \{0,1\}^{2p(n)}$, its columns are triples $(x, z', y') \in \{0,1\}^{n+2p(n)}$. Let us say that

$$\text{row } (y, z) \text{ beats column } (x, z', y') \quad \Longleftrightarrow \quad V(x, y, z') = 1 \ \text{ or } \ V(x, y', z) = 0.$$

The row (y_n, z_n) is irrefutable: if $x \in L$, then $V(x, y_n, z') = 1$, and if $x \notin L$, then $V(x, y', z_n) = 0$. By Corollary 5.3 we compute a list $R = \{(y_1, z_1), \ldots, (y_k, z_k)\}$ such that every column is beaten by some element of R. Let $Y = (y_1, \ldots, y_k)$.

Claim. For every $x \in L \cap \{0,1\}^n$ and every z', some $i \le k$ satisfies $V(x, y_i, z') = 1$.

Proof of the claim. Consider the column (x, z', y_n). Since $x \in L$, $V(x, y_n, z_i) = 1$ for all i, so the second alternative fails for every i. Hence the first holds for some i. □

Stage 2: a collective no-witness. Define

$$W(x, z) = 1 \quad \Longleftrightarrow \quad V(x, y_i, z) = 0 \ \text{ for all } i \le k,$$

a predicate computable in polynomial time given Y. By the claim, $W(x, z) = 0$ for all z if $x \in L \cap \{0,1\}^n$. If $x \in \{0,1\}^n \setminus L$, then $W(x, z_n) = 1$, because z_n beats every row on x. Hence $x \notin L$ iff $\exists z \ W(x, z) = 1$, for $x \in \{0,1\}^n$, and (2) holds with $w^* = z_n$. Lemma 6.1, with $\alpha = Y$, gives a list $z_1^*, \ldots, z_m^*$ such that every $x \in \{0,1\}^n \setminus L$ satisfies $W(x, z_j^*) = 1$ for some j.

Decoding. The advice for length n is $(Y, z_1^*, \ldots, z_m^*)$, padded to a fixed polynomial length. The decoder rejects x iff some $j \le m$ satisfies $V(x, y_i, z_j^*) = 0$ for all $i \le k$. If $x \notin L$, such a j exists. If $x \in L$, then $W(x, z) = 0$ for every z, so no such j exists. Both stages run in polynomial time with the oracle, and the correctness argument applies to every oracle consistent with SSE. □

Corollary 6.4 (the oblivious chain)**.**

$$\mathbf{OGTD}[\mathbf{prMA}] \subseteq \mathbf{OL}_2^{\mathbf{P}} \subseteq \mathbf{O}_2^{\mathbf{P}} \subseteq \mathbf{OGTD}[\mathbf{prSBP}] \subseteq \mathbf{P}^{\mathbf{prSBP}} \cap \mathbf{P/poly}.$$

Moreover, $\mathbf{P}^{\mathbf{prOMA}} \subseteq \mathbf{O}_2^{\mathbf{P}} \subseteq \mathbf{OGTD}[\mathbf{prSBP}]$ *and* $\mathbf{OGTD}[\mathbf{prS}_2^{\mathbf{P}}] \subseteq \mathbf{OGTD}[\mathbf{prSBP}]$.

Proof. Theorems 3.5 and 6.3, (F2), (F4), Theorem 5.4 and Proposition 2.11(a). □

The class **OM** of Section 4 lies at the bottom of this chain: $\mathbf{OM} \subseteq \mathbf{OGTD}[\mathbf{sOprMA}] \subseteq \mathbf{OGTD}[\mathbf{prMA}]$.

Remark 6.5. The containments of Corollary 6.4 mirror the uniform chain $\mathbf{P}^{\mathbf{prOMA}} \subseteq \mathbf{P}^{\mathbf{prMA}} \subseteq \mathbf{L}_2^{\mathbf{P}} \subseteq \mathbf{S}_2^{\mathbf{P}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$ (Figure 2). There is one asymmetry. $\mathbf{S}_2^{\mathbf{P}} \subseteq \mathbf{P}^{\mathbf{prS}_2^{\mathbf{P}}}$ holds trivially, but we do not know whether $\mathbf{O}_2^{\mathbf{P}} \subseteq \mathbf{OGTD}[\mathbf{prS}_2^{\mathbf{P}}]$. Theorem 6.3 therefore does not follow from Theorem 3 by a reduction; it uses the Goldreich–Wigderson lemma with approximate counting directly, twice. Using instead the $\mathbf{FP}^{\mathbf{prAM}}$ procedure of [CR11] for matrices with an irrefutable row would give only the weaker $\mathbf{O}_2^{\mathbf{P}} \subseteq \mathbf{OGTD}[\mathbf{prAM}]$.

7 Open problems

1. *Lower bounds below* **OMIN**. Does **OM**, or even **OGTD**[**prOMA**], contain a language outside $\mathbf{SIZE}[n^k]$ for every k? For **OM**, this would give fixed-polynomial lower bounds for both $\mathbf{P}^{\mathbf{sOprMA}}$ and $\mathbf{ZPP}^{\mathbf{ONP}}$. For **OGTD**[**prOMA**], it would give such lower bounds for $\mathbf{P}^{\mathbf{prOMA}}$.

2. *The role of the tester.* In **OM**, the **co-ONP** tester certifies the generated advice; it is used only for the containment in $\mathbf{ZPP}^{\mathbf{ONP}}$ (Theorem 4.5). Can it be dropped, i.e., is $\mathbf{OGTD}[\mathbf{sOprMA}] \subseteq \mathbf{ZPP}^{\mathbf{ONP}}$? Can the certified advice always be made unique, i.e., is $\mathbf{OUM} = \mathbf{OM}$? A positive answer to the first question would also make the tester-free class **OGTD**[**sOprMA**] lie inside all three deepest collapse classes.

3. *The three deepest collapse classes.* No containment among **OMIN**, $\mathbf{P}^{\mathbf{sOprMA}}$ and $\mathbf{ZPP}^{\mathbf{ONP}}$ is known. Since **ONP** is contained in the last two classes, $\mathbf{P}^{\mathbf{sOprMA}} \subseteq \mathbf{OMIN}$ or $\mathbf{ZPP}^{\mathbf{ONP}} \subseteq \mathbf{OMIN}$ would imply $\mathbf{ONP} \subseteq \mathbf{OMIN}$, which is open [HV25, HVW26]. Conversely, $\mathbf{OMIN} \subseteq \mathbf{P}^{\mathbf{sOprMA}}$ or $\mathbf{OMIN} \subseteq \mathbf{ZPP}^{\mathbf{ONP}}$ would give fixed-polynomial lower bounds for these classes. If $\mathbf{NP} \subseteq \mathbf{P/poly}$, then $\mathbf{ONP} \subseteq \mathbf{OMIN}$ (see Corollary A.3). By Theorem 6.2, $\mathbf{ONP} \subseteq \mathbf{OGTD}[\mathbf{prSBP}]$ unconditionally.

4. *Symmetric alternation versus linear orders.* Is $\mathbf{O}_2^{\mathbf{P}} \subseteq \mathbf{OL}_2^{\mathbf{P}}$, i.e., $\mathbf{O}_2^{\mathbf{P}} = \mathbf{OL}_2^{\mathbf{P}}$? Is $\mathbf{S}_2^{\mathbf{P}} \subseteq \mathbf{L}_2^{\mathbf{P}}$, i.e., $\mathbf{S}_2^{\mathbf{P}} = \mathbf{L}_2^{\mathbf{P}}$? Is $\mathbf{O}_2^{\mathbf{P}} \subseteq \mathbf{OGTD}[\mathbf{prS}_2^{\mathbf{P}}]$?

5. *Promise oracles.* Chakaravarthy and Roy [CR11] asked whether $\mathbf{P}^{\mathbf{prS}_2^{\mathbf{P}}} \subseteq \mathbf{S}_2^{\mathbf{P}}$. For $\mathbf{P}^{\mathbf{prS}_2^{\mathbf{P}}}$, Theorem 3 improves their upper bound $\mathbf{P}^{\mathbf{prAM}}$ to $\mathbf{P}^{\mathbf{prSBP}}$. For $\mathbf{S}_2^{\mathbf{P}}$, our earlier paper [HV25] shows that it contains $\mathbf{P}^{\mathbf{prMA}}$, via its subclass $\mathbf{L}_2^{\mathbf{P}}$ defined by Korten and Pitassi [KP24]. However, the original question remains open.

6. *Zero error.* We proved that $(\mathbf{S}_2^{\mathbf{P}} \subseteq)\ \mathbf{P}^{\mathbf{prS}_2^{\mathbf{P}}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$. At the same time, Cai's result [Cai07] shows $\mathbf{S}_2^{\mathbf{P}} \subseteq \mathbf{ZPP}^{\mathbf{NP}}$ (in fact, $\mathbf{P}^{\mathbf{prS}_2^{\mathbf{P}}} \subseteq \mathbf{ZPP}^{\mathbf{NP}}$ is sketched in [CR11, Theorem 4.16] and also follows from [FIKU08]). This raises a natural question: what is the relationship between $\mathbf{ZPP}^{\mathbf{NP}}$ and $\mathbf{P}^{\mathbf{prSBP}}$? For example, can we strengthen our theorem to $\mathbf{ZPP}^{\mathbf{NP}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$?

7. *Parallel queries.* Can the procedure of Theorem 5.2 be made non-adaptive? This would give $\mathbf{S}_2^{\mathbf{P}} \subseteq \mathbf{P}_{\parallel}^{\mathbf{prSBP}} = \mathbf{BPP}_{\mathbf{path}}$ [OS18, HV25]. For minimum finding, i.e., for $\mathbf{L}_2^{\mathbf{P}} \subseteq \mathbf{P}_{\parallel}^{\mathbf{prSBP}}$, this question was asked in [HV25, Section 6]. Either containment would imply $\mathbf{P}^{\mathbf{NP}} \subseteq \mathbf{PP}$ [HHT97], which is open and fails relative to an oracle [Bei94]. Hence a positive answer requires non-relativizing techniques.

AI methodology

This paper started as a follow-up to our earlier paper [HV25], where the question of the smallest Karp–Lipton collapse was investigated and the earlier containment $\mathbf{L}_2^{\mathbf{P}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$ was proved. We used GPT-6 Astra and Claude Opus 5.5 for assistance at most stages of the preparation of this paper, so these tools materially affected all sections. Namely, we shared our private notes with these tools, discussed proofs with them in an unstructured manner, asked Astra-6 to verify the results, and asked Opus 5.5 to draft both the conference submission and the preprint version (and also to keep them in sync) in a dialogue with us. Opus 5.5 formalized, at our request, the definition of the class **sOprMA** (even though it was somewhat straightforward from the definitions of our previous classes) and the precise definition of the **OGTD** classes (after we had formulated it in principle). We the authors take full responsibility for the paper, including the correctness and originality of all content, which we verified. The final text was edited by the human authors.

Acknowledgments

We are indebted to Ryan Williams, with whom we recently finished our previous paper [HVW26] on the topic and discussed multi-stage computations while working on it. The first author would like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the programme 'Logical Foundations of Complexity Theory', where part of the work on this paper was undertaken. This event was supported by EPSRC grant EP/Z000580/1. We greatly appreciate an open discussion with Oliver Korten during the workshop LFCW02 in the framework of this programme. We thank the scientific organizers (Sam Buss and Jan Krajíček) for doing their excellent job, and the other participants for contributing to the stimulating atmosphere of the event. We also thank Scott Aaronson, Yaroslav Alekseev, and John Hitchcock for conversations on related topics.

References

[Aar07] S. Aaronson. The learnability of quantum states. *Proc. R. Soc. A*, 463(2088):3089–3114, 2007.

[AD14] S. Aaronson and A. Drucker. A full characterization of quantum advice. *SIAM J. Comput.*, 43(3):1131–1183, 2014.

[AGHK11] B. Aydinlioğlu, D. Gutfreund, J. M. Hitchcock, and A. Kawachi. Derandomizing Arthur–Merlin games and approximate counting implies exponential-size lower bounds. *Comput. Complex.*, 20(2):329–366, 2011.

[AKSS95] V. Arvind, J. Köbler, U. Schöning, and R. Schuler. If NP has polynomial-size circuits, then MA=AM. *Theor. Comput. Sci.*, 137(2):279–282, 1995.

[BCG+96] N. H. Bshouty, R. Cleve, R. Gavaldà, S. Kannan, and C. Tamon. Oracles and queries that are sufficient for exact learning. *J. Comput. Syst. Sci.*, 52(3):421–433, 1996.

[Bei94] R. Beigel. Perceptrons, PP, and the polynomial hierarchy. *Comput. Complex.*, 4:339–349, 1994.

[BF99] H. Buhrman and L. Fortnow. One-sided versus two-sided error in probabilistic computation. In *STACS 1999*, pages 100–109, 1999.

[BGM06] E. Böhler, C. Glaßer, and D. Meister. Error-bounded probabilistic computations between MA and AM. *J. Comput. Syst. Sci.*, 72(6):1043–1076, 2006.

[Cai07] J.-Y. Cai. $S_2^p \subseteq ZPP^{NP}$. *J. Comput. Syst. Sci.*, 73(1):25–35, 2007.

[Can96] R. Canetti. More on BPP and the polynomial-time hierarchy. *Inf. Process. Lett.*, 57(5):237–241, 1996.

[CHLR26] L. Chen, S. Hirahara, Z. Li, and H. Ren. Symmetric exponential time requires near-maximum circuit size. *J. ACM*, 73(1):4:1–4:21, 2026.

[CHR24] L. Chen, S. Hirahara, and H. Ren. Symmetric exponential time requires near-maximum circuit size. In *STOC 2024*, pages 1990–1999, 2024.

[CHV92] J.-Y. Cai, L. A. Hemachandra, and J. Vyskoč. Promise problems and access to unambiguous computation. Technical Report TR 419, Department of Computer Science, University of Rochester, April 1992. Conference version in *MFCS 1992*, LNCS 629, pages 162–171.

[CMMW19] L. Chen, D. M. McKay, C. D. Murray, and R. R. Williams. Relations and equivalences between circuit lower bounds and Karp–Lipton theorems. In *CCC 2019*, LIPIcs 137, pages 30:1–30:21, 2019.

[CR06] V. T. Chakaravarthy and S. Roy. Oblivious symmetric alternation. In *STACS 2006*, pages 230–241, 2006.

[CR11] V. T. Chakaravarthy and S. Roy. Arthur and Merlin as oracles. *Comput. Complex.*, 20(3):505–558, 2011.

[ESY84] S. Even, A. L. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Inf. Control*, 61(2):159–173, 1984.

[FGM+89] M. Fürer, O. Goldreich, Y. Mansour, M. Sipser, and S. Zachos. On completeness and soundness in interactive proof systems. *Advances in Computing Research*, 5:429–442, 1989.

[FIKU08] L. Fortnow, R. Impagliazzo, V. Kabanets, and C. Umans. On the complexity of succinct zero-sum games. *Comput. Complex.*, 17(3):353–376, 2008.

[FSW09] L. Fortnow, R. Santhanam, and R. Williams. Fixed-polynomial size circuit bounds. In *CCC 2009*, pages 19–26, 2009.

[Gav95] R. Gavaldà. Bounding the complexity of advice functions. *J. Comput. Syst. Sci.*, 50(3):468–475, 1995.

[GGKN25] M. Gilboa, P. W. Goldberg, E. Koutsoupias, and N. Nisan. Complexity of unambiguous problems in Σ_2^P. arXiv:2510.19084, 2025 (version 3, 2026).

[GGLS26] K. Gajulapalli, S. Ghentiyala, Z. Li, and S. Saraogi. Downward self-reducibility in the total function polynomial hierarchy. In *SODA 2026*, pages 5117–5150, 2026. ECCC TR25-121.

[GLS26] S. Ghentiyala, Z. Li, and N. Stephens-Davidowitz. Range avoidance, Arthur–Merlin, and TFNP. In *STOC 2026*, pages 1925–1936, 2026. Revised version: ECCC TR25-210, revision 1.

[GLV24] K. Gajulapalli, Z. Li, and I. Volkovich. Oblivious complexity classes revisited: Lower bounds and hierarchies. In *FSTTCS 2024*, LIPIcs 323, pages 23:1–23:19, 2024.

[GM15] O. Goldreich and O. Meir. Input-oblivious proof systems and a uniform complexity perspective on P/poly. *ACM Trans. Comput. Theory*, 7(4):16:1–16:13, 2015.

[GS86] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *STOC 1986*, pages 59–68, 1986.

[GS88] J. Grollmann and A. L. Selman. Complexity measures for public-key cryptosystems. *SIAM J. Comput.*, 17(2):309–335, 1988.

[GW99] O. Goldreich and A. Wigderson. Improved derandomization of BPP using a hitting set generator. In *RANDOM-APPROX 1999*, LNCS 1671, pages 131–137, 1999.

[HHT97] Y. Han, L. A. Hemaspaandra, and T. Thierauf. Threshold computation and cryptographic security. *SIAM J. Comput.*, 26(1):59–78, 1997.

[HS15] E. A. Hirsch and D. Sokolov. On the probabilistic closure of the loose unambiguous hierarchy. *Inf. Process. Lett.*, 115(9):725–730, 2015.

[HV25] E. A. Hirsch and I. Volkovich. Upper and lower bounds for the linear ordering principle. ECCC TR25-142, 2025 (full version: revision 2, October 2026). Conference version in *STACS 2026*, LIPIcs 364, pages 52:1–52:19, 2026.

[HVW26] E. A. Hirsch, I. Volkovich, and R. Williams. Input-oblivious complexity classes in the second level of the polynomial hierarchy. ECCC TR26-250, revision 1, 2026.

[Käm91] J. Kämper. Nonuniform proof systems: a new framework to describe nonuniform and probabilistic complexity classes. *Theor. Comput. Sci.*, 85(2):305–331, 1991.

[Kan82] R. Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Inf. Control*, 55(1–3):40–56, 1982.

[KKMP21] R. Kleinberg, O. Korten, D. Mitropolsky, and C. Papadimitriou. Total functions in the polynomial hierarchy. In *ITCS 2021*, LIPIcs 185, pages 44:1–44:18, 2021.

[KL80] R. M. Karp and R. J. Lipton. Some connections between nonuniform and uniform complexity classes. In *STOC 1980*, pages 302–309, 1980. Journal version: Turing machines that take advice, *L'Enseignement Mathématique*, 28:191–209, 1982.

[Kor22] O. Korten. The hardest explicit construction. In *FOCS 2021*, pages 433–444. IEEE, 2022.

[KP24] O. Korten and T. Pitassi. Strong vs. weak range avoidance and the linear ordering principle. In *FOCS 2024*, pages 1388–1407, 2024. Full version: ECCC TR24-076.

[KS85] K.-I. Ko and U. Schöning. On circuit-size complexity and the low hierarchy in NP. *SIAM J. Comput.*, 14(1):41–51, 1985.

[KT94] J. Köbler and T. Thierauf. Complexity-restricted advice functions. *SIAM J. Comput.*, 23(2):261–275, 1994.

[KW98] J. Köbler and O. Watanabe. New collapse consequences of NP having small circuits. *SIAM J. Comput.*, 28(1):311–324, 1998.

[Li24] Z. Li. Symmetric exponential time requires near-maximum circuit size: Simplified, truly uniform. In *STOC 2024*, pages 2000–2007, 2024.

[MGD24] S. C. Marshall, C. Gyurik, and V. Dunjko. On bounded advice classes. arXiv:2405.18155, 2024.

[MPR26] D. Miloschewsky, S. Podder, and D. Rudolph. Promises should be taken seriously: On relativization with promise problems. arXiv:2609.07945, 2026.

[OS18] R. O'Donnell and A. C. C. Say. The weakness of CTC qubits and the power of approximate counting. *ACM Trans. Comput. Theory*, 10(2):5:1–5:22, 2018.

[RS98] A. Russell and R. Sundaram. Symmetric alternation captures BPP. *Comput. Complex.*, 7(2):152–162, 1998.

[RW26] H. Ren and R. Williams. Near-maximum circuit lower bounds for exponential time with Merlin–Arthur queries. To appear in *FOCS 2026*. ECCC TR26-118, 2026.

[SU06] R. Shaltiel and C. Umans. Pseudorandomness for approximate counting and sampling. *Comput. Complex.*, 15(4):298–341, 2006.

[SW14] R. Santhanam and R. Williams. On uniformity and circuit lower bounds. *Comput. Complex.*, 23(2):177–205, 2014.

[ZF87] S. Zachos and M. Fürer. Probabilistic quantifiers vs. distrustful adversaries. In *FSTTCS 1987*, LNCS 287, pages 443–455, 1987.

A Other routes to the collapses

The main text derives the collapse to **OMIN** (Corollary 3.11) from the collapse to $\mathbf{P}^{\mathbf{prMA}}$ of [CR11], and the collapse to **OM** (Theorem 4.6) from the collapses to **OMIN** and to $\mathbf{P}^{\mathbf{sOprMA}}$. This appendix gives other routes: through approximate counting instead of the collapse of [CR11] (Section A.1), through the collapse to $\mathbf{ZPP}^{\mathbf{NP}}$ [BCG+96, KW98], which avoids promise problems (Section A.2), and to **OM** without **OMIN** (Section A.3). Section A.4 discusses relativization.

A.1 Through approximate counting

Remark A.1. The containment $\mathbf{PH} \subseteq \mathbf{P}^{\mathbf{prMA}}$ under $\mathbf{NP} \subseteq \mathbf{P/poly}$, used in the proof of Corollary 3.11, can also be obtained from Theorem 3 instead of [CR11]. Under the hypothesis, $\mathbf{PH} = \mathbf{S}_2^{\mathbf{P}}$ by Sengupta's argument [Cai07], and $\mathbf{S}_2^{\mathbf{P}} \subseteq \mathbf{P}^{\mathrm{SSE}}$ by Theorem 3. Moreover, SSE has an Arthur–Merlin protocol [GS86], which can be made perfectly complete [FGM+89]. In it, Merlin's messages can be replaced by circuits for the search version of SAT, as in [AKSS95] and [HV25, Proposition 4.1]. As in the proof of Lemma 4.2, the correct circuits then dominate every other message on every input and every random string. This gives $\mathrm{SSE} \in \mathbf{sOprMA}$ under the hypothesis, and even $\mathbf{PH} \subseteq \mathbf{P}^{\mathrm{SSE}} \subseteq \mathbf{P}^{\mathbf{sOprMA}}$. Thus the collapse passes through Theorem 3, with a single fixed

oracle problem. In particular, Corollary 3.11, the collapse $\mathbf{PH} = \mathbf{P}^{\mathbf{sOprMA}}$ of Lemma 4.2, and hence Theorem 4.6 can be proved without the collapse of [CR11].

A.2 Without promise problems

Both routes to the collapse $\mathbf{PH} = \mathbf{P}^{\mathbf{sOprMA}}$ (through [CR11] and Lemma 4.2, or through Remark A.1) end with a promise oracle. By Lemma 4.1, $\mathbf{P}^{\mathbf{sOprMA}} \subseteq \mathbf{BPP}^{\mathbf{ONP}}$, so such an oracle can be traded for randomness and a language oracle. The collapse of Köbler and Watanabe [KW98] reaches a class of this kind directly, without promise problems: if $\mathbf{NP} \subseteq \mathbf{P/poly}$, then $\mathbf{PH} = \mathbf{ZPP}^{\mathbf{NP}} \subseteq \mathbf{BPP}^{\mathbf{NP}}$. It remains to derandomize $\mathbf{BPP}^{\mathbf{NP}}$ inside $\mathbf{L}_2^{\mathbf{P}}$, and (F3) makes this possible. Lemma A.2 is the case $A = \mathrm{SAT}$ of [HVW26, Lemma 6.12], which states $\mathbf{BPP}^A \subseteq \mathbf{P}^{A,\mathbf{OL}_2^{\mathbf{P}}}$ for $A \in \mathbf{P/poly}$, combined with Lemma 3.1; we include the short proof to keep the paper self-contained for Definition 2.7.

Lemma A.2 (cf. [HVW26, Lemma 6.12])**.** *If* $\mathbf{NP} \subseteq \mathbf{P/poly}$*, then* $\mathbf{BPP}^{\mathbf{NP}} \subseteq \mathbf{L}_2^{\mathbf{P}}$.

Proof. Let $L \in \mathbf{BPP}^{\mathrm{SAT}}$ via an oracle machine M running in time $t(n)$, so that $\Pr_r[M^{\mathrm{SAT}}(x,r) = L(x)] \geq 3/4$ for every x. By hypothesis, $\mathrm{SAT} \cap \{0,1\}^m$ has circuits of size at most $m^d + d$ for a constant d. Fix x of length n. Replacing the oracle gates of the natural SAT-oracle circuit for $r \mapsto M^{\mathrm{SAT}}(x,r)$ by circuits for SAT yields an ordinary circuit with $t(n)$ inputs and size at most $s(n)$, for a polynomial s independent of x. Let $N = \max(s(n), t(n), 8)$, and let $g_1, \ldots, g_m$ be the list of (F3) on 1^N, computed with oracle $B \in \mathbf{OL}_2^{\mathbf{P}}$, with $t(n)$-bit prefixes g_i'. Then $\Pr_i[M^{\mathrm{SAT}}(x, g_i') = L(x)] \geq 3/4 - 1/8 > 1/2$. So $L(x)$ is the majority of $M^{\mathrm{SAT}}(x, g_i')$, which is computable in $\mathbf{P}^{\mathrm{SAT},B} \subseteq \mathbf{L}_2^{\mathbf{P}}$ by (F1), (F2) and Lemma 3.1. The circuit for SAT is used only in the analysis. □

Corollary A.3 (cf. [HV25, Corollary 2])**.** *If* $\mathbf{NP} \subseteq \mathbf{P/poly}$*, then* $\mathbf{PH} = \mathbf{L}_2^{\mathbf{P}}$ *for Definition 2.7, and* $\mathbf{ONP} \subseteq \mathbf{OL}_2^{\mathbf{P}}$.

Proof. By [KW98] and Lemma A.2, $\mathbf{PH} = \mathbf{ZPP}^{\mathbf{NP}} \subseteq \mathbf{BPP}^{\mathbf{NP}} \subseteq \mathbf{L}_2^{\mathbf{P}} \subseteq \mathbf{PH}$. Hence, for every $L \in \mathbf{PH}$, the tally language T_L of Lemma 3.9 is in $\mathbf{L}_2^{\mathbf{P}}$, and $L \in \mathbf{P}^{\mathbf{TALLY} \cap \mathbf{L}_2^{\mathbf{P}}} = \mathbf{OL}_2^{\mathbf{P}}$ by Theorem 3.3. Thus $\mathbf{PH} = \mathbf{OL}_2^{\mathbf{P}}$; in particular, $\mathbf{ONP} \subseteq \mathbf{NP} \subseteq \mathbf{OL}_2^{\mathbf{P}}$. □

Remark A.4.

1. Corollary A.3 also follows from Corollary 3.11, but its proof is a second proof of Theorem 2. It uses neither the collapse of [CR11] nor Theorem 3.5: all its oracles are languages (SAT and the language B of (F3)), so the tally argument of Theorem 3.3 applies directly. The proof of Corollary 3.11 gives, in addition, $\mathbf{PH} = \mathbf{OGTD}[\mathbf{prMA}]$.

2. Unconditionally, it is open whether $\mathbf{ONP} \subseteq \mathbf{OL}_2^{\mathbf{P}}$ [HV25, HVW26]. By Corollary A.3, a negative answer would imply $\mathbf{NP} \not\subseteq \mathbf{P/poly}$. For $\mathbf{NP}$ in place of $\mathbf{ONP}$, the question is equivalent to $\mathbf{NP} \subseteq \mathbf{P/poly}$ (Corollary 3.13).

A.3 The collapse to OM without OMIN

The collapse to $\mathbf{OM}$ (Theorem 4.6) can also be proved without $\mathbf{OMIN}$, along the lines of the proof of the collapse to $\mathbf{ZPP}^{\mathbf{NP}}$ [BCG+96, KW98]. The advice is a stack of SAT-search circuits, one for each quantifier block, in which each circuit may use the previous ones. A failure of a circuit in the stack has a short witness, so the advice is certified in $\mathbf{co\text{-}NP}$, which equals $\mathbf{co\text{-}ONP}$ under the hypothesis. The generator finds the lexicographically first certified stack by a prefix search with a $\mathbf{\Sigma}_2^{\mathbf{P}}$ oracle, which is in $\mathbf{P}^{\mathbf{sOprMA}}$ under the hypothesis. This proof gives a unique generator output but not a unique accepted advice.

A.4 Relativization

Lemmas 3.1, 3.2, 3.8, 3.9, Proposition 2.11 and Theorem 3.10 relativize. Hence Corollary 3.11 relativizes whenever the cited ingredients (F3) and the collapse of [CR11] do; their proofs consist of circuit constructions that work for oracle circuits as well (one should, however, carefully recheck all the dependencies for being relativizable). For Theorem 2, (F3) suffices: its second proof (Corollary A.3) replaces the collapse of [CR11] by that of [KW98], whose proof relativizes, and Lemma A.2 relativizes whenever (F3) does. We do not use these facts.

B Further consequences

B.1 Derandomizing approximate counting

We say that *approximate counting is in* $\mathbf{FP^{NP}}$ if SSE is consistent with a language in $\mathbf{P^{NP}}$. Equivalently, given a circuit C and a rational $\varepsilon > 0$, a $(1+\varepsilon)$-approximation of $\#C$ is computable with an $\mathbf{NP}$ oracle in time polynomial in $|C|$ and $1/\varepsilon$ (Lemma 2.6).

Corollary B.1. *Suppose that approximate counting is in* $\mathbf{FP^{NP}}$. *Then:*

(i) $\mathbf{P^{prS_2^P}} = \mathbf{P^{NP}}$; *in particular,* $\mathbf{S_2^P} = \mathbf{P^{NP}}$;

(ii) $\mathbf{O_2^P} \subseteq \mathbf{OGTD[NP]} = \mathbf{P^{TALLY \cap P^{NP}}}$, *i.e., every language in* $\mathbf{O_2^P}$ *has polynomial-size circuits constructible in* $\mathbf{FP^{NP}}$;

(iii) *for every* k, *some language with polynomial-size circuits constructible in* $\mathbf{FP^{NP}}$ *has no circuits of size* n^k; *in particular,* $\mathbf{P^{NP}} \not\subseteq \mathbf{SIZE}[n^k]$;

(iv) *if moreover* $\mathbf{NP} \subseteq \mathbf{P/poly}$, *then* $\mathbf{PH} = \mathbf{OGTD[NP]} = \mathbf{P^{NP}}$.

Proof. Let $A \in \mathbf{P^{NP}}$ be consistent with SSE. Proposition 2.11(d), with M the identity machine, gives $\mathbf{P}^{\mathrm{SSE}} \subseteq \mathbf{P}^A \subseteq \mathbf{P^{NP}}$ and $\mathbf{OGTD}[\mathrm{SSE}] \subseteq \mathbf{OGTD}[A]$. By Proposition 2.11(b), $\mathbf{OGTD}[A] = \mathbf{P^{TALLY \cap P^A}} \subseteq \mathbf{P^{TALLY \cap P^{NP}}} = \mathbf{OGTD[NP]}$. Now (i) follows from Theorem 5.4 and $\mathbf{P^{NP}} \subseteq \mathbf{L_2^P} \subseteq \mathbf{S_2^P}$, and (ii) from Theorem 6.3. For (iii), $\mathbf{OL_2^P} \subseteq \mathbf{O_2^P} \subseteq \mathbf{OGTD[NP]}$ by (ii), and $\mathbf{OL_2^P} \not\subseteq \mathbf{SIZE}[n^k]$ by (F2). For (iv), $\mathbf{PH} = \mathbf{OL_2^P} \subseteq \mathbf{OGTD[NP]} \subseteq \mathbf{P^{NP}}$ by Corollary 3.11. □

Remark B.2. Shaltiel and Umans [SU06] observed that Cai's algorithm [Cai07] is a reduction to sampling. Hence it can be derandomized under a hardness assumption for $\mathbf{E^{NP}}$ that is implied by the one used to derandomize $\mathbf{AM}$, which gives $\mathbf{S_2^P} = \mathbf{P^{NP}}$ under this assumption [SU06, Theorem 4.9]. Item (i) of Corollary B.1 requires only the derandomization of approximate counting, and it also covers $\mathbf{P^{prS_2^P}}$. Conversely, Aydinlioğlu, Gutfreund, Hitchcock and Kawachi [AGHK11] showed that derandomizing approximate counting in this way implies exponential-size circuit lower bounds for $\mathbf{E^{NP}}$. Item (iii) can be viewed as a fixed-polynomial, "uniform-circuit" analogue of that implication.

B.2 Function classes, exponential time, and other classes inside $\mathbf{S_2^P}$

By Theorem 5.4, $\mathbf{FP^{prS_2^P}} \subseteq \mathbf{FP^{prSBP}}$. In particular, every search problem with a single-valued $\mathbf{FP^{prS_2^P}}$ algorithm, for instance a search problem whose output-bit language $\{(x,i) : \text{bit } i \text{ of the output on } x \text{ is } 1\}$ is in $\mathbf{S_2^P}$, has a single-valued $\mathbf{FP^{prSBP}}$ algorithm. By a standard padding argument, Theorem 3 scales up to exponential time: $\mathbf{S_2^E} \subseteq \mathbf{E^{prSBP}}$, where in $\mathbf{E^{prSBP}}$ the oracle queries may have length $2^{O(n)}$. Together with the near-maximum lower bound for

$\mathbf{S}_2^{\mathbf{E}}$ [CHR24, Li24] this gives $\mathbf{E}^{\mathbf{prSBP}} \not\subseteq \mathbf{SIZE}[2^n/n]$, which was proved in [HV25] via $\mathbf{L}_2^{\mathbf{E}} \subseteq \mathbf{E}^{\mathbf{prSBP}}$. Finally, Theorem 3 applies to every class known to lie in $\mathbf{S}_2^{\mathbf{P}}$. For instance, the classes **PTW** (Polynomial Tournament Winner), **PCW** $\subseteq$ **PTW** (Polynomial Condorcet Winner) and **PMA** (Polynomial Majority Argument) of unambiguous $\mathbf{\Sigma}_2^{\mathbf{P}}$ problems are contained in $\mathbf{S}_2^{\mathbf{P}}$ [GGKN25, Theorems 3.3, 4.2 and 5.3], and hence in $\mathbf{P}^{\mathbf{prSBP}}$. Since $\mathbf{L}_2^{\mathbf{P}} \subseteq \mathbf{PTW}$ [GGKN25, Theorem 3.4] (their $\mathbf{L}_2^{\mathbf{P}}$ is defined by linear orders, as ours), already the containment $\mathbf{PTW} \subseteq \mathbf{P}^{\mathbf{prSBP}}$ strengthens $\mathbf{L}_2^{\mathbf{P}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$ [HV25]. Similarly, approximating the value of a succinctly given zero-sum game within an additive error is complete for promise-$\mathbf{S}_2^{\mathbf{P}}$ [FIKU08]. Hence such approximations, obtained by binary search, can be computed in $\mathbf{FP}^{\mathbf{prSBP}}$.

B.3 Minimum finding

As a special case, the Goldreich–Wigderson lemma with approximate counting (Theorem 5.2) gives another minimum-finding algorithm, and thus another proof of LOP $\in \mathbf{FP}^{\mathbf{prSBP}}$ and $\mathbf{L}_2^{\mathbf{P}} \subseteq \mathbf{P}^{\mathbf{prSBP}}$ [HV25, Theorem 3.14].

Corollary B.3. *Given a circuit E that defines a strict linear order $\prec$ on $\{0,1\}^a$, the minimum of $\prec$ is computable in $\mathbf{FP}^{\mathrm{SSE}}$; the output is the same for every oracle consistent with* SSE. *Consequently,* LOP $\in \mathbf{FP}^{\mathbf{prSBP}}$.

Proof. Consider the matrix $V(u,w) = 1$ iff $u = w$ or $u \prec w$. Its minimum μ is an irrefutable row, so Corollary 5.3 gives R such that every column is beaten by some row of R. For the column μ, some $u \in R$ satisfies $u = \mu$ or $u \prec \mu$; hence $\mu \in R$, and μ is the $\prec$-least element of R, which we output. For LOP, given an arbitrary circuit E, first search for a violation of the axioms of strict linear orders (irreflexivity, totality, asymmetry, transitivity) using SAT $\in \mathbf{P}^{\mathrm{SSE}}$ and search-to-decision; if there is none, output the minimum. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il