Electronic Colloquium on Computational Complexity, Report No. 108 (2022)

Hardness Self-Amplification from Feasible Hard-Core Sets

Shuichi Hirahara * and Nobutaka Shimizu †

July 16, 2022

Abstract

We consider the question of *hardness self-amplification*: Given a Boolean function f that is hard to compute on a $o(1)$-fraction of inputs drawn from some distribution, can we prove that f is hard to compute on a $(\frac{1}{2} - o(1))$-fraction of inputs drawn from the same distribution? We prove hardness self-amplification results for natural distributional problems studied in fine-grained average-case complexity, such as the problem of counting the number of the triangles modulo 2 in a random tripartite graph and the online vector-matrix-vector multiplication problem over $\mathbb{F}_2$. More generally, we show that any problem that can be decomposed into "computationally disjoint" subsets of inputs admits hardness self-amplification. This is proved by generalizing the security proof of the Nisan–Wigderson pseudorandom generator, in which case *nearly disjoint* subsets of inputs are considered.

At the core of our proof techniques is a new notion of *feasible hard-core set*, which generalizes Impagliazzo's hard-core set [Impagliazzo, FOCS'95]. We show that any weak average-case hard function f has a *feasible hard-core set* H: any small H-oracle circuit (that is allowed to make queries q to H if $f(q)$ can be computed without the oracle) fails to compute f on a $(\frac{1}{2} - o(1))$-fraction of inputs in H.

*National Institute of Informatics. Email: s_hirahara@nii.ac.jp

†Tokyo Institute of Technology Email: shimizu.n.ah@m.titech.ac.jp

ISSN 1433-8092

Contents

1 Introduction

Average-case complexity quantifies the hardness of a function in terms of the difficulty of evaluating it on a certain fraction of inputs. Depending on the fraction of hard inputs, we obtain two different notions, *weak* average-case hardness and *strong* average-case hardness: A function f is said to be *weakly average-case hard* if any efficient algorithm fails to solve f on a δ-fraction of inputs, where $\delta > 0$ is a small parameter. A Boolean function f is said to be *strongly average-case hard* if any efficient algorithm fails to solve f on a $(1/2-\epsilon)$-fraction of inputs for a small $\epsilon > 0$. The average-case complexity of a function depends greatly on whether we use strong or weak average-case notions, as any *biased* function cannot be strongly average-case hard.[1] The theory of average-case complexity [BT06] would become very robust if there is a general proof technique that connects weak average-case hardness of f and strong average-case hardness of f, ideally for natural problems f of practical interest.

There are general proof techniques, called *hardness amplification*, that transform any weakly average-case hard function f into a strongly average-case hard function g. For example, Yao's XOR lemma [Yao82; GNW11] states that if a Boolean function f cannot be computed by small circuits on a δ-fraction of inputs, then the function $g := f^{\oplus k}$ defined as $f^{\oplus k}(x_1, \dots, x_k) := f(x_1) \oplus \cdots \oplus f(x_k)$ cannot be computed by small circuits on a $(\frac{1}{2}-\epsilon)$-fraction of inputs for small parameters δ and $\epsilon > 0$. Hardness amplification theorems have had fundamental impacts on the theory of computation, especially on cryptography [Yao82] and derandomization [Imp95; IW97]. The proof techniques developed in these lines of research are geared to obtaining better parameters (e.g., trade-offs between the parameters k, ϵ, and δ [GNW11], a small input length of g [Imp95; IW97], small advice complexity [IJKW10], the monotonicity of g with respect to f [ODo04; Tre05; HVV06]); consequently, hardness-amplified functions g tend to be highly artificial.

In this paper, we broaden the applicability of the proof techniques of hardness amplification, by developing a general framework that enables us to show hardness amplification for *natural* problems f and g over *natural* distributions over instances of f and g. Our framework, in fact, allows us to show *hardness self-amplification*: The problem f and the hardness-amplified problem g (as well as the input distributions) are identical. We prove that several natural problems studied in fine-grained complexity admit hardness self-amplification.

1.1 Our Results

Before presenting our general framework, we provide examples of hardness self-amplification results that follow from the framework, while reviewing literature. For a finite set S, we write $x \sim S$ to denote that x is sampled uniformly at random from S. Throughout this paper, we use circuits as a computational model.[2]

Parity of Triangles (Shown in Section 8). We consider the problem of counting the number of triangles modulo 2 in a random tripartite graph. A graph $G = (V_1 \cup V_2 \cup V_3, E)$ is *tripartite* if every edge $e \in E$ lies between V_i and V_j for $i \neq j$. A triple of vertices v_1, v_2, v_3 forms a partite triangle in G if $\{v_i, v_j\} \in E$ for every $i \neq j$ and $v_i \in V_i$ for every $i \in [3]$.[3] Suppose $n = |V_1| = |V_2| = |V_3|$. For every $n \in \mathbb{N}$, let $\mathsf{TriParity}_n \colon \{0,1\}^{3n^2} \to \{0,1\}$ be the parity of the number of partite triangles

[1] We say that a function $f \colon \{0,1\}^n \to \{0,1\}$ is *biased* if $\Pr_{x \sim \{0,1\}}[f(x) = 1] \not\approx \frac{1}{2}$. Otherwise, f is said to be *nearly balanced*.

[2] It is important for our results that a computational model is non-uniform. Whether our results can be extended to uniform computational models is an important open question.

[3] We write $[m] := \{1, \dots, m\}$ for $m \in \mathbb{N}$.

contained in the input graph, formally defined as

$$\mathsf{TriParity}_n(x) := \bigoplus_{\substack{v_1 \in V_1 \\ v_2 \in V_2 \\ v_3 \in V_3}} \prod_{1 \le i < j \le 3} x[v_i, v_j], \tag{1}$$

where we identify the input $x \in \{0,1\}^{3n^2}$ with the tripartite graph by regarding x as the edge indicator. It is not hard to observe that $\mathsf{TriParity}_n$ can be computed by an $O(n^\omega)$-time algorithm where $\omega < 2.37286$ is the square matrix multiplication exponent [AW21]. The running time of this algorithm is deemed to be optimal: The *k-clique hypothesis* [LWW18] asserts that detecting a k-clique in a given n-vertex graph requires $n^{\omega k/3 - o(1)}$ time in worst-case. This hypothesis implies that computing $\mathsf{TriParity}_n$ requires $n^{\omega - o(1)}$ time in worst-case, as the k-clique subgraph detection problem can be efficiently reduced to the problem of computing the parity of the number of k-clique subgraphs in the worst case (see, e.g., [BBB21, Lemma A.1]). We consider the average-case complexity of computing $\mathsf{TriParity}_n(x)$ where the input $x \sim \{0,1\}^{3n^2}$ is drawn uniformly at random. In other words, the input is a random tripartite graph where all possible edges occur with probability $1/2$ independently.

Worst-case-to-average-case reductions for subgraph counting prblems on Erdős–Rényi random graphs and its variants have recently attracted much attention [GR18; BBB21; Gol20; DLW20; HS21]. Boix-Adserá, Brennan, and Bresler [BBB21] showed that if there is a $T(n)$-time algorithm that counts k-clique subgraphs in a $(1 - 1/\operatorname{polylog}(n))$-fraction of Erdős–Rényi random graphs, then there is a $T(n) \cdot \operatorname{polylog}(n)$-time randomized algorithm that counts k-cliques in every n-vertex graph. Goldreich [Gol20] presented a simplified reduction of [BBB21] in the case of counting k-cliques modulo 2: If there is a $T(n)$-time algorithm that computes the parity of k-cliques in a $(1 - 2^{-k^2})$-fraction of graphs, then there is a randomized $O(T(n))$-time algorithm that computes the parity of k-cliques in every graph. An important open question, raised in [BBB21; Gol20], is to improve the *error tolerance* of the worst-case-to-average-case reductions; for example, the reduction of [Gol20] can tolerate an error 2^{-k^2} of an average-case solver.

Why is it difficult to make the reductions error-tolerant? Goldreich [Gol20] noted that it is non-trivial to prove even the fact that the parity of the number of k-cliques in an Erdős–Rényi random graph is nearly balanced. Although this fact follows from a general result of Kolaitis and Kopparty [KK13], any connection from worst-case hardness to strong average-case hardness for a function f must prove the property of being balanced implicitly: If a function f is not *nearly balanced* (i.e., the probability that $f(x) = 1$ over a random input x is not close to $1/2$), then a trivial algorithm that always outputs either 0 or 1 succeeds with probability $\gg 1/2$, which can be combined with an error-tolerant worst-case-to-average-case reduction for f. Thus, any worst-case-to-average-case reduction for f that can tolerant an error $\approx 1/2$ can be seen as a "computational proof" of the fact that the function f is nearly balanced. The fact that $\mathsf{TriParity}_n(x)$ is 1 with probability $1/2 \pm 2^{-\Omega(n)}$ over a uniformly random x follows from the "statistical" result of [KK13].[4]

As an application of our general framework, we prove that $\mathsf{TriParity}_n$ admits hardness self-amplification: If there is a circuit of size s that computes $\mathsf{TriParity}_n$ on a $\left(\frac{1}{2} + \delta\right)$-fraction of inputs, then there is a circuit of size $O(s + n^2)$ that computes $\mathsf{TriParity}_n$ on a $(1 - \epsilon)$-fraction of inputs, where $\delta, \epsilon > 0$ are arbitrary small constants. By combining this with [BBB21; Gol20], we obtain the following error-tolerant worst-case-to-average-case reduction for $\mathsf{TriParity}_n$.

Theorem 1.1. *For any constant $\delta > 0$, if there exists a circuit C of size s satisfying*

$$\Pr_{x \sim \{0,1\}^{3n^2}}[C(x) = \mathsf{TriParity}_n(x)] \ge \frac{1}{2} + \delta,$$

[4] For completeness, we present a proof in Appendix A.

then there exists a randomized circuit C' of size $O(n^2+s)$ satisfying

$$\Pr_{C'}[C'(x) = \mathsf{TriParity}_n(x)] \geq \frac{2}{3}$$

for every $x \in \{0,1\}^{3n^2}$.

Note that the success probability $1/2+\delta$ in Theorem 1.1 is nearly optimal since a random guess achieves the success probability $1/2$. In other words, Theorem 1.1 indicates that one cannot do better than the random guess unless the parity of triangle subgraphs can be efficiently solved in the worst case. The success probability $2/3$ in the conclusion of Theorem 1.1 can be amplified to any constant less than 1, as the success probability of any randomized algorithm can be easily amplified by repetition.

Online Vector-Matrix-Vector Problem (Shown in Section 9). In the *Online Vector-Matrix-Vector Multiplication* (OuMv) introduced by Henzinger, Krinninger, Nanongkai, and Saranurak [HKNS15], we are initially given a matrix M and each query consists of a pair of vectors u_i, v_i. Our task is to compute $u_i^\top M v_i$ one by one, where the multiplications are over Boolean semiring. It is conjectured that solving OuMv for n queries requries $n^{3-o(1)}$ time. Henzinger et al. [HKNS15] established fine-grained complexity of several dynamic problems based on this conjecture. OuMv has been well studied in the context of fine-grained complexity of dynamic problems. Several lower bounds for OuMv and related problems have been known for *cell-probe* model [CKLM18; CKL18; LW17] in which the computational costs are measured by the size of the data structure and the number of cells probed by the algorithm answering a given query (in other words, we do not care the cost of the time for constructing the data structure and for answering the query).

Here, we focus on OuMv over $\mathbb{F}_2$ on a uniformly random matrix and vectors. There is a randomized reduction from OuMv over Boolean semiring to OuMv over $\mathbb{F}_2$, which exploits the well-known isolation technique [HLS22, Lemma 2.1]. We identify the OuMv problem over $\mathbb{F}_2$ with the function $\mathsf{OuMv}_n\colon \{0,1\}^{n\times n}\times\{0,1\}^{2n}\to\{0,1\}$ defined as

$$\mathsf{OuMv}_n(M,u,v) = u^\top M v, \tag{2}$$

where the operations are over $\mathbb{F}_2$. Here, the matrix M is given at the preprocess stage and a pair of vectors u, v are given as a query.

Henzinger, Lincoln, and Saha [HLS22] presented a uniform worst-case-to-average-case reduction for OuMv over $\mathbb{F}_2$: If an average-case data structure computes OuMv_n for $(1-\epsilon)$-fraction of inputs, then there is a randomized worst-case data structure that computes OuMv_n with probability $1-8\epsilon$ for all inputs.

Very recently, Asadi, Golovnev, Gur, and Shinkar [AGGS22] developed a different framework based on additive combinatorics and proved that the Online Matrix-Vector Multiplication problem (OMv), which is closely related to OuMv, admits an error-tolerant worst-case-to-average-case reduction for uniform computational model. Their framework is applicable to multi-output problems, but may not be applicable to decision problems, such as OuMv.[5] Our framework is applicable to decision problems and enables us to present such a reduction for OuMv in a nonuniform model of circuits.

Specifically, we consider the following nonuniform model for static data structure problems: Let $\mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ be the class of circuits consists of preprocess part of size s_{pre}, data structure of length

[5]The framework of [AGGS22] relies on the fact that OMv admits an efficient verifier that checks the correctness of the computation. Such a verifier can be constructed for multi-output problems, such as the matrix multiplication and OMv.

ℓ, and query-answer part of size s_{ans}. More formally, $C \in \mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ is a circuit that can be written as $C(x;q) = C_{\mathrm{ans}}(C_{\mathrm{pre}}(x), q)$ for a preprocess circuit $C_{\mathrm{pre}}\colon \{0,1\}^m \to \{0,1\}^\ell$ of size s_{pre} given $x \in \{0,1\}^m$ as a static data and a query-answer circuit $C_{\mathrm{ans}}\colon \{0,1\}^\ell \times \{0,1\}^n \to \{0,1\}$ of size s_{ans} given $q \in \{0,1\}^n$ as a query. Note that ℓ corresponds to the size of the data structure.

Theorem 1.2. *For any constant $\delta > 0$, if there exists a circuit $C \in \mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ satisfying $\Pr_{(M,u,v)\sim\{0,1\}^{n^2+2n}}[C(M;u,v) = \mathsf{OuMv}_n(M,u,v)] \geq 1/2+\delta$, then there exists a randomized circuit $C' \in \mathcal{C}(O(s_{\mathrm{pre}}), O(\ell), O(n+s_{\mathrm{ans}}))$ satisfying $\Pr_{C'}[C'(M;u,v) = \mathsf{OuMv}_n(M,u,v)] \geq 2/3$ for every $(M,u,v) \in \{0,1\}^{n\times n} \times \{0,1\}^{2n}$.*

Generalized OuMv (Shown in Section 9). Jin and Xu [JX22] introduced the OuMv_k hypothesis, which is an extension of the OuMv hypothesis to rank-k tensors. In OuMv_k, we are given a rank-k tensor $M \in \{0,1\}^{n^k}$ at the preprocessing stage. Each query consists of k vectors $x_1, \ldots, x_k \in \{0,1\}^k$. Our task is to compute

$$\mathsf{OuMv}_n^{(k)}(M, x_1, \ldots, x_k) := \sum_{i_1\in[n],\ldots,i_k\in[n]} M(i_1,\ldots,i_k)x_1(i_1)\cdots x_k(i_k), \tag{3}$$

where the operations are over $\mathbb{F}_2$. Note that $\mathsf{OuMv}_n = \mathsf{OuMv}_n^{(k)}$ for $k = 2$. Jin and Xu [JX22] considered OuMv_k over Boolean semiring and presented various lower bounds based on the conjecture that OuMv_k for n queries requires $n^{1+k-o(1)}$ time.

In this paper, we consider OuMv_k over $\mathbb{F}_2$. It is easy to see that OuMv_k over Boolean semiring can be randomizedly reduced to OuMv_k over $\mathbb{F}_2$ by the usual isolating technique as the special case of $k = 2$ is shown in [HLS22, Lemma 2.1]. Our technique for proving Theorem 1.2 can be extended to derive the following result.

Theorem 1.3. *For any constant $\delta > 0$, if there exists a circuit $C \in \mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ satisfying $\Pr_{(M,u_1,\ldots,u_k)\sim\{0,1\}^{n^k+kn}}[C(M;u_1,\ldots,u_k) = \mathsf{OuMv}_n^{(k)}(M,u_1,\ldots,u_k)] \geq 1/2+\delta$, then there exists a randomized circuit $C' \in \mathcal{C}(O(s_{\mathrm{pre}}), O(\ell), O(n^{k-1}+s_{\mathrm{ans}}))$ satisfying $\Pr_{C'}[C'(M;u_1,\ldots,u_k) = \mathsf{OuMv}_n^{(k)}(M,u_1,\ldots,u_k)] \geq 2/3$ for every $(M,u_1,\ldots,u_k) \in \{0,1\}^{n^k} \times \{0,1\}^{kn}$.*

1.2 A General Framework of Hardness (Self-)Amplification

More generally, we show that any problem that can be written as the sum of "computationally disjoint" subsets of inputs admits hardness self-amplification. Specifically, for an input $x \in \{0,1\}^n$ and a subset $S \subseteq [n]$, let $x|_S \in \{0,1\}^{|S|}$ denote the string obtained by concatenating all the bits of x indexed by S. Let $g\colon \{0,1\}^n \to \{0,1\}$ be a function that can be written as

$$g(x) = \bigoplus_{i=1}^{k} f(x|_{S_i}) \tag{4}$$

for some function $f\colon \{0,1\}^\ell \to \{0,1\}$ and for some subsets $S_1, \ldots, S_k$ of size ℓ. For a size parameter $s \in \mathbb{N}$, we say that a family of subsets $S_1, \ldots, S_k$ is a *s-computational design on f* if for every distinct pair $i \neq j \in [k]$ and for every $x|_{S_i\setminus S_j} \in \{0,1\}^{|S_i\setminus S_j|}$, there exists a circuit of size s that computes the function f restricted by $x|_{S_i\setminus S_j}$, i.e., one that maps $x|_{S_i\cap S_j}$ to $f(x)$. Our general framework of hardness amplification can be stated as follows.

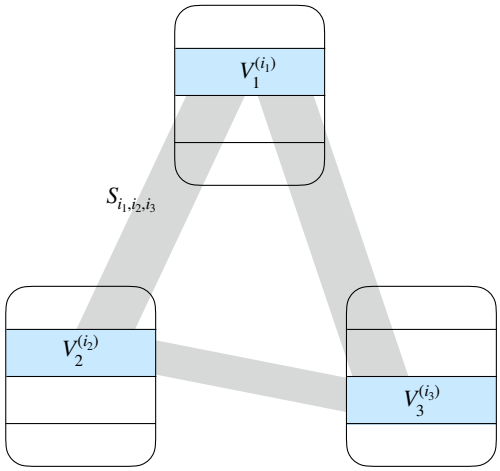

Figure 1: The indices $i_1, i_2, i_3 \in [a]$ specify three vertex sets $V_1^{(i_1)}, V_2^{(i_2)}$, and $V_3^{(i_3)}$

Theorem 1.4 (see also Theorem 7.1)**.** *Assume that g can be written as (4) for some s'-computational design $S_1, \ldots, S_k$ on f such that there are at least k' disjoint subsets in $S_1, \ldots, S_k$. If any circuit of size s fails to compute f on a δ-fraction of inputs, then any circuit of size s' fails to compute g on a $(1/2 - \epsilon)$-fraction of inputs, where*

$$\epsilon = \exp\left(-\delta k'\right) \quad \text{and} \quad s = s' \cdot k^{O(k^2/\epsilon^2)}.$$

This result generalizes Yao's XOR lemma, which corresponds to the case when $S_1, \ldots, S_k$ are disjoint subsets. Our hardness amplification theorem generalizes Yao's XOR lemma to highly correlated inputs. For example, Theorem 1.1 is proved by applying it to

$$\mathsf{TriParity}_n(x) = \bigoplus_{m \in [a]^3} \mathsf{TriParity}_{n/a}(x|_{S_m}) \tag{5}$$

for some family $\{S_m\}_{m \in [a]^3}$ of subsets. More specifically, we partition each part V_i of a tripartite graph $G = (V_1 \cup V_2 \cup V_3, E)$ into a disjoint subsets $V_i^{(1)}, \ldots, V_i^{(a)}$ of size n/a. For each $m = (i_1, i_2, i_3) \in [a]^3$, we define $S_{i_1,i_2,i_3} \subseteq [3n^2]$ to be the indices of the edges in $V_1^{(i_1)} \cup V_2^{(i_2)} \cup V_3^{(i_3)}$; see Figure 1. Although the family of the subsets is far from being disjoint (for example, $|S_{1,1,1} \cap S_{1,1,2}| = n^2$), we observe that $\{S_m\}_{m \in [a]^3}$ is an $O(n^2)$-computational design on $\mathsf{TriParity}_{n/a}$: For any fixed $x|_{S_i \setminus S_j}$, the function that takes $x|_{S_i \cap S_j}$ as input and outputs $\mathsf{TriParity}_{n/a}(x)$ is a linear function, which can be clearly computed by linear-sized circuits. Using this "computational disjointness", Theorem 1.4 shows that weak average-case hardness of $\mathsf{TriParity}_{n/a}$ can be amplified to strong average-case hardness of $\mathsf{TriParity}_n$.

Our proof techniques build on and generalize those developed in the line of research on *derandomized hardness amplification theorems*. A celebrated theorem of Impagliazzo and Wigderson [IW97], which is a culmination of [Yao82; BM84; NW94; BFNW93; Imp95], states that $\mathsf{P} = \mathsf{BPP}$ if $\mathsf{E} = \mathsf{DTIME}(2^{O(n)})$ cannot be computed by circuits of size $2^{o(n)}$. The key technical contribution of [IW97] is to prove a derandomized version of Yao's XOR lemma, which states that if any circuit of size $2^{o(n)}$ fails to compute a function $f\colon \{0,1\}^n \to \{0,1\}$ on a $(1/3)$-fraction of inputs, then there exists a function $g\colon \{0,1\}^{O(n)} \to \{0,1\}$ such that any circuit of size $2^{o(n)}$ fails to compute g on a

$(1/2 - 2^{-o(n)})$-fraction of inputs. Here, the function g is defined as

$$g(x, y) = \bigoplus_{m=1}^{k} f(x|_{S_m} \oplus G(y))$$

for some family of subsets $S_m \subseteq [O(n)]$ and for some function G.[6] In the literature of derandomized hardness amplification, whether g is natural or not is not important; the only requirement is that $g \in \mathsf{E}$ if $f \in \mathsf{E}$. Thus, the family $\{S_m\}_{m\in[k]}$ can be chosen to be nearly disjoint subsets of $[O(n)]$. We say that a family $\{S_m\}_{m\in[k]}$ of subsets is an *s-combinatorial design* if $|S_i \cap S_j| \leq \log s$ for every distinct pair $i \neq j \in [k]$. Nisan and Wigderson [NW94] showed that there exists a $2^{o(n)}$-combinatorial design that can be efficiently computed. In general, for an s-combinatorial design $\{S_m\}_{m\in[k]}$, Impagliazzo and Wigderson [IW97] showed that if f is weakly average-case hard for circuits of size $s^{O(1)} \cdot n^{O(1)}$, then g is strongly average-case hard for circuits of size s. Theorem 1.4 generalizes such derandomized hardness amplification theorems:[7] whereas previous results are applicable to *s-combinatorial designs*, our results are applicable to *s-computational designs* on f. Note that any s-combinatorial design is also an s-computational design on f for every function f because any function on $\log s$-bit inputs can be computed by circuits of size $O(2^{\log s}/\log s) \leq s$; moreover, the converse is not true, as the family of subsets in (5) is an example of an $O(n^2)$-computational design on $\mathsf{TriParity}_{n/a}$ that is not a 2^{n^2-1}-combinatorial design.

2 Proof Overview

We now present proof ideas of Theorem 1.4, which generalizes derandomized XOR lemmas from *combinatorial* designs to *computational* designs. At the core of our proofs is a new notion of feasible hard-core set, which generalizes Impagliazzo's hard-core set [Imp95]. In Section 2.1, we review the proof techniques of derandomized hardness amplification. In Section 2.2, we introduce the notion of feasible hard-core set. In Section 2.3, we outline the proof of the existence of a feasible hard-core set for any mildly average-case-hard function.

2.1 A Review of Derandomized Hardness Amplification

Our starting point is an elegant exposition of [IW97] presented by Healy, Vadhan, and Viola [HVV06], which is based on the literature on hardness amplification within NP [ODo04; Tre03]. Let $f\colon \{0,1\}^n \to \{0,1\}$, $g\colon \{0,1\}^m \to \{0,1\}$, and $\sigma\colon \{0,1\}^k \to \{0,1\}$ be functions and $S_1, \ldots, S_k \subseteq [m]$ be subsets of size n. Suppose that for every $z \in \{0,1\}^m$,

$$g(z) = \sigma(f(z|_{S_1}), \ldots, f(z|_{S_k})).$$

Note that (4) corresponds to the case of $\sigma(y_1, \ldots, y_k) := \bigoplus_{i=1}^{k} y_i$. Hardness amplification can be proved for every function σ whose expected bias is small; thus, we present such a general proof here. We first assume that the family $\{S_i\}_{i\in[k]}$ is an *s-combinatorial design* for a small parameter $s \in \mathbb{N}$ and outline a standard proof of derandomized hardness amplification.

Impagliazzo [Imp95] showed that any mildly average-case hard function f has a dense *hard-core set* H on which f is strongly average-case hard. Specifically, we say that f is δ-hard on $H \subseteq \{0,1\}^n$

[6] Theorem 1.4 requires the technical condition that there are k' disjoint subsets in $S_1, \ldots, S_k$. This condition is circumvented in [IW97] because of the usage of the function G.

[7] We note that the dependence of the size parameters s and s' in Theorem 1.4 is significantly worse than [IW97]. Improving this is an interesting open question.

if any small circuit fails to compute f on a δ-fraction of inputs in H. *Impagliazzo's hard-core lemma* states that if f is δ-hard on $\{0,1\}^n$, then there exists a "hard-core set" H of size $\delta 2^n$ such that f is $(1/2-\epsilon)$-hard on H.

Let $f\colon \{0,1\}^n \to \{0,1\}$ be a δ-hard function on $\{0,1\}^n$. Let f_H denote a random function such that $f_H(x) = f(x)$ if $x \notin H$ and $f_H(x)$ is a uniformly random bit if $x \in H$. Intuitively, f_H can be thought as an idealized version of f which is information-theoretically hard on inputs in H. Impagliazzo's hard-core lemma implies the existence of a large set H such that $(x, f(x))$ and $(x, f_H(x))$ are computationally indistinguishable, i.e.,

$$(x, f(x)) \approx_c (x, f_H(x)), \tag{6}$$

where $x \sim \{0,1\}^n$. Here, for random variables X and Y, we denote by $X \approx_c Y$ that X and Y are computationally indistinguishable by small circuits; i.e., for any small circuit C,

$$|\Pr[C(X) = 1] - \Pr[C(Y) = 1]| \leq \epsilon$$

for a small parameter $\epsilon > 0$.

Using a hybrid argument of Nisan and Wigderson [NW94], (6) implies

$$(z, f(z|_{S_1}), \ldots, f(z|_{S_k})) \approx_c (z, f_H(z|_{S_1}), \ldots, f_H(z|_{S_k})) \tag{7}$$

for $z \sim \{0,1\}^m$. Specifically, the contrapositive can be proved as follows. Assume that (7) does not hold. Then, by a triangle inequality, there exists some index $i \in [k]$ such that

$$(z, f(z|_{S_1}), \ldots, f(z|_{S_i}), \ldots, f_H(z|_{S_k})) \not\approx_c (z, f(z|_{S_1}), \ldots, f_H(z|_{S_i}), \ldots, f_H(z|_{S_k})). \tag{8}$$

By an averaging argument, $z|_{[m]\setminus S_i}$ and the randomness in $f_H(z|_{S_{i+1}}), \ldots, f_H(z|_{S_k})$ can be fixed. Now, we use the assumption that $\{S_i\}_{i\in[k]}$ is an s-combinatorial design, i.e., $|S_i \cap S_j| \leq \log s$ for every distinct pair (i,j). Since $z|_{[m]\setminus S_i}$ is fixed, given $x := z|_{S_i}$ as input, the output

$$(f(z|_{S_1}), \ldots, f(z|_{S_{i-1}}), f_H(z|_{S_{i+1}}), \ldots, f_H(z|_{S_k}))$$

can be computed by a circuit of size $O(k \cdot 2^{\log s}/\log s) \leq O(ks)$, as any function on $\log s$-bit inputs can be computed by a circuit of size $O(2^{\log s}/\log s)$. By combining this circuit with the circuit that distinguishes the two distributions of (8), we obtain a circuit witnessing $(x, f(x)) \not\approx_c (x, f_H(x))$. This completes a proof sketch of the contrapositive of (6) $\Rightarrow$ (7). We mention in passing that Nisan and Wigderson [NW94] used the same hybrid argument to construct a pseudorandom generator based on a strongly average-case hard function.

By (7), we obtain

$$(z, g(z)) = (z, \sigma(f(z|_{S_1}), \ldots, f(z|_{S_k}))) \approx_c (z, \sigma(f_H(z|_{S_1}), \ldots, f_H(z|_{S_k})))$$

because applying an efficiently computable function σ preserves computational indistinguishability. Finally, we argue that $(z, \sigma(f_H(z|_{S_1}), \ldots, f_H(z|_{S_k})))$ is statistically close to (z, b), where $z \sim \{0,1\}^m$ and $b \sim \{0,1\}$. For simplicity, assume that $\sigma(y_1, \ldots, y_k) = \bigoplus_{i=1}^k y_i$, in which case if $z|_{S_i} \in H$ for some $i \in [k]$, the output of σ is completely uniform. Using the assumption of Theorem 1.4 that there are k' disjoint subsets in $S_1, \ldots, S_k$, the probability that there exists no i such that $z|_{S_i} \in H$ is bounded by

$$(1-\delta)^{k'} \leq \exp(-\delta k').$$

Thus, the statistical distance between $(z, \sigma(f_H(z|_{S_1}), \ldots, f_H(z|_{S_k})))$ and $(z, b) \sim \{0,1\}^n \times \{0,1\}$ is at most $\exp(-\delta k')$, which is small. We conclude that

$$(z, g(z)) \approx_c (z, b),$$

which is equivalent to saying that any small circuit C satisifes

$$\Pr_{z \sim \{0,1\}^m}[C(z) = g(z)] \leq \frac{1}{2} + \epsilon$$

for a small parameter $\epsilon > 0$. This completes the proof sketch of a derandomized hardness amplification theorem.

2.2 An Extension to Computational Designs

We now wish to extend the proof sketch presented above to the case that $\{S_i\}_{i \in [k]}$ is an *s-computational design* on f. In this case, it is possible that $|S_i \cap S_j| \geq s$ for some distinct pair (i, j); thus, the proof of (6) $\Rightarrow$ (7) may blow up the circuit size exponentially in s. We need to revise this proof in order to extend the proof for s-combinatorial designs to s-computational designs.

A First Attempt. Let us assume for a moment that H is *efficiently recognizable*, i.e., the characteristic function of H can be computed by a small circuit of size s. In this case, it is easy to observe that the proof of (6) $\Rightarrow$ (7) is applicable to an s-computational design on f. Indeed, fix $z|_{[m] \setminus S_i}$ and the randomness in $f_H(z|_{S_{i+1}}), \ldots, f_H(z|_{S_k})$. Consider a function that takes $x = z|_{S_i}$ as input and outputs

$$(f(z|_{S_1}), \ldots, f(z|_{S_{i-1}}), f_H(z|_{S_{i+1}}), \ldots, f_H(z|_{S_k})).$$

We claim that this function can be computed by a circuit of size $O(sk)$. It is evident that $(f(z|_{S_1}), \ldots, f(z|_{S_{i-1}}))$ can be computed by a circuit of size $O(sk)$ because of the definition of an s-computational design. In order to compute $f_H(z|_{S_j})$ for every $j > i$, we use the assumption that the characteristic function of H can be computed by a circuit of size s. By the definition of f_H, the output of $f_H(z|_{S_j})$ is defined to be $f(z|_{S_j})$ if $z|_{S_j} \notin H$ and to be a uniformly random bit if $z|_{S_j} \in H$. This can be computed by a circuit of size $O(s)$ if H is efficiently recognizable.

Can we justify the assumption that a hard-core set H is efficiently recognizable? Unfortunately, Reingold, Trevisan, Tulsiani, and Vadhan [RTTV08] gave the following simple counterexample: Let $f\colon \{0,1\}^n \to \{0,1\}$ be a random function that outputs $f(x) = 1$ with probability $1-\delta$ independently for every $x \in \{0,1\}^n$. This function is δ-hard for any small (say, polynomial-size) circuit. For any set $H \subseteq \{0,1\}^n$ with $|H| \geq \delta 2^n$, we have $|H \cap f^{-1}(1)|/|H| \approx 1 - \delta$ with probability $2^{-\Theta(2^n)}$ by the Hoeffding inequality. By taking the union bound over all efficiently recognizable H (e.g., whose characteristic function can be computed by polynomial-size circuits), we obtain a function f such that f is δ-hard and f is biased on every efficiently recognizable H. This counterexample shows that the function f does not have any efficiently recognizable hard-core set.

A Second Attempt. The counterexample above exploits the definition of a hard-core set H such that a function f must be nearly balanced on H (i.e., $|H \cap f^{-1}(0)| \approx |H \cap f^{-1}(1)| \approx |H|/2$). Reingold et al. [RTTV08] proved the existence of an efficiently recognizable hardcore set by relaxing the condition of being balanced.

Theorem 2.1 (Restatement of Theroem 3.1 of [RTTV08]). *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a δ-hard function on $\{0,1\}^n$ for circuits of size s. Then, there exists $H \subseteq \{0,1\}^n$ with $|H| \geq \delta 2^n$ such that*

$$\Pr_{x\sim H}[C(x) = f(x)] \leq \max\left\{\frac{|H \cap f^{-1}(0)|}{|H|}, \frac{|H \cap f^{-1}(1)|}{|H|}\right\} + \epsilon$$

for any circuit of size $s' = O(\delta^2 \epsilon s)$ and $\delta/2 \leq |H \cap f^{-1}(0)|/|H| \leq 1 - \delta/2$. Moreover, there exists a circuit of size s that can decide whether $x \in H$ or not on input $x \in \{0,1\}^n$.

Can we use this theorem for our purpose? Unfortunately, this approach does not work for the following two reasons.

1. Since f can be biased on the hard-core set H of Theorem 2.1, it does not necessarily imply that $(x, f(x)) \approx_c (x, f_H(x))$.
2. The circuit size s' is always smaller than the size s of the circuit that decides H. In contrast, to prove (6) $\Rightarrow$ (7), we need to show that if $(z, f(z)) \approx_c (z, f_H(z))$ for circuits of size s', then $(z, f(z|_{S_1}), \ldots, f(z|_{S_k})) \approx_c (z, f_H(z|_{S_1}), \ldots, f_H(z|_{S_k}))$ for circuits of size s'', where we would like to maximize s''. Using that H is decidable by a circuit of size s, it can be shown that if $(x, f(x)) \approx_c (x, f_H(x))$ for circuits of size s', then $(z, f(z|_{S_1}), \ldots, f(z, |_{S_k})) \approx_c (z, f_H(z|_{S_1}), \ldots, f_H(z|_{S_k}))$ for circuits of size $s'' = s' - O(sk)$. However, since $s \geq s'$, we have $s'' \leq 0$; thus, the conclusion is meaningless.

Although the first issue could be fixed using [SS93], the second issue appears to be an inherent limitation of this approach.

Our Solution: Feasible Hard-Core Set. Our key technical contribution is to generalize Impagliazzo's hard-core set to a *feasible hard-core set lemma*. In the original notion of hard-core set, H is said to be a hard-core set of f for circuits of size s if for every circuit C of size s,

$$\Pr_{x\sim H}[C(x) = f(x)] \leq \frac{1}{2} + \epsilon$$

for a small parameter $\epsilon > 0$. Ideally, we would like to generalize this to H-oracle circuits C^H, i.e.,

$$\Pr_{x\sim H}[C^H(x) = f(x)] \leq \frac{1}{2} + \epsilon.$$

Here, an *H-oracle circuit* C^H is a circuit with H-oracle gates, which compute the characteristic function of H. It is easy to observe that the proof of (6) $\Rightarrow$ (7) works for H-oracle circuits. Unfortunately, we failed to prove the existence of a hard-core set H for H-oracle circuits. However, by imposing an appropriate restriction on H-oracle circuits, we are able to generalize Impagliazzo's hard-core lemma and prove the hardness self-amplification theorem.

The restriction on H-oracle circuits C^H is as follows: Informally, we require that C^H can make a query q to an oracle only if C^H can compute $f(q)$. To formalize this idea, it is easier to assume that C^H is a *k-query nonadaptive* oracle circuit, i.e., there exists a small circuit Q_i that, on input x, outputs the i-th query q_i of C^H for each $i \in [k]$. We say that an oracle circuit C^H of size s is *f-trapdoor* if for every $i \in [k]$, there exists a circuit of size s that computes $f(Q_i(x)))$ on input x. In other words, an f-trapdoor oracle circuit is allowed to make a query q only if $f(q)$ can be computed without an oracle.

Now, we introduce a new notion of feasible hard-core set. We say that H is a *f-feasible $(1/2-\epsilon)$-hard-core set* for k-query circuits of size s if H is a hard-core set for f-feasible k-query H-oracle circuits of size s, i.e., for every f-trapdoor k-query H-oracle circuit C^H of size s,

$$\Pr_{x\sim H}[C^H(x) = f(x)] \leq \frac{1}{2} + \epsilon.$$

It is not hard to observe that the proof of (6) $\Rightarrow$ (7) works for f-trapdoor k-query H-oracle circuits. Indeed, for all $j \in [k] \setminus \{i\}$, the function $z|_{S_i} \mapsto f(z|_{S_j})$ can be computed by a circuit of size s if $z|_{[m]\setminus S_i}$ is fixed; thus, an f-trapdoor circuit can ask a query $z|_{S_j}$ to an oracle H and can decide whether $z|_{S_j} \in H$ or not, which enables the computation of $f_H(z|_{S_j})$. It remains to prove the existence of a feasible hard-core set.

2.3 The Existence of a Feasible Hard-core Set

We now state the generalization of Impagliazzo's hard-core lemma to the feasible hard-core lemma.

Lemma 2.2 (feasible hard-core set; see also Lemma 5.4)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a δ-hard function on $\{0,1\}^n$ for circuits of size s and suppose $\delta^2 2^n \geq 8\cdot 10^4$. Then, there exists an f-feasible $(1/2-\epsilon)$-hard-core set $H \subseteq \{0,1\}^n$ of size $\delta 2^n$ for k-query circuits of size s', where*

$$s' = s \cdot \exp\left(-(\delta\epsilon)^{-O(1)} \cdot \log k\right) - O\left(n/(\delta\epsilon)^{O(1)}\right).$$

We prove this by generalizing the original proof of Impagliazzo's hard-core lemma [Imp95], which is based on a boosting algorithm.[8] The proof of the contrapositive of the feasible hard-core set lemma proceeds by constructing hard-core sets $H_1, \ldots, H_T$ inductively. The first hard-core set is $H_1 = \{0,1\}^n$ and H_t is defined by using $H_1, \ldots, H_{t-1}$. Using the assumption that H_t is not a feasible hard-core set, we obtain some f-trapdoor k-query H_t-oracle circuit $C_t^{H_t}$ that computes f on a $(1/2+\epsilon)$-fraction of inputs in H_t. The proof of Impagliazzo's hard-core lemma [Imp95] enables us to show that the circuit $C' = \mathsf{majority}(C_1^{H_1}, \ldots, C_T^{H_T})$ performs well over the uniform distribution. However, this circuit has oracle access to $H_1, \ldots, H_T$. To complete the proof, we need to make C' oracle-free. To this end, we make an induction hypothesis that $C_1^{H_1}, \ldots, C_{t-1}^{H_{t-1}}$ can be computed by small circuits $C'_1, \ldots, C'_{t-1}$ without any oracle. We claim the existence of a small circuit C'_t that simulates $C_t^{H_t}$ without any oracle. Now, we exploit the property that $C_t^{H_t}$ is an f-trapdoor circuit. By the property of the f-trapdoor circuit, for any query q of $C_t^{H_t}$, there exists a small circuit that computes $f(q)$ without any oracle. Thus, to simulate $C_t^{H_t}$, it suffices to decide whether $q \in H_t$ using $f(q)$ as advice. In the boosting algorithm, whether $q \in H_t$ depends on the "performance" of previous circuits $C_1^{H_1}(q), \ldots, C_{t-1}^{H_{t-1}}(q)$; we need to count the number of indices $i \in \{1, \ldots, t-1\}$ such that $C_i^{H_i}(q) = f(q)$. Using the advice $f(q)$ and the induction hypothesis that $C_i^{H_i}$ can be computed by the small circuit C'_i, the number of such indices i can be counted efficiently. Details can be found in Section 5.

We note that the parameters in our feasible hard-core set lemma are significantly worse than standard hard-core lemmas: the dependence on δ and ϵ is exponential. However, for our applications, it suffices to choose $\delta^{-1}, \epsilon^{-1}$ and k to be constants independent of n, in which case the loss in the circuit size s is negligible. Whether the dependence on parameters can be improved is an interesting open question.

[8] Interestingly, the proof based on the minimax theorem does not work; see Section 5.2.

2.4 Conclusion and Open Problems

This paper presents a general framework of hardness amplification for a function that can be written as the sum of "computatuonally disjoint" components, which extends the nearly-disjoint generator of Nisan and Wigderson [NW94]. This general framework enables us to obtain strong fine-grained average-case hardness of natural problems.

This paper leaves many open questions and intriguing research directions. The first one is to extend our hardness amplification framework to non-Boolean functions. For example, can we amplify the hardness of counting triangles modulo q for $q > 2$? Can we amplify the hardness of the matrix multiplication using our framework? The very recent work of Asadi, Golovnev, Gur, and Shinkar [AGGS22] developed a different framework and showed a hardness-self amplification theorem for the matrix multiplication. Interestingly, the framework of [AGGS22] works well for non-Boolean functions, whereas our framework works well for Boolean functions. It would be an intriguing research direction to generalize both frameworks to obtain a general framework that works in both settings.

The difficulty of extending our framework to non-Boolean functions is that boosting algorithms for proving the hard-core lemma requires a circuit that correctly computes f on at least $(1/2+\epsilon)$-fraction in any large set H (regardless of whether f is Boolean or not). Although it is possible to encode a multi-output function as a Boolean function using Hadamard encoding or any error-correcting code (as is common in the literature of hardness amplification), the goal of obtaining a *natural* strongly average-case hard problem prevents us from using such proof techniques.

The second question concerns the range of applications: Can we apply our hardness amplification framework to other famous problems? For example, counting zero-weight triangles and counting t-cliques over Erdős–Rényi random graph $G(n, 1/2)$ have been well studied in the literature of fine-grained average-case complexity. Despite effort, we could not construct computational designs for these problems.

The third direction is to obtain a *uniform* hardness amplification in order to establish the strong average-case hardness against uniform algorithms. The main obstacle to this goal is the construction of the computational design. Although it is easy to extend the notion of computational design to uniform computational models, we are not aware of a problem that admits such design. For example, in Section 1.2, we exploit the fact that $\mathsf{TriParity}(x)$ becomes a linear function over $O(n^2)$ variables if the input is partially fixed, but the coefficients of the linear function may not be computed by uniform algorithms. There are proof techniques in the literature on derandomization [IW01; TV07] that exploit downward self-reducibility and random self-reducibility to make reductions uniform, which may be useful to investigate this research direction.

2.5 Organization

This paper is organized as follows: In Section 3, we briefly mention related works. In Section 4, we present our notations and definitions. In Section 5, we present the first technical tool: feasible hardcore lemma. In Section 6, we present the second technical tool: computational design. In Section 7, we prove our new hardness amplification result using the technical tools. In Sections 8 and 9, we apply our general result to concrete problems and prove Theorems 1.1 to 1.3.

3 Related Work

3.1 Relation to Coding Theory

A worst-case-to-average-case reduction for a function f can be seen as the following task: Compute $f(x)$ for an input $x \in X$ given a function $C\colon X \to Y$ that is ϵ-close[9] to f as oracle. We usually assume that $f \in \mathcal{F}$ for some class of "tractable" functions. This task is referred to as *local decoding for $\mathcal{F}$ within relative distance ϵ*. Here, the target function f (i.e., the function in $\mathcal{F}$ that is ϵ-close to the given oracle C) is promised to be unique.

However, in the task of local decoding, the error tolerance ϵ must be so small that the uniqueness condition holds. To obtain reductions for smaller ϵ, consider the following task: Given a function $C\colon X \to Y$ as oracle, enumerate all functions $f \in \mathcal{F}$ that are ϵ-close to C. This task is referred to as *local list decoding for $\mathcal{F}$ within relative distance ϵ*.

Local (list) decoding for $\mathcal{F} = \mathsf{RM}_{n,d,\mathbb{F}_q}$ being the family of degree-d n-variate polynomials over $\mathbb{F}_q$ has been extensively studied for decades [GL89; Sud97; GRS00; STV01; GKZ08; BL15], motivated by a wide range of applications including derandomization, PCP theorems, and cryptography.

One may run a local list decoding algorithm using an average-case solver C as oracle and obtain a worst-case solver C' for f. This idea works if we use local list-decoding in *implicit form* in the sense that the decoder outputs a list of *oracle machines* $M_1^C, \dots, M_\ell^C$ one of which computes the target function f (in other words, the correct machine can be seen as a succinct representation of f). Such computational task for $\mathcal{F} = \mathsf{RM}_{n,d,\mathbb{F}_q}$ with large q can be done by the work of [STV01, Theorem 29]. Sudan, Trevisan, and Vadhan [STV01] presented a local list-decoding algorithm for a Reed–Muller code $\mathcal{F} = \mathsf{RM}_{n,d,\mathbb{F}_q}$ with large q and used it to present an alternative proof of the theorem of [IW97] without using XOR lemmas. We note that there are exponentially many polynomials at the distance of 2^{-d} over $\mathbb{F}_2$ [KLP12]. This means that even list decoding algorithms that enumerate all the polynomials close to a given string in an explicit form cannot be used to give worst-case-to-average-case reductions. To the best of our knowledge, no efficient local list-decoding algorithm for small $\mathbb{F}_q$ is known: The currently known local list-decoders for $\mathsf{RM}_{n,d,\mathbb{F}_q}$ with small q [GKZ08; BL15] recover all coefficients of f, which requires a running time of at least n^d.

There is another line of works that puts a restriction on queries made by the decoder. Bafna, Srinivasan, and Sudan [BSS20] presented a local decoding algorithm for $\mathsf{RM}_{n,d,\mathbb{F}_q}$ within relative distance $2^{-O(qd)}$ that makes at most $2^{O(qd)}$ queries in $\{0,1\}^n$. Decoders in this setting can be used to obtain a worst-case-to-average-case reduction for graph problems. The worst-case-to-average-case reductions of Boix-Adserá, Brennan, and Bresler [BBB21] and subsequent works [DLW20; HS21] can be seen as a local decoder for a specific family $\mathcal{F} \subseteq \mathsf{RM}_{n,d,\mathbb{F}_q}$ (called *good low-degree polynomials* by Dalirrooyfard, Lincoln, and Williams [DLW20]) that makes queries in $\{0,1\}^n$ within relative distance $1/(\log n)^{\Theta(d)}$. Unfortunately, this decoder has small relative distance (i.e., small error tolerance). Boix-Adserá, Brennan, and Bresler [BBB21] and Goldreich [Gol20] obtained a local decoder for good low-degree polynomials over $\mathbb{F}_2$ within constant relative distance. Actually, the decoder of [BSS20] already implies this result (see Appendix B for details).

3.2 Subgraph Counting on Random Graphs

In an H-subgraph counting problem, we are given a graph and asked to count the number of subgraphs that is isomorphic to H. This is a fundamental task of graph algorithms and has been widely investigated for decades.

[9] A function f over X is *ϵ-close* to a function g over X if $\Pr_{x\sim X}[f(x) \neq g(x)] \leq \epsilon$.

There is a recent progress on average-case lower bounds for the subgraph counting on random graphs. Goldreich and Rothblum [GR18] presented a worst-case-to-average-case reduction for K_t-subgraph counting for fixed t, where the input is drawn from uniform distribution over some set S of n-vertex graphs. Although their reduction has a large error tolerance of $1 - 1/\operatorname{polylog}(n)$, the input distribution of the average-case solver is somewhat artificial due to their reduction steps. Boix-Adserá, Brennan, and Bresler [BBB21] and subsequent works [DLW20; HS21] reduced H-counting and similar problems to that on Erdős–Rńeyi random graph. Goldreich [Gol20] obtained a simple worst-case-to-average-case reduction for the parity variant of the K_t-counting. Hirahara and Shimizu [HS21] presented the framework of hardness amplification in fine-grained complexity for a variant of H-counting problems and improved the error tolerance, though the input distribution of the average-case solver is somewhat artificial (specifically, it is the disjoint union of $n^{o(1)}$ random bipartite graphs).

3.3 Fine-grained Average-case Complexity

In a pioneering work, Ball, Rosen, Sabin, and Vasudevan [BRSV17] investigated the fine-grained average-case complexity of problems and constructed Proof of Work systems based on them. To this end, they encoded famous problems in fine-grained complexity (e.g., Orthogonal Vectors, All-Pairs Shortest Paths, 3SUM) as low-degree polynomial evaluation and then obtained worst-case-to-average-case reductions for the encoded problems. They hope for constructing a one-way function under a worst-case hardness assumption in fine-grained complexity setting. Motivated by this research direction, LaVigne, Lincoln, and Williams [LLW19] constructed a cryptographic key exchange scheme in the fine-grained complexity setting under the average-case hardness assumption on Zero-k-Clique (the problem of finding a zero-weight k-clique in an edge-weighted graph). In view of this, one might think there is a tremendous gap in average-case complexity between "counting" and "finding" problems: Is counting much harder than finding? Interestingly, this is not the case for some problems: Dalirrooyfard, Lincoln, and Williams [DLW20] reduced the counting zero-k-cliques to finding a zero-k-clique.

3.4 Other Natural Problems

The permanent $\mathsf{perm}(M)$ of a matrix $M \in \mathbb{F}_q^{n\times n}$ is defined by $\mathsf{perm}(M) = \sum_{s\in S_n} \prod_{i\in[n]} M_{i,s(i)}$ where S_n is the set of all permutations over $[n]$. It is widely known that perm admits a worst-case-to-average-case reduction (oftenly reffered to as random self-reducibility) [Lip91] and subsequent works [GLRSW91; GS92; FL92; CPS99] improved the error tolerance. In particular, Cai, Pavan, and Sivakumar [CPS99] proved that if $\mathsf{perm}(M)$ for a $(1/\mathrm{poly}(n))$-fraction of $M \sim \mathbb{F}_q^{n\times n}$ over large finite field $\mathbb{F}_q$ can be computed then $\mathsf{perm}(M)$ for any input $M \in \mathbb{F}_q^{n\times n}$ can be computed by a randomized algorithm with a polynomial overhead in running time. A worst-case-to-average-case reduction for computing the permanent over a small finite field is not known (note that the permanent over $\mathbb{F}_2$ is equal to the determinant and can be computed in polynomial time).

The matrix multiplication over a finite ring R is a classical example that admits a worst-case-to-average-case reduction [BLR93]: Let A be the average-case solver that correctly compute the matrix multiplication for 99% of inputs. For given matrices $M, N \sim \mathbb{F}_q^{n\times n}$, sample two random matrices $S, T \sim R^{n\times n}$ and output $A(M-S, N-T) + A(M-S, T) + A(S, N-T) + A(S, T)$. This algorithm succeeds 96% over the choice of S, T for any inputs by the union bound. Very recently, Asadi, Golovnev, Gur, and Shinkar [AGGS22] improved the error tolerance using additive combinatorics.

4 Preliminaries

For $n \in \mathbb{N}$, let $[n] := \{1, \dots, n\}$. We denote by U_S the uniform distribution over a finite set S. We use the shorthand notation $U_n = U_{\{0,1\}^n}$. We use $x \sim D$ to denote that x is sampled according to a distribution D. For a finite set S, we use $x \sim S$ for the shorthand of $x \sim U_S$. For a distribution D over $\{0,1\}^n$ and function $h\colon \{0,1\}^n \to \{0,1\}^m$, let $h(D)$ be the distribution of $h(X)$ for $X \sim D$. A *random function* is a function that is sampled according to some distribution over a set of functions.

A (random) function $f\colon \{0,1\}^n \to \{0,1\}$ is *δ-hard for size s over a distribution D* if $\Pr[C(x) = f(x)] \leq 1 - \delta$ for any size-s circuit C, where the probability is taken over random choices of $x \sim D$ (and f). Two distributions D_1 and D_2 are *ϵ-indistinguishable* for size s if $|\mathbf{E}_{x\sim D_1}[C(x)] - \mathbf{E}_{y\sim D_2}[C(y)]| \leq \epsilon$ for any size-s circuit C. We invoke the following well-known results.

Lemma 4.1 ([Yao82]; see, e.g., Lemma 2.3 of [HVV06])**.** *If $f\colon \{0,1\}^n \to \{0,1\}$ is $(1/2 - \epsilon)$-hard over U_n for size s, then the distributions $(x, f(x))$ for $x \sim U_n$ and U_{n+1} are ϵ-indistinguishable for size $s - O(1)$. Converesely, if $(x, f(x))$ for $x \sim U_n$ and U_{n+1} are ϵ-indistinguishable, then f is $(1/2 - \epsilon)$-hard over U_n for size $s - O(1)$.*

Lemma 4.2 (folklore)**.** *Let $C\colon \{0,1\}^n \to \{0,1\}$ be a circuit of size s and D_1, D_2 be distributions over $\{0,1\}^n$ that are ϵ-indistinguishable for size s'. Then, $C(D_1)$ and $C(D_2)$ are ϵ-indistinguishable for size $s - s'$.*

Proof Sketch. If C' is the distinguisher for $C(D_1)$ and $C(D_2)$ of size $s - s'$, then the circuit $C' \circ C$ has size s and distinguishes D_1 and D_2. □

We identify a set $H \subseteq \{0,1\}^n$ with the characteristic function, i.e., for $x \in \{0,1\}^n$, $H(x) = 1$ if $x \in H$ and $H(x) = 0$ otherwise. We say that H is *δ-dense* if $|H| \geq \delta 2^n$. In an oracle circuit C^H, the circuit C can access to $H(x)$.

A measure is a function $M\colon \{0,1\}^n \to [0,1]$. The *size* of M is $|M| := \sum_{x\in\{0,1\}^n} M(x)$ and the *relative size* of M is $\mu(M) := |M|/2^n$. A measure M is *δ-dense* if $\mu(M) \geq \delta$. Let U_M be the distribution that assigns $M(a)/|M|$ to each $a \in \{0,1\}^n$. We use $x \sim M$ for the shorthand of $x \sim U_M$. Note that, if $M(x) \in \{0,1\}$ for every $x \in \{0,1\}^n$, then M can be seen as a binary indicator for some subset $H \subseteq \{0,1\}^n$ and we have $|H| = |M|$ and $U_M = U_H$. In this sense, a measure can be seen as a continuous relaxation of a set. For a measure M, an oracle circuit C^M can access $M(q)$ (with sufficient precision, say, $O(\log n)$ bits) for each query $q \in \{0,1\}^n$. Actually, throughout the paper, our circuit uses oracle M to toss a coin with head probability $M(q)$ for a query q.

5 Feasible Hard-core Set

In this section, we define the notion of feasible hard-core set, which is our key ingredient.

Definition 5.1 (Trapdoor Oracle Circuit)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a function. An f-*trapdoor oracle circuit *is an oracle circuit $C^{\mathcal{O}}$ such that $C^{\mathcal{O}}(x)$ given x as input makes nonadaptive queries $q_1(x), \dots, q_k(x)$ such that $f(q_i(x))$ for given x can be computed by a circuit smaller than $C^{\mathcal{O}}$ for all $i \in [k]$ (here, the size of $C^{\mathcal{O}}$ does not take the size of the oracle gate into account).*

Definition 5.2 (Feasible Hard-core)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$, $\epsilon \in (0,1)$, and $k \in \{0\} \cup \mathbb{N}$. A set (measure) H is a* feasible ϵ-hard-core set (measure) for size s with k-queries *if $\Pr_{x\sim H}[C^H(x) = f(x)] \leq 1/2 + \epsilon$ for any size-s k-query f-trapdoor oracle circuit C^H.*

Note that the definition above allows the case of $k = 0$ in which the corresponding circuit would be oracle-free. Therefore, the notion of feasible hard-core set is a generalization of the hard-core set of [Imp95].

5.1 Feasible Hard-core Lemma

The original proof of the hard-core lemma by Impagliazzo [Imp95] consists of two steps. The first and ingredient step is the proof of the existence of a hard-core measure M. Then, a standard probabilistic argument shows that the random set H where every $x \in \{0,1\}^n$ belongs to H independently with probability $M(x)$ satisfies $\Pr_H[\forall \text{small } C, \Pr_{x\sim H}[C(x) = f(x)] \approx \Pr_{x\sim M}[C(x) = f(x)]] > 0$. This implies that the random set H is indeed the hard-core set with positive probability. Actually, the first step still works for our feasible hard-core setting as we will see in Lemma 5.3, while the second part does not. Though a feasible hardcore measure (Lemma 5.3) suffices to prove our hardness amplification result, we will work on showing the existence of a feasible hardcore set in Lemma 5.4.

Lemma 5.3 (Feasible Hard-core Measure)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a δ-hard function over U_n for size s. Then, there is a δ-dense feasible $(1/2-\epsilon)$-hard-core measure M for size s' with k-queries, where $s' = \frac{s}{O(t)+O(kt)^t}$ for $t = \delta^{-2}\epsilon^{-2}$.*

Proof. We invoke the proof of the original hard-core lemma by Impagliazzo [Imp95]. We prove the contraposition. Suppose for every δ-dense measure M that there is a size-s k-query f-trapdoor oracle circuit C^M_M satisfying $\Pr_{x\sim M}[C^M_M(x) = f(x)] \geq 1/2 + \epsilon$. We construct an oracle-free circuit C' that satisfies $\Pr_{x\sim U_n}[C'(x) = f(x)] > 1 - \delta$, which contradicts to the hardness of f. Our circuit C' is of the form $C'(x) = \mathsf{maj}(C_1, \ldots, C_t)$ for some $t \leq \epsilon^{-2}\delta^{-2}$ (maj denotes the majority function).

Let $\gamma = \delta\epsilon$. Start with the constant measure $M_1 \equiv 1$ (i.e., uniform distribution over all inputs). For a measure M_i, consider $C^{M_i}_{M_i}$. The circuit $C^{M_i}_{M_i}$ is an oracle circuit at this point but we will claim that it can be transformed into an oracle-free circuit C_i at a cost of size. Let $R_i(x) = \sum_{j=1}^{i}[C_j(x) = f(x)]$, where $[Z] = 1$ if Z holds and $[Z] = -1$ otherwise. For $a \in \mathbb{Z}$, let $m(a) = 1$ if $a \leq 0$, $m(a) = 0$ if $a \geq 1/\gamma$, and $m(a) = 1 - \gamma a$ otherwise. Define the measure M_{i+1} by $M_{i+1}(x) = m(R_i(x))$ for every $x \in \{0,1\}^n$. Terminate this process if $\mu(M_{i+1}) \leq \delta$. Suppose we obtained $C_1, \ldots, C_t$ and let $C' = \mathsf{maj}(C_1, \ldots, C_t)$. If $C'(x) \neq f(x)$ then $R_t(x) \leq 0$ and thus $M_{t+1}(x) = 1$. Therefore, $\Pr_{x\sim U_n}[C'(x) = f(x)] \geq 1 - \mu(M_{t+1}) \geq 1 - \delta$.

We bound t by the same argument in [Imp95]. Let $P_t = \sum_{j\in[t]}\sum_{x\in\{0,1\}^n} M_j(x)[C_j(x) = f(x)]$. First, observe that $P_t = \sum_{j\in[t]} |M_j|(2\Pr_{x\sim D_{M_j}}[C_j(x) = f(x)] - 1) \geq 2^{n+1}\gamma t$. On the other hand, we claim $P_t \leq 2^n(t\gamma/2 + 1/\gamma)$. To see this, fix $x \in \{0,1\}^n$ and consider the edge-weighted digraph $G_x = (V, A)$ defined by the vertex set $V := \{R_j(x)\colon j = 0, \ldots, t\}$ (here we set $R_0(x) = 0$), edge set $A := \{(R_{j-1}(x), R_j(x))\colon j = 1, \ldots, t\}$ where each edge $(R_{j-1}(x), R_j(x))$ is associated with a weight $M_j(x)(R_j(x) - R_{j-1}(x)) = m(R_{j-1})(R_j(x) - R_{j-1}(x))$. Note that G can be seen as a walk visiting $R_0(x), R_1(x), \ldots, R_t(x)$ in this order. Consider the sum of edge-weights in G_x. Suppose G_x contains a pair edges of the form $(a, a+1)$ and $(a+1, a)$. This pair contributes at most $|m(a) - m(a+1)| \leq \gamma$ to the sum. After removing such (at most $t/2$) pairs, the remaining edges are of the form either $\{(a, a+1), \ldots, (a+\ell-1, a+\ell)\}$ or $\{(a, a-1), \ldots, (a-\ell+1, a-\ell)\}$. The former type has weight $m(a) + \cdots + m(a+\ell-1) \leq 1/\gamma$, whereas the latter one has $-m(a) - \cdots - m(a-\ell+1) \leq 0$. Thus, the total weight of G_x is at most $t\gamma/2 + 1/\gamma$. Since P_t is the sum of total edge weight of G_x over all $x \in \{0,1\}^n$, we have $P_t \leq 2^n(t\gamma/2 + 1/\gamma)$. Therefore, $2^{n+1}\gamma t \leq P_t \leq 2^n(t\gamma/2 + 1/\gamma)$, which implies $t \leq 2/(3\gamma^2) \leq \delta^{-2}\epsilon^{-2}$.

Finally, we bound the size of C'. Suppose for simplicity that $C^{M_i}_{M_i}$ makes exactly k queries for each M_i. Note that the oracle circuit $C^{M_1}_{M_1}$ can be a circuit C_1 of size s by replacing the oracle gate

with the constant 1. Suppose that C_j can be transformed into a size-s_j circuit for each $j \in [i]$. We assume $s_1 \leq \cdots \leq s_i$ for simplicity. We transform the size-s oracle circuit $C_{M_{i+1}}^{M_{i+1}}$ into an oracle-free circuit of size s_{i+1}. For given input $x \in \{0,1\}^n$, the circuit $C_{M_{i+1}}^{M_{i+1}}$ runs as follows: First, run its query-making part to construct queries $q_1(x), \ldots, q_k(x)$. Then, access the oracle M_{i+1} to obtain $M_{i+1}(b_1), \ldots, M_{i+1}(b_k)$. Finally, decide the output depending on these responses. The first and last steps can be done by a size-s circuit. Thus, it remains to compute $M_{i+1}(q_j(x))$ without oracle for all $j \in [k]$. By definition, $M_{i+1}(q_j(x)) = m(R_i(x))$ for $R_i(x) = \sum_{\ell \in [i]} [C_\ell(q_j(x)) = f(q_j(x))]$, which can be computed by an oracle-free circuit of size $O(\sum_{\ell \in [i]} s_\ell) = O(ts_i)$ (note that $f(q_j(x))$ can be computed by a size-s circuit). Therefore, $C_{M_{i+1}}^{M_{i+1}}$ can be transformed into a circuit C_{i+1} of size at most $s + O(kts_i)$. By solving the recursion $s_{i+1} \leq s + O(kts_i)$ and $s_1 = s$, we obtain $s_i \leq (1 + i(Lkt)^i) \cdot s$ for some absolute constant $L > 0$. Therefore, $C' = \mathsf{maj}(C_1, \ldots, C_t)$ has size at most $O(s(t + t^2(Lkt)^t))$. The claim follows since $t \leq \delta^{-2}\epsilon^{-2}$. □

Lemma 5.4 (Feasible Hard-core Set)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a δ-hard function over U_n for size s and suppose $\delta^2 2^n \geq 8 \cdot 10^4$. Then, there is a δ-dense feasible $(1/2 - \epsilon)$-hard-core set $H \subseteq \{0,1\}^n$ for size s' with k-queries, where $s' = \frac{s - O(nt \log(\delta^{-1}\epsilon^{-1}))}{O(t) + O(kt)^t}$ for $t = \delta^{-2}\epsilon^{-2}$.*

Proof. We prove the contraposition. Suppose for every δ-dense set H that there is a size-s k-query f-trapdoor oracle circuit C_H^H satisfying $\Pr_{x \sim H}[C_H^H(x) = f(x)] \geq 1/2 + \gamma$. We construct a circuit C' satisfying $\Pr_{x \sim U_n}[C'(x) = f(x)] > 1 - \delta$, which contradicts to the assumption of the lemma. The proof is almost identical to that of Lemma 5.3. The circuit C' is of the form $C'(x) = \mathsf{maj}(C_1, \ldots, C_t)$ for some $t \leq \delta^{-2}\epsilon^{-2}$.

Consider the following randomized procedure that defines a sequence of circuits $(C_i)_{i \in \mathbb{N}}$. Start with the constant measure $M_1 \equiv 1$. For a measure M_i, consider the random set $H_i \subseteq \{0,1\}^n$ defined by $x \in H_i$ with probability $M_i(x)$ for all $x \in \{0,1\}^n$. Here, we assume that the random variables $(H_i(x))_{x \in \{0,1\}^n}$ are pairwise independent for the sake of later argument. If $|H_i| < \delta 2^n$, let C_i be an arbitrary size-s circuit (say, the circuit that always outputs 0). Otherwise, let C_i be the oracle circuit $C_{H_i}^{H_i}$. Let $R_i(x) = \sum_{j=1}^{i} [C_{H_i}^{H_i}(x) = f(x)]$. Define the measure M_{i+1} by $M_{i+1}(x) = 1$ if $R_i(x) \leq 0$, $M_{i+1}(x) = 0$ if $R_i(x) \geq 1/\gamma$, and $M_{i+1}(x) = 1 - \gamma R_i(x)$ otherwise, where $\gamma := \delta\epsilon$. Terminate this procedure when $\mu(M_{i+1}) < 1.01\delta$.

Suppose we obtained $C_1, \ldots, C_t$ by the procedure above and let $C' = \mathsf{maj}(C_1, \ldots, C_t)$. If $C'(x) \neq f(x)$ then $R_t(x) \leq 0$ and thus $M_{t+1}(x) = 1$. Therefore, $\Pr_{x \sim U_n}[C'(x) = f(x)] \geq 1 - \mu(M_{t+1}) < 1 - 1.01\delta$.

We claim $t \leq \delta^{-2}\epsilon^{-2}$. Consider the potential $P_t = \sum_{x \in \{0,1\}^n} \sum_{i=1}^{t} H_i(x)[C_i(x) = f(x)]$, where we recall that $[Z] = 1$ if Z holds and $[Z] = -1$ otherwise. Note that P_t is a random variable due to the random choice of H_i. On one hand, we have

$$\begin{aligned} P_t &\geq \sum_{i \in [t]\colon |H_i| \geq \delta 2^n} |H_i| \sum_{x \in H_i} [C_i(x) = f(x)] \\ &\geq \delta 2^n \cdot \sum_{i \in [t]\colon |H_i| \geq \delta 2^n} (2 \Pr_{x \sim H_i}[C_i(x) = f(x)] - 1) \\ &\geq \delta\epsilon 2^{n+1} \cdot |\{i \in [t]\colon |H_i| \geq \delta 2^n\}|. \end{aligned}$$

In the last inequality, we used the property of C_i. By the pairwise independence of $(H_i(x))_{x \in \{0,1\}^n}$, we have $\mathbf{Var}[|H_i(x)|] = \sum_{x \in \{0,1\}^n} M_i(x)(1 - M_i(x)) \leq \frac{2^n}{4}$. From the Chebyshev inequality, we obtain $\Pr[|H_i(x)| \geq \delta 2^n] \geq 1 - \frac{\mathbf{Var}[|H_i|]}{(0.01\delta 2^n)^2} \geq 1 - \frac{10^4}{\delta^2 2^{n+2}} \geq \frac{1}{2}$ by our assumption on δ.

On the other hand, fix $x \in \{0,1\}^n$ and consider the edge-weighted digraph $G_x = (V, A)$ defined by the vertex set $V = \{R_j \colon j = 0, \dots, t\}$ (here we set $R_0(x) = 0$) and edge set $A = \{(R_{j-1}(x), R_j(x)) \colon j \in [t]\}$. An edge $(R_{j-1}(x), R_j(x))$ is associated with a weight $H_j(x)(R_j(x) - R_{j-1}(x)) \in \{0, \pm 1\}$ (note that the weight is a random variable). Note that G can be seen as a walk on $\mathbb{Z}$ that visits $R_0(x), R_1(x), \dots, R_t(x)$ in this order. Then, the sum of edge weights of G is equal to $S := \sum_{i \in [t]} H_i(x)[C_i(x) = f(x)]$. For $a \in \mathbb{Z}$, let $m(a) = 0$ if $a \le 0$, $m(a) = 1 - \gamma a$ if $0 < a < 1/\gamma$, and $m(a) = 1$ if $1/\gamma \le a$. Suppose that G contains a pair of edges of the form $(a, a+1)$ and $(a+1, a)$. This pair in expectation contributes $\mathbf{E}[H_i(x) - H_j(x)] = m(a) - m(a+1) \le \gamma$ to S for some $i, j \in [t]$. Summing up these contributions over at most $t/2$ such pairs, the expected contribution of such pairs in G to S is at most $\gamma t/2$. If we remove these pairs from G, the remaining part is a directed path visiting a sequence of vertices either $(0, 1, \dots, \ell)$ or $(0, -1, \dots, -\ell)$ for some $\ell \in \mathbb{N}$. The former pattern has a total weight at most $1/\gamma$ since an edge $(b, b+1)$ has weight 0 if $b > 1/\gamma$. The latter one has a total weight at most 0. Therefore, we have $\mathbf{E}[S] \le \gamma t/2 + 1/\gamma$ and thus $\mathbf{E}[P_t] \le 2^n(\gamma t/2 + 1/\gamma)$. Combining the upper and lower bounds of P_t, we obtain

$$\delta \epsilon 2^n \le \mathbf{E}[P_t] \le 2^n(\gamma t/2 + 1/\gamma),$$

which implies $t = O(\delta^{-2}\epsilon^{-2})$.

Finally, it suffices to show that we can make C' oracle-free. This can be done by the same argument as the proof of Lemma 5.3 except for sampling H_i. The key difference is that the circuit $C_{H_i}^{H_i}$ has a table $(H_i(z))_{z \in \{0,1\}^n}$ as oracle and looks at $H_i(q)$ when it makes a query q, as opposed to the oracle circuit $C_{M_i}^{M_i}$ in Lemma 5.3 that samples $\mathrm{Ber}(M_i(q))$ when it makes a query q. For example, if the computations $C_{H_i}^{H_i}(x)$ and $C_{H_i}^{H_i}(y)$ for $x \neq y$ make a query q, then the response $H_i(q)$ of the oracle must be the same value, while this consistence does not necessarily hold in the oracle circuit $C_{M_i}^{M_i}$. To ensure this consistency efficiently, the circuit C_i first samples a random function $h \in \mathcal{H}$ from a pairwise independent hash family $\mathcal{H}$ and then use h to sample $H_i(q)$ for every query q. Note that $M_i(q)$ is represented by some $B = O(t \log \gamma^{-1})$ bits. Let $W = \{0, \dots, 2^B - 1\}$. Then, it suffices to sample pairwise independent random variables $(X_z)_{z \in \{0,1\}^n}$ such that each merginal distribution is U_W. Sample $r_0 \sim U_W$ and $r = (r_1, \dots, r_n) \sim U_{W^n}$. Let $h \colon \{0,1\}^n \to V$ be the random function defined as $h(x) = \left(r_0 + \sum_{i \in [n]} x_i r_i\right) \bmod 2^B$. Then, the family $(h(x))_{x \in \{0,1\}^n}$ are pairwise independent over the random choice of h and thus has the desired property. The construction of h can be done using $O(nB) = O(nt \log \gamma^{-1})$ random bits. Suppose that $C_{H_i}^{H_i}$ can transformed to an oracle-free circuit of size s_i. By the argument in the proof of Lemma 5.3 combined with the size for sampling h, we have $s_{i+1} \le s + O(kts_i) + nt \log \gamma^{-1}$ and $s_1 = s$. Therefore, $C' = \mathsf{maj}(C_1, \dots, C_t)$ has size at most $O((s + nt \log(\delta^{-1}\epsilon^{-1}))(t + t^2(Lkt)^t))$ for some constant $L > 0$. □

5.2 Does Nisan's Proof Work?

Another well-known approach to the proof of the hard-core lemma exploits the minimax theorem of Neumann, presented by Nisan [Imp95]. We briefly outline the argument and explain why this approach fails to prove the feasible hard-core lemma.

Let $f \colon \{0,1\}^n \to \{0,1\}$ be a function and consider the following two-player zero-sum game. Player 1 chooses a δ-dense set H and Player 2 chooses a size-s' oracle circuit C. Let $P(H, C) = \Pr_{x \sim H}[C^H(x) = f(x)]$ be the outcome of the game. The objective of Player 1 (Player 2) is to obtain the small (resp., large) outcome of the game. Two players can take mixed strategy; Player 1 chooses a distribution $\mathcal{H}$ over all δ-dense sets and then samples a set $H \sim \mathcal{H}$. Similarly, Player 2 decides a distribution $\mathcal{C}$ over size-s' oracle circuits and then samples a circuit $C \sim \mathcal{C}$. The expected

outcome of the game is $c(\mathcal{H},\mathcal{C}) := \mathbf{E}_{H\sim\mathcal{H},C\sim\mathcal{C}}[P(H,C)]$. Then, the minimax theorem implies

$$\min_{\mathcal{H}}\max_{\mathcal{C}} c(\mathcal{H},\mathcal{C}) = \max_{\mathcal{C}}\min_{\mathcal{H}} c(\mathcal{H},\mathcal{C}).$$

Therefore, one of the following cases holds.

Case 1. $\min_{\mathcal{H}}\max_{\mathcal{C}} c(\mathcal{H},\mathcal{C}) < 1/2+\epsilon$**.** Let $\mathcal{H}$ be the distribution that attains the minimum. Then, it holds that $\max_C \mathbf{E}_{H\sim\mathcal{H}}\left[\Pr_{x\sim H}[C^H(x)=f(x)]\right] < 1/2+\epsilon$. Note that sampling $x\sim H$ for $H\sim\mathcal{H}$ yields a distribution $D_{\mathcal{H}}$ of x over $\{0,1\}^n$. Let $M(x)=\delta 2^n D_{\mathcal{H}}(x)$ be a measure.

If C is oracle-free, then the event $C^H(x)=f(x)$ does not depends on H. Therefore, we would have $\mathbf{E}_{H\sim\mathcal{H}}\left[\Pr_{x\sim H}[C(x)=f(x)]\right] = \Pr_{x\sim M}[C(x)=f(x)]$ and thus the measure M is the hard-core measure against any small oracle-free circuit. However, the same argument does not yield a feasible hard-core measure since the event $C^H(x)=f(x)$ depends on the choice of H, that is, we cannot obtain the equality of the form $\mathbf{E}_{H\sim\mathcal{H}}\left[\Pr_{x\sim H}[C^H(x)=f(x)]\right] = \Pr_{x\sim M}[C^H(x)=f(x)]$.

Case 2. $\max_{\mathcal{C}}\min_{\mathcal{H}} c(\mathcal{H},\mathcal{C}) \geq 1/2+\epsilon$**.** Let $\mathcal{C}$ be the distribution of circuits that attains the maximum. In the oracle-free setting, the well-known proof in [Imp95] claims that $C' = \mathsf{maj}(C_1,\ldots,C_t)$ for i.i.d. random circuits $C_1,\ldots,C_t\sim\mathcal{C}$ approximates f well on U_n. In the feasible hard-core setting, there arise two questions: the way of defining C' (note that $\mathcal{C}$ is the distribution of oracle circuits but C' must be oracle-free) and the proof of the performance of C' on U_n. Indeed, the proof of the performance of C' on approximating f crucially depends on the fact that C' is independent of H.

6 Nearly Disjoint Generator and Computational Design

A (k,n,ℓ)-*collection* is a family $\mathcal{S}=\{S_1,\ldots,S_k\}$ of subsets such that $S_i\in\binom{[\ell]}{n}$ for all $i\in[k]$. Let $f\colon\{0,1\}^n\to\{0,1\}$ be a function and (A,B) be a partition of $[n]$. For $z\in\{0,1\}^B$, we denote by $f(\cdot|z)\colon\{0,1\}^A\to\{0,1\}$ the function defined by $f(y|z)=f(y,z)$. Intuitively speaking, computing $f(y|z)$ is the task of computing $f(x)$ given a partial input $z=x|_A\in\{0,1\}^A$ in advance.

Definition 6.1 (Computational Design)**.** *Let $f\colon\{0,1\}^n\to\{0,1\}$ and $\mathcal{S}=\{S_1,\ldots,S_k\}$ be a (k,n,ℓ)-collection. We say that $\mathcal{S}$* is a *s-computational design for $f\colon\{0,1\}^n\to\{0,1\}$ if, for every $i\neq j$ and every $z\in\{0,1\}^{S_i\setminus S_j}$, $f(\cdot|z)$ can be computed by a size-s circuit C_z.*

For example, suppose that a (k,n,ℓ)-collection $\mathcal{S}=\{S_1,\ldots,S_k\}$ satisfies $|S_i\cap S_j|\leq d$ for any $i\neq j$. Then, $\mathcal{S}$ is a $2^{O(d)}$-computational design for any function.

For a function f and measure M, define $f_M\colon\{0,1\}^n\to\{0,1\}$ as the random function where each $f_M(x)$ is the output of the following procedure: Toss a coin with head probability $M(x)$. If the coin is head, then output a uniform random bit U_1. Otherwise, output $f(x)$. Let $f_M^k\colon(\{0,1\}^n)^k\to\{0,1\}^k$ be the direct product of k independent copies of f_M. For a (k,n,ℓ)-collection $\mathcal{S}$, define the function $\mathrm{ND}_{\mathcal{S}}\colon\{0,1\}^\ell\to(\{0,1\}^n)^k$ by

$$\mathrm{ND}_{\mathcal{S}}(\sigma) = (\sigma|_{S_1},\ldots,\sigma|_{S_k}).$$

Recall that, for a function $f\colon\{0,1\}^n\to\{0,1\}$,

$$f^k\circ\mathrm{ND}_{\mathcal{S}}(\sigma) = (f(\sigma|_{S_1}),\ldots,f(\sigma|_{S_k})).$$

Lemma 6.2. *Let $k, \ell \in \mathbb{N}$ be parameters. Let $M\colon \{0,1\}^n \to [0,1]$ be any measure and $\mathcal{S}$ be a (k,n,ℓ)-collection that is an s-computational design for $f\colon \{0,1\}^n \to \{0,1\}$. Suppose there is a circuit D of size s_D satisfying*

$$\mathop{\mathbf{E}}_{\sigma\sim U_\ell}\left[D(\sigma, f^k \circ \mathrm{ND}_{\mathcal{S}}(\sigma))\right] - \mathop{\mathbf{E}}_{\substack{\sigma\sim U_\ell \\ f_M^k}}\left[D(\sigma, f_M^k \circ \mathrm{ND}_{\mathcal{S}}(\sigma))\right] > \epsilon.$$

Then there exists a randomized size-$O(sk + s_D)$ k-query f-trapdoor oracle circuit C^M satisfying

$$\mathop{\mathbf{E}}_{\substack{x\sim U_n \\ C^M}}[C^M(x, f(x))] - \mathop{\mathbf{E}}_{\substack{x\sim U_\ell \\ f_M \\ C^M}}[C^M(x, f_M(x))] \geq \frac{\epsilon}{k}.$$

Proof. For every $i = 0, 1, \ldots, k$, let

$$H_i(\sigma) = (\sigma, f_M(x_1), \ldots f_M(x_i), f(x_{i+1}), \ldots f(x_k)), \tag{9}$$

be the hybrid distribution for $\sigma \sim \{0,1\}^\ell$ and $(x_1, \ldots, x_k) = \mathrm{ND}_{\mathcal{S}}(\sigma)$. Note that $H_0(\sigma) = (\sigma, f^k \circ \mathrm{ND}_{\mathcal{S}}(\sigma))$ and $H_k(\sigma) = (\sigma, f_M^k \circ \mathrm{ND}_{\mathcal{S}}(\sigma))$. Since the circuit D distinguishes the distributions H_0 and H_k, we have

$$\epsilon < \mathop{\mathbf{E}}_{y\sim H_0(U_l)}[D(y) = 1] - \mathop{\mathbf{E}}_{y\sim H_k(U_\ell)}[D(y)] = \sum_{i=1}^{k}\left(\mathop{\mathbf{E}}_{y\sim H_{i-1}(U_\ell)}[D(y)] - \mathop{\mathbf{E}}_{y\sim H_i(U_\ell)}[D(y) = 1]\right).$$

Therefore, $\mathbf{E}_{i\sim[k]}[\mathbf{E}_{y\sim H_{i-1}(U_\ell)}[D(y)] - \mathbf{E}_{y\sim H_i(U_\ell)}[D(y)]] > \epsilon/k$. For $i \in [k]$, partition the random seed $\sigma \sim U_\ell$ into $\sigma = (x, z)$ for $x := \sigma|_{S_i} \sim \{0,1\}^n$ and $z := \sigma|_{[\ell]\setminus S_i} \sim \{0,1\}^{\ell-n}$ so that the distinguishing property of D can be rewritten as

$$\mathop{\mathbf{E}}_{\substack{i\sim[k] \\ (x,z)\sim\{0,1\}^\ell}}\left[\mathop{\mathbf{E}}_{H_{i-1}}[D(H_{i-1}(x,z))] - \mathop{\mathbf{E}}_{H_i}[D(H_i(x,z))]\right] > \epsilon/k.$$

Here, the probability takes the randomness of the probabilistic function f_M inside H_i and H_{i-1} into account. By averaging, we can fix $i \in [k]$ and $z \in \{0,1\}^{[\ell]\setminus S_i}$ satisfying

$$\mathop{\mathbf{E}}_{\substack{x\sim U_n \\ H_{i-1}}}[D(H_{i-1}(x,z))] - \mathop{\mathbf{E}}_{\substack{x\sim U_n \\ H_i}}[D(H_i(x,z))] > \frac{\epsilon}{k}. \tag{10}$$

For each $j \in [k] \setminus \{i\}$, let $E_j\colon \{0,1\}^{S_i} \to \{0,1\}$ be the probabilistic function defined by

$$E_j(x) = \begin{cases} f_M(x|_{S_i\cap S_j}, z|_{S_j\setminus S_i}) & \text{if } j < i, \\ f(x|_{S_i\cap S_j}, z|_{S_j\setminus S_i}) & \text{if } j > i. \end{cases}$$

Note that $E_j(x)$ depends only on $x|_{S_i\cap S_j}$ and the distribution $E_j(U_{S_i})$ coincides with the j-th component of the input drawn from $H_i(U_{S_i}, z)$. By the definition of H_i, the inequality (10) can be rewritten as

$$\begin{aligned}&\mathop{\mathbf{E}}_{\substack{x\sim U_n \\ E_1,\ldots,E_{i-1}}}[D(x, z, E_1(x), \ldots, E_{i-1}(x), f(x), E_{i+1}(x), \ldots, E_k(x))] \\ &\quad - \mathop{\mathbf{E}}_{\substack{x\sim U_n \\ E_1,\ldots,E_{i-1},f_M}}[D(x, z, E_1(x), \ldots, E_{i-1}(x), f_M(x), E_{i+1}(x), \ldots, E_k(x))] > \frac{\epsilon}{k}.\end{aligned} \tag{11}$$

Let C^M be the oracle circuit that, for given input $(x, b) \in \{0,1\}^{n+1}$,

$$C^M(x, b) = D(x, z, E_1(x), \ldots, E_{i-1}(x), b, E_{i+1}(x), \ldots, E_k(x)).$$

The circuit C^M uses the oracle access to M to compute $E_j(x) = f_M(x|_{S_j \cap S_i}, z|_{S_j \setminus S_i})$ for $j < i$ (Note that $f(x|_{S_j \cap S_i}, z|_{S_j \setminus S_i})$ can be computed by a size-s circuit since z is fixed and $\mathcal{S}$ is an s-computational design for f). Therefore, C^M is a size-$O(ks + s_D)$ k-query f-trapdoor oracle circuit. Moreover, C^M has the desired distinguishing property from (11). □

7 Hardness Amplification

The *intersection graph* $G_{\mathcal{S}}$ of a (k, n, ℓ)-collection $\mathcal{S}$ is the graph on vertex set $[k]$ where a pair of vertices $i, j \in [k]$ forms an edge if and only if $S_i \cap S_j \neq \emptyset$.

Theorem 7.1. *Let $n, k, \ell \in \mathbb{N}$ and $\delta \in (0, 1]$ be arbitrary. Let $f\colon \{0,1\}^n \to \{0,1\}$. Let $\mathcal{S}$ be a (k, n, ℓ)-collection that is a s-computational design for f. Let $d_{\max}$ be the maximum degree of the intersection graph of $\mathcal{S}$. Let $g = \oplus_k \circ f^k \circ \mathrm{ND}_{\mathcal{S}}$.*

If f is δ-hard for size s_f, then g is $(1/2 - \epsilon)$-hard for size s_g, where ϵ, s_f and s_g satisfy

$$\epsilon \geq \exp\left(-\frac{\delta k}{d_{\max} + 1}\right),$$
$$s_f \leq (sk + s_g)k^{O(k^2/\epsilon^2)}.$$

7.1 Information-Theoretical Hardness of $\oplus_k \circ f_M^k \circ \mathrm{NW}_{\mathcal{S}}$

Definition 7.2. *For a binary random variable X, let* $\mathrm{Bias}[X] := |\Pr[X = 1] - \Pr[X = 0]|$ *be the* bias *of X. For a probabilistic function $h\colon \{0,1\}^n \to \{0,1\}$, let* $\mathrm{ExpBias}[h] := \mathbf{E}_{x \sim U_n}[\mathrm{Bias}[h(x)]]$. *Here,* Bias *concerns the randomness inside $h(x)$ for every fixed x.*

Lemma 7.3 (e.g., [HVV06])**.** *Any random function f is $(1/2 - \mathrm{ExpBias}[f]/2)$-hard for any size.*

Lemma 7.4. *Let $\mathcal{S}$ be a (k, n, ℓ)-collection and $\alpha(G_{\mathcal{S}})$ be the size of the maximum independent set of $G_{\mathcal{S}}$. Let M be a measure of density δ. Then, for any $k > 0$ and $f\colon \{0,1\}^n \to \{0,1\}$,*

$$\mathrm{ExpBias}[\oplus_k \circ f_M^k \circ \mathrm{ND}_{\mathcal{S}}] \leq \exp(-\delta\alpha(G_{\mathcal{S}})).$$

Proof. Fix arbitrary $x_1, \ldots, x_k \in \{0,1\}^n$ and let $\lambda := \prod_{i \in [k]}(1 - M(x_i))$. Then, we can write the distribution of $\bigoplus_{i \in [k]} f_M(x_i)$ (over the random choice of all $f_M(x_i)$) as the convex combination $\bigoplus_{i \in [k]} f_M(x_i) = \lambda 1_A + (1 - \lambda)U_1$, where 1_A denotes the Dirac measure on $A := \bigoplus_{i \in [k]} f(x_i)$. Identifying a distribution over $\{0,1\}$ as a vector $[0,1]^2$, we have

$$\begin{aligned}
\mathrm{ExpBias}[\oplus_k \circ f_M^k \circ \mathrm{ND}_{\mathcal{S}}] &= \mathop{\mathbf{E}}_{(x_1,\ldots,x_k) = \mathrm{ND}_{\mathcal{S}}(U_\ell)}[\mathrm{Bias}[\|\lambda 1_A + (1 - \lambda)U_1 - U_1\|_1]] \\
&= \mathop{\mathbf{E}}_{x_1,\ldots,x_k}[\lambda \|1_A - U_1\|_1] \\
&\leq 2 \mathop{\mathbf{E}}_{x_1,\ldots,x_k}\left[\prod_{i=1}^{m}(1 - M(x_i))\right].
\end{aligned}$$

Here, $\|\cdot\|_1$ denotes the ℓ^1-norm.

Let $U \subseteq [k]$ be an independent set of $G_{\mathcal{S}}$ of size $\alpha(G_{\mathcal{S}})$. For a random seed $\sigma \in \{0,1\}^{\ell}$, let $(x_1, \ldots, x_k) = \mathrm{ND}_{\mathcal{S}}(\sigma)$. Note that, if $\sigma \sim U_\ell$, the random variables $(x_i)_{i \in U}$ are mutually indenpendent and $x_i \sim U_n$ for every $i \in U$. Therefore, we obtain

$$\begin{aligned}\mathop{\mathbf{E}}_{x_1,\ldots,x_k}\left[\prod_{i\in[k]}(1-M(x_i))\right] &\le \mathop{\mathbf{E}}_{x_1,\ldots,x_k}\left[\prod_{i\in U}(1-M(x_i))\right] \\ &= \prod_{i\in U}\mathop{\mathbf{E}}_{x\sim U_n}[1-M(x)] \\ &= (1-\delta)^{|U|} \le \exp(-|U|\delta).\end{aligned}$$

□

Corollary 7.5. *Under the assumption of Lemma 7.4, the function $\bigoplus_k \circ f_M^k \circ \mathrm{ND}_{\mathcal{S}}$ is $(1/2 - \exp(-\delta k/(d_{\max}+1)))$-hard for any size, where $d_{\max}$ is the maximum degree of the intersection graph $G_{\mathcal{S}}$.*

Proof. Since any k-vertex graph with maximum degree $d_{\max}$ has an independent set of size at least $k/d_{\max}$, the claim follows from Lemmas 7.3 and 7.4 □

7.2 Next-Bit Predictor

We prove that an efficient distinguisher D for the distributions $(x, f(x))$ and $(x, f_M(x))$ yields a small circuit that computes $f(x)$ for $x \sim M$. The proof is identical to the argument of Yao's next-bit predictor [Yao82].

Lemma 7.6. *Let $M\colon \{0,1\}^n \to [0,1]$ be any δ-dense measure. Suppose that there exists a randomized oracle circuit D^M satisfying*

$$\Pr_{\substack{x\sim U_n\\ D^M}}[D^M(x,f(x))=1] - \Pr_{\substack{x\sim U_n\\ f_M\\ D^M}}[D^M(x,f_M(x))=1] \ge \epsilon.$$

Then, the randomized oracle circuit $P^M(x) := 1 \oplus c \oplus D^M(x,c)$ for $c \sim U_1$ satisfies

$$\Pr_{\substack{x\sim M\\ P^M}}[P^M(x)=f(x)] \ge \frac{1}{2} + \frac{\epsilon}{\delta}.$$

Proof. Note that $P^M(x)$ outputs c whenever $D^M(x,c)=1$. Since $c = f(x)$ with probability $1/2$, we have

$$\begin{aligned}\Pr_{\substack{x\sim M\\ P^M}}[P^M(x)=f(x)] &= \frac{1}{2}\Pr[D^M(x,f(x))=1] + \frac{1}{2}\Pr[D^M(x,1-f(x))=0] \\ &= \frac{1}{2} + \frac{1}{2}\mathbf{E}[D^M(x,f(x)) - D^M(x,1-f(x))]\end{aligned}$$

Therefore, it suffices to show $\mathbf{E}_{x\sim M, D^M}[D^M(x,f(x)) - D^M(x,1-f(x))] \ge \frac{2\epsilon}{\delta}$.

For $x \in \{0,1\}^n$, let $b_x \sim \mathrm{Ber}(M(x))$. Consider the distribution (x, b_x) for $x \sim U_n$. Note that the distribution of x conditioned on $b_x = 1$ is M. To see this, for any fixed $a \in \{0,1\}^n$, note that

$$\Pr_{x\sim U_n}[x=a|b_x=1] = \frac{\Pr[b_x=1|x=a]\Pr[x=a]}{\Pr_{x\sim U_n}[b_x=1]} = \frac{M(a)}{\delta 2^n}.$$

We regard b_x as the indicator that $f_M(x) = U_1$ and thus rewrite $f_M(x) = b_x c + (1 - b_x) f(x)$ for $c \sim U_1$. Note that $f(x) = f_M(x)$ whenever $b_x = 0$. From the assumption, we have

$$\begin{aligned}
\epsilon &\leq \mathop{\mathbf{E}}_{x \sim U_n, D^M} [D^M(x, f(x)) - D^M(x, f_M(x))] \\
&= \Pr[b_x = 1] \mathop{\mathbf{E}}_{\substack{x \sim U_n \\ c \sim U_1 \\ D^M}} [D^M(x, f(x)) - D^M(x, c) | b_x = 1] \\
&= \delta \cdot \mathop{\mathbf{E}}_{\substack{x \sim M \\ c \sim U_1 \\ D^M}} [D^M(x, f(x)) - D^M(x, c)] \\
&= \frac{\delta}{2} \mathop{\mathbf{E}}_{\substack{x \sim M \\ D^M}} [D(x, f(x)) - D(x, 1 - f(x))].
\end{aligned}$$

This completes the proof. □

7.3 Putting All Together

We combine Lemmas 5.3, 6.2, 7.4 and 7.6 to prove Theorem 7.1.

Proof of Theorem 7.1. Let M be any δ-dense measure and $\mathcal{S}$ be a (k, n, ℓ)-collection that is a s-computational design for f. Let $d_{\max}$ be the maximum degree of the graph $G_{\mathcal{S}}$. Let $g = \oplus_k \circ f^k \circ \mathrm{ND}_{\mathcal{S}}$ and $g' = \oplus_k \circ f_M^k \circ \mathrm{ND}_{\mathcal{S}}$.

We prove the contraposition. Suppose g is not $(1/2 - \epsilon)$-hard for size s_g. From Lemma 4.1, there is a circuit D of size $s_g + O(1)$ satisfying

$$\mathop{\mathbf{E}}_{\sigma \sim U_\ell} [D(\sigma, g(\sigma))] - \mathop{\mathbf{E}}_{\substack{\sigma \sim U_\ell \\ b \sim U_1}} [D(\sigma, b)] > \epsilon$$

On the other hand, from Corollary 7.5, g' is $\left(\frac{1}{2} - \frac{1}{2}\exp\left(-\frac{\delta k}{d_{\max}+1}\right)\right)$-hard for any size. Hence, the circuit D satisfies

$$\mathop{\mathbf{E}}_{\sigma \sim U_n} [D(\sigma, g'(\sigma))] - \mathop{\mathbf{E}}_{\substack{\sigma \sim U_\ell \\ b \sim U_1}} [D(\sigma, b)] < \frac{1}{2} \exp\left(-\frac{\delta k}{d_{\max} + 1}\right).$$

and thus we have $\mathbf{E}_{\sigma \sim U_\ell} [D(\sigma, g(\sigma))] - \mathbf{E}_{\sigma \sim U_\ell} [D(\sigma, g'(\sigma))] > \epsilon' := \epsilon - \frac{1}{2}\exp\left(-\frac{\delta k}{d_{\max}+1}\right)$. From Lemma 4.2, there is a circuit D' of size $s_g + O(k)$ satisfying

$$\mathop{\mathbf{E}}_{\sigma \sim U_\ell} \left[D'(\sigma, f^k \circ \mathrm{ND}_{\mathcal{S}}(\sigma))\right] - \mathop{\mathbf{E}}_{\sigma \sim U_\ell} \left[D'(\sigma, f_M^k \circ \mathrm{ND}_{\mathcal{S}}(\sigma))\right] > \epsilon'.$$

From Lemma 6.2, there exists a randomized admissible oracle circuit C^M of size $O(sk + s_g)$ that makes k queries and satisfies

$$\mathop{\mathbf{E}}_{\substack{x \sim U_n \\ C^M}} \left[C^M(x, f(x))\right] - \mathop{\mathbf{E}}_{\substack{x \sim U_n \\ C^M}} \left[C^M(x, f_M(x))\right] \geq \frac{\epsilon'}{k}.$$

From Lemma 7.6, there exists a randomized size-$O(sk + s_g)$ k-query f-trapdoor oracle circuit P^M satisfying

$$\Pr_{\substack{x \sim M \\ P^M}} \left[P^M(x) = f(x)\right] \geq \frac{1}{2} + \frac{\epsilon'}{\delta k}.$$

Finally, from (the contraposition of) Lemma 5.3, there exists a size $(sk + s_g) \cdot k^{O(k^2/\epsilon'^2)}$ circuit C satisfying $\Pr_{x\sim U_n}[C(x) = f(x)] \geq 1-\delta$. Note that, if $\epsilon \geq \exp\left(-\frac{\delta k}{d_{\max}+1}\right)$, then $\epsilon' \geq \frac{\epsilon}{2}$. This implies Theorem 7.1. □

8 Triangle Parity on Random Tripartite Graph

We consider the hardness of computing $\mathsf{TriParity}_n(x)$ defined by (1) for $x \sim \{0,1\}^{3n^2}$. Previous works [BSS20; BBB21; Gol20] proves the following worst-case-to-average-case reduction for triangle counting[10].

Theorem 8.1 (Worst-case-to-average-case Reduction)**.** *There exists an absolute constant $\epsilon_0 > 0$ satisfying the following: If there exists a $T(n)$-time algorithm A satisfying*

$$\Pr_{x\sim\{0,1\}^{3n^2}}[A(x) = \mathsf{TriParity}_n(x)] \geq 1 - \epsilon_0$$

for all n, then, there exists an $O(T(n))$-time randomized algorithm A' such that, for any n and any input $x \in \{0,1\}^{3n^2}$,

$$\Pr_{A'}[A'(x) = \mathsf{TriParity}_n(x)] \geq 2/3.$$

This section is devoted to proving the following result.

Theorem 8.2 (Hardness Self Amplification for Triangle Parity)**.** *For any $\delta, \epsilon > 0$, there exists $a = O(\sqrt{\log(1/\epsilon)/\delta})$ satisfying the following: Suppose that there exists a size-s circuit C satisfying*

$$\Pr_{x\sim\{0,1\}^{3n^2}}[C(x) = \mathsf{TriParity}_n(x)] \geq \frac{1}{2} + \delta.$$

Then there exists a circuit C' of size $(n^2 + s)a^{O(a^6/\epsilon^2)}$ such that

$$\Pr_{x\sim\{0,1\}^{3(n/a)^2}}[D(x) = \mathsf{TriParity}_{n/a}(x)] \geq 1 - \epsilon.$$

Proof. We rewrite $\mathsf{TriParity}_n$ as $\mathsf{TriParity}_n = \oplus_k \circ \mathsf{TriParity}^k_{n'} \circ \mathrm{ND}_{\mathcal{S}}$ for a $O(n^2)$-computational design $\mathcal{S}$ for $\mathsf{TriParity}_{n'}$, where $k = a^3$ and $n' = n/a$. Then we apply Theorem 7.1.

Let $a = a(\delta, \epsilon) > 0$ be sufficiently large parameter that will be specified later. We assume a divides n for simplicity. For each $i \in [3]$, divide V_i into a disjoint subsets $V_i^{(1)}, \ldots, V_i^{(a)} \subseteq V_i$ each of size n/a. For each $i_1, i_2, i_3 \in [a]$, let $S_{i_1,i_2,i_3} \subseteq [3n^2]$ be the subset such that the restriction $x|_{S_{i_1,i_2,i_3}} \in \{0,1\}^{3(n/a)^2}$ for input x denotes the edge indicator vector of the induced subgraph of x induced by $V_1^{(i_1)} \cup V_2^{(i_2)} \cup V_3^{(i_3)}$ (see Figure 1). Then we can rewrite $\mathsf{TriParity}_n$ as

$$\begin{aligned}
\mathsf{TriParity}_n(x) &= \bigoplus_{i_1,i_2,i_3\in[a]} \bigoplus_{\substack{v_1\in V_1^{(i_1)}\\ v_2\in V_2^{(i_2)}\\ v_3\in V_3^{(i_3)}}} \prod_{1\leq a<b\leq 3} x[v_a, v_b] \\
&= \bigoplus_{i_1,i_2,i_3\in[a]} \mathsf{TriParity}_{n/a}(x|_{S_{i_1,i_2,i_3}}) \qquad (12)\\
&= \oplus_{a^3} \circ \mathsf{TriParity}^{a^3}_{n/a} \circ \mathrm{ND}_{\mathcal{S}},
\end{aligned}$$

[10] Actually, the results of [BBB21; Gol20] concern an Erdős–Rényi random graph $G(n, 1/2)$ but the same technique works for random tripartite graph. For completeness, we prove Theorem 8.1 in Appendix B using the general local-decoding algorithm of [BSS20]

where $\mathcal{S} := (S_{i_1,i_2,i_3})_{i_1,i_2,i_3\in[a]}$.

We claim that $\mathcal{S}$ is an $O(n^2)$-computational design for $\mathsf{TriParity}_{n/a}$. Fix different triple of indices $(i_1,i_2,i_3),(j_1,j_2,j_3) \in [a]^3$ with $(i_1,i_2,i_3) \neq (j_1,j_2,j_3)$ and fix $z \in \{0,1\}^{S_{i_1,i_2,i_3}\setminus S_{j_1,j_2,j_3}}$. Consider the function $\mathsf{TriParity}_{n/a}(w|z)$ for given $w \in \{0,1\}^{S_{i_1,i_2,i_3}\cap S_{j_1,j_2,j_3}}$, where we recall that $\mathsf{TriParity}_{n/a}(\cdot|z)\colon w \mapsto \mathsf{TriParity}_{n/a}(w,z))$. Note that the string $(w,z) \in \{0,1\}^{S_{i_1,i_2,i_3}}$ represents the subgraph of x induced by $V_{i_1} \cup V_{i_2} \cup V_{i_3}$. Thus the function $\mathsf{TriParity}_{n/a}(w|z)$ equals to the parity of the number of triangle subgraphs contained in the induced subgraph given by S_{i_1,i_2,i_3} but edges inside S_{j_1,j_2,j_3} are fixed.

Suppose that $i_c \neq j_c$ for all $c \in [3]$. Then, $S_{i_1,i_2,i_3} \cap S_{j_1,j_2,j_3} = \emptyset$ and thus $\mathsf{TriParity}_{n/a}(w,z)$ is a constant function. Similarly, $S_{i_1,i_2,i_3} \cap S_{j_1,j_2,j_3} = \emptyset$ if $i_1 = j_1$, $i_2 \neq j_2$, and $i_3 \neq j_3$. In both cases, $\mathsf{TriParity}_{n/a}(w|z)$ can be computed by a constant size circuit.

Suppose that $i_1 = j_1$, $i_2 = j_2$, but $i_3 \neq j_3$. In this case, the input w denotes edges in $E(V_1^{(i_1)}, V_2^{(i_2)})$ and the fixed string z denotes edges in $E(V_1^{(i_1)}, V_3^{(i_3)}) \cup E(V_3^{(i_3)}, V_2^{(i_2)})$, where $E(S,T)$ is the set of edges lying between S and T. Therefore, $\mathsf{TriParity}_{n/a}(w|z)$ is a linear function, i.e.,

$$\mathsf{TriParity}_{n/a}(w|z) = \bigoplus_{u\in V_{i_1}, v\in V_{i_2}} A_{uv} w[u,v]$$

where $A_{uv} \in \{0,1\}$ is a constant that depends on the fixed string z (specifically, A_{uv} is the parity of the number of uv-paths that passes through a vertex in V_3). Therefore, $\mathsf{TriParity}_{n/a}(w|z)$ can be computed by a size $O(n^2)$ circuit and $\mathcal{S}$ is the desired computational design.

Consider the intersection graph G of $\mathcal{S}$. Two vertices $S_{i_1,i_2,i_3}, S_{j_1,j_2,j_3} \in \mathcal{S}$ forms an edge in G if their intersection is nonempty. This occurs if $|\{i_1,i_2,i_3\} \cap \{j_1,j_2,j_3\}| \geq 2$. Therefore, the maximum degree $d_{\max}$ of G is at most $3a$.

Finally, we apply Theorem 7.1. For any $\delta,\epsilon > 0$, take $a = a(\delta,\epsilon) = O(\sqrt{\log(1/\epsilon)/\delta})$ such that $\epsilon \geq \exp(-\delta a^3/(3a+1))$ holds. Then, from the contrapositive of Theorem 7.1, if $\mathsf{TriParity}_n$ can be solved by a size s_g circuit with success probability $1/2+\epsilon$, then $\mathsf{TriParity}_{n/a}$ can be solved by a size s_f circuit with success probability $1-\delta$, where $s_f \leq (n^2a^3 + s_g)\cdot a^{O(a^6/\epsilon^2)} = (n^2+s_g)a^{O(a^6/\epsilon^2)}$. □

Combining Theorems 8.1 and 8.2, we immediately obtain Theorem 1.1.

Theorem 8.3 (Reminder of Theorem 1.1)**.** *For any constant $\delta > 0$, there exists a constant $a = a(\delta) > 0$ satisfying the following: If there exists a size-s circuit C satisfying $\Pr_{x\sim\{0,1\}^{3n^2}}[C(x) = \mathsf{TriParity}_n(x)] \geq 1/2+\delta$, then there exists a randomized circuit C' of size $O(n^2+s)$ satisfying $\Pr_{C'}[C'(x) = \mathsf{TriParity}_n(x)] \geq 2/3$ for every $x \in \{0,1\}^{3n^2}$.*

Proof. Let C be the size-s circuit satisfying $\Pr_{x\sim\{0,1\}^{3n^2}}[C(x) = \mathsf{TriParity}_n(x)] \geq 1/2+\delta$ and $\epsilon_0 > 0$ be the constant mentioned in Theorem 8.1. Then, from Theorem 8.2 with letting $\epsilon = \epsilon_0$, we have a randomized circuit C_1 of size $O(n^2+s)$ satisfying $\Pr_{C_1,x\sim\{0,1\}^{3(n/a)^2}}[C_1(x) = \mathsf{TriParity}_{n/a}(x)] \geq 1-\epsilon_0$ (note that $a = a(\delta,\epsilon)$ is a constant). Then, from Theorem 8.1, we have a circuit C_2 of size $O(n^2+s)$ satisfying $\Pr_{C_2}[C_2(x) = \mathsf{TriParity}_{n/a}(x)] \geq 2/3$. By running C'' for many times (with independent random seeds) and taking majority, we can increase this success probability from $2/3$ to $1-1/(3a^3)$. Finally, we reduce $\mathsf{TriParity}_n$ to $\mathsf{TriParity}_{n/a}$ using (12). Let C' be the circuit defined by $C'(x) = \oplus_{i_1,i_2,i_3\in[a]} C_2(x|_{S_{i_1,i_2,i_3}})$, where S_{i_1,i_2,i_3} is defined in the proof of Theorem 8.2. By the union bound, it holds with probability $2/3$ that $C_2(x|_{S_{i_1,i_2,i_3}}) = \mathsf{TriParity}_{n/a}(x|_{S_{i_1,i_2,i_3}})$ for all $i_1,i_2,i_3 \in [a]$. Therefore, from (12), we have $\Pr_{C'}[C'(x) = \mathsf{TriParity}_n(x)] \geq 2/3$. □

8.1 Can We Extend to k-Clique Counting?

It is known by [BBB21; Gol20] that Theorem 8.1 can be extend to the parity of k-Clique subgraphs in a random k-partite graph (indeed, the result of [BSS20] immediately implies a worst-case-to-average-case reduction for any small subgraph on Erdős–Rényi random graph $G(n, 1/2)$ with a constant error tolerance!). In view of this, it is natural to ask for an extension of Theorem 8.2 to the parity of k-clique or more general graphs.

Unfortunately this is a nontrivial task for our framework. The most difficult part is to construct a computational design. Recall that the proof of Theorem 8.2 is obtained by first constructing a computational design $\mathcal{S}$ and then applying Theorem 7.1. The important property of counting triangle subgraphs is that the function f counting the number of trianles in a tripartite graph becomes linear function if edges are partially fixed, which ensures a computational design. We are not aware of whether we can apply the same argument for counting k-clique subgraphs.

9 Online Vector-Matrix-Vector Multiplication Problem

9.1 Framework

We introduce a formal framework for computational complexity of static data structure problems. A (decision) static data structure problem is specified by a function $f\colon \{0,1\}^m \times \{0,1\}^n \to \{0,1\}$, where the first m bits of input correspond to the static data and the rest n bits correspond to the query. We write $f(x;q)$ to specify the static data x and query q. A circuit $C\colon \{0,1\}^{m+n} \to \{0,1\}$ has *preprocess size* s_{pre}*, data structure size* ℓ*, and answer size* s_{ans} if there exist circuits $C_{\mathrm{pre}}\colon \{0,1\}^m \to \{0,1\}^\ell$ of size s_{pre} and $C_{\mathrm{ans}}\colon \{0,1\}^m \times \{0,1\}^\ell \to \{0,1\}$ of size s_{ans} such that $C(x;q) := C_{\mathrm{ans}}(C_{\mathrm{pre}}(x), q)$ for all $x \in \{0,1\}^m, q \in \{0,1\}^n$. The class of such circuits is denoted by $\mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$. The circuit $C_{\mathrm{pre}}(x)$ can be seen as a preprocess in the sense that C_{pre} outputs a string representing a data structure for given offline input x. Then, C_{ans} receives a query q and the data structure $C_{\mathrm{pre}}(x)$ to compute $f(x;q)$. Note that we do not care update and therefore our framework do not capture dynamic problems. Any circuit $C\colon \{0,1\}^{m+n} \to \{0,1\}$ of size s has preprocess size m, data structure size m, and answer size s by setting $C_{\mathrm{pre}}(x) = x$ and $C_{\mathrm{ans}}(x) = C(x)$. This corresponds to the case that each query is dealt with an offline algorithm (circuit).

Lemma 9.1. *For any $C_1, \dots, C_t \in (s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ and any circuit $D\colon \{0,1\}^t \to \{0,1\}$ of size s_D, the function $(x,q) \mapsto D(C_1(x;q), \dots, C_t(x;q))$ can be computed by a circuit $D' \in \mathcal{C}(ts_{\mathrm{pre}}, t\ell, ts_{\mathrm{ans}} + s_D)$.*

Proof. Write $C_i(x;q) = C^i_{\mathrm{ans}}(C^i_{\mathrm{pre}}(x), q)$ for two circuits C^i_{pre} and C^i_{ans}. Consider $D'_{\mathrm{pre}}(x) := (C^1_{\mathrm{pre}}(x), \dots, C^t_{\mathrm{pre}}(x)) \in \{0,1\}^{t\ell}$ and $D'_{\mathrm{ans}}(y_1, \dots, y_t, q) = D(C^1_{\mathrm{ans}}(y_1, q), \dots, C^t_{\mathrm{ans}}(y_t, q))$. Then, the circuit $D'(x;q) = D'_{\mathrm{ans}}(D'_{\mathrm{pre}}(x), q) \in \mathcal{C}(ts_{\mathrm{pre}}, t\ell, ts_{\mathrm{ans}} + s_D)$ satisfies the claim. □

9.2 Hardness Amplification for Static Data Structure Problems

Consider a function $f\colon \{0,1\}^m \times \{0,1\}^n \to \{0,1\}$. Let $\mathcal{D}_{\mathrm{pre}}, \mathcal{D}_{\mathrm{ans}}$ be distributions over $\{0,1\}^m$ and $\{0,1\}^n$, respectively. We say that f is δ-hard for $\mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ over $(\mathcal{D}_{\mathrm{pre}}, \mathcal{D}_{\mathrm{ans}})$ if, for $C \in \mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$, $\Pr_{x\sim\mathcal{D}_{\mathrm{pre}}, q\sim\mathcal{D}_{\mathrm{ans}}}[C(x;q) = f(x;q)] \le 1 - \delta$. Unless otherwise noted, $\mathcal{D}_{\mathrm{pre}}$ and $\mathcal{D}_{\mathrm{ans}}$ are uniform distributions if we just say that f is δ-hard.

Our hardness amplification result (Theorem 7.1) holds against circuits in $\mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ since the class is closed under taking majority with a slight blow-up of size due to Lemma 9.1. Let

$f\colon \{0,1\}^{m+n} \to \{0,1\}$ and $g\colon \{0,1\}^{m'+n'} \to \{0,1\}$ be functions satisfying

$$g(x;q) = \oplus_k \circ f^k \circ \mathrm{ND}_{\mathcal{S}}(x;q) = \bigoplus_{i\in[k]} f\left(x|_{S'_i}; q|_{S''_i}\right),$$

where $\mathcal{S} = (S_i)_{i\in[k]}$ is a $(k, n+m, n'+m')$-collection, $S'_i = S_i \cap [m']$ and $S''_i = S_i \cap ([m'+n'] \setminus [m'])$ satisfy $|S_i| = m$ and $|S'_i| = n$.

Recall that $\mathcal{S} = (S'_i \cup S''_i)_{i\in[k]}$ is a s-computational design for f if, for every $i \neq j$ and every fixed $w \in \{0,1\}^{S'_i \setminus S'_j}, z \in \{0,1\}^{S''_i \setminus S''_j}$, the function $f\left(x|_{S'_i\cap S'_j}, w; q|_{S''_i\cap S''_j}, z\right) : \{0,1\}^{S'_i\cap S'_j} \times \{0,1\}^{S''_i\cap S''_j} \to \{0,1\}$ can be computed by a circuit $C_{w,z}$ of size s. (Note that the circuit $C_{w,z}$ does not necessarily belong to the class $\mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$.)

Theorem 9.2. *Let $n, k \in \mathbb{N}$ and $\delta \in (0,1]$ be arbitrary. Let $f\colon \{0,1\}^{n+m} \to \{0,1\}$. Let $\mathcal{S}$ be a $(k, n+m, n'+m')$-collection that is a s-computational design for f. Let $d_{\max}$ be the maximum degree of the intersection graph of $\mathcal{S}$. Let $g = \oplus_k \circ f^k \circ \mathrm{ND}_{\mathcal{S}}$.*

If f is δ-hard for $\mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$, then g is $(1/2-\epsilon)$-hard for size $\mathcal{C}(s'_{\mathrm{pre}}, \ell', s'_{\mathrm{ans}})$ for some $s'_{\mathrm{pre}}, \ell', s'_{\mathrm{ans}}$, where the parameters satisfy

$$\begin{aligned}
&\epsilon \geq \exp\left(-\frac{\delta k}{d_{\max}+1}\right),\\
&s_{\mathrm{pre}} = O(\delta^{-2}\epsilon^{-2} s'_{\mathrm{pre}}),\\
&\ell = O(\delta^{-2}\epsilon^{-2}\ell'),\\
&s_{\mathrm{ans}} = (sk + s'_{\mathrm{ans}}k)k^{O(k^2\epsilon^{-2})}.
\end{aligned}$$

Proof Sketch. Since the proof is identical to that of Theorem 7.1 (see Section 7.3) we outline only the sketch here.

Suppose that g is not $(1/2-\epsilon)$-hard for $\mathcal{C}(s'_{\mathrm{pre}}, \ell', s'_{\mathrm{ans}})$. Then, by a slight modification of Lemma 4.1, we obtain a circuit $D \in \mathcal{C}(s'_{\mathrm{pre}}, \ell', s'_{\mathrm{ans}} + O(1))$ satisfying

$$\mathop{\mathbf{E}}_{\sigma\sim U_{n'+m'}}[D(\sigma, g(\sigma))] - \mathop{\mathbf{E}}_{\substack{\sigma\sim U_{n'+m'}\\ b\sim U_1}}[D(\sigma, b)] > \epsilon.$$

Here, for the input $\sigma = (x;q) \in \{0,1\}^{n'+m'}$ and $b \in \{0,1\}$, the circuit D can be written as $D(x;q,b) = D_{\mathrm{ans}}(D_{\mathrm{pre}}(x), q, b)$. Specifically, the circuit $D(x;q,b)$ outputs 1 if $g(x;q) = b$ and the uniform random bit otherwise.

For a δ-dense measure $M\colon \{0,1\}^{n+m} \to [0,1]$, let $g'(x;q)$ be the function defined by $g'(x;q) := \oplus_k \circ f_M^k \circ \mathrm{ND}_{\mathcal{S}} = \bigoplus_{i\in[k]} f_M\left(x|_{S'_i}; q|_{S''_i}\right)$, where f_M is the random function defined by $f_M(a;b) = f(a;b)$ with probability $1 - M(a,b)$ and $f_M(a;b) = U_1$ with probability $M(a,b)$ (recall that U_1 denotes the uniform random bit). Then, by the information-theoretical bound, the function g' is $\left(\frac{1}{2} - \frac{1}{2}\exp\left(-\frac{\delta k}{d_{\max}+1}\right)\right)$-hard for any size. Therefore, the circuit D distinguishes $g(\sigma)$ and $g'(\sigma)$, i.e.,

$$\mathop{\mathbf{E}}_{\sigma}[D(\sigma, g(\sigma)] - \mathop{\mathbf{E}}_{\sigma}[D(\sigma, g'(\sigma)] > \epsilon' := \epsilon - \frac{1}{2}\exp\left(-\frac{\delta k}{d_{\max}+1}\right).$$

Then, by Lemma 4.2, we obtain a circuit $D' \in \mathcal{C}(s'_{\mathrm{pre}}, \ell, s'_{\mathrm{ans}} + O(k))$ satisfying

$$\mathop{\mathbf{E}}_{\sigma}[D'(\sigma, f^k \circ \mathrm{ND}_{\mathcal{S}}(\sigma)] - \mathop{\mathbf{E}}_{\sigma}[D'(\sigma, f_M^k \circ \mathrm{ND}_{\mathcal{S}}(\sigma)] > \frac{\epsilon}{2},$$

where D' is of the form $D'(x;q,b_1,\dots,b_k) = D'_{\mathrm{ans}}(D'_{\mathrm{pre}}(x),q,b_1,\dots,b_k)$.

From the proof of Lemma 6.2, we have a trapdoor oracle circuit $C^M \in \mathcal{C}(s'_{\mathrm{pre}},\ell,s'_{\mathrm{ans}}+O(ks))$ that makes at most k queries and satisfies

$$\mathop{\mathbf{E}}_{\substack{(x;q)\sim U_{n+m}\\ C^M}}[C^M(x;q,f(x,q))] - \mathop{\mathbf{E}}_{\substack{(x;q)\sim U_{n+m}\\ C^M}}[C^M(x;q,f_M(x,q))] > \frac{\epsilon'}{k}.$$

Note that, in the proof of Lemma 6.2, the circuit C^M is of the form

$$\begin{aligned} C^M(x;q,b) &= D'(x;q,C_1(x;q),\dots,C_{i-1}(x;q),b,C_{i+1}(x;q),\dots,C_k(x;q)) \\ &= D'_{\mathrm{ans}}(D'_{\mathrm{pre}}(x),q,C_1(x;q),\dots,C_{i-1}(x;q),b,C_{i+1}(x;q),\dots,C_k(x;q)). \end{aligned}$$

where C_i are size-s circuits (since $\mathcal{S}$ is a computational design). This construction blows up only the "answer" part.

From Lemma 7.6, we have a randomized trapdoor oracle circuit $P^M \in \mathcal{C}(s'_{\mathrm{pre}},\ell,s'_{\mathrm{ans}}+O(ks))$ satisfying

$$\Pr_{\substack{(x;q)\sim M\\ P^M}}[P^M(x;q) = f(x;q)] \geq \frac{1}{2} + \frac{\epsilon'}{\delta k}.$$

Finally, from the proof of Lemma 5.3, there exists a circuit $C \in \mathcal{C}(s_{\mathrm{pre}},\ell,s_{\mathrm{ans}})$ that approximates f well over the uniform distribution, where

$$\begin{aligned} s_{\mathrm{pre}} &= O(ts'_{\mathrm{pre}}), \\ \ell &= O(t\ell'), \\ s_{\mathrm{ans}} &= (sk + s'_{\mathrm{ans}}k)\cdot k^{O(t)} \end{aligned}$$

for some $t = O(k^2\epsilon^{-2})$. Note that, in the proof of Lemma 5.3, we claimed that the circuit $C' = \mathsf{maj}(C_1,\dots,C_t)$ performs well where $C_1,\dots,C_t$ are circuits chosen by a suitable way. If $C_1,\dots,C_t \in \mathcal{C}(s_{\mathrm{pre}},\ell,s_{\mathrm{ans}})$, then $C' \in \mathcal{C}(ts_{\mathrm{pre}},t\ell,ts_{\mathrm{ans}}+O(t))$ from Lemma 9.1. □

9.3 Hardness Amplification for OuMv

Let OuMv_n be the function defined by (2). With this notation, the (circuit) OuMv Conjecture can be stated as follows: Any circuit $C \in \mathcal{C}(s_{\mathrm{pre}},\ell,s_{\mathrm{ans}})$ of $C(M;u,v) = \mathsf{OuMv}_n(M,u,v)$ satisfies $s_{\mathrm{pre}} + ns_{\mathrm{ans}} = n^{3-o(1)}$.

We consider the average-case complexity of computing OuMv_n where the matrix $M \sim \{0,1\}^{n\times n}$ and vectors $u,v \sim \{0,1\}^n$ are uniformly at random.

For completeness, we prove the following worst-case-to-average-case reduction (on uniform computational model) for this problem that was already given by Henzinger et al. [HLS22]. The argument is essentially based on [BLR93].

Theorem 9.3. *Suppose that there is a pair $(A_{\mathrm{pre}},A_{\mathrm{ans}})$ of algorithms runs in time $T_{\mathrm{pre}}(n)$ and $T_{\mathrm{ans}}(n)$ respectively satisfying*

$$\Pr_{M\sim\{0,1\}^{n\times n},u,v\sim\{0,1\}^n}[A_{\mathrm{ans}}(A_{\mathrm{pre}}(M),u,v) = \mathsf{OuMv}_n(M,u,v)] \geq 1-\epsilon$$

for all $n \in \mathbb{N}$. Then there is a pair $A' = (A'_{\mathrm{pre}}, A'_{\mathrm{ans}})$ of randomized algorithms runs in time $O(T_{\mathrm{pre}}(n))$ and $O(T_{\mathrm{ans}}(n))$ respectively satisfying

$$\Pr_{A'_{\mathrm{pre}}, A'_{\mathrm{ans}}}[A'_{\mathrm{ans}}(A'_{\mathrm{pre}}(M), u, v) = \mathsf{OuMv}_n(M, u, v)] \geq 1 - 8\epsilon$$

for all $n \in \mathbb{N}$ and $M \in \{0,1\}^{n \times n}, u, v \in \{0,1\}^n$.

Moreover, if the output of A_{pre} has length at most $\ell(n)$, then that of A'_{pre} is at most $2\ell(n)$.

Proof. Let $A_{\mathrm{pre}}, A_{\mathrm{ans}}$ be the algorithms satisfying the assumption. The randomized algorithm $A'_{\mathrm{pre}}(M)$ given input M runs as follows: Sample $R_1 \sim \{0,1\}^{n \times n}$ and let $R_2 = M - R_1$ and output $(A_{\mathrm{pre}}(R_1), A_{\mathrm{pre}}(R_2))$. The randomized algorithm $A'_{\mathrm{ans}}(u, v)$ given input $u, v \in \{0,1\}^n$ runs as follows: Sample $r_1, r'_1 \sim \{0,1\}^n$ independently and let $r_2 = u - r_1$ and $r'_2 = v - r'_1$. Then, output $\sum_{i \in [2], j \in [2], k \in [2]} A_{\mathrm{ans}}(A_{\mathrm{pre}}(R_i), r_j, r'_k)$. By the union bound, with probability $1 - 8\epsilon$, we have $A_{\mathrm{ans}}(A_{\mathrm{pre}}(R_i), r_j, r'_k) = \mathsf{OuMv}_n(R_i, r_j, r'_k)$ for all $i, j, k \in [2]$. If this holds, we have $A'_{\mathrm{ans}}(A'_{\mathrm{pre}}(M), u, v) = \sum_{i,j,k \in [2]} \mathsf{OuMv}_n(R_i, r_j, r'_k) = \mathsf{OuMv}_n(M, u, v)$. □

Theorem 9.4. *For any $\delta, \epsilon > 0$, there exists $a = a(\delta, \epsilon)$ satisfying the following: Suppose there exists a circuit $C \in \mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ such that*

$$\Pr_{M,u,v}[C(M; u, v) = \mathsf{OuMv}_n(M, u, v)] \geq \frac{1}{2} + \delta.$$

Then, there exists a circuit $C' \in \mathcal{C}(s'_{\mathrm{pre}}, \ell', s'_{\mathrm{ans}})$ such that

$$\Pr_{\substack{M \sim \{0,1\}^{(n/a) \times (n/a)} \\ u,v \sim \{0,1\}^{n/a}}}[C'(M; u, v) = \mathsf{OuMv}_{n/a}(M, u, v)] \geq 1 - \epsilon,$$

where $s'_{\mathrm{pre}} = O(a^4 \epsilon^{-2} s_{\mathrm{pre}})$, $\ell' = O(a^4 \epsilon^{-2} \ell)$, and $s'_{\mathrm{ans}} = O(n + s_{\mathrm{ans}}) a^{O(a^4/\epsilon^2)}$.

Proof. We rewrite OuMv_n as $\mathsf{OuMv}_n = \oplus_k \circ (\mathsf{OuMv}_{n'})^k \circ \mathrm{ND}_{\mathcal{S}}$ for an $O(n)$-computational design $\mathcal{S}$ for $\mathsf{OuMv}_{n'}$, where $k = a^2$ and $n' = n/a$. Then we apply Theorem 9.2.

Let $a = a(\delta, \epsilon)$ be a sufficiently large parameter that will be specified later. We assume a divides n for simplicity. Let R, C denote the row and column set of the given matrix M. Then, we have $u \in \{0,1\}^R, v \in \{0,1\}^C$, and $M \in \{0,1\}^{R \times C}$. Let $R_1, \ldots, R_a$ and $C_1, \ldots, C_a$ be a partition of R and C such that each R_i and C_i has the same size n/a, respectively. Let $M_{i,j} = M|_{R_i \times C_j} \in \{0,1\}^{R_i \times C_j}$ be the submatrix of M given by $R_i \times C_j$. Similarly, for given vectors $u \in \{0,1\}^R$ and $v \in \{0,1\}^C$, let $u_i = u|_{R_i} \in \{0,1\}^{R_i}$ and $v_j = v|_{C_j} \in \{0,1\}^{C_j}$ be the restriction of u on R_i and v on C_j, respectively. Then, we have

$$\mathsf{OuMv}_n(M, u, v) = \bigoplus_{i,j \in [a]} \mathsf{OuMv}_{n/a}(M_{i,j}, u_i, v_j) = \oplus_k \circ \mathsf{OuMv}^k_{n/a} \circ \mathrm{ND}_{\mathcal{S}}, \tag{13}$$

where $\mathcal{S} = (S_{i,j})_{i,j \in [a]}$ for $S_{i,j} = (R_i \times C_j) \cup R_i \cup C_j \subseteq (R \times C) \cup R \cup C$.

We claim that $\mathcal{S}$ defined above is an $O(n)$-computational design for $\mathsf{OuMv}_{n'}$. Fix distinct $S_{i,j}, S_{i',j'} \in \mathcal{S}$. Our task is to compute $\mathsf{OuMv}_{n/a}(M_{i,j}, u_i, v_j) = u_i^\top M_{i,j} v_j$ by an $O(n)$-size circuit where the input is partially hardwired. If $i \neq i'$ and $j \neq j'$, then $S_{i,j} \cap S_{i',j'} = \emptyset$ and thus the input is completely hardwired. If $i = i'$ and $j \neq j'$, then $M_{i,j}$ and v_j are hardwired. In this case, the function $\mathsf{OuMv}_{n/a}(M_{i,j}, u_i, v_j) = u_i^\top w$ for a fixed w is a linear function and thus computed by an $O(n)$-size circuit. Therefore, $\mathcal{S}$ is the desired computational design.

Consider the intersection graph $G_{\mathcal{S}}$ of $\mathcal{S}$. Two distinct vertices $S_{i,j}, S_{i',j'}$ form an edge if $i = i'$ or $j = j'$. Therefore, $G_{\mathcal{S}}$ has maximum degree at most $2a$.

Finally, we apply Theorem 9.2. For any $\delta, \epsilon > 0$, take $a = a(\delta, \epsilon) = O(\delta^{-1} \log(1/\epsilon))$ such that $\epsilon > \exp(-\delta a^2/(2a+1))$ holds. Then, from the contrapositive of Theorem 9.2, we obtain the claim. □

Combining Theorems 9.3 and 9.4, we obtain Theorem 1.2.

Theorem 9.5 (Reminder of Theorem 1.2)**.** *For any constant $\delta > 0$, there exists a constant $a = a(\delta) > 0$ satisfying the following: If there exists a circuit $C \in \mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ satisfying $\Pr_{(M,u,v)\sim\{0,1\}^{n^2+2n}}[C(M;u,v) = \mathsf{OuMv}_n(M,u,v)] \geq 1/2+\delta$, then there exists a randomized circuit $C' \in \mathcal{C}(O(s_{\mathrm{pre}}), O(\ell), O(n+s_{\mathrm{ans}}))$ satisfying $\Pr_{C'}[C(M;u,v) = \mathsf{OuMv}_n(M,u,v)] \geq 2/3$ for every $(M,u,v) \in \{0,1\}^{n\times n} \times \{0,1\}^{2n}$.*

Proof. Let $C \in \mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ be the circuit of the assumption, i.e., $\Pr_{(M,u,v)\sim U_{n^2+2n}}[C(M;u,v) = \mathsf{OuMv}_n(M,u,v)] \geq 1/2+\delta$. Then, from Theorem 9.4 with letting $\epsilon = 0.01$, we have a circuit $C_1 \in \mathcal{C}(O(s_{\mathrm{pre}}, O(\ell), O(n+s_{\mathrm{ans}}))$ satisfying $\Pr_{(M,u,v)}[C_1(M;u,v) = \mathsf{OuMv}_{n/a}(M,u,v)] \geq 0.99$, where $M \sim \{0,1\}^{(n/a)\times(n/a)}$ and $u, v \sim \{0,1\}^{n/a}$ (note that $a = a(\delta,\epsilon)$ is a constant). Finally, from Theorem 9.3, we have a circuit C_2 such that $\Pr_{C_2}[C_2(M;u,v) = \mathsf{OuMv}_{n/a}(M,u,v)] \geq 2/3$ for any input $(M,u,v) \in \{0,1\}^{(n/a)\times(n/a)} \times \{0,1\}^{2(n/a)}$. By repetition, we can amplify this success probability and thus we may assume that the success probability of C_2 is $1 - 1/(3a^2)$. Note that this repetition occurs a linear-blows in size parameters $s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}}$ by Lemma 9.1. Let C' be the circuit defined by $C'(M;u,v) = \oplus_{i,j\in[a]} C_2(M_{i,j};u_i,v_j)$. By the union bound and (13), we have $\Pr_{C'}[C'(M;u,v) = \mathsf{OuMv}_n(M,u,v)] \geq 2/3$ for any input M,u,v. □

9.4 Hardness Amplification for OuMv_k over $\mathbb{F}_2$

Let $\mathsf{OuMv}^{(k)}$ be the function defined by (3). We consider the average-case complexity of computing $\mathsf{OuMv}^{(k)}$ for random tensor $M \sim \{0,1\}^{n^k}$ and vectors $u_1, \ldots, u_k \sim \{0,1\}^n$.

The following worst-case-to-average-case reduction is an immediate extension of Theorem 9.3.

Theorem 9.6. *Suppose that there is a pair $(A_{\mathrm{pre}}, A_{\mathrm{ans}})$ of algorithms runs in time $T_{\mathrm{pre}}(n)$ and $T_{\mathrm{ans}}(n)$ respectively satisfying*

$$\Pr_{M\sim\{0,1\}^{n^k}, u_1,\ldots,u_k\sim\{0,1\}^n}[A_{\mathrm{ans}}(A_{\mathrm{pre}}(M), u_1, \ldots, u_k) = \mathsf{OuMv}_n^{(k)}(M, u_1, \ldots, u_k)] \geq 1-\epsilon$$

for all $n \in \mathbb{N}$. Then there is a pair $A' = (A'_{\mathrm{pre}}, A'_{\mathrm{ans}})$ of randomized algorithms runs in time $O(2^k T_{\mathrm{pre}}(n))$ and $O(2^k T_{\mathrm{ans}}(n))$ respectively satisfying

$$\Pr_{A'_{\mathrm{pre}}, A'_{\mathrm{ans}}}[A'_{\mathrm{ans}}(A'_{\mathrm{pre}}(M), u_1, \ldots, u_k) = \mathsf{OuMv}_n(M, u_1, \ldots, u_k)] \geq 1 - 2^{k+1}\epsilon$$

for all $n \in \mathbb{N}$ and $M \in \{0,1\}^{n^k}, u_1, \ldots, u_k \in \{0,1\}^n$.

Moreover, if the output of A_{pre} has length at most $\ell(n)$, then that of A'_{pre} is at most $2\ell(n)$.

Proof. Let $A_{\mathrm{pre}}, A_{\mathrm{ans}}$ be the algorithms satisfying the assumption. The randomized algorithm $A'_{\mathrm{pre}}(M)$ given input M runs as follows: Sample $R_0 \sim \{0,1\}^{n^k}$ and let $R_1 = M - R_0$ and output $(A_{\mathrm{pre}}(R_0), A_{\mathrm{pre}}(R_1))$. The randomized algorithm $A'_{\mathrm{ans}}(u,v)$ given input $u_1, \ldots, u_k \in \{0,1\}^n$ runs as follows: Sample $r_0^{(1)}, \ldots, r_0^{(k)} \sim \{0,1\}^n$ independently and let $r_1^{(i)} = u_i - r_0^{(i)}$. Then, output

$\sum_{(i_0,\dots,i_k)\in\{0,1\}^{k+1}} A_{\mathrm{ans}}(A_{\mathrm{pre}}(R_{i_0}), r^{(1)}_{i_1},\dots,r^{(k)}_{i_k})$. By the union bound, with probability at least $1-2^{k+1}\epsilon$, we have $A_{\mathrm{ans}}(A_{\mathrm{pre}}(R_{i_0}), r^{(1)}_{i_1},\dots,r^{(k)}_{i_k}) = \mathsf{OuMv}^{(k)}_n(R_{i_0}, r^{(1)}_{i_1},\dots,r^{(k)}_{i_k})$ for all $(i_0,\dots,i_k)\in\{0,1\}^{k+1}$. If this holds, the output of A'_{ans} is correct. □

Theorem 9.7. *For any $\delta,\epsilon>0$, there exists $a=a(\delta,\epsilon)$ satisfying the following: Suppose there exists a circuit $C\in\mathcal{C}(s_{\mathrm{pre}},\ell,s_{\mathrm{ans}})$ such that*

$$\Pr_{M,u_1,\dots,u_k}[C(M;u_1,\dots,u_k)=\mathsf{OuMv}^{(k)}_n(M,u_1,\dots,u_k)]\ge\frac{1}{2}+\delta.$$

Then, there exists a circuit $C'\in\mathcal{C}(s'_{\mathrm{pre}},\ell',s'_{\mathrm{ans}})$ such that

$$\Pr_{\substack{M\sim\{0,1\}^{(n/a)^k}\\ u_1,\dots,u_k\sim\{0,1\}^{n/a}}}[C'(M;u_1,\dots,u_k)=\mathsf{OuMv}^{(k)}_{n/a}(M,u_1,\dots,u_k)]\ge 1-\epsilon,$$

where $s'_{\mathrm{pre}}=O(a^4\epsilon^{-2}s_{\mathrm{pre}})$, $\ell'=O(a^4\epsilon^{-2}\ell)$, and $s'_{\mathrm{ans}}=O(n^{k-1}+s_{\mathrm{ans}})a^{O(a^4/\epsilon^2)}$.

Proof. The proof is almost identical to that of Theorem 9.4. We rewrite $\mathsf{OuMv}^{(k)}_n$ as $\mathsf{OuMv}^{(k)}_n = \oplus_{a^k}\circ(\mathsf{OuMv}^{(k)}_{n'})^{a^k}\circ\mathrm{ND}_{\mathcal{S}}$ for an $O(n^{k-1})$-computational design $\mathcal{S}$ for $\mathsf{OuMv}^{(k)}_{n'}$. Then we apply Theorem 9.2.

Let $a=a(\delta,\epsilon)$ be a sufficiently large parameter that will be specified later. We assume a divides n for simplicity. Let $I_1\times\cdots\times I_k$ denote the index set of the given matrix M. Thus we have $M\in\{0,1\}^{I_1\times\cdots\times I_k}$, $u_i\in\{0,1\}^{I_i}$ for $i\in[k]$. Divide each I_i into a sets $I_{i,1},\dots,I_{i,a}$ each of size n/a. For each $\mathbf{j}=(j_1,\dots,j_k)\in[a]^k$, let $M_{\mathbf{j}}=M|_{I_{1,j_1}\times\cdots\times I_{k,j_k}}\in\{0,1\}^{I_{1,j_1}\times\cdots\times I_{k,j_k}}$ be the subtensor of M induced by $I_{1,j_1}\times\cdots\times I_{k,j_k}$. Similarly, for given vectors $u_i\in\{0,1\}^{I_i}$ and $j\in[a]$, let $u_{i,j}=u_i|_{I_{i,j}}\in\{0,1\}^{I_{i,j}}$ the restriction of u_i on $I_{i,j}$. Then, we have

$$\begin{aligned}\mathsf{OuMv}^{(k)}_n(M,u_1,\dots,u_k)&=\bigoplus_{\mathbf{j}=(j_1,\dots,j_k)\in[a]^k}\mathsf{OuMv}^{(k)}_{n/a}(M_{\mathbf{j}},u_{1,j_1},\dots,u_{k,j_k})\\&=\oplus_{a^k}\circ(\mathsf{OuMv}^{(k)}_n)^{a^k}\circ\mathrm{ND}_{\mathcal{S}},\end{aligned}\tag{14}$$

where $\mathcal{S}=(S_j)_{j\in[a]^k}$ for $S_{j_1,\dots,j_k}=(I_{1,j_1}\times\cdots\times I_{k,j_k})\cup I_{1,j_1}\cup\cdots\cup I_{k,j_k}$.

We claim that $\mathcal{S}$ defined above is an $O(n^{k-1})$-computational design for OuMv_n. Fix distinct $S_{\mathbf{j}},S_{\mathbf{j}'}\in\mathcal{S}$. We claim that $\mathsf{OuMv}^{(k)}_{n/a}(M_{\mathbf{j}},u_{1,j_1},\dots,u_{k,j_k})$ can be computed by an $O(n^{k-1})$-size circuit if the bits indexed by $S_{\mathbf{j}}\setminus S_{\mathbf{j}'}$ of the input is hardwired. In this setting, $M_{\mathbf{j}}$ is hardwired since $\mathbf{j}\neq\mathbf{j}'$. Moreover, there is an index $l\in[k]$ satisfying $j_l\neq j'_l$. For simplicity, suppose $l=k$ for simplicity (the case of other l can be dealt with the same way). Then, $u_{k,j_k}\in\{0,1\}^{n/a}$ is hardwired. Let $W\in\{0,1\}^{(n/a)^{k-1}}$ be the rank-$(k-1)$ tensor satisfying $W(i_1,\dots,i_{k-1})=\sum_{i\in[n/a]}M_{\mathbf{j}}(i_1,\dots,i_{k-1},x)u_{k,j_k}(i)$. Note that W can be hardwired with $(n/a)^{k-1}=O(n^{k-1})$ bits and thus

$$\mathsf{OuMv}^{(k)}_{n/a}(M_{\mathbf{j}},u_{1,j_1},\dots,u_{k,j_k})=\bigoplus_{(i_1,\dots,i_{k-1})\in[n/a]^{k-1}}W(i_1,\dots,i_{k-1})\prod_{s\in[k-1]}u_{s,j_s}$$

can be computed by an $O(n^{k-1})$-size circuit. Therefore, $\mathcal{S}$ is the desired computational design.

Consider the intersection graph $G_{\mathcal{S}}$ of $\mathcal{S}$. Two distinct vertices $S_{\mathbf{j}},S_{\mathbf{j}'}$ form an edge if and only if $j_i=j'_i$ for some $i\in[k]$. Therefore, $G_{\mathcal{S}}$ has maximum degree at most ka.

Finally, we apply Theorem 9.2. For any $\delta,\epsilon>0$, take $a=a(\delta,\epsilon)$ such that $\epsilon>\exp(-\delta a^k/(ka+1))$ holds. Then, from the contrapositive of Theorem 9.2, we obtain the claim. □

Theorem 9.8 (Reminder of Theorem 1.3)**.** *Let $\delta > 0$ be any constant. Suppose that there exists a circuit $C \in \mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ satisfying*

$$\Pr_{(M,u_1,\ldots,u_k)\sim\{0,1\}^{n^k+kn}}[C(M;u_1,\ldots,u_k) = \mathsf{OuMv}_n^{(k)}(M,u_1,\ldots,u_k)] \geq 1/2+\delta.$$

Then, there exists a randomized circuit $C' \in \mathcal{C}(O(s_{\mathrm{pre}}), O(\ell), O(n^{k-1}+s_{\mathrm{ans}}))$ satisfying

$$\Pr_{C'}[C'(M;u_1,\ldots,u_k) = \mathsf{OuMv}_n^{(k)}(M,u_1,\ldots,u_k)] \geq 2/3$$

for every $(M,u_1,\ldots,u_k) \in \{0,1\}^{n^k} \times \{0,1\}^{kn}$.

Proof. Let $C \in \mathcal{C}(s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}})$ be the circuit of the assumption. From Theorem 9.7 with $\epsilon = 0.1 \times 2^{-k-1}$, we have a circuit $C_1 \in \mathcal{C}(O(s_{\mathrm{pre}}, O(\ell), O(n^{k-1}+s_{\mathrm{ans}}))$ satisfying $\Pr_{(M,\mathbf{u})}[C_1(M;\mathbf{u}) = \mathsf{OuMv}_{n/a}^{(k)}(M,\mathbf{u})] \geq 1-0.1\times 2^{-k-1}$, where $M \sim \{0,1\}^{(n/a)^k}$ and $\mathbf{u} = (u_1,\ldots,u_k) \sim \{0,1\}^{k(n/a)}$ (note that $a = a(\delta,\epsilon)$ is a constant). From Theorem 9.6, we have a circuit $C_2 \in \mathcal{C}(O(s_{\mathrm{pre}}, O(\ell), O(n^{k-1}+s_{\mathrm{ans}}))$ satisfying $\Pr_{C_2}[C_2(M;\mathbf{u}) = \mathsf{OuMv}_{n/a}^{(k)}(M,\mathbf{u})] \geq 0.9$ for any input $(M,\mathbf{u}) \in \{0,1\}^{(n/a)^k} \times \{0,1\}^{k(n/a)}$. By repetition, we can amplify this success probability and thus we may assume that the success probability of C_2 is $1-1/(3a^k)$. Note that this repetition occurs a linear-blows in size parameters $s_{\mathrm{pre}}, \ell, s_{\mathrm{ans}}$ by Lemma 9.1. Let C' be the circuit defined by $C'(M;\mathbf{u}) = \oplus_{\mathbf{j}\in[a]^k} C_2(M_{\mathbf{j}}; u_{1,j_1},\ldots,u_{k,j_k})$. By the union bound over $\mathbf{j} \in [a]^k$ and (13), we have $\Pr_{C'}[C'(M;\mathbf{u}) = \mathsf{OuMv}_n^{(k)}(M,\mathbf{u})] \geq 2/3$ for any input $M,\mathbf{u}$. □

References

[AGGS22] Vahid R. Asadi, Alexander Golovnev, Tom Gur, and Igor Shinkar. "Worst-Case to Average-Case Reductions via Additive Combinatorics". In: *arXiv 2202.08996 (to appear in STOC22)* (2022). URL: https://arxiv.org/abs/2202.08996.

[AW21] Josh Alman and Virginia Vassilevska Williams. "A Refined Laser Method and Faster Matrix Multiplication". In: *Proceedings of Symposium on Discrete Algorithms (SODA)* (2021), pp. 522–539. DOI: 10.1137/1.9781611976465.32. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611976465.32.

[BBB21] Enric Boix-Adserá, Matthew Brennan, and Guy Bresler. "The Average-Case Complexity of Counting Cliques in Erdös–Rényi Hypergraphs". In: *SIAM Journal on Computing (Special Section FOCS2019)* (2021), pp. 39–80. ISSN: 0097-5397. DOI: 10.1137/20M1316044.

[BFNW93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. "BPP Has Subexponential Time Simulations Unless EXPTIME has Publishable Proofs". In: *Computational Complexity* 3 (1993), pp. 307–318. DOI: 10.1007/BF01275486.

[BL15] Abhishek Bhowmick and Shachar Lovett. "The List Decoding Radius of Reed-Muller Codes over Small Fields". In: *Proceedings of Symposium on Theory of Computing (STOC)* (2015), pp. 277–285. DOI: 10.1145/2746539.2746543.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. "Self-testing/correcting with applications to numerical problems". In: *Journal of Computer and System Sciences* 47 (3 1993), pp. 549–595. ISSN: 00220000. DOI: 10.1016/0022-0000(93)90044-W. URL: https://www.sciencedirect.com/science/article/pii/002200009390044W.

[BM84] Manuel Blum and Silvio Micali. "How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits". In: *SIAM Journal on Computing* 13.4 (1984), pp. 850–864. DOI: 10.1137/0213053.

[BRSV17] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. "Average-case fine-grained hardness". In: *Proceedings of Symposium on Theory of Computing (STOC)* (2017), pp. 483–496. DOI: 10.1145/3055399.3055466.

[BSS20] Mitali Bafna, Srikanth Srinivasan, and Madhu Sudan. "Local decoding and testing of polynomials over grids". In: *Random Structures & Algorithms* 57 (3 2020), pp. 658–694. ISSN: 1042-9832. DOI: 10.1002/rsa.20933.

[BT06] Andrej Bogdanov and Luca Trevisan. "Average-Case Complexity". In: *Foundations and Trends in Theoretical Computer Science* 2.1 (2006). DOI: 10.1561/0400000004.

[CKL18] Diptarka Chakraborty, Lior Kamma, and Kasper Green Larsen. "Tight cell probe bounds for succinct Boolean matrix-vector multiplication". In: *Proceedings of Symposium on Theory of Computing (STOC)* (2018), pp. 1297–1306. DOI: 10.1145/3188745.3188830. URL: https://dl.acm.org/doi/10.1145/3188745.3188830.

[CKLM18] Arkadev Chattopadhyay, Michal Koucký, Bruno Loff, and Sagnik Mukhopadhyay. "Simulation beats richness: new data-structure lower bounds". In: *Proceedings of Symposium on Theory of Computing (STOC)* (2018), pp. 1013–1020. DOI: 10.1145/3188745.3188874.

[CPS99] Jin-Yi Cai, A. Pavan, and D. Sivakumar. "On the Hardness of Permanent". In: *Proceedings of Symposium on Theoretical Aspects of Computer Science (STACS)* (1999), pp. 90–99. DOI: 10.1007/3-540-49116-3_8. URL: https://link.springer.com/chapter/10.1007/3-540-49116-3_8.

[DLW20] Mina Dalirrooyfard, Andrea Lincoln, and Virginia Vassilevska Williams. "New Techniques for Proving Fine-Grained Average-Case Hardness". In: *Proceedings of Symposium on Foundations of Computer Science (FOCS)* (2020), pp. 774–785. DOI: 10.1109/FOCS46700.2020.00077.

[FL92] Uriel Feige and Carsten Lund. "On the hardness of computing the permanent of random matrices (extended abstract)". In: *Proceedings of Symposium on Theory of Computing (STOC)* (1992), pp. 643–654. DOI: 10.1145/129712.129775.

[GKZ08] Parikshit Gopalan, Adam R. Klivans, and David Zuckerman. "List-decoding reed-muller codes over small fields". In: *Proceedings of Symposium on Theory of Computing (STOC)* (2008), pp. 265–274. DOI: 10.1145/1374376.1374417.

[GL89] O. Goldreich and L. A. Levin. "A hard-core predicate for all one-way functions". In: *Proceedings of Symposium on Theory of Computing (STOC)* (1989), pp. 25–32. DOI: 10.1145/73007.73010.

[GLRSW91] Peter Gemmell, Richard Lipton, Ronitt Rubinfeld, Madhu Sudan, and Avi Wigderson. "Self-testing/correcting for polynomials and for approximate functions". In: *Proceedings of Symposium on Theory of Computing (STOC)* (1991), pp. 33–42. DOI: 10.1145/103418.103429.

[GNW11] Oded Goldreich, Noam Nisan, and Avi Wigderson. "On Yao's XOR-Lemma". In: *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation* (2011), pp. 273–301. DOI: 10.1007/978-3-642-22670-0_23.

[Gol20] Oded Goldreich. "On Counting t-Cliques Mod 2". In: *ECCC TR20-104* (2020).

[GR18] Oded Goldreich and Guy Rothblum. "Counting t-Cliques: Worst-Case to Average-Case Reductions and Direct Interactive Proof Systems". In: *Proceedings of Symposium on Foundations of Computer Science (FOCS)* (2018). DOI: 10.1109/FOCS.2018.00017.

[GRS00] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. "Learning Polynomials with Queries: The Highly Noisy Case". In: *SIAM Journal on Discrete Mathematics* 13 (4 2000), pp. 535–570. ISSN: 0895-4801. DOI: 10.1137/S0895480198344540.

[GS92] Peter Gemmell and Madhu Sudan. "Highly resilient correctors for polynomials". In: *Information Processing Letters* 43 (4 1992), pp. 169–174. ISSN: 00200190. DOI: 10.1016/0020-0190(92)90195-2.

[HKNS15] Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. "Unifying and Strengthening Hardness for Dynamic Problems via the Online Matrix-Vector Multiplication Conjecture". In: *Proceedings of Symposium on Theory of Computing (STOC)* (2015), pp. 21–30. DOI: 10.1145/2746539.2746609. URL: https://dl.acm.org/doi/10.1145/2746539.2746609.

[HLS22] Monika Henzinger, Andrea Lincoln, and Barna Saha. "The Complexity of Average-Case Dynamic Subgraph Counting". In: *Proceedings of Symposium on Discrete Algorithms (SODA)* (2022), pp. 459–498. DOI: 10.1137/1.9781611977073.23.

[HS21] Shuichi Hirahara and Nobutaka Shimizu. "Nearly Optimal Average-Case Complexity of Counting Bicliques Under SETH". In: *Proceedings of Symposium on Discrete Algorithms (SODA)* (2021), pp. 2346–2365. DOI: 10.1137/1.9781611976465.140.

[HVV06] Alexander Healy, Salil Vadhan, and Emanuele Viola. "Using Nondeterminism to Amplify Hardness". In: *SIAM Journal on Computing* 35 (4 2006), pp. 903–931. ISSN: 0097-5397. DOI: 10.1137/S0097539705447281. URL: https://epubs.siam.org/doi/abs/10.1137/S0097539705447281?mobileUi=0.

[IJKW10] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. "Uniform Direct Product Theorems: Simplified, Optimized, and Derandomized". In: *SIAM J. Comput.* 39.4 (2010), pp. 1637–1665. DOI: 10.1137/080734030.

[Imp95] R. Impagliazzo. "Hard-core distributions for somewhat hard problems". In: *Proceedings of Foundations of Computer Science (FOCS)* (1995), pp. 538–545. DOI: 10.1109/SFCS.1995.492584. URL: https://ieeexplore.ieee.org/document/492584.

[IW01] Russell Impagliazzo and Avi Wigderson. "Randomness vs Time: Derandomization under a Uniform Assumption". In: *J. Comput. Syst. Sci.* 63.4 (2001), pp. 672–688. DOI: 10.1006/jcss.2001.1780.

[IW97] Russell Impagliazzo and Avi Wigderson. "¡i¿P = BPP¡/i¿ if ¡i¿E¡/i¿ requires exponential circuits". In: *Proceedings of Symposium on Theory of Computing (STOC)* (1997), pp. 220–229. DOI: 10.1145/258533.258590. URL: https://dl.acm.org/doi/10.1145/258533.258590.

[JX22] Ce Jin and Yinzhan Xu. "Tight Dynamic Problem Lower Bounds from Generalized BMM and OMv". In: *arXiv:2202.11250 (to appear at STOC22)* (2022). URL: https://arxiv.org/abs/2202.11250.

[KK13] Phokion G. Kolaitis and Swastik Kopparty. "Random graphs and the parity quantifier". In: *Journal of the ACM* 60 (5 2013), pp. 1–34. ISSN: 0004-5411. DOI: 10.1145/2528402.

[KLP12] Tali Kaufman, Shachar Lovett, and Ely Porat. "Weight Distribution and List-Decoding Size of Reed-Muller Codes". In: *IEEE Trans. Inf. Theory* 58.5 (2012), pp. 2689–2696. DOI: 10.1109/TIT.2012.2184841.

[Lip91] Richard Lipton. "New directions in testing". In: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 2 (1991), pp. 191–202.

[LLW19] Rio LaVigne, Andrea Lincoln, and Virginia Vassilevska Williams. "Public-Key Cryptography in the Fine-Grained Setting". In: *Proceedings of Annual International Cryptology Conference (CRYPTO)* (2019), pp. 605–635. DOI: 10.1007/978-3-030-26954-8_20.

[LW17] Kasper Green Larsen and Ryan Williams. "Faster Online Matrix-Vector Multiplication". In: *Proceedings of Symposium on Discrete Algorithms (SODA)* (2017), pp. 2182–2189. DOI: 10.1137/1.9781611974782.142. URL: https://epubs.siam.org/doi/10.1137/1.9781611974782.142.

[LWW18] Andrea Lincoln, Virginia Vassilevska Williams, and Ryan Williams. "Tight Hardness for Shortest Cycles and Paths in Sparse Graphs". In: *Proceedings of Symposium on Discrete Algorithms (SODA)* (2018), pp. 1236–1252. DOI: 10.1137/1.9781611975031.80.

[NW94] Noam Nisan and Avi Wigderson. "Hardness vs randomness". In: *Journal of Computer and System Sciences* 49.2 (1994), pp. 149–167.

[ODo04] Ryan O'Donnell. "Hardness amplification within NP". In: *J. Comput. Syst. Sci.* 69.1 (2004), pp. 68–94. DOI: 10.1016/j.jcss.2004.01.001.

[RTTV08] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil Vadhan. "Dense Subsets of Pseudorandom Sets". In: *Proceedings of Symposium on Foundations of Computer Science (FOCS)* (2008), pp. 76–85. DOI: 10.1109/FOCS.2008.38. URL: https://ieeexplore.ieee.org/document/4690942.

[SS93] A. W. Schrift and Adi Shamir. "Universal Tests for Nonuniform Distributions". In: *J. Cryptol.* 6.3 (1993), pp. 119–133. DOI: 10.1007/BF00198461.

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. "Pseudorandom Generators without the XOR Lemma". In: *Journal of Computer and System Sciences* 62 (2 2001), pp. 236–266. ISSN: 00220000. DOI: 10.1006/jcss.2000.1730.

[Sud97] Madhu Sudan. "Decoding of Reed Solomon Codes beyond the Error-Correction Bound". In: *Journal of Complexity* 13 (1 1997), pp. 180–193. ISSN: 0885064X. DOI: 10.1006/jcom.1997.0439.

[Tre03] Luca Trevisan. "List-Decoding Using The XOR Lemma". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*. 2003, pp. 126–135. DOI: 10.1109/SFCS.2003.1238187.

[Tre05] Luca Trevisan. "On uniform amplification of hardness in NP". In: *Proceedings of the Symposium on Theory of Computing (STOC)*. 2005, pp. 31–38. DOI: 10.1145/1060590.1060595.

[TV07] Luca Trevisan and Salil P. Vadhan. "Pseudorandomness and Average-Case Complexity Via Uniform Reductions". In: *Computational Complexity* 16.4 (2007), pp. 331–364. DOI: 10.1007/s00037-007-0233-x.

[Yao82] Andrew Chi-Chih Yao. "Theory and Application of Trapdoor Functions". In: *In Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (1982), pp. 80–91.

A Balancedness of $\mathsf{TriParity}_n$

Recall that $\mathsf{TriParity}_n\colon \{0,1\}^{3n^2} \to \{0,1\}$ is the number of triangles modulo 2 contained in the tripartite graph given by x (see (1) for definition). In this section, we prove that $\mathsf{TriParity}(x)$ for $x \sim \{0,1\}^{3n^2}$ is nearly balanced using the technique of Kolaitis and Kopparty [KK13], who proved that the number of arbitrary fixed subgraphs in an Erdős–Rényi random graph $G(n,p)$ with any constant $p > 0$ modulo a fixed constant is nearly balanced. We first recall the following result that is a special case of [KK13, Lemma 3.3].

Lemma A.1. *Let $Z_1, \dots, Z_m \in \{0,1\}$ be i.i.d. random variables such that $\mathbf{E}[Z_1] = p \in (0,1)$. Consider the random variable $Q(Z_1, \dots, Z_m)$, where $Q \in \mathbb{F}_2[z_1, \dots, z_m]$ is a multivariate degree-d polynomial written as*

$$Q(z_1, \dots, z_m) = \sum_{F \in \mathcal{F}} \prod_{i \in F} z_i$$

for $\mathcal{F} \subseteq \binom{[m]}{d}$.

Suppose that there exists $\mathcal{E} = \{E_1, \dots, E_r\} \subseteq \mathcal{F}$ satisfying (i) $E_i \cap E_j = \emptyset$ for all $i \neq j$, and (ii) $|F \cap (\cup_{i \in [r]} E_i)| < d$ for every $F \in \mathcal{F} \setminus \mathcal{E}$. Then, $\left|\mathbf{E}\left[(-1)^{\mathsf{TriParity}_n(x)}\right]\right| \leq 2^{-\Omega(r)}$, where the hidden constant in $\Omega(\cdot)$ depends on p and d.

Proposition A.2. *For $x \sim \{0,1\}^{3n^2}$, $\mathsf{TriParity}_n(x) = 1$ with probability $\frac{1}{2} \pm 2^{-\Omega(n)}$.*

Proof. We apply Lemma A.1. Note that $\mathsf{TriParity}_n(x)$ can be written as the degree-3 polynomial over i.i.d. binary random variables $(x_i)_{i \in [3n^2]}$. Specifically, write

$$\mathsf{TriParity}_n(x) = \sum_{u \in U, v \in V, w \in W} x_{uv} x_{vw} x_{wu} = \sum_{F \in \mathcal{F}} \prod_{e \in F} x_e,$$

where U, V, W are the partite vertex set with $|U| = |V| = |W| = n$ and $\mathcal{F} = \{\{uv, vw, wu\}\colon u \in U, v \in V, w \in W\}$ is the set of all possible triangles (here, we identify uv with the edge $\{u, v\}$). Write $U = \{u_1, \dots, u_n\}$, $V = \{v_1, \dots, v_n\}$, and $W = \{w_1, \dots, w_n\}$. Let $\mathcal{E} = \{\{u_i v_i, v_i w_i, w_i u_i\}\colon i \in [n]\}$ be the set of n vertex-disjoint triangles. Note that for any disjoint $E, E' \in \mathcal{E}$, $E \cap E' = \emptyset$ and for any $F \in \mathcal{F} \setminus \mathcal{E}$, $|F \cap \bigcup_{E \in \mathcal{E}} E| \leq 2$. Therefore, from Lemma A.1, we have

$$|\Pr[\mathsf{TriParity}_n(x) = 1] - \Pr[\mathsf{TriParity}_n(x) = 0]| = \left|\mathbf{E}\left[(-1)^{\mathsf{TriParity}_n(x)}\right]\right| \leq 2^{-\Omega(n)}.$$

□

B Local Decoding over Grids

In this section, we prove Theorem 8.1 using the following local decoder of $\mathsf{RM}_{n,d,\mathbb{F}_q}$ ([BSS20, Theorem 5.4]) of Bafna, Srinivasan, and Sudan. Note that we can obtain the worst-case-to-average-case reduction for the parity of t-clique subgraphs over $G(n,1/2)$ of [BBB21; Gol20] by the same way.

Lemma B.1. *For a finite field $\mathbb{F}$ of characteristic q, let $f\colon \{0,1\}^n \to \mathbb{F}$ be a n-variate degree-d polynomial. Then, there are an integer $k = k(d,q) \le qd$ and an $O(\binom{2k}{k} + n)$-time randomized oracle algorithm $A^{\mathcal{O}}$ that makes at most $\binom{2k}{k}$ nonadaptive queries and, if the oracle $\mathcal{O}$ satisfies $\Pr_{x\sim\{0,1\}^n}[\mathcal{O}(x) = f(x)] \ge 1 - 1/\left(3 \times \binom{2k}{k}\right)$, then $\Pr[A^{\mathcal{O}}(x) = f(x)] \ge 2/3$ for any $x \in \{0,1\}^n$ where the probability is taken for the randomness of $A^{\mathcal{O}}$.*

Proof Sketch. For given input $x \in \{0,1\}^n$, the algorithm $A^{\mathcal{O}}$ runs as follows: Let $k = k(q,d)$ be the smallest power of q that is strictly greater than d. Let $h\colon [n] \to [2k]$ be a uniform random map (each $h(j)$ is independently chosen uniformly at random from $[2k]$). For $y \in \{0,1\}^{2k}$, define $z = z(y) \in \{0,1\}^n$ by $z_i = y_{h(i)} \oplus x_j$. Then, output $\binom{d+k}{k}^{-1} \sum_{y\in B} \mathcal{O}(z(y))$ (the operations are over $\mathbb{F}$), where $B \subseteq \{0,1\}^{2k}$ is the set of binary vector y that is balanced (i.e., exactly k elements of y is one) and the last $k-d$ bits are zero. Note that $A^{\mathcal{O}}$ makes at most $|B| \le \binom{2k}{k}$ nonadaptive queries.

To see the correctness, consider the function $g(y_1,\dots,y_{2k}) := f(z_1,\dots,z_n)$. Note that $g(y) = f(z) = \mathcal{O}(z)$ with probability $1 - \left(3 \times \binom{2k}{k}\right)$ for each $y \in B$ (over the choice of h) since h is a uniformly random and y is balanced. Therefore, by the union bound, the output $A^{\mathcal{O}}(x)$ is equal to $\binom{d+k}{k}^{-1} \sum_{y\in B} g(y)$ with probability at least $2/3$. We refer the proof of the fact that $\binom{d+k}{k}^{-1} \sum_{y\in B} g(y) = g(0) = f(x)$ to the proof of [BSS20, Lemma 5.1]. □

Theorem B.2 (Reminder of Theorem 8.1)**.** *There exists an absolute constant $\epsilon_0 > 0$ satisfying the following: If there exists a $T(n)$-time algorithm A satisfying*

$$\Pr_{x\sim\{0,1\}^{3n^2}}[A(x) = \mathsf{TriParity}_n(x)] \ge 1 - \epsilon_0$$

for all n, then, there exists an $O(T(n))$-time randomized algorithm A' such that, for any n and any input $x \in \{0,1\}^{3n^2}$,

$$\Pr_{A'}[A'(x) = \mathsf{TriParity}_n(x)] \ge 2/3.$$

Proof. The oracle algorithm obtained by applying Lemma B.1 with f being $\mathsf{TriParity}_n$ and $\mathcal{O}$ being the average-case solver A is the worst-case solver. □

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