

On the Composition of Randomized Query Complexity and Approximate Degree

Sourav Chakraborty* Chandrima Kayal† Manaswi Paraashar‡ Rajat Mittal§
Swagato Sanyal¶ Nitin Saurabh ||

July 8, 2023

Abstract

For any Boolean functions f and g , the question whether $R(f \circ g) = \tilde{\Theta}(R(f) \cdot R(g))$, is known as the composition question for the randomized query complexity. Similarly, the composition question for the approximate degree asks whether $\widetilde{\deg}(f \circ g) = \tilde{\Theta}(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g))$. These questions are two of the most important and well-studied problems in the field of analysis of Boolean functions, and yet we are far from answering them satisfactorily.

It is known that the measures compose if one assumes various properties of the outer function f (or inner function g). This paper extends the class of outer functions for which R and $\widetilde{\deg}$ compose.

A recent landmark result (Ben-David and Blais, 2020) showed that $R(f \circ g) = \Omega(\text{noisy}R(f) \cdot R(g))$. This implies that composition holds whenever $\text{noisy}R(f) = \tilde{\Theta}(R(f))$. We show two results:

1. When $R(f) = \Theta(n)$, then $\text{noisy}R(f) = \Theta(R(f))$. In other words, composition holds whenever the randomized query complexity of the outer function is full.
2. If R composes with respect to an outer function, then $\text{noisy}R$ also composes with respect to the same outer function.

On the other hand, no result of the type $\widetilde{\deg}(f \circ g) = \Omega(M(f) \cdot \widetilde{\deg}(g))$ (for some non-trivial complexity measure $M(\cdot)$) was known to the best of our knowledge. We prove that

$$\widetilde{\deg}(f \circ g) = \tilde{\Omega}(\sqrt{\text{bs}(f)} \cdot \widetilde{\deg}(g)),$$

where $\text{bs}(f)$ is the block sensitivity of f . This implies that $\widetilde{\deg}$ composes when $\widetilde{\deg}(f)$ is asymptotically equal to $\sqrt{\text{bs}(f)}$.

It is already known that both R and $\widetilde{\deg}$ compose when the outer function is symmetric. We also extend these results to weaker notions of symmetry with respect to the outer function.

*Indian Statistical Institute, Kolkata, India. sourav@isical.ac.in

†Indian Statistical Institute, Kolkata, India. chandrimakayal2012@gmail.com

‡Aarhus University, Denmark. manaswi.isi@gmail.com

§Indian Institute of Technology Kanpur, India. rmittal@cse.iitk.ac.in

¶Indian Institute of Technology Kharagpur, India. sanyalswagato@gmail.com

|| Indian Institute of Technology, Hyderabad, India. 3295.nitin@gmail.com

Contents

1	Introduction	2
2	Our Results and Techniques	3
2.1	Lower bounds on $\widetilde{R}(f \circ g)$ when the outer function has full randomized query complexity	3
2.2	Lower bound on $\widetilde{\deg}(f \circ g)$ in terms of block sensitivity of f and $\widetilde{\deg}(g)$	5
2.3	Composition results when the outer functions has some symmetry	7
3	Preliminaries	8
3.1	Standard definitions and functions	8
3.1.1	Standard definition of complexity measures	8
3.1.2	For the composition of R	9
4	Results about composition of R	11
4.1	Composition for functions with $R(f) = \Theta(n)$	12
4.2	Additional implications of Observation 4.2	12
4.3	Proof of Corollary 2.5	14
5	Composition of approximate degree in terms of block sensitivity	14
6	Conclusion	17
A	Some important Boolean functions	20
A.1	Some properties of SINK and Rubinstein function	20
B	Approximate degree of Promise-OR composed with different inner functions	21
C	Composition theorems for strongly-k-junta symmetric outer functions	24
C.1	Composition of approximate degree for \sqrt{n} -junta symmetric functions	25
D	Proof of Theorem 2.4	26

1 Introduction

For studying the complexity of Boolean functions, a number of simple complexity measures (like decision tree complexity, randomized query complexity, degree, certificate complexity and so on) have been studied over the years. (Refer to the survey [Bd02] for an introduction to complexity measures of Boolean functions.) Understanding how these measures are related to each other [ABK16, ABK⁺21, ABB⁺17, Hua19], and how they behave for various classes of Boolean functions has been a central area of research in complexity theory [Pat92, Dru11, Sun07].

A crucial step towards understanding a complexity measure is: how does the complexity measure behave when two Boolean functions are combined to obtain a new function (i.e., what is the relationship between the measure of the resulting function and the measures of the two individual functions) [BKT19, BDGKW20, GSS16, Tal13]? One particularly natural combination of functions is called *composition*, and it takes a central role in complexity theory.

For any two Boolean functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and $g : \{0, 1\}^m \rightarrow \{0, 1\}$, the composed function $f \circ g : \{0, 1\}^{nm} \rightarrow \{0, 1\}$ is defined as the function

$$f \circ g(x_1, \dots, x_n) = f(g(x_1), \dots, g(x_n)),$$

where $x_i \in \{0, 1\}^m$ for $i \in [n]$. For the function $f \circ g$, the function f is called the outer function and g is called the inner function. See Definition 3.1 for a natural extension to partial functions.

Let $M(\cdot)$ be a complexity measure of Boolean functions. A central question in complexity theory is the following.

Question (Composition question for M). *Is the following true for all Boolean functions f and g :*

$$M(f \circ g) = \widetilde{\Theta}(M(f) \cdot M(g))?$$

The notation $\widetilde{\Theta}(\cdot)$ hides poly-logarithmic factors of the arity of the outer function f .

Composition of Boolean functions with respect to different complexity measures is a very important and useful tool in areas like communication complexity, circuit complexity and many more. To take an example, a popular application of composition is to create new functions demonstrating better separations (refer to [NS94, Tal13, Amb05, GSS16] for some other results of similar flavour).

It is known that the answer to the composition question is in the affirmative for complexity measures like deterministic decision tree complexity [Sav02, Tal13, Mon14], degree [Tal13] and quantum query complexity [Rei11, LMR⁺11, Kim13]. While it is well understood how several complexity measures behave under composition, there are two important measures (even though well studied) for which this problem remains wide open: randomized query complexity (denoted by R) and approximate degree (denoted by $\widetilde{\deg}$) [She12, NS94, Amb05, She13a, BT13, She13b]. (See Definition 3.2 and Definition 3.3 for their respective formal definitions.)

For both R and $\widetilde{\deg}$ the upper bound inequality is known, i.e., $R(f \circ g) = \widetilde{O}(R(f) \cdot R(g))$ (folklore) and $\widetilde{\deg}(f \circ g) = O(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g))$ [She13c]. Thus it is enough to prove the lower bound on the complexity of composed function in terms of the individual functions. Most of the attempts to prove this direction of the question have focused on special cases when the outer function has certain special properties¹.

The initial steps taken towards answering the composition question for R were to show a lower bound by using a weaker measure of outer function than the randomized query complexity. In particular, it was shown that $R(f \circ g) = \Omega(s(f) \cdot R(g))$ [GJPW18, AKK16], where $s(f)$ is the sensitivity of f (Definition 3.4). Since $s(f) = \Theta(R(f))$ for any symmetric function² f , these

¹We note that some works have also studied the composition of R and $\widetilde{\deg}$ when the inner functions have special properties, for example, [ABK16, BK18, AGJ⁺17, GLSS19, Li21, BDBGM22].

²Functions that depend only on the Hamming weight of their input.

results show that R composes when the outer function is a symmetric function (like OR, AND, Majority, Parity, etc.). The lower bound was later improved to obtain $R(f \circ g) = \Omega(\text{fbs}(f) \cdot R(g))$ [BDG⁺20, BDB20a], where $\text{fbs}(f)$ is the fractional block sensitivity of f (Definition 3.5).

Unfortunately, at this stage, there could even be a cubic gap between R and fbs ; the best known bound is $R(f) = O(\text{fbs}(f)^3)$ [ABK⁺21]. Given that there are already known Boolean functions with quadratic gap between $\text{fbs}(f)$ and $R(f)$ (e.g., BKK function [ABK16]), it is natural to consider composition question for randomized query complexity when R is *big* but fbs is *small*. We take a step towards this problem by showing that composition for R holds when the outer function has full randomized query complexity, i.e., $R(f) = \Theta(n)$, where n is the arity of the outer function f .

For composition of $\widetilde{\text{deg}}$, Sherstov [She12] already showed that $\widetilde{\text{deg}}(f \circ g)$ composes when the approximate degree of the outer function f is $\Theta(n)$, where n is the arity of the outer function. Thus approximate degree composes for several symmetric functions (having approximate degree $\Theta(n)$, like Majority and Parity). Though, until recently it was not even clear if $\widetilde{\text{deg}}(\text{OR} \circ \text{AND}) = \Omega(\widetilde{\text{deg}}(\text{OR}) \widetilde{\text{deg}}(\text{AND}))$ (arguably the simplest of composed functions). OR has approximate degree $O(\sqrt{n})$, and thus the result of [She12] does not prove $\widetilde{\text{deg}}$ composition when the outer function is OR (similarly for AND). In a long series of work by [NS94, Amb05, She13a, BT13, She13b], the question was finally resolved; it was later generalized to the case when the outer function is symmetric [BBGK18].

In contrast to R composition, no lower bound on the approximate degree of composed function is known with a weaker measure on the outer function. It is well known that for any Boolean function f , $\sqrt{s(f)} \leq \sqrt{\text{bs}(f)} = O(\widetilde{\text{deg}}(f))$ [NS94]. So a natural step towards proving $\widetilde{\text{deg}}$ composition is: prove a lower bound on $\widetilde{\text{deg}}(f \circ g)$ by $\sqrt{\text{bs}(f)} \cdot \widetilde{\text{deg}}(g)$. We show this result by generalizing the approach of [BBGK18].

While the techniques used for the composition of R and $\widetilde{\text{deg}}$ are quite different, one can still observe similarities between the classes of outer functions for which the composition of R and $\widetilde{\text{deg}}$ is known to hold respectively. We further expand these similarities, by extending the classes of outer functions for which the composition theorem hold.

2 Our Results and Techniques

It is well-known, by amplification, that $R(f \circ g) = O(R(f) \cdot R(g) \cdot \log R(f))$. In the case of approximate degree, Shrestov [She13c] showed that $\widetilde{\text{deg}}(f \circ g) = O(\widetilde{\text{deg}}(f) \cdot \widetilde{\text{deg}}(g))$. So, to answer the composition question for R (or $\widetilde{\text{deg}}$), we are only concerned about proving a lower bound on $R(f \circ g)$ (or $\widetilde{\text{deg}}(f \circ g)$) in terms of $R(f)$ and $R(g)$ (or $\widetilde{\text{deg}}(f)$ and $\widetilde{\text{deg}}(g)$) respectively.

We split our results into three parts. In the first part we prove the tight lower bound on $R(f \circ g)$ when the outer function has full randomized complexity. In the second part we provide a tight lower bound on $\widetilde{\text{deg}}(f \circ g)$ in terms of $\text{bs}(f)$ and $\widetilde{\text{deg}}(g)$. Our results on the lower bound of $R(f \circ g)$ and $\widetilde{\text{deg}}(f \circ g)$ are summarized in Table 1. Finally, we also prove composition theorems for R and $\widetilde{\text{deg}}$ when the outer functions have a slightly relaxed notion of symmetry.

2.1 Lower bounds on $R(f \circ g)$ when the outer function has full randomized query complexity

Sherstov [She12] proved that $\widetilde{\text{deg}}(f \circ g) = \Omega(\widetilde{\text{deg}}(f) \cdot \widetilde{\text{deg}}(g))$ when the approximate degree of the outer function f is $\Theta(n)$, where n is the arity of f . Though, a corresponding result for the case of randomized query complexity was not known. Our main result is to prove the corresponding theorem for randomized query complexity.

	In terms of $\text{bs}(f)$	In terms of arity of f
R	$R(f \circ g) = \widetilde{\Omega}(\text{bs}(f) \cdot R(g))$ [GJPW18]	$R(f \circ g) = \widetilde{\Omega}(R(f) \cdot R(g))$ when $R(f) = \Theta(n)$ Theorem 2.1
$\widetilde{\text{deg}}$	$\widetilde{\text{deg}}(f \circ g) = \widetilde{\Omega}(\sqrt{\text{bs}(f)} \cdot \widetilde{\text{deg}}(g))$ Theorem 2.6	$\widetilde{\text{deg}}(f \circ g) = \widetilde{\Omega}(\widetilde{\text{deg}}(f) \cdot \widetilde{\text{deg}}(g))$ when $\widetilde{\text{deg}}(f) = \Theta(n)$ [She12]

Table 1: Composition of R and $\widetilde{\text{deg}}$ depending on the complexity of the outer function in terms of block-sensitivity and arity.

Theorem 2.1. *Let f be a partial Boolean function on n -bits such that $R(f) = \Theta(n)$. Then for all partial functions g , we have*

$$R(f \circ g) = \Omega(R(f) \cdot R(g)).$$

The proof of this theorem is given in Section 4. Notice, since $R(f \circ g) = O(R(f) \cdot R(g) \log R(f))$ (by error reduction), Theorem 2.1 says that composition of R holds when the randomized query complexity of the outer function, f , is $\Theta(n)$. Next, we give main ideas behind the proof of the above theorem.

Ideas behind proof of Theorem 2.1 A crucial complexity measure that we use for the proof of Theorem 2.1 is called the *noisy randomized query complexity*, first introduced by Ben-David and Blais [BDB20b] in order to study randomized query complexity. Noisy randomized query complexity can be seen as a generalization of randomized query complexity where the algorithm can query a bit with any bias and only pays proportionally to the square of the bias in terms of cost (see Definition 3.8). They give the following characterization of $\text{noisyR}(f)$ (the noisy randomized query complexity of f).

Theorem 2.2 (Ben-David and Blais [BDB20b]). *For all partial functions f on n -bits, we have*

$$\text{noisyR}(f) = \Theta\left(\frac{R(f \circ \text{GapMaj}_n)}{n}\right), \quad (1)$$

where GapMaj_n is the *Gap-Majority* function on n bits whose input is promised to have Hamming weight either $(n/2 + 2\sqrt{n})$ (in which case the output is -1) or $(n/2 - 2\sqrt{n})$ (in which case the output is 1).

We want to point out that the arity of f and Gap-Majority is the *same* in Theorem 2.2. Towards a proof of Theorem 2.1, we first make the following crucial observation.

Observation 2.3. *Let f be a partial Boolean function on n bits. If $t(n) \geq 1$ is a non-decreasing function of n and*

$$\text{noisyR}(f) = \Omega\left(\frac{R(f \circ \text{GapMaj}_{t(n)})}{t(n)}\right),$$

then $R(f \circ g) = \Omega((R(f) \cdot R(g))/t(n))$ for all partial functions g .

In particular choosing $t(n)$ to be $(\log n)$, if the outer function f satisfies

$$\text{noisyR}(f) = \Omega\left(\frac{R(f \circ \text{GapMaj}_{\log n})}{\log n}\right). \quad (2)$$

then the above observation gives $R(f \circ g) = \Omega((R(f) \cdot R(g))/(\log n))$ for all partial functions g .

The Observation 2.3 allows us to approach the composition question for randomized query complexity in a conceptually fresh manner. The goal of proving that randomized query complexity composes for a function or a class of functions, say upto $(\log n)$ -factor, reduces to showing that Equation 2 holds for that function or class of functions for $t(n) = \log n$.

We are able to show that Equation 2 holds for all non-decreasing functions $t(n)$ with a slight overhead.

Theorem 2.4. *Let f be a partial function on n bits and let $t \geq 1$, then $R(f \circ \text{GapMaj}_t) = O(t \cdot \text{noisyR}(f) + n)$.*

Notice that this is a generalization of Ben-David and Blais' characterization of noisyR given by Theorem 2.2 in one direction. To give an idea of the proof, their characterization (Theorem 2.2) shows that any noisy oracle algorithm for f can be simulated using only two biases, 1 and $1/\sqrt{n}$ (where n is the arity of f), with only constant overhead. We generalize this by showing that the same simulation works with a slight overhead even when the bias $1/\sqrt{n}$ is replaced by a bias $1/\sqrt{t}$, for some $t \geq 1$. A proof the above theorem is provided in Appendix D.

This seemingly inconsequential generalization allows us to complete the proof of Theorem 2.1, i.e. if for an n -bit partial function f , $R(f) = \Theta(n)$, then $R(f \circ g) = \tilde{\Theta}(R(f) \cdot R(g))$ for all partial functions g (see Section 4 for details).

Furthermore, Theorem 2.4 even sheds light on *the composition question for noisyR*. A corollary of this theorem is that if R composes with respect to an outer function, then noisyR also composes with respect to the same outer function (see Section 4 for a proof).

Corollary 2.5. *Let f be a partial Boolean function. If $R(f \circ g) = \tilde{\Theta}(R(f) \cdot R(g))$ for all partial functions g then $\text{noisyR}(f \circ g) = \tilde{\Theta}(\text{noisyR}(f) \cdot \text{noisyR}(g))$.*

2.2 Lower bound on $\widetilde{\text{deg}}(f \circ g)$ in terms of block sensitivity of f and $\widetilde{\text{deg}}(g)$

As discussed in the introduction, the composition question for $\widetilde{\text{deg}}$ is only known to hold when the outer function f is symmetric [BBGK18] or has high approximate degree [She12]. There are also no known lower bounds on $\widetilde{\text{deg}}(f \circ g)$ in terms of weaker measures of f and $\widetilde{\text{deg}}(g)$. Compare this with the situation with respect to composition of R. It was shown in [GJPW18] that $R(f \circ g) = \Omega(s(f)R(g))$, where $s(f)$ denotes the sensitivity of f . This was later strengthened to $\Omega(\text{fbs}(f)R(g))$ [BDG⁺20, BDB20a], where $\text{fbs}(f)$ is the fractional block sensitivity of f .

In this second part we show analogous lower bounds on approximate degree of composed function $f \circ g$. Our main result here is the following.

Theorem 2.6. *For all non-constant (possibly partial)³ Boolean functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and $g : \{0, 1\}^m \rightarrow \{0, 1\}$, we have*

$$\widetilde{\text{deg}}(f \circ g) = \tilde{\Omega}(\sqrt{\text{bs}(f)} \cdot \widetilde{\text{deg}}(g)).$$

We first note that the above theorem is tight in terms of block sensitivity, i.e., we cannot have $\widetilde{\text{deg}}(f \circ g) = \tilde{\Omega}(\text{bs}(f)^c \cdot \widetilde{\text{deg}}(g))$ for any $c > 1/2$. This is because the OR function over n bits witnesses the tight quadratic separation between $\widetilde{\text{deg}}$ and bs, i.e., $\widetilde{\text{deg}}(\text{OR}_n) = \Theta(\sqrt{n}) = \Theta(\sqrt{\text{bs}(\text{OR}_n)})$ [NS94].

We also get the following composition theorem as a corollary. It says that the composition for $\widetilde{\text{deg}}$ holds when the outer function has minimal approximate degree with respect to its block sensitivity. Recall, $\widetilde{\text{deg}}(f) = \Omega(\sqrt{\text{bs}(f)})$ [NS94].

³For definitions of block sensitivity and approximate degree in the context of partial functions, please see Definitions 3.4 and 3.9.

Corollary 2.7. *For all Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ with $\widetilde{\deg}(f) = \Theta(\sqrt{\text{bs}(f)})$ and for all $g : \{0, 1\}^m \rightarrow \{0, 1\}$, we have $\widetilde{\deg}(f \circ g) = \widetilde{\Theta}(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g))$.*

This complements a result of Sherstov [She12, Theorem 6.6], which shows that composition of $\widetilde{\deg}$ holds when the outer function has maximal $\widetilde{\deg}$ with respect to its arity.

We further note that Corollary 2.7 covers new set of composed functions $f \circ g$ for which the composition theorem for $\widetilde{\deg}$ doesn't follow from the known results [BBGK18, She12]. For example, consider the Rubinstein function RUB with arity n (Definition A.3) as the outer function f . It is clearly not a symmetric function. It also doesn't have high approximate degree, i.e., $\widetilde{\deg}(\text{RUB}) = O(\sqrt{n} \log n)$ (Lemma A.7). Therefore, the composition of $\widetilde{\deg}(\text{RUB} \circ g)$ doesn't follow from the existing results. However, it follows from Corollary 2.7, since $\text{bs}(\text{RUB}) = \Omega(n)$ and so $\widetilde{\deg}(\text{RUB}) = \widetilde{\Theta}(\sqrt{\text{bs}(\text{RUB})})$.

Another example is the sink function SINK over $\binom{n}{2}$ variables (Definition A.2), which is also not a symmetric function. Furthermore, its approximate degree is $O(\sqrt{n} \log n)$ (Lemma A.7). Therefore, the composition of $\widetilde{\deg}(\text{SINK} \circ g)$ also doesn't follow from the existing results. Again, it follows from Corollary 2.7, since $\text{bs}(\text{SINK}) = \Theta(n)$ (Observation A.4) and $\widetilde{\deg}(\text{SINK}) = \widetilde{\Theta}(\sqrt{n})$.

Ideas behind proof of Theorem 2.6 We will first sketch the proof ideas in the case when f and g are total Boolean functions, and then explain how to extend it to partial functions too.

Our starting point is the well known Nisan-Szegedy's embedding of PrOR over $\text{bs}(f)$ many bits in a Boolean function f [NS94]. Carrying out this transformation in $f \circ g$ embeds $\text{PrOR}_{\text{bs}(f)} \circ (g_1, \dots, g_{\text{bs}(f)})$ into $f \circ g$, where $g_1, \dots, g_{\text{bs}(f)}$ are different partial functions such that $\widetilde{\text{bdeg}}(g_i) \geq \widetilde{\deg}(g)$ for all $i \in [\text{bs}(f)]^4$. Since the transformation is just substitutions of variables by constants, we further have

$$\widetilde{\deg}(f \circ g) \geq \widetilde{\text{bdeg}}(\text{PrOR}_{\text{bs}(f)} \circ (g_1, \dots, g_{\text{bs}(f)})). \quad (3)$$

It now looks like that we can appeal to the composition theorem for PrOR (Theorem 3.13) [BBGK18] to obtain our theorem. However, there is a technical difficulty – Theorem 3.13 doesn't hold for *different* inner *partial* functions. It only deals with a single total inner function. We therefore generalize the proof of Theorem 3.13 to obtain the following general version of the composition theorem for PrOR.

Theorem 2.8. *For any partial Boolean functions g_1, g_2, \dots, g_n , we have*

$$\widetilde{\text{bdeg}}(\text{PrOR}_n \circ (g_1, g_2, \dots, g_n)) = \Omega \left(\frac{\sqrt{n} \cdot \min_{i=1}^n \widetilde{\text{bdeg}}(g_i)}{\log n} \right).$$

We can now obtain our main theorem from Eq. (3) and Theorem 2.8. The proof of Theorem 2.8 is a generalization of Theorem 3.13. For lack of space, we present it in Appendix B.

We end this part with a comment on how to extend our main theorem to partial functions. Note that with the appropriate definition of block sensitivity (Definition 3.4), the embedding of Nisan-Szegedy's carries through and then the rest of the proof is identical.

⁴ $\widetilde{\text{bdeg}}$ is the notion of approximate degree in the context of partial functions. For a formal definition, see Definition 3.9.

2.3 Composition results when the outer functions has some symmetry

The class of symmetric functions capture many important function like OR, AND, Parity and Majority. Recall that a function is symmetric when the function value only depends on the Hamming weight of the input; in other words, a function is symmetric iff its value on an input remains unchanged even after permuting the bits of the input. As noted earlier, both for $\widetilde{\text{deg}}$ and $\widetilde{\text{deg}}$, composition was known to hold when the outer function was symmetric.

A natural question is, whether one can prove composition theorems when the outer function is *weakly* symmetric (it is symmetric with respect to a weaker notion of symmetry). In this paper we consider one such notion of symmetry – junta-symmetric functions.

Definition 2.9 (*k-junta symmetric function*). *A function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is called a k-junta symmetric function if there exists a set \mathcal{J} of size k of variables such that the function value depends on assignments to the variables in \mathcal{J} as well as on the Hamming weight of the whole input.*

k -junta symmetric functions can be seen as a mixture of symmetric functions and k -juntas. This class of functions has been considered previously in literature, particularly in [CFG12, BWY15] where these functions plays a crucial role. [CFG12] even presents multiple characterisations of k -junta symmetric functions for constant k . Note that by definition an arbitrary k -junta (i.e., a function that depend on k variables) is also a k -junta symmetric function, since we can consider the dependence on Hamming weight to be trivial. Thus, this notion loses out on the symmetry of the function considered. We, therefore, consider the class of *strongly* k -junta symmetric functions.

Definition 2.10 (*Strongly k-junta symmetric function*). *A k-junta symmetric function is called strongly k-junta symmetric if every variable is influential. In other words, there exists a setting to the junta variables such that the function value depends on the Hamming weight of the whole input in a non-trivial way.*

We prove that if the outer function is strongly \sqrt{n} -junta symmetric (“strongly” indicating that the dependence on the Hamming weight is non-trivial) then $\widetilde{\text{deg}}$ composes. On the other hand, Theorem 2.1 implies that R composes for any strongly k -junta symmetric functions (as long as $n - k = \Theta(n)$).

Theorem 2.11. *For any strongly k-junta symmetric function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and any Boolean function $g : \{0, 1\}^m \rightarrow \{0, 1\}$, we have*

- $\widetilde{\text{deg}}(f \circ g) = \widetilde{\Theta}(\widetilde{\text{deg}}(f) \cdot \widetilde{\text{deg}}(g))$ where $k = O(\sqrt{n})$.
- $\text{R}(f \circ g) = \widetilde{\Theta}(\text{R}(f) \cdot \text{R}(g))$ where $n - k = \Theta(n)$.

For the lack of space, the proof of the above theorem is given in Appendix C. Note that if one is able to prove the above theorem for k -junta-symmetric functions (without the requirement of “strongly”) for any non-constant k then we would have the full composition theorem.

Organization of the paper

We have formally defined complexity measures and Boolean functions needed for our results in Section 3. Section 4 contains proofs of our results related to the composition of randomized query complexity (Theorem 2.1). In Section 5 we give the proof of our result for the composition of approximate degree (Theorem 2.6). Finally, the results about composition of functions with weak notion of symmetry are in Appendix C.

3 Preliminaries

Notations: We will use $[n]$ to represent the set $\{1, \dots, n\}$. For any (possibly partial) Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1, *\}$ we will denote by $\text{Dom}(f)$ the set $f^{-1}(\{0, 1\})$. The arity of f is the number of variables - in this case n . A Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1, *\}$ is said to be total if $\text{Dom}(f) = \{0, 1\}^n$. Any function (not otherwise stated) will be a total Boolean function.

For any $x \in \{0, 1\}^n$, we will use $|x|$ to denote the number of 1s in x , that is, the Hamming weight of the string x . The string x^i denotes the modified string x with the i -th bit flipped. Similarly, x^B is defined to be the string such that all the bits whose index is contained in the set $B \subseteq [n]$ are flipped in x .

Following is a formal definition of (partial) function composition.

Definition 3.1 (Generalized composition of functions). *For any (possibly partial) Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1, *\}$ and n (possibly partial) Boolean functions g_1, g_2, \dots, g_n , define the (possibly partial) composed function*

$$f \circ (g_1, g_2, \dots, g_n)(x_1, x_2, \dots, x_n) = f(g_1(x_1), g_2(x_2), \dots, g_n(x_n)),$$

where g_i 's can have different arities and, moreover, if $x_i \notin \text{Dom}(g_i)$ for any $i \in [n]$ or the string $(g_1(x_1), g_2(x_2), \dots, g_n(x_n)) \notin \text{Dom}(f)$, then the function $f \circ g$ outputs $*$.

In this paper we use the standard definitions of various complexity measures like randomized query complexity, sensitivity, block-sensitivity, fractional block sensitivity, noisy randomized query complexity, approximate degree and bounded approximate degree and the partial function Promise-OR. We present the formal definitions in the following subsections.

3.1 Standard definitions and functions

3.1.1 Standard definition of complexity measures

We look at many different complexity measures in the paper, let us start with the formal definition of R and $\widetilde{\text{deg}}$.

Definition 3.2 (Randomized query complexity (R)). *Let $f : \{0, 1\}^n \rightarrow \{0, 1, *\}$ be a (possibly partial) Boolean function. A randomized query algorithm A computes f if $\forall x \in \text{Dom}(f), \Pr[A(x) \neq f(x)] \leq 1/3$, where the probability is over the internal randomness of the algorithm. The cost of the algorithm A , $\text{cost}(A)$, is the number of queries made in the worst case over any input as well as internal randomness. The randomized query complexity of f , denoted by $R(f)$, is defined as*

$$R(f) = \min_{A \text{ computes } f} \text{cost}(A).$$

Definition 3.3 (Approximate degree ($\widetilde{\text{deg}}$)). *A polynomial $p : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to approximate a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ if $|p(x) - f(x)| \leq 1/3, \forall x \in \{0, 1\}^n$. The approximate degree of f , $\widetilde{\text{deg}}(f)$, is the minimum possible degree of a polynomial which approximates f .*

Note that the constant $1/3$ in the above definitions can be replaced by any constant strictly smaller than $1/2$ which changes $\widetilde{\text{deg}}(f)$ by only a constant factor.

Other than R and $\widetilde{\text{deg}}$, two important related measures are sensitivity ($s(f)$) and block sensitivity ($\text{bs}(f)$). While the sensitivity and block sensitivity of a total function is well defined, we note that for the case of partial functions there are at least two valid ways of extending the definition from total functions to partial functions. All our results in this paper will hold for partial functions with the following definitions of sensitivity and block sensitivity.

Definition 3.4. The sensitivity $s(f, x)$ of a function $f : \{0, 1\} \rightarrow \{0, 1, *\}$ on x is the maximum number s such that there are indices $i_1, i_2, \dots, i_s \in [n]$ with $f(x^{i_j}) = 1 - f(x)$, for all $1 \leq j \leq s$. Here x^i is obtained from x by flipping the i^{th} bit. The sensitivity of f is defined to be $s(f) = \max_{x \in \text{Dom}(f)} s(f, x)$.

Similarly, the block sensitivity $\text{bs}(f, x)$ of a function $f : \{0, 1\} \rightarrow \{0, 1, *\}$ on x is the maximum number b such that there are disjoint sets $B_1, B_2, \dots, B_b \subseteq [n]$ with $f(x^{B_j}) = 1 - f(x)$ for all $1 \leq j \leq b$. Recall x^{B_j} is obtained from x by flipping all bits inside B_j . The block sensitivity of f is defined to be $\text{bs}(f) = \max_{x \in \text{Dom}(f)} \text{bs}(f, x)$.

In the definition of block sensitivity, the constraint that the blocks has to be disjoint can be relaxed by extending the definition to “fractional blocks”. This gives the measure of fractional block sensitivity.

Definition 3.5. The fractional block sensitivity $\text{fbs}(f, x)$ of a function $f : \{0, 1\} \rightarrow \{0, 1, *\}$ on x is the maximum value of $\sum_{j=1}^b p_j$ such that there are sets $B_1, B_2, \dots, B_b \subseteq [n]$ and $p_1, \dots, p_b \in (0, 1]$ satisfying the following two conditions.

- For each $1 \leq j \leq b$, $f(x^{B_j}) = 1 - f(x)$, and
- For each $1 \leq i \leq n$, $\sum_{j: i \in B_j} p_j \leq 1$.

The fractional block sensitivity of f is defined to be $\text{fbs}(f) = \max_{x \in \text{Dom}(f)} \text{fbs}(f, x)$.

3.1.2 For the composition of R

The function Gap-Majority has played an important role in the study of composition of R.

Definition 3.6 (Gap-Majority). The function $\text{GapMaj}_t : \{0, 1\}^t \rightarrow \{0, 1, *\}$ is a partial function with arity t such that

$$\text{GapMaj}_t(x) = \begin{cases} 1 & \text{if } |x| = t/2 + 2\sqrt{t}, \\ 0 & \text{if } |x| = t/2 - 2\sqrt{t}, \\ * & \text{otherwise.} \end{cases}$$

It can be shown that $R(\text{GapMaj}_t) = \Theta(t)$ [BDB20b].

In regards to the composition question of R, one of the most significant complexity measures (defined by Ben-David and Blais [BDB20b]) is that of noisyR. We first define the noisy oracle model.

Definition 3.7 (Noisy Oracle Model and Noisy Oracle Access to a String ([BDB20b])). For $b \in \{0, 1\}$, a noisy oracle to b takes a parameter $-1 \leq \gamma \leq 1$ as input and returns a bit b' such that $\Pr[b' = b] = (1 + \gamma)/2$. The cost of one such query is γ^2 . Each query to noisy oracle returns independent bits.

For $x = (x_1, \dots, x_n) \in \{0, 1\}^n$, noisy oracle access to x is access to n independent noisy oracles, one for each bit x_i , $i \in [n]$.

Next, we define the noisy oracle model of computation.

Definition 3.8 (Noisy Oracle Model of Computation ([BDB20b])). Let $f : \{0, 1\}^n \rightarrow \{0, 1, *\}$ be a partial Boolean function. A noisyR query algorithm A computes f if for all $x \in \text{Dom}(f)$, $\Pr[A(x) \neq f(x)] \leq 1/3$, where A is a randomized algorithm given noisy oracle access to x , and the probability is over both noisy oracle calls and the internal randomness of the algorithm A . The cost of the algorithm A for an input x is the sum of the cost of all noisy oracle calls made by A on

x , and the cost of A , $\text{cost}(A)$, is the maximum cost over all $x \in \text{Dom}(f)$. The noisyR randomized query complexity of f , denoted by $\text{noisyR}(f)$, is defined as

$$\text{noisyR}(f) = \min_{A \text{ computes } f} \text{cost}(A).$$

Again, $1/3$ in the above definition can be replaced by any constant $< 1/2$. If only queries with $\gamma = 1$ are allowed in the noisy query model, then we obtain the usual randomized algorithm for f , thus $\text{noisyR}(f) = O(\text{R}(f))$.

For the composition of $\widetilde{\text{deg}}$ The definition of $\widetilde{\text{deg}}$ can naturally be extended to partial functions f by restricting the definition to hold only for inputs in $\text{Dom}(f)$, i.e., it doesn't specify the value of the approximating polynomial on inputs *not* in $\text{Dom}(f)$. So the approximating polynomial can take arbitrarily large values on points outside the domain. However, for the purpose of understanding the composition of approximate degree of Boolean functions (or even total Boolean functions) one needs a measure of approximate degree of partial Boolean functions which is bounded on all the points of the Boolean cube.

Definition 3.9 (Bounded approximate degree ($\widetilde{\text{bdeg}}$)). *For a partial Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1, *\}$, the bounded approximate degree of f ($\widetilde{\text{bdeg}}(f)$) is the minimum possible degree of a polynomial p such that*

- $|p(x) - f(x)| \leq 1/3, \quad \forall x \in \text{Dom}(f)$, and
- $0 \leq p(x) \leq 1 \quad \forall x \in \{0, 1\}^n$.

In other words, we take the minimum possible degree of a polynomial which is bounded for all possible inputs ($p(x) \in [0, 1]$ for all $x \in \{0, 1\}^n$), and it approximates f in the usual sense over $\text{Dom}(f)$.

Over the years people have tried to study the composition of $\widetilde{\text{deg}}$ with different outer functions. In this context the following restriction of OR is an important partial function:

Definition 3.10 (Promise-OR). *Promise-OR (denoted by PrOR_n) is the function $\text{PrOR}_n : \{0, 1\}^n \rightarrow \{0, 1, *\}$ such that:*

$$\text{PrOR}_n(x) = \begin{cases} 0 & \text{if } |x| = 0, \\ 1 & \text{if } |x| = 1, \\ * & \text{otherwise.} \end{cases}$$

Some useful previous results: We will also be crucially using a few results from prior works in our proofs. The following are a couple of useful results on noisyR.

Lemma 3.11 ([BDB20b]). *Let f be a non-constant partial Boolean function then $\text{noisyR}(f) = \Omega(1)$.*

Theorem 3.12 ([BDB20b]). *For all partial functions f and g , $\text{R}(f \circ g) = \Omega(\text{noisyR}(f) \cdot \text{R}(g))$.*

We will also be using the following theorem of [BBGK18] regarding the composition question of $\widetilde{\text{bdeg}}$ when the outer function is PrOR_n . Informally, we will call it the Promise-OR composition theorem.

Theorem 3.13 ([BBGK18]). *For any Boolean function $g : \{0, 1\}^m \rightarrow \{0, 1\}$ we have,*

$$\widetilde{\text{bdeg}}(\text{PrOR}_n \circ g) = \Omega\left(\sqrt{n} \cdot \widetilde{\text{deg}}(g) / \log n\right).$$

4 Results about composition of R

This section is devoted to the results related to the composition of randomized query complexity. Our main result states that composition of R holds if the outer function has full randomized query complexity (Theorem 2.1). As mentioned in the proof idea, the proof critically depends on the notion of noisy randomized query complexity and its properties (introduced by Ben-David and Blais [BDB20b]).

Recall the definition of noisy randomized query complexity of a function f from Definition 3.8. As mentioned in the introduction (Theorem 2.2), Ben-David and Blais [BDB20b] proved that

$$\text{noisyR}(f) = \Theta\left(\frac{R(f \circ \text{GapMaj}_n)}{n}\right), \quad (4)$$

where GapMaj_n is the Gap-Majority function on n bits. Note that Ben-David and Blais proved Equation 4 when the arity of functions f and Gap-Majority is the same. We show that if Equation 4 can be generalized for Gap-Majority functions of arbitrary arity for some outer function f , then randomized query complexity composes for the function f . We restate the following observation from the introduction.

Observation 2.3. *Let f be a partial Boolean function on n bits. If $t(n) \geq 1$ is a non-decreasing function of n and*

$$\text{noisyR}(f) = \Omega\left(\frac{R(f \circ \text{GapMaj}_{t(n)})}{t(n)}\right),$$

then $R(f \circ g) = \Omega((R(f) \cdot R(g))/t(n))$ for all partial functions g .

Proof. Suppose $\text{noisyR}(f) = \Omega\left(\frac{R(f \circ \text{GapMaj}_{t(n)})}{t(n)}\right)$, since $R(f \circ \text{GapMaj}_t) \geq R(f)$, we have $\text{noisyR}(f) = \Omega(R(f)/t(n))$. Theorem 3.12 implies that a lower bound on noisyR translates to a lower bound on $R(f \circ g)$. We have,

$$\begin{aligned} R(f \circ g) &= \Omega(\text{noisyR}(f) \cdot R(g)) && \text{(Theorem 3.12)} \\ &= \Omega\left(\frac{R(f) \cdot R(g)}{t(n)}\right). && \square \end{aligned}$$

Observation 2.3 follows from the above observation by choosing $t(n)$ to be a small function of n .

We restate from Section 1 our generalized characterization of noisyR (i.e., generalization of Equation 4, see Appendix D for its proof).

Theorem 2.4. *Let f be a partial function on n bits and let $t \geq 1$, then $R(f \circ \text{GapMaj}_t) = O(t \cdot \text{noisyR}(f) + n)$.*

This allows us to show that if for an n -bit partial function f , $R(f) = \Theta(n)$, then $R(f \circ g) = \tilde{\Theta}(R(f) \cdot R(g))$ for all partial functions g (Theorem 2.1).

The proof of Theorem 2.1 is discussed in Section 4.1. A corollary of this theorem is that if R composes with respect to an outer function, then noisyR also composes with respect to the same outer function (Corollary 2.5).

We give proof of Theorem 2.1 in the next section and prove Corollary 2.5 in Section 4.3. We need the following theorem for these proofs, which lower bounds $R(f \circ g)$ in terms of $R(f)$ and $R(g)$.

Theorem 4.1 ([GLSS19]). *Let f and g be partial functions then $R(f \circ g) = \Omega(R(f) \cdot \sqrt{R(g)})$.*

4.1 Composition for functions with $R(f) = \Theta(n)$

We restate the theorem below.

Theorem 2.1. *Let f be a partial Boolean function on n -bits such that $R(f) = \Theta(n)$. Then for all partial functions g , we have*

$$R(f \circ g) = \Omega(R(f) \cdot R(g)).$$

Proof. From Theorem 4.1 we have a lower bound on the randomized query complexity of $(f \circ \text{GapMaj}_t)$:

$$R(f \circ \text{GapMaj}_t) = \Omega(R(f) \cdot \sqrt{t}). \quad (5)$$

On the other hand, Theorem 2.4 gives an upper bound of $O(t \cdot \text{noisyR}(f) + n)$ on $R(f \circ \text{GapMaj}_t)$. Thus, choosing $t = \left(\frac{C \cdot n}{\text{noisyR}(f)}\right)$ for a large enough constant C , we have

$$R(f) \cdot \sqrt{\frac{n}{\text{noisyR}(f)}} = O\left(\frac{n}{\text{noisyR}(f)} \cdot \text{noisyR}(f) + n\right).$$

This implies that

$$R(f) = O\left(\sqrt{n \cdot \text{noisyR}(f)}\right). \quad (6)$$

Thus, if $R(f) = \Theta(n)$, then $\text{noisyR}(f) = \Omega(R(f))$, which implies composition from Theorem 3.12. \square

Notice that Equation 6 is equivalent to the following observation.

Observation 4.2. *Let f be a partial Boolean function on n -bits. Then, $\text{noisyR}(f) = \Omega\left(\frac{R(f)^2}{n}\right)$.*

When $R(f) = \Theta(n)$, we have already seen that Observation 4.2 implies composition of randomized query complexity when the outer function is f .

Though, Observation 4.2 implies a more general result. When $R(f)$ is close to n (arity of f), Observation 4.2 places a limit on the gap between $R(f)$ and $\text{noisyR}(f)$ (consequently on the violation of composition with outer function being f). These implications are formally discussed in the next Section 4.2.

4.2 Additional implications of Observation 4.2

Without loss of generality we can assume $R(f \circ g) = \Omega(R(g))$ (note that this is true when f is non-constant).

Ben-David and Blais [BDB20b] gave a counterexample for composition, but the arity of the used function was very high compared to the randomized query complexity. They observed that the composition can still be true in the weaker sense:

$$R(f \circ g) = \Omega\left(\frac{R(f) \cdot R(g)}{\log n}\right).$$

Observation 4.2 shows that a much weaker composition result is true.

Corollary 4.3. *Let f and g be partial functions on n and m bits respectively. If $R(f \circ g) = \Omega(R(g))$, then*

$$R(f \circ g) = \Omega\left(\frac{R(f) \cdot R(g)}{\sqrt{n}}\right).$$

Proof.

$$\begin{aligned} \mathsf{R}(f \circ g) &= \Omega(\text{noisy}\mathsf{R}(f) \cdot \mathsf{R}(g)) && \text{(Theorem 3.12)} \\ &= \Omega\left(\frac{\mathsf{R}(f)^2 \cdot \mathsf{R}(g)}{n}\right). && (7) \end{aligned}$$

Where the last equality follows from Observation 4.2 ⁵ Now there are two cases:

- **Case 1.** $\mathsf{R}(f) = O(\sqrt{n})$. In this case $\mathsf{R}(f)/\sqrt{n} = O(1)$ and since we assumed $\mathsf{R}(f \circ g) = \Omega(\mathsf{R}(g))$, the claim follows from Equation 7.
- **Case 2.** $\mathsf{R}(f) = \Theta(n^{1/2} \cdot t(n))$ where $t(n)$ is a strictly increasing function of n . Thus,

$$\frac{\mathsf{R}(f)^2 \cdot \mathsf{R}(g)}{n} = \Omega(t(n)^2 \cdot \mathsf{R}(g)) = \Omega\left(\frac{\mathsf{R}(f) \cdot \mathsf{R}(g)}{\sqrt{n}}\right).$$

Again, the claim follows from Equation 7. □

The weaker composition, Corollary 4.3, implies that if $\mathsf{R}(f)$ and $\mathsf{R}(g)$ are comparable to the arity of these functions, the randomized query complexity of $f \circ g$ is “not far” from the conjectured randomized query complexity $\mathsf{R}(f) \cdot \mathsf{R}(g)$. In other words, if there is a large polynomial separation between $\mathsf{R}(f \circ g)$ and $(\mathsf{R}(f) \cdot \mathsf{R}(g))$, then $\mathsf{R}(f)$ and $\mathsf{R}(g)$ can not be too large.

Corollary 4.4. *Let f and g be partial functions such that f is a function on n -bits and g is a function on $t(n)$ -bits where $t(n)$ is a strictly increasing function of n . If $\mathsf{R}(f) = \Theta(n^\beta)$, $\mathsf{R}(g) = \Theta(n^\gamma)$ and $\mathsf{R}(f \circ g) = O((\mathsf{R}(f) \cdot \mathsf{R}(g))^\alpha)$, where $\alpha < 1$ is a constant, then $(1 - \alpha)(\alpha + \beta) < 1/2$.*

Proof. For some constants A and B we have

$$A \cdot \frac{\mathsf{R}(f) \cdot \mathsf{R}(g)}{\sqrt{n}} \leq \mathsf{R}(f \circ g) \leq B \cdot (\mathsf{R}(f) \cdot \mathsf{R}(g))^\alpha,$$

where the first inequality follows from Corollary 4.3 and second from assumption. Assigning the values of $\mathsf{R}(f)$ and $\mathsf{R}(g)$ in terms on n we have,

$$\begin{aligned} A \cdot n^{\beta+\gamma-1/2} &\leq B \cdot n^{\alpha(\beta+\gamma)} \\ n^{(1-\alpha)(\beta+\gamma)-1/2} &\leq \frac{B}{A}. \end{aligned}$$

which implies, for large enough n , $(1 - \alpha)(\beta + \gamma) \leq 1/2$. □

A special case of the above corollary is when arity and randomized query complexity of g are superpolynomial in n . In this case a polynomial gap between $\mathsf{R}(f \circ g)$ and $(\mathsf{R}(f) \cdot \mathsf{R}(g))$ is not possible.

Another implication of Theorem 2.1 is that composition of R for an outer function f implies the composition of $\text{noisy}\mathsf{R}$ for outer function being f (Corollary 2.5).

⁵Sherstov [She12] proved that for Boolean functions f and g , $\widetilde{\text{deg}}(f \circ g) = \Omega((\widetilde{\text{deg}}(f))^2 \widetilde{\text{deg}}(g)/n)$. Thus in Equation 7 we prove the same result but in the randomized world.

4.3 Proof of Corollary 2.5

First, we need the following lemma which follows from Theorem 2.4, Theorem 3.12 and Lemma 3.11.

Lemma 4.5. *Let f be a partial function on n bits and let $t = \Omega(n)$. Then*

$$\text{noisyR}(f) = \Theta\left(\frac{\text{R}(f \circ \text{GapMaj}_t)}{t}\right).$$

Proof. From Theorem 2.4 we have for all $t \geq 1$, $\text{R}(f \circ \text{GapMaj}_t) = O(t \cdot \text{noisyR}(f) + n)$. Since we have assumed $t = \Omega(n)$ and $\text{noisyR}(f) = \Omega(1)$ (Lemma 3.11), we get $\text{R}(f \circ \text{GapMaj}_t) = O(t \cdot \text{noisyR}(f))$. Thus, $\text{noisyR}(f) = \Omega\left(\frac{\text{R}(f \circ \text{GapMaj}_t)}{t}\right)$.

The upper bound $\text{noisyR}(f) = O\left(\frac{\text{R}(f \circ \text{GapMaj}_t)}{t}\right)$ follows from Theorem 3.12 and the fact that $\text{R}(\text{GapMaj}_t) = \Theta(t)$. \square

Now we prove that if R composes for f then noisyR composes for that f . For convenience, we recall the statement of the corollary from the introduction.

Corollary 2.5. *Let f be a partial Boolean function. If $\text{R}(f \circ g) = \tilde{\Theta}(\text{R}(f) \cdot \text{R}(g))$ for all partial functions g then $\text{noisyR}(f \circ g) = \tilde{\Theta}(\text{noisyR}(f) \cdot \text{noisyR}(g))$.*

Proof. From Theorem 2.2, we have

$$\text{noisyR}(f \circ g) = \Theta\left(\frac{\text{R}((f \circ g) \circ \text{GapMaj}_{mn})}{mn}\right).$$

Since $(f \circ g) \circ h = f \circ (g \circ h)$, the right hand side of the above expression is equal to

$$\Theta\left(\frac{\text{R}(f \circ (g \circ \text{GapMaj}_{mn}))}{mn}\right).$$

The proof follows from the assumption that R composes and Lemma 4.5.

$$\begin{aligned} \text{noisyR}(f \circ g) &= \Theta\left(\frac{\text{R}(f) \cdot \text{R}(g \circ \text{GapMaj}_{mn})}{mn}\right) && \text{(assuming R composes)} \\ &= \Theta(\text{R}(f) \cdot \text{noisyR}(g)) && \text{(from Lemma 4.5)} \\ &= \Theta(\text{noisyR}(f) \cdot \text{noisyR}(g)). && \text{(assuming R composes)} \end{aligned}$$

\square

5 Composition of approximate degree in terms of block sensitivity

In this section we study the composition question for approximate degree. Recall that the composition question asks: whether for all Boolean functions f and g

$$\widetilde{\text{deg}}(f \circ g) = \widetilde{\Omega}(\widetilde{\text{deg}}(f) \widetilde{\text{deg}}(g))?$$

Following our discussion from the introduction, we know that the above composition is known to hold for only two sub-classes of outer functions, namely symmetric functions [BBGK18] and functions with high approximate degree [She12]. It is thus natural to seek weaker lower bounds to make progress towards the composition question. One way to weaken the expression on the right-hand side would be to replace the measure $\widetilde{\text{deg}}(f)$ by a weaker measure (like $\sqrt{\text{s}(f)}$, $\sqrt{\text{bs}(f)}$ or $\sqrt{\text{fbs}(f)}$). Here we will establish one such lower bound of $\sqrt{\text{bs}(f)} \widetilde{\text{deg}}(g)$.

We restate our theorem now.

Theorem 2.6. *For all non-constant (possibly partial)⁶ Boolean functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and $g : \{0, 1\}^m \rightarrow \{0, 1\}$, we have*

$$\widetilde{\deg}(f \circ g) = \widetilde{\Omega}(\sqrt{\text{bs}(f)} \cdot \widetilde{\deg}(g)).$$

We note that many analogous results are known in the setting of composition of R; see, for example, [GJPW18, BDG⁺20, BDB20a, BK18, AGJ⁺17, GLSS19, BDBGM22]. To the best of our knowledge, this is the first such result in the setting of $\widetilde{\deg}$. We present only a proof sketch here; most of the technical parts of the proof appear in Appendix B.

Further, we present the sketch of the proof in two parts. For simplicity, in the first part we sketch a proof of the lower bound $\sqrt{s(f)} \widetilde{\deg}(g)$ for total function f , and then in the second part we modify the arguments to obtain Theorem 2.6.

We begin with a proof sketch for a lower bound of $\sqrt{s(f)} \widetilde{\deg}(g)$. Let $x \in \{0, 1\}^n$ be an input having the maximum sensitivity with respect to f , and $S \subseteq [n]$ be the set of sensitive bits at x ($|S| = s(f)$). Consider the subfunction f' obtained from f by fixing the set of variables *not* in S according to x . By construction, f' is defined over $s(f)$ many variables and is fully sensitive at the input $x|_S$ given by x restricted to the indices in S . Since f' is a subfunction of f and g is non-constant, we have $\widetilde{\deg}(f \circ g) \geq \widetilde{\deg}(f' \circ g)$.

Notice that f' at the neighbourhood of x , in the Boolean cube, is the partial function PrOR (Definition 3.10) or its negation. Therefore, we have $\widetilde{\deg}(f \circ g) \geq \widetilde{\deg}(f' \circ g) \geq \widetilde{\text{bdeg}}(\text{PrOR}_{|S|} \circ g)$ (see Definition 3.9 for a definition of the bounded approximate degree). We can now invoke the composition theorem for PrOR (Theorem 3.13) [BBGK18] to obtain our lower bound:

$$\widetilde{\deg}(f \circ g) \geq \widetilde{\deg}(f' \circ g) \geq \widetilde{\text{bdeg}}(\text{PrOR}_{|S|} \circ g) = \widetilde{\Omega}(\sqrt{s(f)} \widetilde{\deg}(g)).$$

However, there is a technical issue with our argument above. When we claimed that f' looks like a PrOR function we were not quite correct. Technically, it is a Shifted-PrOR function $\text{PrOR}_{|S|}^{x|_S}$, where $\text{PrOR}_n^a(y_1, y_2, \dots, y_n) := \text{PrOR}_n(y_1 \oplus a_1, y_2 \oplus a_2, \dots, y_n \oplus a_n)$ for $a \in \{0, 1\}^n$. Formally, we have

$$\widetilde{\deg}(f \circ g) \geq \widetilde{\deg}(f' \circ g) \geq \widetilde{\text{bdeg}}(\text{PrOR}_{|S|}^{x|_S} \circ g) = \widetilde{\text{bdeg}}(\text{PrOR}_{|S|} \circ (g_1, \dots, g_{|S|})), \quad (8)$$

where $g_i = g$ or $\neg g$ depending on the corresponding i -th bit in $x|_S$.

We, therefore, need a composition theorem for PrOR with *different* inner functions, while Theorem 3.13 requires that all the inner functions be the same. In fact, we would need a more general composition theorem with *different* inner *partial* functions, which we restate below. This generalization is crucially used when dealing with block sensitivity.

Theorem 2.8. *For any partial Boolean functions g_1, g_2, \dots, g_n , we have*

$$\widetilde{\text{bdeg}}(\text{PrOR}_n \circ (g_1, g_2, \dots, g_n)) = \Omega\left(\frac{\sqrt{n} \cdot \min_{i=1}^n \widetilde{\text{bdeg}}(g_i)}{\log n}\right).$$

The proof of Theorem 2.8 is a generalization of proof of Theorem 3.13. For the sake of completeness and reader's convenience, we present the proof in Appendix B (Theorem B.7)⁷.

Now returning to Eq. (8) and using Theorem 2.8, we obtain the desired lower bound:

$$\widetilde{\deg}(f \circ g) \geq \widetilde{\deg}(f' \circ g) \geq \widetilde{\text{bdeg}}(\text{PrOR}_{|S|}^{x|_S} \circ g) = \widetilde{\Omega}(\sqrt{s(f)} \widetilde{\deg}(g)).$$

We are now ready to present the modifications required to improve the lower bound to $\widetilde{\Omega}(\sqrt{\text{bs}(f)} \widetilde{\deg}(g))$.

⁶For definitions of block sensitivity and approximate degree in the context of partial functions, please see Definitions 3.4 and 3.9.

⁷For a nearly optimal generalization see Theorem B.8 in Appendix B.

Proof of Theorem 2.6. Let $b = \text{bs}(f)$ and $a = (a_1, a_2, \dots, a_n)$ be an input where f achieves the maximum block sensitivity. Further, let B_1, B_2, \dots, B_b be disjoint minimal sets of variables that achieves the block sensitivity at a , i.e., $f(a) \neq f(a^{B_i})$ for all $i \in [b]$. Recall, a^{B_i} denotes the Boolean string obtained from a by flipping the bits at all the indices given by B_i . Define a partial function $f' : \{0, 1\}^n \rightarrow \{0, 1, *\}$ such that,

$$f'(x) = \begin{cases} 0 & \text{if } x = a, \\ 1 & \text{if } x = a^{B_i}, \text{ for some } i \in [b], \\ * & \text{otherwise.} \end{cases}$$

Note that f contains f' or its negation as a sub function. Thus, $\widetilde{\text{deg}}(f \circ g) \geq \widetilde{\text{bdeg}}(f' \circ g)$.

Since g is non-constant, we can fix the indices *not* in $\bigcup_{i=1}^b B_i$ according to a to obtain $f'' \circ g$. We would now like to embed PrOR_b over the remaining variables in f'' . For this purpose we define the following partial functions: for every $i \in [b]$, let $I_i : \{0, 1\}^{B_i} \rightarrow \{0, 1, *\}$ be such that

$$I_i(x) = \begin{cases} 0 & \text{if } x = a|_{B_i}, \\ 1 & \text{if } x = a^{B_i}|_{B_i}, \\ * & \text{otherwise.} \end{cases}$$

Now observe that $f'' \circ g$ can be rewritten as $\text{PrOR}_b \circ (I_1 \circ g, \dots, I_b \circ g)$. We therefore have

$$\begin{aligned} \widetilde{\text{deg}}(f \circ g) &\geq \widetilde{\text{bdeg}}(f' \circ g) \geq \widetilde{\text{bdeg}}(f'' \circ g) = \widetilde{\text{bdeg}}(\text{PrOR}_b \circ (I_1 \circ g, \dots, I_b \circ g)) \\ &= \Omega\left(\frac{\sqrt{b} \cdot \min_i \widetilde{\text{bdeg}}(I_i \circ g)}{\log b}\right) = \widetilde{\Omega}\left(\sqrt{b} \cdot \widetilde{\text{deg}}(g)\right), \end{aligned}$$

where the second last equality follows from Theorem 2.8 and the last equality uses the fact $\widetilde{\text{bdeg}}(I_i \circ g) \geq \widetilde{\text{deg}}(g)$ for all i , which in turn follows from each I_i being non-constant. \square

We end this section with few final remarks. As a corollary to Theorem 2.6 we have the following composition for $\widetilde{\text{deg}}$ when the outer function has minimal approximate degree with respect to its block sensitivity.

Corollary 2.7. *For all Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ with $\widetilde{\text{deg}}(f) = \Theta(\sqrt{\text{bs}(f)})$ and for all $g : \{0, 1\}^m \rightarrow \{0, 1\}$, we have $\widetilde{\text{deg}}(f \circ g) = \widetilde{\Theta}(\widetilde{\text{deg}}(f) \cdot \widetilde{\text{deg}}(g))$.*

We also note that the set of Boolean functions with $\widetilde{\text{deg}}(f) = \Theta(\sqrt{\text{bs}(f)})$ includes examples of *non-symmetric* functions f with *low* approximate degree. In other words, when such functions are outer function in a composed function then the composition of $\widetilde{\text{deg}}$ doesn't follow from the known results [BBGK18, She12]. For example, consider the Rubinstein function RUB with arity n (Definition A.3). It is a non-symmetric function with $\widetilde{\text{deg}}(\text{RUB}) = O(\sqrt{n} \log n)$ (Lemma A.7). Thus, the composition of $\widetilde{\text{deg}}(\text{RUB} \circ g)$ does not follow from [BBGK18] or [She12]. However, $\widetilde{\text{deg}}(\text{RUB}) = \widetilde{\Theta}(\sqrt{\text{bs}(\text{RUB})})$, and hence from Corollary 2.7, $\widetilde{\text{deg}}(\text{RUB} \circ g) = \widetilde{\Theta}(\widetilde{\text{deg}}(\text{RUB}) \widetilde{\text{deg}}(g))$.

Another example is given by the SINK function on $\Theta(n^2)$ variables (Definition A.2). Its approximate degree is $O(\sqrt{n} \log n)$ (Lemma A.7), while $\text{bs}(\text{SINK}) = \Theta(n)$. Thus, $\widetilde{\text{deg}}(\text{SINK} \circ g) = \widetilde{\Theta}(\widetilde{\text{deg}}(\text{SINK}) \widetilde{\text{deg}}(g))$ follows from Corollary 2.7.

As stated in the introduction, we recall that Theorem 2.6 is tight in terms of block-sensitivity, i.e., the lower bound can not be improved to $\widetilde{\Omega}(\text{bs}(f)^c \cdot \widetilde{\text{deg}}(g))$ for some $c > 1/2$.

6 Conclusion

While our work makes progress on the composition problem for R and $\widetilde{\text{deg}}$, the main problems of whether $\widetilde{\text{deg}}$ and R composes for any pair of Boolean functions still remains open. In this light, we would like to highlight some questions that can be useful stepping stones towards the main questions.

We showed that the composition question for R is equivalent to the following open question (which is a generalization of Ben-David and Blais [BDB20b] result):

Open question 6.1. *Let $f : \{0, 1\}^n \rightarrow \{0, 1, *\}$ be a Boolean function. Then, is it true that for arbitrary t , $\text{noisy}R(f) = \Theta(R(f \circ \text{GapMaj}_t)/t)$?*

In case of approximate degree composition, a natural question is whether $\sqrt{\text{bs}(f)}$ can be replaced by some other complexity measures. In this regards we state the following open problems:

Open question 6.2. *For all Boolean functions f and g , can we prove either of the following:*

- $\widetilde{\text{deg}}(f \circ g) = \Omega(\sqrt{\text{deg}(f)} \cdot \widetilde{\text{deg}}(g))$?
- $\widetilde{\text{deg}}(f \circ g) = \Omega(\sqrt{\text{fbs}(f)} \cdot \widetilde{\text{deg}}(g))$?

Recently, in [SYZ04, Sun07, Dru11, Cha11, CKP22], the classes of transitive functions got a lot of attention as natural generalization of the classes of symmetric functions. Can the result for symmetric functions be extended to transitive functions?

Open question 6.3. *Can we prove that $\widetilde{\text{deg}}$ and R compose when the outer function is transitive?*

References

- [ABB⁺17] Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs. Separations in query complexity based on pointer functions. *Journal of the ACM*, 64(5):32:1–32:24, 2017.
- [ABK16] Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *STOC*, pages 863–876, 2016.
- [ABK⁺21] Scott Aaronson, Shalev Ben-David, Robin Kothari, Shrivastava Rao, and Avishay Tal. Degree vs. approximate degree and quantum implications of Huang’s sensitivity theorem. In *STOC*, pages 1330–1342, 2021.
- [AGJ⁺17] Anurag Anshu, Dmitry Gavinsky, Rahul Jain, Srijita Kundu, Troy Lee, Priyanka Mukhopadhyay, Miklos Santha, and Swagato Sanyal. A composition theorem for randomized query complexity. In *FSTTCS*, volume 93, pages 10:1–10:13, 2017.
- [AKK16] Andris Ambainis, Martins Kokainis, and Robin Kothari. Nearly optimal separations between communication (or query) complexity and partitions. In *CCC*, volume 50, pages 4:1–4:14, 2016.
- [Amb05] Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory Comput.*, 1(1):37–46, 2005.
- [BBGK18] Shalev Ben-David, Adam Bouldan, Ankit Garg, and Robin Kothari. Classical lower bounds from quantum upper bounds. In *FOCS*, pages 339–349, 2018.
- [Bd02] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

- [BDB20a] Shalev Ben-David and Eric Blais. A tight composition theorem for the randomized query complexity of partial functions. *arXiv:2002.10809*, 2020.
- [BDB20b] Shalev Ben-David and Eric Blais. A tight composition theorem for the randomized query complexity of partial functions: Extended abstract. In *FOCS*, pages 240–246, 2020.
- [BDBGM22] Shalev Ben-David, Eric Blais, Mika Göös, and Gilbert Maystre. Randomised composition and small-bias minimax. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 624–635. IEEE, 2022.
- [BDG⁺20] Andrew Bassilakis, Andrew Drucker, Mika Göös, Lunjia Hu, Weiyun Ma, and Li-Yang Tan. The power of many samples in query complexity. In *ICALP*, volume 168, pages 9:1–9:18, 2020.
- [BDGKW20] Shalev Ben-David, Mika Göös, Robin Kothari, and Thomas Watson. When Is Amplification Necessary for Composition in Randomized Query Complexity? In *APPROX/RANDOM 2020*, volume 176, pages 28:1–28:16, 2020.
- [BK18] Shalev Ben-David and Robin Kothari. Randomized query complexity of sabotaged and composed functions. *Theory Comput.*, 14(1):1–27, 2018.
- [BKT19] Mark Bun, Robin Kothari, and Justin Thaler. Quantum algorithms and approximating polynomials for composed functions with shared inputs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 662–678. SIAM, 2019.
- [BNRdW07] Harry Buhrman, Ilan Newman, Hein Röhrig, and Ronald de Wolf. Robust polynomials and quantum algorithms. *Theory Comput. Syst.*, 40(4):379–395, 2007.
- [BT13] Mark Bun and Justin Thaler. Dual lower bounds for approximate degree and Markov-Bernstein inequalities. In *ICALP*, volume 7965, pages 303–314, 2013.
- [BWY15] Eric Blais, Amit Weinstein, and Yuichi Yoshida. Partially symmetric functions are efficiently isomorphism testable. *SIAM J. Comput.*, 44(2):411–432, 2015.
- [CFGM12] Sourav Chakraborty, Eldar Fischer, David García-Soriano, and Arie Matsliah. Junta-symmetric functions, hypergraph isomorphism and crunching. In *CCC*, pages 148–158, 2012.
- [Cha11] Sourav Chakraborty. On the sensitivity of cyclically-invariant boolean functions. *Discret. Math. Theor. Comput. Sci.*, 13(4):51–60, 2011.
- [CKP22] Sourav Chakraborty, Chandrima Kayal, and Manaswi Paraashar. Separations between combinatorial measures for transitive functions. In *ICALP*, volume 229, pages 36:1–36:20, 2022.
- [CMS20] Arkadev Chattopadhyay, Nikhil S. Mande, and Suhail Sherif. The log-approximate-rank conjecture is false. *Journal of the ACM*, 67, 2020.
- [Dru11] Andrew Drucker. Block sensitivity of minterm-transitive functions. *Theor. Comput. Sci.*, 412(41):5796–5801, 2011.

- [Fel91] William Feller. *An Introduction to Probability Theory and Its Applications*. Wiley Series in Probability and Statistics, 1991.
- [GJPW18] Mika Göös, T. S. Jayram, Toniann Pitassi, and Thomas Watson. Randomized communication versus partition number. *ACM Trans. Comput. Theory*, 10(1), 2018.
- [GLSS19] Dmitry Gavinsky, Troy Lee, Miklos Santha, and Swagato Sanyal. A Composition Theorem for Randomized Query Complexity via Max-Conflict Complexity. In *ICALP*, volume 132, pages 64:1–64:13, 2019.
- [GSS16] Justin Gilmer, Michael E. Saks, and Srikanth Srinivasan. Composition limits and separating examples for some Boolean function complexity measures. *Combinatorica*, 36(3):265–311, 2016.
- [Hua19] Hao Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *Annals of Mathematics*, 190(3):949–955, 2019.
- [Kim13] Shelby Kimmel. Quantum adversary (upper) bound. *Chicago Journal of Theoretical Computer Science*, 2013(4), April 2013.
- [Li21] Yaqiao Li. Conflict complexity is lower bounded by block sensitivity. *Theor. Comput. Sci.*, 856:169–172, 2021.
- [LMR⁺11] Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *FOCS*, pages 344–353, 2011.
- [Mon14] Ashley Montanaro. A composition theorem for decision tree complexity. *Chicago Journal of Theoretical Computer Science*, 2014(6), July 2014.
- [Nis89] Noam Nisan. Crew prams and decision trees. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 327–335, 1989.
- [NS94] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Comput. Complex.*, 4:301–313, 1994.
- [Pat92] Ramamohan Paturi. On the degree of polynomials that approximate symmetric boolean functions. In *STOC*, page 468–474, 1992.
- [Rei11] Ben W. Reichardt. *Reflections for quantum query algorithms*, pages 560–569. SODA. SIAM, 2011.
- [Rub95] David Rubinfeld. Sensitivity vs. block sensitivity of Boolean functions. *Combinatorica*, 15(2):297–299, 1995.
- [Sav02] Petr Savický. On determinism versus unambiguous nondeterminism for decision trees. *Electron. Colloquium Comput. Complex.*, 9(009), 2002.
- [She12] A. A. Sherstov. Strong direct product theorems for quantum communication and query complexity. *SIAM J. Comput.*, 41(5):1122–1165, 2012.
- [She13a] Alexander A. Sherstov. Approximating the AND-OR tree. *Theory Comput.*, 9:653–663, 2013.
- [She13b] Alexander A. Sherstov. The intersection of two halfspaces has high threshold degree. *SIAM J. Comput.*, 42(6):2329–2374, 2013.

- [She13c] Alexander A. Sherstov. Making polynomials robust to noise. *Theory Comput.*, 9:593–615, 2013.
- [Sun07] Xiaoming Sun. Block sensitivity of weakly symmetric functions. *Theor. Comput. Sci.*, 384(1):87–91, 2007.
- [SYZ04] Xiaoming Sun, Andrew Chi-Chih Yao, and Shengyu Zhang. Graph properties and circular functions: How low can quantum query complexity go? In *CCC*, pages 286–293, 2004.
- [Tal13] Avishay Tal. Properties and applications of boolean function composition. In *ITCS*, page 441–454, 2013.
- [Tal14] Avishay Tal. Shrinkage of De Morgan formulae by spectral techniques. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 551–560. IEEE, 2014.

A Some important Boolean functions

In this section, we define some important Boolean functions that have been used in the paper. We start with Multiplexer Function or Addressing Function.

Definition A.1 (Multiplexer Function or Addressing Function). *The function $\text{MUX} : \{0, 1\}^{k+2^k} \rightarrow \{0, 1\}$ with input $(x_0, \dots, x_{k-1}, y_0, \dots, y_{2^k-1})$ outputs the bit y_t , where $t = \sum_{i=0}^{k-1} x_i 2^i$.*

The following function was defined in [CMS20].

Definition A.2 (SINK). *Consider a tournament defined on k vertices with $\binom{k}{2}$ variables such that, for $i < j$, if $x_{ij} = 1$ then there is an outgoing edge from i to j . A vertex $i \in [n]$ is a sink-vertex if all edges incident to it are incoming edges. For $x \in \binom{k}{2}$, $\text{SINK}(x) = 1$ if there a vertex $i \in [n]$ such that i is a sink-vertex, and 0 otherwise.*

Definition A.3 (Rubinstein function [Rub95]). *Let $g : \{0, 1\}^k \rightarrow \{0, 1\}$ be such that $g(x) = 1$ iff x contains two consecutive ones and the rest of the bits are 0. The Rubinstein function, denoted by $\text{RUB} : \{0, 1\}^{k^2} \rightarrow \{0, 1\}$, is defined to be $\text{RUB} = \text{OR}_k \circ g$.*

A.1 Some properties of SINK and Rubinstein function

Following is an easy observation.

Observation A.4. *The sensitivity of $\text{SINK} : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ is at least $(k - 1)$. Consider a tournament on vertices v_1, \dots, v_k such that v_1 is a sink-vertex and (v_2, \dots, v_k, v_2) is a cycle. Observe that flipping any edge incident to v_1 changes the value of the function from 1 to 0. In particular, $\text{bs}(\text{SINK}) \geq \text{s}(\text{SINK}) = \Omega(k)$.*

We now want to give an upper bound on the approximate degree of these two functions. For this, we first need the following generalization of approximate degree.

Definition A.5 (ε -Error Approximate Degree ($\widetilde{\text{deg}}_\varepsilon$)). *A polynomial $p : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to ε -approximate a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ if*

$$|p(x) - f(x)| \leq \varepsilon, \quad \forall x \in \{0, 1\}^n.$$

The ε -approximate degree of f ($\widetilde{\text{deg}}_\varepsilon(f)$) is the minimum possible degree of a polynomial which ε -approximates f .

It is known (see [BNRdW07] also [Tal14, Appendix A]) that given a polynomial p that approximates a Boolean function f to error $1/3$, one can come up with a polynomial p' that ε -approximates f such that $\deg(p') = O(\deg(p) \log(1/\varepsilon))$.

Also, the following theorem will be used in this section.

Theorem A.6 ([She13c]). *For all Boolean functions f and g , $\widetilde{\deg}(f \circ g) = O(\widetilde{\deg}(f)\widetilde{\deg}(g))$.*

We now upper bound the approximate degree of SINK and RUB.

Lemma A.7. *For the sink function $\text{SINK} : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ and the Rubinstein function $\text{RUB} : \{0, 1\}^{k^2} \rightarrow \{0, 1\}$ we have*

1. $\widetilde{\deg}(\text{SINK}) = O(\sqrt{k} \log k)$, and
2. $\widetilde{\deg}(\text{RUB}) = O(k \log k)$.

Proof. We first upper bound the approximate degree of SINK. For every vertex $i \in [n]$, there is a sink at that vertex if and only if all edges incident to that vertex are incoming. This is equivalent to saying that the variables x_{ij} , when $j > i$, are 0 and x_{ji} , when $j < i$, are 1. This can be computed by an AND_k function and we call this function $\text{AND}_k^{(i)}$. Also, since any tournament has at most one sink vertex, SINK can be expressed as the sum of $\text{AND}_k^{(i)}$'s, for $i \in [k]$. Recall that the approximate degree of AND_k is $O(\sqrt{k})$, which, from Theorem A.6, implies that $1/(3k)$ -error approximate degree of AND_k is $O(\sqrt{k} \log k)$. Replacing each $\text{AND}_k^{(i)}$ with $1/(3k)$ -error approximating polynomial gives a $1/3$ -error approximating polynomial for SINK with degree $O(\sqrt{k} \log k)$.

Now we upper bound the approximate degree of Rubinstein function. Recall that $\text{RUB} : \{0, 1\}^{k^2} \rightarrow \{0, 1\}$ is defined as $\text{RUB} = \text{OR}_k \circ g$, where $g : \{0, 1\}^k \rightarrow \{0, 1\}$ is a function such that $g(x) = 1$ if and only if x contains two consecutive 1's and the rest of the bits are 0. Observe that there are only $(k - 1)$ inputs on which g takes value 1, call these inputs z_1, \dots, z_{k-1} . Let $\text{AND}_k^{(i)}$ be the Boolean function which evaluates to 1 if and only if its input is z_i . Thus g can be expressed as the sum of $\text{AND}_k^{(i)}$'s, for $i \in [k]$. By a similar argument as in the last paragraph, the $1/3$ -error approximate degree of g can be bounded by $O(\sqrt{k} \log k)$. From Theorem A.6, this implies an $O(k \log k)$ upper bound on the approximate degree of RUB. \square

B Approximate degree of Promise-OR composed with different inner functions

In this section we show that the approximate degree composes when the outer function is PrOR and the inner functions are (possibly) different partial functions. The proof is essentially a straightforward generalization of the proof of Theorem 3.13 [BBGK18, Theorem 16 (arXiv version)]. However, for the sake of completeness and reader's convenience, we give an overview of the proof here. We will need some definitions and theorems from [BBGK18] which we state now. We start with the definition of a problem called "singleton combinatorial group testing". It generalizes the combinatorial group testing problem.

Definition B.1 (Singleton CGT). *Let D be the set of all $w \in \{0, 1\}^{2^n}$ for which there exists an $x \in \{0, 1\}^n$ such that for all $S \subseteq [n]$ satisfying $\sum_{i \in S} x_i \in \{0, 1\}$, we have $\sum_{i \in S} x_i = w_S$. Note that for all $w \in D$, the string x is uniquely defined by $x_i = w_{\{i\}}$. Let us denote this string by $x(w)$. We then define the partial function $\text{SCGT}_{2^n} : D \rightarrow \{0, 1\}^n$ by $\text{SCGT}_{2^n}(w) = x(w)$.*

Theorem B.2 ([BBGK18, Theorem 19 (arXiv version)]). *The bounded-error quantum query complexity of SCGT_{2^n} is $\Theta(\sqrt{n})$.*

For a formal Definition of bounded error quantum query complexity we refer the survey by [Bd02]. Before we state the next result that we need from [BBGK18] we are defining robustness of a polynomial to input noise.

Definition B.3 (Robustness to input noise). *For any function $f : \{0, 1\}^n \rightarrow \{0, 1, *\}$ we say a polynomial $p : \{0, 1\}^n \rightarrow \mathbb{R}$ approximately computes f with δ -robustness where $\delta \in [0, \frac{1}{2})$ if for any $x \in \text{Dom}(f)$ and $\Delta \in [-\delta, \delta]^n$, we have $|f(x) - p(\Delta + x)| \leq \frac{1}{3}$.*

Now we are ready to state the next result.

Theorem B.4 ([BBGK18, Theorem 17 (arXiv version)]). *For a partial Boolean function f , there exists a bounded multilinear polynomial p of degree $O(Q(f))$ that approximates f with robustness $\Omega(1/Q(f)^2)$ where $Q(f)$ is the bounded error quantum query complexity of the function f .*

We refer [BBGK18] for more details about robustness of a polynomial induces by quantum algorithm. We also need the existence of a multilinear robust polynomial for $\text{XOR}_n \circ \text{SCGT}_{2^n}$, which follows from Theorems B.2 and B.4 above, where $\text{XOR}_n \circ \text{SCGT}_{2^n}$ is the parity of n output bits of SCGT_{2^n} .

Theorem B.5 ([BBGK18, Theorem 20 (arXiv version)]). *There is a real polynomial p of degree $O(\sqrt{n})$ over 2^n variables $\{w_S\}_{S \subseteq [n]}$ and a constant $c \geq 10^{-5}$ such that for any input $w \in \{0, 1\}^{2^n}$ with $\text{XOR}_n \circ \text{SCGT}_{2^n}(w) \neq *$ and any $\Delta \in [-c/n, c/n]^{2^n}$,*

$$|p(w + \Delta) - \text{XOR}_n \circ \text{SCGT}_{2^n}(w)| \leq 1/3.$$

Furthermore, p is multilinear and for all $w \in \{0, 1\}^{2^n}$, $p(w) \in [0, 1]$.

We also need the following result of Sherstov that shows composition holds for the approximate degree of the parity of n different functions.

Theorem B.6 ([She12, Theorem 5.9]). *For any partial Boolean functions f_1, \dots, f_n , we have*

$$\widetilde{\text{bdeg}}(\text{XOR} \circ (f_1, \dots, f_n)) = \Omega\left(\sum_{i=1}^n \widetilde{\text{bdeg}}(f_i)\right).$$

We are now ready to prove Theorem 2.8 which we restate below.

Theorem B.7. *For any partial Boolean functions f_1, f_2, \dots, f_n , we have*

$$\widetilde{\text{bdeg}}(\text{PrOR}_n \circ (f_1, f_2, \dots, f_n)) = \Omega\left(\frac{\sqrt{n} \cdot \min_{i=1}^n \widetilde{\text{bdeg}}(f_i)}{\log n}\right).$$

Furthermore the following upper bound also holds,

$$\widetilde{\text{bdeg}}(\text{PrOR}_n \circ (f_1, f_2, \dots, f_n)) = O\left(\sqrt{n} \cdot \max_{i=1}^n \widetilde{\text{bdeg}}(f_i) \cdot \log n\right).$$

Proof. The upper bound follows by first amplifying the approximation of inner function to within error $\Theta(1/n)$ and then composing with the polynomial given by Theorem B.4 for PrOR.

For amplification one can use the univariate amplification polynomial of degree $O(\log(1/\varepsilon))$ that maps $[0, 1/3]$ to $[0, \varepsilon]$, $[2/3, 1]$ to $[1 - \varepsilon, 1]$, and $[1/3, 2/3]$ to $[0, 1]$ given by [BNRdW07, Lemma 1].

We now give an overview of the lower bound proof following [BBGK18].

Let q be an approximating polynomial for $\text{PrOR} \circ (f_1, \dots, f_n)$ of degree $T := \widetilde{\text{bdeg}}(\text{PrOR} \circ (f_1, \dots, f_n))$, which is also bounded on all Boolean inputs outside the promise. Let q' be the polynomial obtained from q by amplifying the approximation to within error c/n on all inputs in the promise, where c is the constant from Theorem B.5. Note that the degree of q' is $O(T \cdot \log n)$, and it remains bounded on all possible Boolean inputs.

We can assume without loss of generality that f_1, f_2, \dots, f_n are non-constant partial Boolean functions. Therefore for each f_i there exists an input y such that $f_i(y) = 0$. For all $S \subseteq [n]$, we now define a polynomial q'_S using q that approximates $\text{PrOR} \circ (f_i)_{i \in S}$ by setting the variables of each $f_i, i \notin S$, to an input where it evaluates to 0. Note that the degree of q'_S is bounded by the degree of q' .

Now consider the polynomial p over 2^n variables $\{w_S\}_{S \subseteq [n]}$ given by Theorem B.5. Let r be the polynomial obtained from p by replacing the variables w_S by polynomials q'_S , i.e., $r = p \circ (q'_S)_{S \subseteq [n]}$. Clearly the degree of r is $O(T\sqrt{n} \log n)$.

It can now be argued that r approximates $\text{XOR} \circ (f_1, \dots, f_n)$ to error within $1/3$. A slight care is needed in this argument for p expects inputs which are Δ -close to Boolean values $\{0, 1\}$. However, it may happen that some q'_S , though bounded in $[0, 1]$, is not close to Boolean values $\{0, 1\}$. This is where we will use the fact that p is also *multilinear*, and hence the value of p on a convex combination of Boolean inputs is equal to the convex combination of values of p on the Boolean inputs. Hence, if p works correctly when all inputs are in $\{0, 1\}$, then it must also be correct on inputs in $[0, 1]$. A final thing to note is that any invalid input (to p) in $[0, 1]^{2^n}$ can be written as a convex combination of valid inputs.

Thus, we have

$$\widetilde{\text{bdeg}}(\text{XOR} \circ (f_1, \dots, f_n)) = O(T\sqrt{n} \log n) = O(\sqrt{n} \log n \cdot \widetilde{\text{bdeg}}(\text{PrOR} \circ (f_1, \dots, f_n))).$$

Whereas from Theorem B.6 we have

$$\widetilde{\text{bdeg}}(\text{XOR} \circ (f_1, \dots, f_n)) = \Omega \left(\sum_{i=1}^n \widetilde{\text{bdeg}}(f_i) \right) = \Omega \left(n \cdot \min_i \widetilde{\text{bdeg}}(f_i) \right).$$

Combining the two, we obtain the lower bound

$$\widetilde{\text{bdeg}}(\text{PrOR} \circ (f_1, \dots, f_n)) = \Omega \left(\frac{\sqrt{n} \cdot \min_{i=1}^n \widetilde{\text{bdeg}}(f_i)}{\log n} \right).$$

□

We will now use this weak bound to establish nearly optimal bound for the approximate degree of PrOR composed with n different *partial* functions. This will again be a simple generalization of OR composed with different functions [BBGK18, Theorem 37]. For the sake of completeness, we work out some of the details.

Theorem B.8. *For any partial Boolean functions f_1, f_2, \dots, f_n , we have*

$$\widetilde{\text{bdeg}}(\text{PrOR}_n \circ (f_1, f_2, \dots, f_n)) = \tilde{\Theta} \left(\sqrt{\sum_{i=1}^n \widetilde{\text{bdeg}}(f_i)^2} \right),$$

when the lcm of $\widetilde{\text{bdeg}}(f_i)^2$ for $i \in [n]$ is $\Theta(\max_i \widetilde{\text{bdeg}}(f_i)^2)$.

Proof. As mentioned before, the proof is merely working out the details of [BBGK18, Theorem 37] while keeping in mind that we are working with *partial* functions.

Let $F = \text{PrOR}_n \circ (f_1, f_2, \dots, f_n)$, $d_i = \widetilde{\text{bdeg}}(f_i)^2$ for $i \in [n]$, and ℓ be the lcm of d_i 's. Now consider the function $G = \text{PrOR}_\ell \circ F$. From Theorem B.7, we have the following bounds on $\widetilde{\text{bdeg}}(G)$ up to constants

$$\frac{\sqrt{\ell} \cdot \widetilde{\text{bdeg}}(F)}{\log \ell} \leq \widetilde{\text{bdeg}}(G) \leq \sqrt{\ell} \cdot \widetilde{\text{bdeg}}(F) \cdot \log \ell. \quad (9)$$

Now using the associativity of PrOR we can rewrite G as

$$G = \text{PrOR}_{n\ell} \circ (\underbrace{f_1, \dots, f_1}_{\ell \text{ times}}, \dots, \underbrace{f_n, \dots, f_n}_{\ell \text{ times}}). \quad (10)$$

Further regrouping f_i 's, we can rewrite G as follows

$$G = \text{PrOR}_d \circ (\underbrace{\text{PrOR}_{\ell/d_1} \circ f_1, \dots, \text{PrOR}_{\ell/d_1} \circ f_1}_{d_1 \text{ times}}, \dots, \underbrace{\text{PrOR}_{\ell/d_n} \circ f_n, \dots, \text{PrOR}_{\ell/d_n} \circ f_n}_{d_n \text{ times}}), \quad (11)$$

where $d = \sum_{i=1}^n d_i$. Now using Theorem B.7 and $\sqrt{d_i} = \widetilde{\text{bdeg}}(f_i)$, we obtain following bounds for $\text{PrOR}_{\ell/d_i} \circ f_i$ (up to constants)

$$\frac{\sqrt{\ell}}{\log(\ell/d_i)} \leq \widetilde{\text{bdeg}}(\text{PrOR}_{\ell/d_i} \circ f_i) \leq \sqrt{\ell} \cdot \log(\ell/d_i). \quad (12)$$

Now consider (11) and using Theorem B.7 along with (12), we obtain

$$\frac{\sqrt{d\ell}}{\log d \cdot \log \ell} \leq \widetilde{\text{bdeg}}(G) \leq \sqrt{d\ell} \cdot \log \ell \cdot \log d \quad (13)$$

Now from (13) and (9) it follows

$$\frac{\sqrt{d}}{\log d \cdot \log^2 \ell} \leq \widetilde{\text{bdeg}}(F) \leq \sqrt{d} \cdot \log^2 \ell \cdot \log d.$$

□

C Composition theorems for strongly- k -junta symmetric outer functions

In this section we will prove the composition result of $\widetilde{\text{deg}}$ and R when the outer function has some amount of symmetry. Of course, there are various notion of symmetry. Traditionally a function is said to have the maximum amount of symmetry when the function value is invariant under any permutation of the variables. Such functions are called symmetric. Symmetric functions are very well studied in the literature of Boolean function analysis. In the terms of composition theorems of $\widetilde{\text{deg}}$ and R it was proved in [BBGK18] and [GJPW18] that $\widetilde{\text{deg}}$ and R respectively composes when the outer function is symmetric.

In terms of weaker notions of symmetry there are various possible definitions. In this paper we consider the case of strongly- k -junta symmetric functions. The composition theorem for $\widetilde{\text{deg}}$ when the outer function is strongly- k -junta symmetric (Theorem 2.11(Part(i))) is presented in Appendix C.1. The proof of the composition theorem for R when the outer function is strongly- k -junta symmetric (Theorem 2.11(Part(ii))) follows easily from Theorem 2.1.

Observation C.1. For any strongly k -junta symmetric function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and any Boolean function $g : \{0, 1\}^m \rightarrow \{0, 1\}$, we have $R(f \circ g) = \tilde{\Omega}(R(f) \cdot R(g))$ where $n - k = \Theta(n)$.

Proof. There exists an assignment of the k -bits such that the resulting function is a non-constant symmetric function on $(n - k)$ bits. Since the sensitivity of the restricted function is $\Omega(n)$, the randomized query complexity is also $\Omega(n)$ (see [Nis89]). Hence, from Theorem 2.1 the result follows. \square

C.1 Composition of approximate degree for \sqrt{n} -junta symmetric functions

A crucial result that we use in the prove of composition theorem of $\widetilde{\deg}$ is the following result from [Pat92].

Theorem C.2 ([Pat92]). For any non-constant symmetric function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, let k be the closest integer to $n/2$ such that f takes different values on inputs of Hamming weight k and $k + 1$. Define,

$$\gamma(f) = \begin{cases} k & \text{if } k \leq n/2, \\ n - k & \text{otherwise.} \end{cases}$$

Then

$$\widetilde{\deg}(f) = \Theta\left(\sqrt{n(\gamma(f) + 1)}\right).$$

Using the result of [Pat92] we prove the following proposition about the approximate degree of a k -junta symmetric function. Recall the multiplexer function from Definition A.1.

Proposition C.3. For any k -junta symmetric function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, we have $\widetilde{\deg}(f) = \Omega\left(\sqrt{(n - k)\gamma_{\max}}\right)$ and $\widetilde{\deg}(f) = O\left(\max\{k, \sqrt{(n - k)\gamma_{\max}}\}\right)$, where $\gamma_{\max} = \max_{i \in \{0, 1\}^k} \{\gamma(f_i)\}$ such that f_i is the symmetric function obtained by restricting the junta variables according to i .

Proof. Fixing the junta variables in f we obtain a symmetric function on $n - k$ variables with approximate degree $\Omega(\sqrt{(n - k)\gamma_{\max}})$ (Theorem C.2), which in turn implies the same lower bound on $\widetilde{\deg}(f)$.

For the upper bound, we obtain an approximating polynomial for f by composing the (exact) polynomial for the multiplexer function $\text{MUX} : \{0, 1\}^{k+2^k} \rightarrow \{0, 1\}$ with the approximating polynomials for different symmetric functions obtained by restricting the k junta variables. Therefore, $\widetilde{\deg}(f) = k + O(\sqrt{(n - k)\gamma_{\max}}) = O\left(\max\{k, \sqrt{(n - k)\gamma_{\max}}\}\right)$. \square

As mentioned earlier, the composition of $\widetilde{\deg}$ when the outer function is symmetric was proved in [BBGK18]. The following is their result that we crucially use in the proof of Theorem 2.11.

Theorem C.4 ([BBGK18]). For any symmetric Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and any Boolean function $g : \{0, 1\}^m \rightarrow \{0, 1\}$ we have,

$$\widetilde{\deg}(f \circ g) = \tilde{\Omega}(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g))$$

We now present the proof of Part (i) of Theorem 2.11, that the proof of composition of $\widetilde{\deg}$ when the outer function is strongly- k -junta symmetric.

Proof of Theorem 2.11(Part (i)). Since f is a strongly- k -junta symmetric function so there exists a setting of the k junta variables such that the resulting function is a non-constant symmetric function. Let f' be the symmetric function obtained by restricting the junta variables of f so that f' is non-constant. Then by Theorem C.2 the approximate degree of f' is $\Omega(\sqrt{(n-k)\gamma_{\max}})$. Then clearly we have

$$\widetilde{\deg}(f \circ g) \geq \widetilde{\deg}(f' \circ g) = \widetilde{\Omega}(\widetilde{\deg}(f') \cdot \widetilde{\deg}(g)) = \widetilde{\Omega}(\sqrt{(n-k)\gamma_{\max}} \cdot \widetilde{\deg}(g)), \quad (14)$$

where the first equality follows from Theorem C.4. Now from Proposition C.3 we know that $\widetilde{\deg}(f) = O(\sqrt{(n-k)\gamma_{\max}})$ if $k = O(\sqrt{(n-k)\gamma_{\max}})$, which is satisfied when $k = O(\sqrt{n})$. Thus from (14) we obtain

$$\widetilde{\deg}(f \circ g) = \widetilde{\Omega}(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g)).$$

□

D Proof of Theorem 2.4

Let us start by recalling the theorem we want to prove in this section.

Theorem 2.4. *Let f be a partial function on n bits and let $t \geq 1$, then $R(f \circ \text{GapMaj}_t) = O(t \cdot \text{noisyR}(f) + n)$.*

We mention two lemmas from Ben-David and Blais [BDB20a] that we need for the proof of this theorem. The first lemma shows that given a noisy oracle with γ bias, it can simulate an oracle of $\gamma' \geq \gamma$ bias by making approximately $\left(\frac{\gamma'}{\gamma}\right)^2$ queries.

Lemma D.1 ([BDB20a]). *Let $\gamma \in [-1/3, 1/3]$ be nonzero and $k \leq 1/\gamma^2$ be an odd positive integer. If we take k independent samples from Bernoulli $((1 + \gamma)/2)$ and take their majority, the resultant distribution is Bernoulli $((1 + \gamma')/2)$ where $\text{sign}(\gamma') = \text{sign}(\gamma)$ and*

$$\frac{\sqrt{k}|\gamma|}{3} \leq |\gamma'| \leq 3\sqrt{k}|\gamma|.$$

The following lemma shows that it is enough to query the noisy oracle with just two biases.

Lemma D.2 ([BDB20a]). *Let f be a partial function and let A be an optimal noisy randomized algorithm for A of cost $\text{noisyR}(f)$. Then there is another noisy randomized algorithm \widehat{A} of cost $O(\text{noisyR}(f))$, which only queries its noisy oracles with parameter either $\gamma = 1$ or $\gamma = \widehat{\gamma}$, where $\widehat{\gamma} > 0$ is the smallest bias that A uses on any input.*

Another important lemma needed for the proof concerns the property of a random walk on a line where the coin is biased with some probability.

Lemma D.3. *Define a random walk on a line where the coin gives head with probability $\frac{1+\widehat{\gamma}}{2}$. The random walk start at 0 and stops if it reaches T or $-T$. Conditioned on the fact that we reach T before $-T$, let the expected number of steps of the walk be μ_T , then*

$$\mu_T = \frac{T}{\widehat{\gamma}} - \frac{2T}{\widehat{\gamma}}(1 - \widehat{\gamma})^T \left(\frac{(1 + \widehat{\gamma})^T - (1 - \widehat{\gamma})^T}{(1 + \widehat{\gamma})^{2T} - (1 - \widehat{\gamma})^{2T}} \right). \quad ([\text{Fel91}, \text{Chapter XIV}])$$

Moreover, for $T = \Theta(1/(\sqrt{t} \cdot \widehat{\gamma}))$, we have

- $\mu_{2T} = \Omega(1/(t \cdot \hat{\gamma}^2))$, and
- $\mu_{2T} \leq 12\mu_T$.

For completeness, we present the proof after the proof of Theorem 2.4

Proof of Theorem 2.4. From Lemma D.2, we can assume that f can be computed by a noisy randomized algorithm \hat{A} of cost $O(\text{noisyR}(f))$ that makes queries with only two biases: 1 and $\hat{\gamma}$. We will now simulate \hat{A} , that uses bias 1 and $\hat{\gamma}$, with a noisyR algorithm B that uses bias 1 and $1/\sqrt{t}$, where $t \geq 1$.

Case 1: The first case is when $\hat{\gamma} \geq 1/\sqrt{t}$. In this case we can use multiple oracle calls of bias $1/\sqrt{t}$ to simulate one oracle call of bias $\hat{\gamma}$. To do this, algorithm B makes $O(t\hat{\gamma}^2)$ calls of bias $1/\sqrt{t}$ and takes their majority. By Lemma D.1, the algorithm B obtains a bit of bias slightly greater than $1/\sqrt{t}$. Then B adds some bias to this bit to obtain a bit of bias exactly $\hat{\gamma}$. The cost paid by B is $O(t\hat{\gamma}^2 \cdot 1/t) = O(\hat{\gamma}^2)$.

Case 2: We now consider the case when $\hat{\gamma} < 1/\sqrt{t}$. In this case we wish to generate many bits of low bias ($\hat{\gamma}$) using a single bit of high bias ($1/\sqrt{t}$). Consider the following random walk based sampling procedure.

- **Setup.** Let

$$T = \left\lfloor \frac{1}{5\sqrt{t}\hat{\gamma}} \right\rfloor. \quad (15)$$

The random walk will take place on a line that is marked with integral multiples of T , with T being right to 0, $-T$ being left to 0 and so on. Also, $(T - 1)$ points are marked between every two adjacent integral multiples of T . The random walk starts at 0. Also, let

$$R = \left(\frac{1 + \hat{\gamma}}{1 - \hat{\gamma}} \right)^T.$$

1. Toss a coin of bias

$$\delta' = \frac{R - 1}{R + 1}.$$

2. If the result of the toss is 1, then sample a sequence of bits using a $\hat{\gamma}$ -bias coin consisting of w 1's and z 0's, conditioned on $w - z = T$. This sampling procedure can be simulated by picking a random walk from the set of $\hat{\gamma}$ -biased random walks on the line which reach T before $-T$ (starting at 0).
3. If the result of the toss is 0, then sample a sequence of bits using a $\hat{\gamma}$ -bias coin consisting of w 1's and z 0's, conditioned on $w - z = -T$. This sampling procedure can be simulated by picking a random walk from the set of $\hat{\gamma}$ -biased random walks on the line which reach $-T$ before T (starting at 0).

We now prove the correctness of the above protocol. A crucial observation in [BDB20a] is that conditioned on a set of sequences, all of which reach T before $-T$, the probability of choosing each sequence is the same whether we choose bias $\hat{\gamma}$ or $-\hat{\gamma}$ for Step 2. and Step 3. above. The reason is that the probability for any sequence of walks with w 1's and z 0's such that $w - z = T$, is R times more likely when we choose a $\hat{\gamma}$ biased coin than when we choose a $-\hat{\gamma}$ biased coin. Furthermore, the probability of a $\hat{\gamma}$ -biased walk starting at 0 and reaching T is $R/(R + 1)$. This is because if

the probability of reaching T is p , then probability of reaching $-T$ is p/R . Since $p + p/R = 1$, $p = R/(R+1)$. Similarly, the probability that a $-\hat{\gamma}$ -biased coin reaches T is $1/(R+1)$ and reaches $-T$ is $R/(R+1)$. Thus δ' is chosen such that the probability of being correct is exactly $R/(1+R)$. Also, assuming that $\hat{\gamma}$ is smaller than $1/10$, the bias δ' can be shown to be smaller than but within a constant factor of $1/\sqrt{t}$ (we refer to [BDB20a, Theorem 4] for details). Thus, by adding noise, the bias $1/\sqrt{t}$ can be converted to bias δ' .

Let μ_T denote the expected number of bits that one run of the above sampling procedure generates. Lemma D.3 implies that μ_T is lower bounded by $\Omega(1/(t \cdot \hat{\gamma}^2))$.

In summary, for one iteration of the Step 1. to 3. of the sampling procedure, the following are true: a) The choice of δ' in Step 1. implies correct distribution (from bias $\hat{\gamma}$) of generating the coin tosses, and b) an expected number of $\Omega(1/(t\hat{\gamma}^2))$ $\hat{\gamma}$ -biased samples are generated. c) Cost paid is δ^2 .

Before moving further let us mention explicitly how algorithm B works: B runs algorithm \hat{A} on the given input. For each $i \in [n]$:

- If \hat{A} makes a query of bias 1 then B also queries that bit with bias 1. The value of such bits are known with certainty and no other queries to these bits are made.
- When \hat{A} makes a query of bias $\hat{\gamma}$ then
 - If B has $\hat{\gamma}$ biased bits available, then B uses these bits to run \hat{A} . Otherwise B generates the above sampling procedure to generate $\hat{\gamma}$ biased bits.

Clearly, the correctness of B follows from the correctness of \hat{A} . In order to upper bound the expected cost of B , we upper bound the expected number of times B runs the above sampling procedure for each index $i \in [n]$. Recall that our goal is to show that the total expected cost is bounded by:

$$O(\text{noisyR}(f) + n/t).$$

Fix a $k \in [n]$ and consider the noisy query algorithm B that uses bias $1/\sqrt{t}$ and 1 to simulate the algorithm \hat{A} (that uses bias $\hat{\gamma}$ and 1). For a run, say r , of algorithm \hat{A}

- Let $T_{r,k}$ be the number of times algorithm \hat{A} queries the k th bit in the r -th run and let

$$T_k = \mathbb{E}_r[T_{r,k}].$$

Observe that

$$\sum_k T_k = O(\text{noisyR}(f)/\hat{\gamma}^2). \tag{16}$$

- Let $X_1, X_2, \dots, X_{\ell_r}$ be the number of bits sampled by B , while simulating the r -th run of \hat{A} , where B makes ℓ_r independent calls to the above sampling procedure in order to meet the demand of $T_{r,k}$.

Let

$$L_{k,r} = (X_1 + \dots + X_{\ell_r}) - T_{r,k}, \tag{17}$$

denote the number of surplus bits (i.e. not used in simulation) generated by the algorithm B . Thus,

$$\begin{aligned} \mathbb{E}[X_1 + \dots + X_{\ell_r}] &= \mathbb{E}[T_{r,k}] + \mathbb{E}[L_{k,r}] \\ &= T_k + \mathbb{E}[L_{k,r}], \end{aligned} \tag{18}$$

where the expectation in the first equation is over both the run r and the randomness of the sampling procedure.

We now upper bound $\mathbb{E}[L_{k,r}]$. Note that $L_{k,r}$ denotes the number of bits generated by the following walk on line: a $\hat{\gamma}$ -biased random walk on line starts at some point x_r , where $-T < x_r < T$ depends on X_1, \dots, X_{ℓ_r} and $T_{r,k}$, and stops after reaching either $-T$ or T . The expected value of $L_{k,r}$ is upper bounded by the expected value of a $\hat{\gamma}$ -biased random walk on line starts at x_r and stops after reaching either $-(T + |x_r|)$ or $(T + |x_r|)$. Since $-T < x_r < T$, $\mathbb{E}[L_{k,r}]$ is upper bounded by the expected value of a $\hat{\gamma}$ -biased random walk on line starts at 0 and stops after reaching either $-2T$ or $2T$.

From Lemma D.3 we have the following bound on μ_{2T} in terms of μ_T ,

$$\mu_{2T} \leq 12\mu_T.$$

The above expression, combined with Equation 18, implies that for every $k \in [n]$

$$\mathbb{E}[X_1 + \dots + X_{\ell_r}] \leq T_k + 12\mu_T, \tag{19}$$

where the expectation is over both r (i.e. randomness used by \hat{A}) and the randomness of the sampling procedure.

Next, we upper bound the expected number of times B calls the sampling procedure, i.e. we upper bound $\mathbb{E}[\ell_r]$. Let I_t denote the random variable such that $I_t = 0$ if $t > \ell_r$ and $I_t = 1$ otherwise. Thus,

$$\begin{aligned} \mathbb{E}\left[\sum_{i=1}^{\ell_r} X_i\right] &= \mathbb{E}\left[\sum_{i=1}^{\infty} X_i I_i\right] \\ &= \sum_{i=1}^{\infty} \mathbb{E}[X_i I_i] \\ &= \sum_{i=1}^{\infty} \Pr[I_i = 1] \mathbb{E}[X_i | I_i = 1] \\ &= \sum_{i=1}^{\infty} \Pr[\ell_r \geq i] \mathbb{E}[X_i | \ell_r \geq i] \\ &= \mathbb{E}[X_i] \sum_{i=1}^{\infty} \Pr[\ell_r \geq i] \\ &= \mathbb{E}[X_i] \mathbb{E}[\ell_r]. \end{aligned}$$

In the second last equation, we have used $\mathbb{E}[X_i | \ell_r \geq i] = \mathbb{E}[X_i]$. This is because the event $\ell_r \geq i$ depends on X_1, \dots, X_{i-1} but not on X_i . Since $\mathbb{E}[X_i] = \mu_T$ for all i , along with Equation 19 we have

$$\mathbb{E}[\ell_r] \leq T_k / \mu_T + 12.$$

Summing over all k and using Equation 16, the expected number of queries made by algorithm B is upper bounded by

$$O\left(\frac{\text{noisyR}(f)}{\hat{\gamma}^2 \mu_T} + 12n\right).$$

Thus the expected cost of B is upper bounded by

$$O\left(\frac{\text{noisyR}(f)}{t\hat{\gamma}^2\mu_T} + \frac{12n}{t}\right).$$

Since $\mu_T = \Omega(1/(t\hat{\gamma}^2))$, the above quantity is upper bounded by

$$O\left(\text{noisyR}(f) + \frac{n}{t}\right).$$

In order to complete the proof of the theorem, we show how to simulate B to obtain a randomized query algorithm B' for $f \circ \text{GapMaj}_t$ of cost $O(t \cdot \text{noisyR}(f) + n)$. The algorithm B' simulates B in the following manner:

1. if b queries i -th bit of f with bias 1, for some $i \in [n]$, then B' queries all the bits of i -th Gap-Majority inner function.
2. if B queries i -th bit of f with bias $1/\sqrt{t}$, for some $i \in [n]$, then B' queries a random bit of i -th Gap-Majority inner function.

The correctness of B' follows directly from the simulation and the correctness of B . Also, in both the cases (1. and 2.), cost paid by B' is t times the cost paid by B . Thus

$$\text{R}(f \circ \text{GapMaj}_t) = O(t \cdot \text{noisyR}(f) + n). \quad (20)$$

□

Proof of Lemma D.3. As mentioned in the statement of the lemma, the equality

$$\mu_T = \frac{T}{\hat{\gamma}} - \frac{2T}{\hat{\gamma}}(1 - \hat{\gamma})^T \left(\frac{(1 + \hat{\gamma})^T - (1 - \hat{\gamma})^T}{(1 + \hat{\gamma})^{2T} - (1 - \hat{\gamma})^{2T}} \right), \quad (21)$$

follows from [Fel91, Chapter XIV].

Next, we prove the second part of the lemma. The following inequalities are easy to observe.

$$\begin{aligned} (1 + \hat{\gamma})^T - (1 - \hat{\gamma})^T &\leq \frac{2\hat{\gamma}T}{1 - \hat{\gamma}^2T^2}, \\ (1 + \hat{\gamma})^T - (1 - \hat{\gamma})^T &\geq 2\hat{\gamma}T, \\ (1 - \hat{\gamma})^T &\geq 1 - T\hat{\gamma}. \end{aligned}$$

Using the above inequalities and Equation 21 we get the following bounds on μ_T

$$\frac{T^2}{(1 + \hat{\gamma}T)} \leq \mu_T \leq \frac{T}{\hat{\gamma}} (\hat{\gamma}T + 4\hat{\gamma}^2T^2 - 4\hat{\gamma}^3T^3). \quad (22)$$

For $T = \Theta(1/\sqrt{t} \cdot \hat{\gamma})$, the above expression implies that μ_T is lower bounded by $\Omega(1/(t \cdot \hat{\gamma}^2))$.

Now we use Equation 22 to upper bound μ_{2T} in terms of μ_T :

$$\begin{aligned} \mu_{2T} &\leq \frac{2T}{\hat{\gamma}} (2\hat{\gamma}T + 16\hat{\gamma}^2T^2 - 32\hat{\gamma}^3T^3) \\ &= \frac{T^2}{(1 + \hat{\gamma}T)} \cdot \frac{(1 + \hat{\gamma}T)}{T^2} \cdot \frac{2T}{\hat{\gamma}} (2\hat{\gamma}T + 16\hat{\gamma}^2T^2 - 32\hat{\gamma}^3T^3) \\ &\leq \frac{(1 + \hat{\gamma}T)}{T^2} \cdot \frac{2T}{\hat{\gamma}} (2\hat{\gamma}T + 16\hat{\gamma}^2T^2 - 32\hat{\gamma}^3T^3) \cdot \mu_T \\ &= 2(1 + \hat{\gamma}T) \cdot (2 + 16\hat{\gamma}T - 32\hat{\gamma}^2T^2) \cdot \mu_T. \end{aligned}$$

Since $T = \Theta(1/(\sqrt{t} \cdot \hat{\gamma}))$, for a suitable choice of constant, we have,

$$\mu_{2T} \leq 12\mu_T.$$

□