Black-Box PPP is Not Turing-Closed

Noah Fleming
Memorial University

Stefan Grosser
McGill

Toniann Pitassi
Columbia University

Robert Robere
McGill

June 28, 2024

Abstract

The complexity class PPP contains all total search problems many-one reducible to the PIGEON problem, where we are given a succinct encoding of a function mapping $n+1$ pigeons to n holes, and must output two pigeons that collide in a hole. PPP is one of the "original five" syntactically-defined subclasses of TFNP, and has been extensively studied due to the strong connections between its defining problem — the pigeonhole principle — and problems in cryptography, extremal combinatorics, proof complexity, and other fields. However, despite its importance, PPP appears to be less robust than the other important TFNP subclasses. In particular, unlike all other major TFNP subclasses, it was conjectured by Buss and Johnson that PPP is not closed under *Turing reductions*, and they called for a black-box separation in order to provide evidence for this conjecture. The question of whether PPP contains its Turing closure was further highlighted by Daskalakis in his recent IMU Abacus Medal Lecture.

In this work we prove that PPP is indeed not Turing-closed in the black-box setting, affirmatively resolving the above conjecture and providing strong evidence that PPP is not Turing-closed. In fact, we are able to separate PPP from its *non-adaptive* Turing closure, in which all calls to the PIGEON oracle must be made in parallel. This differentiates PPP from all other important TFNP subclasses, and especially from its closely-related subclass PWPP — defined by reducibility to the *weak* pigeonhole principle — which is known to be non-adaptively Turing-closed. Our proof requires developing new tools for PPP lower bounds, and creates new connections between PPP and the theory of *pseudoexpectation operators* used for Sherali-Adams and Sum-of-Squares lower bounds. In particular, we introduce a new type of pseudoexpectation operator that is precisely tailored for lower bounds against black-box PPP, which may be of independent interest.

Contents

1 Introduction

The class TFNP consists of all *total* NP *search problems*: that is, search problems where potential solutions can be efficiently verified, and also where solutions are guaranteed to exist. TFNP contains many important search problems that we would like to efficiently solve in practice, but which seem to not admit polynomial-time algorithms. Two standard examples are the NASH problem (given a bimatrix game, output a Nash equilibrium of that game) and the FACTORING problem (given a number n, output a prime factor of n). Note that TFNP itself is a *semantic* class, and therefore believed not to admit complete problems [Pud15]. Therefore, in order to study problems inside of TFNP, researchers have defined *syntactic* subclasses by using many-one reductions to certain complete problems of interest. For instance, the NASH problem is complete for the class PPAD [DGP09, CDT09], which is typically defined using the complete problem END-OF-LINE [Pap94].

In this work we study the TFNP subclass PPP, whose defining complete problem is PIGEON.

Definition 1.1. The PIGEON problem is defined as follows. The input is a polynomial-size boolean circuit C encoding a boolean function from $\{0,1\}^n \to \{0,1\}^n$. The output is either any $x \in \{0,1\}^n$ such that $C(x) = 0^n$, or any two strings $x \neq y \in \{0,1\}^n$ such that $C(x) = C(y)$. The class PPP contains all total search problems that are polynomial-time many-one reducible to PIGEON.

PPP is one of the "original five" TFNP subclasses introduced in the 1990s to capture the complexity of interesting total search problems [MP91, JPY88, Pap94, BIK$^+$94] — the other four being PLS, PPA, PPAD, and PPADS. PPP is also one of the most important of the TFNP subclasses. The pigeonhole principle captures strong induction, which is the basic axiom underlying most formal systems for mathematical reasoning. Additionally, PPP has strong connections to other areas such as the theory of lattices and cryptography [Jer16, BJP$^+$19, SZZ18, HV21], extremal combinatorics [BFH$^+$23, PPY23], and propositional proof complexity [BCE$^+$98, BM04, BJ12].

Despite the prominent role of PPP, it seems to lack certain robustness properties that all other natural TFNP classes enjoy. A prominent example of such a property is *closure under Turing reductions*. (A TFNP class $\mathcal{C}$ is closed under Turing reductions if any problem polynomial-time reducible to $\mathcal{C}$ via multiple calls to a problem in $\mathcal{C}$ is also polynomial-time reducible to a single call to $\mathcal{C}$.) The classical TFNP classes are typically defined using closure under *many-one* reductions, although, the original family of black-box separations between these classes, proved by [BCE$^+$98], already hold for the Turing closed variants. The later work of Buss and Johnson [BJ12] asked whether these classical TFNP classes are closed under Turing reductions.They proved that four of the five original TFNP classes PPA, PPAD, PPADS, and PLS are closed under Turing reductions, and they constructed an artificial TFNP subclass that was not Turing closed in the black-box setting. Subsequently, with the exception of PPP, all other natural TFNP classes (e.g., CLS) have been shown to be Turing closed. Thus PPP stands as the only natural TFNP class not known to be Turing closed.

The Turing closure of PPP is not only interesting from a structural point of view, but is also connected to other questions. First, our understanding of the relative complexity of other important TFNP problems connected to PPP has been murky, largely due to the absence of a technique for showing that PPP is closed under multiple calls. For example, in a recent work, Sotiraki, Zampetakis, and Zirdelis [SZZ18] showed that several problems on lattices are *Turing* reducible to PIGEON (but, notably, *not* many-one reducible), including the fundamental n-SVP problem underlying lattice-based cryptography. Another well-studied class of problems related to PPP comes from extremal combinatorics. Problems such as Ramsey's theorem and the Sunflower Theorem are also proven via *iterated* applications of the pigeonhole principle; but again it is unknown if these problems are reducible to a single instance of PPP. (See Section 1.2 for references and more on these connections.)

Secondly, the distinction between many-one and Turing reductions is sometimes important, as it relates to the *circuit depth* of the reduction. For example, an important result in cryptography shows the existence of PRGs in NC^0 (assuming there are one-way functions in NC^1) [AIK04]. However the stretch of their PRGs is sublinear, and known methods to increase the stretch require iterated applications of a version of the pigeonhole principle. It is an important open question whether or not PRGs with polynomial stretch exist in NC^0; for example [JLS21] show that such constructions imply indistinguishability obsfucation.

Thirdly, the Turing closure of PPP has important implications in proof complexity. A body of recent work in the intersection of complexity theory and proof complexity has linked natural TFNP classes to natural propositional proof systems (e.g., [BIK$^+$94, GHJ$^+$22, DR23, BFI23].) This in turn has vastly increased our toolkit for proving black-box separations for TFNP classes, and also raised many new questions. Once again the glaring outlier is PPP, which is the only natural TFNP class that lacks an equivalant characterization by a natural proof system. Indeed, we suspect that this again can be attributed to the lack of robustness of the TFNP class PPP, and thus understanding the Turing closure question for PPP (equivalently, whether or not PPP equals $\mathsf{FP}^{\mathsf{PPP}}$) is a step towards understanding this phenomena.

Recently, the question of the Turing-closure of PPP was highlighted in Daskalakis' IMU Abacus Medal Lecture [Das19, Open Question 7], with a tighter characterization of the complexity of n-SVP singled out as one notable application. Buss and Johnson [BJ12] openly conjectured that $\mathsf{PPP} \neq \mathsf{FP}^{\mathsf{PPP}}$, and they ask whether it is possible to provide evidence for this conjecture via a black-box separation.

Open Problem 1. [BJ12, Das19] Can we provide black-box separations between PPP and $\mathsf{FP}^{\mathsf{PPP}}$?

1.1 Main Results and Technical Highlights

The main contribution of this work is to provide the first complexity-theoretic evidence for the separations between PPP and $\mathsf{FP}^{\mathsf{PPP}}$. We work in the *black-box setting*, where the input to PIGEON is presented as a black-box oracle which we can query, instead of as a boolean circuit (see Section 2 for formal definitions). Our main result is a black-box separation between PPP and $\mathsf{FP}^{\mathsf{PPP}}$, resolving the open problem above.

Theorem 1.2. PPP *is not Turing-closed in the black-box setting.*

We note that in the land of TFNP, *all* currently known inclusion results hold in the black-box setting. This means that black-box separations are significant since they rule out all existing techniques for proving inclusions.

We prove Theorem 1.2 by giving a black-box separation between the PIGEON problem, and following PIGEON $\otimes$ PIGEON problem:

Definition 1.3. The PIGEON $\otimes$ PIGEON problem, also denoted $\text{PIGEON}^{\otimes 2}$, is defined as follows. The input is two boolean circuits C_1, C_2, both encoding functions from $\{0,1\}^n \to \{0,1\}^n$. The output is a solution of PIGEON on both C_1 and on C_2.

Theorem 1.4. $\text{PIGEON}^{\otimes 2}$ *is not black-box reducible to* PIGEON.

In other words we show that one cannot efficiently reduce, in a black-box way, solving *two independent instances* of PIGEON to *one instance* of PIGEON. Since $\text{PIGEON}^{\otimes 2}$ is easily observed to be contained in $\mathsf{FP}^{\mathsf{PPP}}$ — just call the PIGEON oracle twice — Theorem 1.2 follows.

We first remark that our result shows something stronger. Namely, it shows that PPP is not even *non-adaptively Turing closed* in the black-box setting, where the non-adaptive Turing closure is where one is allowed to ask multiple queries to the PIGEON oracle in *parallel*, as opposed to *sequentially*. Recall that in the closely related WEAK-PIGEON problem — the defining problem for the class PWPP — we seek to find a collision in a map from 2^{n+1} pigeons to 2^n holes, rather than 2^n pigeons to $2^n - 1$ holes. Contra to our above result, Jeřábek showed that PWPP *is* non-adaptively Turing closed and so, in particular, Weak-Pigeon$^{\otimes 2}$ many-one reduces to Weak-Pigeon [Jer16]. Thus our result further distinguishes Pigeon and PPP from Weak-Pigeon and PWPP.

Our techniques can be used to prove more than just the above separation. Let Pigeon$_N^M$ for $M > N$ denote the problem of finding a collision in a map from M pigeons to N holes. For example, if $N = 2^n$, then we can write Pigeon $=$ Pigeon$_{N-1}^N$ and Weak-Pigeon $=$ Pigeon$_N^{2N}$ in this notation. As an immediate corollary, we can prove the following strengthening of the above result:

Theorem 1.5. *Let $0 < \varepsilon < 1$ be a universal constant. Then* Pigeon$_N^{N+N^{1-\varepsilon}}$ $\otimes$ Pigeon$_N^{N+N^{1-\varepsilon}}$ *is not black-box reducible to* Pigeon$_{N-1}^N$.

Proof. We prove that Pigeon$_N^{N+1}$ is black-box reducible[1] to Pigeon$_N^{N+N^{1-\varepsilon}}$. This immediately implies that Pigeon$_N^{N+1}$ $\otimes$ Pigeon$_N^{N+1}$ reduces to Pigeon$_N^{N+N^{1-\varepsilon}}$ $\otimes$ Pigeon$_N^{N+N^{1-\varepsilon}}$, and so if the latter problem reduces to Pigeon$_N^{N+1}$ then we contradict our main theorem.

Our reduction is straightforward. Given an instance f of Pigeon$_N^{N+1}$, create $c = N^{1/\varepsilon - 1} = N^{(1/\varepsilon)(1-\varepsilon)}$ parallel copies of the instance $f^{(1)}, f^{(2)}, \cdots, f^{(c)}$ that share the hole 0 but otherwise are independent. Reparametrize by setting $M = N^{1/\varepsilon}$, and the result of this reduction is an instance of

$$\text{Pigeon}_{cN}^{c(N+1)} = \text{Pigeon}_{cN}^{cN+c} = \text{Pigeon}_{N^{1/\varepsilon}}^{N^{1/\varepsilon}+N^{(1/\varepsilon)(1-\varepsilon)}} = \text{Pigeon}_M^{M+M^{1-\varepsilon}},$$

and any solution of this new instance can be used to recover a solution to the original Pigeon$_N^{N+1}$ instance. Finally, since $\varepsilon < 1$ is a universal constant, we note that $M = N^{1/\varepsilon}$ incurs just a polynomial-size blowup in the instance. □

Since Weak-Pigeon$^{\otimes 2}$ $\in$ PPP, this result implies that when we slowly increase the ratio of pigeons to holes, a "phase-transition" happens once we reach mapping $\Theta(N)$ pigeons to N holes, where suddenly the non-adaptive problem switches from being "hard" for PPP to being contained in PWPP. (We note that this aligns with the bounded-depth proof complexity of the pigeonhole principle, where a similar phase transition occurs.)

We can also use our lower bound for Pigeon$^{\otimes 2}$ to prove a hierarchy result, showing that having k non-adaptive queries to Pigeon is more powerful than $k-1$ non-adaptive queries. Formally, let Pigeon$^{\otimes k}$ denote the natural generalization of Pigeon$^{\otimes 2}$ to k copies of Pigeon, where the goal is to output k solutions. Clearly Pigeon$^{\otimes k-1}$ reduces to Pigeon$^{\otimes k}$, since we can just embed a fixed solution on the extra copy of Pigeon$^{\otimes k}$. Complementing this, we have the following:

Theorem 1.6. *For all constant k,* Pigeon$^{\otimes k}$ *is not black-box reducible to* Pigeon$^{\otimes k-1}$.

The above hierarchy theorem follows from using our main technical theorem (Theorem 4.15) as a base case in an inductive argument. It is not clear how to deduce the previous hierarchy theorem immediately from Theorem 1.4 via an inductive argument — it appears that our stronger lower-bound technique is needed in order to make the induction work.

Technical Overview. The tools we use to prove the above separations are developed from lower-bound tools in *propositional proof complexity*. Propositional proof complexity has a very close relationship to TFNP, as each of the defining problems for TFNP subclasses correspond to *existence theorems*: that is, tautologies of the form $\forall x \exists y \phi(x, y)$, where ϕ is a polynomial-time computable predicate. Indeed, the *complexity* and *relative relationships* between TFNP subclasses are intimately related to the *relative provability* of the existence principles defining the classes [BIK$^+$94, Mor01, BM04, BJ12].

[1] Our original proof of this theorem used the techniques from Section 4 in a direct argument. We thank Jiawei Li for pointing out this simplified proof.

A recent line of work has tightened these connections further, by showing actual *equivalences* between *black-box* TFNP *subclasses* and *propositional proof systems* [BJ12, GKRS18, GHJ+22, DR23, BFI23]. Roughly speaking, these results show that for many natural TFNP classes C, there is an associated natural proof system, P_{C} such that the following statement holds:

A total search problem R is contained in the (black-box) TFNP subclass C
if and only if
A propositional encoding of the totality of R can be *efficiently proved* in P_{C}.

This generic connection often allows us to reduce the problem of proving black-box separations to the already-tackled problem of proving a separation between the corresponding proof systems[2]. Indeed, this strategy has been used effectively to provide oracle separations between nearly all of the classical TFNP classes [BIK+94, GHJ+22].

Unfortunately, we cannot appeal directly to a proof complexity separation to black-box separate PIGEON and PIGEON$^{\otimes 2}$, since as mentioned in the Introduction, there is no natural proof system corresponding to PPP. Nonetheless, we are able to bypass this issue, and our main technical contribution is the development of new tools to prove lower bounds against black-box PPP directly, without needing to appeal to the corresponding proof system.

Our new tools are still directly inspired from proof complexity methods, and so we believe that they may be of independent interest. The starting place for our lower bound is the observation that the proof theoretic strength of the pigeonhole principle sits between the well-studied semi-algebraic proof systems Sherali-Adams (SA) and Sums-of-Squares (SOS). The theory of lower bounds in SA and SOS are well-developed: SA proof complexity is characterized by the so-called family of *pseudo-expectation operators* and SOS complexity (which is strictly stronger than SA) is characterized by the stronger family of *positive semi-definite pseudo-expectations* [FKP19]. It is well-known that the pigeonhole principle (underlying PIGEON) requires large SA proofs, and [GM08] give a lower bound by explicitly constructing a pseudo-expectation operator for PIGEON. On the other hand, SOS has short proofs of *both* (the totality of) PIGEON and PIGEON$^{\otimes 2}$ (cf. Appendix A). Therefore, both PPP and PPP $\otimes$ PPP are bracketed above and below by two proof systems that are characterized by pseudo-expecatation operators: SA is "too weak" since it cannot efficiently certify either PPP or PPP $\otimes$ PPP, and SOS is "too strong" since it can efficiently certify both PPP and PPP $\otimes$ PPP.

This suggests the following question:

Is there a variant of pseudoexpectation operators, intermediate in strength between standard pseudoexpectations and PSD pseudoexpectations, which imply lower bounds for black-box PPP*?*

We show that the answer to this question is "*Yes*"!

In Section 3, we introduce and develop a new variant of pseudoexpectation operators that are tailored to proving lower bounds against black-box PPP — we call these *collision-free pseudoexpectation operators*. Importantly, such operators are *not* automatically PSD pseudoexpectations, and thus do not imply lower bounds for SOS proofs. We then prove Theorem 1.4 by constructing a collision-free pseudoexpectation operator for the PIGEON$^{\otimes 2}$ problem (cf. Section 4). While the particular pseudoexpectation operator that we choose for PIGEON$^{\otimes 2}$ is natural — it is an obvious generalization of the pseudoexpectation for PIGEON — proving that it is collision-free requires a delicate and technical proof, combining methods developed in the theory of pseudoexpectations and also in the theory of lower bounds for bounded-depth Frege (particularly, the use of *matching decision trees*). Indeed, this makes some sense as the particular black-box

[2] A recent paper of Buss, Fleming and Impagliazzo [BFI23] shows that this equivalence is general: for any syntactic TFNP class, there is always a corresponding proof system satisfying the above property. However, the proof system may in general not be one that is natural or that has already been defined.

separation we are trying to prove is very thin, being between two problems that are very close in complexity. To see this, observe that the PIGEON problem by itself cannot admit a collision-free pseudoexpectation (of course). However, neither can the problem WEAK-PIGEON $\otimes$ WEAK-PIGEON, due to the result of Jeřábek showing WEAK-PIGEON $\otimes$ WEAK-PIGEON $\in$ PWPP $\subseteq$ PPP [Jer16]. In our view, this makes the separation PIGEON$^{\otimes 2} \not\in$ PPP, and its generalization to $N + N^{1-\varepsilon}$ pigeons (Theorem 1.5), quite interesting and surprising.

We believe that these new notions of pseudoexpectations may help us develop an enriched set of tools for constructing dual certificates that generalize the standard pseudoexpectations needed for refutational SA (SOS) lower bounds. For example, a key tool for proving lower bounds on the extension complexity of approximation algorithms is via lifting lower bounds for a implicational versions of SA and SOS (e.g., [CLRS13, LRS15, KMR17]). Lower bounds for implicational SA are harder than proving lower bounds for SA refutations, and thus the pseudoexpectation properties required here are more demanding. Another example are black-box lower bounds for pseudodeterministic algorithms for NP search problems. These can be proven from degree lower bounds for a certain generalization of SA proofs, whose dual certificate again requires a strengthening of the definition of SA pseudoexpectations [GIPS21].

We refer to Section 3 for a formal discussion of the new type of pseudoexpectation operator, and to Section 4 for a technical overview of our lower bound for PIGEON$^{\otimes 2}$.

1.2 Related Work

Cryptography and PPP. In the original paper that introduced the class PPP, Papadimitriou already observed that an efficient algorithm for PIGEON would allow the inversion of any one-way permutation [Pap94, Proposition 3]. This original observation predicted an entire host of results connecting the complexity of PPP to the hardness of various cryptographic primitives in cryptography. A later result by Jeřábek [Jer16] showed that FACTORING is reducible in *randomized* polynomial-time to PIGEON, and even to its weaker variant WEAK-PIGEON (defined above).

Later works have connected the complexity of PPP to other fundamental cryptographic primitives. Work by Hubáček and Václavek [HV21] showed that various versions of the fundamental *discrete logarithm problem* were PWPP- and PPP-complete. Another central recent work by Sotiraki, Zampetakis, and Zirdelis [SZZ18] showed fundamental connections between PPP and PWPP, *collision-resistant hash functions*, and the hardness of various problems on *lattices*. (Indeed, the connection to collision-resistant hash functions is embodied by the very definition of the WEAK-PIGEON problem: we are given a contracting map from $n+1$ bits to n bits computed by a polynomial-size circuit, and the goal is to find two input strings mapping to the same output string.) The hardness of various lattice problems have been foundational to many cryptographic constructions since the seminal work of Ajtai [Ajt96]. Sotiraki, Zampetakis, and Zirdelis show that BLICHFELDT, a computational problem on lattices closely related to Blichfeldt's theorem on lattices [Bli14], is PPP-Complete. Using this problem, they were able to show that several notable problems on lattices are *Turing* reducible to PIGEON (but, notably, *not* many-one reducible). This includes the problem n-SVP of approximating the shortest vector within a lattice to a factor of n, the hardness of which is foundational to lattice-based cryptography [Reg05]. Indeed, locating the n-SVP question into a natural complexity class is an important open question, and the Turing closure FP$^{\text{PPP}}$ is the best current upper bound on its complexity.

Extremal Combinatorics and PPP. In the class FP$^{\text{PPP}}$, we get to use a PIGEON oracle to solve our problems of interest, and of course later queries to the PIGEON oracle can depend on the responses to earlier queries. There are, however, other ways to employ the pigeonhole principle iteratively, and many such examples come from *extremal combinatorics*. The prototypical example is in the standard proof of *Ramsey's Theorem*, which states that any graph on 2^{2n} vertices either has a clique of size n or an independent set of size n. This proof involves an iterated application of the pigeonhole principle in the following sense: we first pick an arbitrary node u, and then choose the next node from whichever set is larger: either the nodes *adjacent* to u, or the nodes *not adjacent* to u. Repeating this process for $2n$ steps yields a sequence of $2n$ nodes, and now we can see that among these $2n$ nodes there is either a clique of size n or an independent set of size n. Indeed, one can easily turn Ramsey's Theorem into a computational problem, denoted RAMSEY, where we are given an encoding of a graph on 2^{2n} nodes by a polynomial-size circuit C and must output an independent set or clique on the graph. We note that it is currently an open question whether or not RAMSEY reduces to PIGEON, although, it is known that WEAK-PIGEON reduces to RAMSEY, both under randomized reductions [KNY19], and also deterministically to a multicoloured version [KNY19, PPY23]. However, for the converse direction, it is not at all clear how to simulate the above proof of Ramsey's theorem inside PPP — in fact, it is not even clear whether or not RAMSEY $\in \mathsf{FP}^{\mathsf{PPP}}$!

To address this problem, Pasarkar, Papadimitriou, and Yannakakis [PPY23] introduced a new TFNP subclass that they called PLC (other work relating PPP to extremal combinatorics appeared concurrently by Bourneuf et al [BFH$^+$23]). The defining problem of PLC is LONG-CHOICE, which embodies the kind of iterated application of the pigeonhole principle seen in the proof of Ramsey's Theorem. They proved that PLC $\supseteq$ PPP, and also that it contains other interesting search problems related to extremal combinatorics *not* known to lie in PPP. Two such examples of problems that lie in PLC include RAMSEY, which we have already discussed, and SUNFLOWER, which is a computational analogue of the famous *Sunflower Lemma* [ER60]. One of the main open problems suggested by Pasarkar, Papadimitriou, and Yannakakis is to give any complexity theoretic evidence — such as a *black-box separation* — showing that PLC and PPP are distinct classes [PPY23], and thus that "sequential" applications of the pigeonhole principle, in this other sense, are stronger than a "single" application of the pigeonhole principle. In a concurrent work (discussed more below), Jain, Li, Robere, and Xun used similar techniques as in the present paper to separate RAMSEY and LONG-CHOICE from PPP [JLRX23].

Propositional proof complexity, bounded arithmetic, and PPP. The theory of *propositional proof complexity* also has many close ties with TFNP, and particularly with the PIGEON problem. As we have already mentioned above, each of the defining search problems for the TFNP subclasses naturally correspond to *existence theorems*: theorems that can be written in the form $\forall x \exists y \phi(x, y)$. Of course, the existence theorem corresponding to the PIGEON problem is — what else — *the pigeonhole principle*. The pigeonhole principle is perhaps the most well-studied tautology in all of propositional proof complexity, as it provides a difficult example for many propositional proof systems under common study [Hak85, Ajt88, BIK$^+$92, Raz98, Raz01]. The pigeonhole principle lower bounds are also closely related to independence results in theories of bounded arithmetic [Ajt88].

There have been several works which formalize the three-way connections between proof complexity, TFNP, and first-order logic, some of which are important to mention. Buss and Krajíček showed that the total functions computable in the TFNP class PLS are exactly the witnessing functions for the bounded-arithmetic theory T^1_2 [BK94]. Buresh-Oppenheim and Morioka gave a framework for defining syntactic TFNP subclasses using existentially quantified first-order formulas [BM04]. For an example, consider the following existential first-order formula:

$$\exists x, y : f(x) = 0 \vee (x \neq y \wedge f(x) = f(y)).$$

This formula states that either $f(x) = 0$, or there is a $y \neq x$ such that $f(x) = f(y)$ — that is, it encodes the pigeonhole principle. Buresh-Oppenheim and Morioka showed how to turn any such existential sentence ϕ into a defining problem Q_ϕ for a corresponding syntactic TFNP subclass. Moreover, they showed that given two such formulas ϕ, ψ, if the corresponding problems Q_ϕ many-one reduces to Q_ψ, then certain *propositional encodings* of the totality of ϕ can be deduced from the totality of ψ. Their results were improved by the work of Buss and Johnson [BJ12], who strengthened this correspondence in two ways: first, they improved *many-one* reducibility to *Turing* reducibility, and second, they strengthened the implication to an *equivalence*. In other words, Buss and Johnson showed that Q_ϕ is *Turing reducible* to Q_ψ if and only if the propositional encoding of ϕ is provable from ψ in a certain proof system. By the results of Buss and Johnson, one can therefore use our work to show that it is impossible to prove the totality of PIGEON $\otimes$ PIGEON from the totality of PIGEON in a particular proof system. We refer an interested reader to [BJ12] for the details.

Comparison with the work of Jain, Li, Robere, and Xun [JLRX23]. A concurrently appearing work of Jain, Li, Robere, and Xun generalizes the collision-free pseudoexpectation operators that are introduced here. They use this generalization to obtain lower bounds for a hierarchy of classes above PPP called the "Pecking Order". The emblematic class for level t of the Pecking Order is t-PIGEON, where we are given a mapping from $(t-1)n+1$ pigeons to n holes, and the goal is to find a hole containing t pigeons (note PIGEON $=$ 2-PIGEON in this notation). The lower bounds shown by Jain, Li, Robere, and Xun are orthogonal to the lower bounds in the present paper. For instance, they use their generalized collision-free pseudoexpectation operators to show that, in the black-box setting, RAMSEY $\notin$ PPP, and also that UPLC $\not\subseteq$ PPP, where UPLC $\subseteq$ PLC is the "unary" variant of the PLC class introduced above [PPY23]. We note that our lower bound for PIGEON$^{\otimes 2}$ in Theorem 1.4 gives an alternate proof of black-box separation PLC $\not\subseteq$ PPP, since PIGEON$^{\otimes 2}$ is easily verified to be in PLC.

1.3 Paper Organization

The rest of the paper is organized as follows. In Section 2, we introduce some necessary technical preliminaries for black-box TFNP, as well as show that PIGEON$^{\otimes 2}$ is contained in both PLC and FP$^{\mathsf{PPP}}$. In Section 3, we develop our new lower bound tool of collision-free pseudoexpectations in a general setting. Then, in Section 4, we prove Theorem 1.2 and Theorem 1.5. Finally, in Section 5 we prove generalizations of our main theorem (Theorems 1.5 and 1.6).

2 Preliminaries on Black-Box TFNP

A relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ is *total* if for all $x \in \{0,1\}^*$ there is a $y \in \{0,1\}^*$ such that $R(x,y)$ holds. Each total relation corresponds to a *total search problem* in the natural way: given x as input, output any y such that $R(x,y)$ holds. In the classical theory of TFNP, we are interested in the total search problems defined from *polynomial-time computable*, *polynomially-bounded* total relations R. The total search problems of particular interest in this paper are related to the *pigeonhole principle*, whose definition we again recall.

Definition 2.1. The PIGEON problem is defined as follows. The input is a boolean circuit C encoding a function from $\{0,1\}^n \to \{0,1\}^n$. The output is either any $x \in \{0,1\}^n$ such that $C(x) = 0^n$, or, any two strings $x \neq y \in \{0,1\}^n$ such that $C(x) = C(y)$.

The class of search problems polynomial-time mapping reducible to PIGEON is called PPP, and is one of the central classes of study in the theory of TFNP. In this paper, we will be interested in the following generalization of the PIGEON problem.

Definition 2.2. The PIGEON $\otimes$ PIGEON problem, also denoted PIGEON$^{\otimes 2}$, is defined as follows. The input is two boolean circuits C_1, C_2, both encoding functions from $\{0,1\}^n \to \{0,1\}^n$. The output is a solution of PIGEON on both C_1 and on C_2.

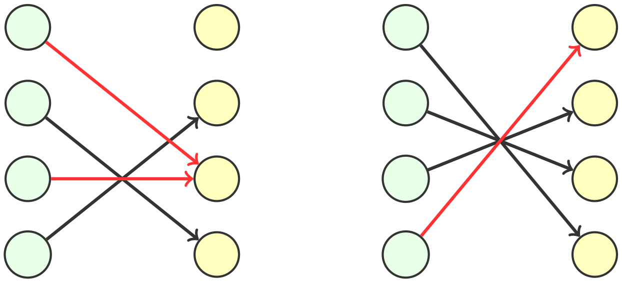

Figure 1: An instance of PIGEON$^{\otimes 2}$. A solution $(1, 3, 4)$ is indicated by the red edges; in the left PIGEON-instance, pigeons 1 and 3 collide in the same hole, while in the right PIGEON-instance, pigeon 4 maps to the forbidden 0-hole.

It is clear that PIGEON$^{\otimes 2}$ lies in the class $\mathsf{FP}^{\mathsf{PPP}}$. This is because, given an instance (C_1, C_2) of PIGEON$^{\otimes 2}$, we can use the PPP oracle to solve each instance separately, and then output both solutions. The main goal of this paper is to give evidence that PIGEON$^{\otimes 2}$ does not lie in PPP. We do this by proving lower bounds for PIGEON$^{\otimes 2}$ in the *black-box model*. For this, we must introduce some preliminaries regarding the black-box model of TFNP.

As defined above, the input to the PIGEON problem is a function $f : \{0,1\}^n \to \{0,1\}^n$ encoded by a polynomial-size circuit. In the black-box setting, we instead presume that the input function f is given by an *oracle*, which we are allowed to query, but cannot investigate the actual computation of the function f.

Definition 2.3. A *total (query) search problem* is a sequence of relations $\mathrm{R} := \{R_n \subseteq \{0,1\}^n \times O_n\}_n$, where O_n are finite sets, such that $\forall x \in \{0,1\}^n \, \exists o \in O_n : R_n(x, o)$. A total search problem R is in TFNP^{dt} if for each $o \in O_n$ there is a poly$(\log(n))$-depth decision tree T_o such that for all $x \in \{0,1\}^n$, $T_o(x) = 1$ iff $(x, o) \in R_n$.

It will be convenient to think of the input as being encoded over a broader domain, such as $[n]$. For example, when considering the PIGEON problem, it is natural to think of the input as being encoded by a function $f : [n+1] \to [n+1]$. We can do this by encoding each of the inputs in binary in the natural way, and this changes the complexity of the reductions in this paper by no more than a $O(\log n)$ factor. Furthermore, it will also often be convenient to refer to a single relation R_n in the sequence R. In an abuse of notation, we refer to R_n as a "total search problem". We will also allow the inputs to R_n to have $n^{O(1)}$ input bits for notational convenience.

With these remarks in mind, we denote by PIGEON^{dt} and $(\text{PIGEON}^{\otimes 2})^{dt}$ the query variants of the PIGEON and PIGEON$^{\otimes 2}$ problems, respectively. The input to PIGEON$_n^{dt}$ is a function $f : [n+1] \to [n+1]$, and the goal is to output either (1) any input i such that $f(i) = 1$, or any $i \neq j$ such that $f(i) = f(j)$ (the input to $(\text{PIGEON}^{\otimes 2})^{dt}$ is defined similarly). Formally speaking, the input to PIGEON$_n^{dt}$ is encoded by a boolean string of length $O(n \log n)$, which encodes $n+1$ "pointers" of pigeons to holes, each of $O(\log n)$ bits each. We will elide this formal low-level encoding, and presume that the query algorithms will always query entire pointers $f(i)$.

We now introduce the definition of a (many-one) reduction in the black-box model.

Definition 2.4. Let $R \subseteq \{0,1\}^n \times O_R$ and $S \subseteq \{0,1\}^m \times O_S$ be total search problems. An *S-formulation of R* is a decision-tree reduction $(f_i, g_o)_{i \in [m], o \in O_S}$ from R to S. Formally, for each $i \in [m]$ and $o \in O_S$ there are functions $f_i \colon \{0,1\}^n \to \{0,1\}$ and $g_o \colon \{0,1\}^n \to O_R$ such that

$$(x, g_o(x)) \in R \impliedby (f(x), o) \in S$$

where $f(x) \in \{0,1\}^m$ is the string whose i^{th} bit is $f_i(x)$. The *depth* of the reduction is

$$d := \max\big(\{D(f_i) : i \in [m]\} \cup \{D(g_o) : o \in O_S\}\big),$$

where $D(h)$ denotes the decision-tree depth of h. The *size* of the reduction is m, the number of input bits to S. The *complexity* of the reduction is $\log m + d$. We write $S^{dt}(R)$ to denote the minimum complexity of an S-formulation of R.

We extend these notations to sequences in the natural way. If R is a single search problem and $\mathrm{S} = (S_m)$ is a sequence of search problems, then we denote by $\mathrm{S}^{dt}(R)$ the minimum of $S_m^{dt}(R)$ over all m. If $\mathrm{R} = (R_n)$ is also a sequence, then we denote by $\mathrm{S}^{dt}(\mathrm{R})$ the function $n \mapsto \mathrm{S}^{dt}(R_n)$.

We denote by PPP^{dt} the class of all total search problems which admit $\mathrm{poly}(\log(n))$-complexity $\textsc{Pigeon}^{dt}$-formulations. In general, we will prove that problems do not black-box reduce to $\textsc{Pigeon}$ by proving that low-complexity $\textsc{Pigeon}$-formulations do not exist. This connection was first noted (using the language of "Type-2 Complexity") by [BCE+98].

3 Collision-Free Pseudoexpectations

In this section we introduce our new lower-bound technique for PPP^{dt} using pseudoexpectation operators. We develop our technique in a general setting, as we believe it may be of independent interest.

We must introduce some notation and also recall some of the theory of multilinear polynomials. All polynomials in this paper will have real coefficients. A polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is *multilinear* if the individual degree of any variable in p is at most 1. The algebra of multilinear polynomials is that of the quotient ring $\mathbb{R}[x_1, \ldots, x_n]/\langle x_i^2 - x_i \rangle_{i=1}^n$. More concretely, we can define the multiplication of two multilinear polynomials as follows: the multilinear polynomial pq is obtained from the multilinear polynomials p and q by multiplying p and q using standard polynomial multiplication, and then lowering the degree of all variables appearing in pq to 1. For instance, $(xy + z) \cdot (xz) = xyz + xz$, as multilinear polynomials. We will always be operating with multilinear polynomials, and presume that all operations are done over this quotient ring.

Let $R_n \subseteq \{0,1\}^n \times O$ be a query total search problem in TFNP^{dt}, and let $x_1, x_2, \ldots, x_n$ denote the n input bits to R_n. A *conjunction* C is a conjunction of boolean literals over $x_1, \ldots, x_n$. It will be particularly convenient to think of a conjunction C as a multilinear polynomial over these variables. For instance, we can encode the conjunction $x \wedge \overline{y} \wedge z$ as the polynomial $x(1-y)z$, which takes the same values as the conjunction over $\{0,1\}$. The *degree* of a conjunction C, denoted $\deg(C)$, is the number of literals occurring in it. Two conjunctions C_1, C_2 are *consistent* if $C_1 C_2 \neq 0$, and an input $x \in \{0,1\}^n$ is *consistent* with C if $C(x) = 1$. A conjunction C *witnesses* the solution $o \in O$ to R_n if for all $x \in R_n$ consistent with C, $(x, o) \in R_n$. We write $R_n \restriction C \subseteq \{0,1\}^{n-\deg(C)} \times O$ to denote the new search problem obtained by restricting the appropriate input bits to R according to the literals in C.

Definition 3.1. Let $\mathcal{P}_{n,d}$ denote the collection of all degree-d real-coefficient multilinear polynomials over the variables $x_1, x_2, \ldots, x_n$. A *degree-d pseudoexpectation operator* is a function $\tilde{\mathbb{E}} : \mathcal{P}_{n,d} \to \mathbb{R}$ satisfying the following properties:

- *Linearity.* $\tilde{\mathbb{E}}[\alpha p + \beta q] = \alpha \tilde{\mathbb{E}}[p] + \beta \tilde{\mathbb{E}}[q]$ for all $\alpha, \beta \in \mathbb{R}$ and all $p, q \in \mathcal{P}_{n,d}$.
- *Normalized.* $\tilde{\mathbb{E}}[1] = 1$.
- *Nonnegativity.* For every degree $\leq d$ conjunction C, $\tilde{\mathbb{E}}[C] \geq 0$.

Let $R \subseteq \{0,1\}^n \times O$ be any total query search problem. We say that $\tilde{\mathbb{E}}$ is a *pseudoexpectation for R* if, in addition to the above three properties, it satisfies the following additional property:

- *R-Nonwitnessing.* $\tilde{\mathbb{E}}[C] = 0$, for any degree $\le d$ conjunction C witnessing a solution to R.

Pseudoexpectation operators were originally introduced to prove lower bounds on *Sherali-Adams degree*, and a stronger pseudoexpectation operator was introduced to prove lower bounds for *Sums-of-Squares degree*. The standard method to construct pseudoexpectation operators is through the use of *pseudodistributions*, the definition of which we recall next.

Definition 3.2. Let $x_1, \dots, x_n$ be a set of boolean variables, and let d be a positive integer. A *degree-d pseudodistribution* over these variables is a family of probability distributions

$$\mathcal{D} = \{\mathcal{D}_S : S \subseteq [n], |S| \le d\},$$

such that the following properties hold:

- For each set $S \subseteq [n]$, $|S| \le d$, $\mathcal{D}_S$ is supported on $\{0,1\}^S$, interpreted as boolean assignments to variables in $\{x_i : i \in S\}$.
- For each $S, T \subseteq [n]$, $|S|, |T| \le d$, we have $\mathcal{D}_S^{S \cap T} = \mathcal{D}_T^{S \cap T} = \mathcal{D}_{S \cap T}$, where $\mathcal{D}_A^B$ for $B \subseteq A$ is the marginal distribution of A to variables indexed by B.

Let $R \subseteq \{0,1\}^n \times O$ be any query total search problem. Then $\mathcal{D}$ is a *pseudodistribution for R* if for every degree $\le d$ conjunction C witnessing a solution to R, no distribution $\mathcal{D}_S$ in $\mathcal{D}$ is supported on a consistent assignment for C.

In other words, a pseudodistribution is an object that "looks like" a probability distribution to an external adversary that can only investigate marginals of up to d bits, but, the distribution is not supported on any assignment that witnesses a solution to R. The following lemma is standard.

Lemma 3.3 ([FKP19]). *Suppose $\mathcal{D}$ is a degree-d pseudodistribution over $x_1, \dots, x_n$ for $R \subseteq \{0,1\}^n \times O$. The operator $\tilde{\mathbb{E}} : \mathcal{P}_{n,d} \to \mathbb{R}$ defined by*

$$\tilde{\mathbb{E}}\left[\prod_{i \in S} x_i\right] = \Pr_{y \sim \mathcal{D}_S}[\forall i \in S : x_i = y_i]$$

and extended to all of $\mathcal{P}_{n,d}$ by linearity is a degree-d pseudoexpectation for $R \subseteq \{0,1\}^n \times O$.

In fact, the two objects are equivalent, but we will only need the above direction of this equivalence in the current work.

We now introduce our new variant of pseudoexpectation operators that are tuned to proving lower bounds for PPP^{dt}-formulations. This definition must be stronger than the definition of a standard pseudoexpectation, since standard pseudoexpectations prove lower bounds for the Sherali-Adams hierarchy, and it is known that $\textsc{Pigeon}^{dt}$ is hard for Sherali-Adams. However, this definition must be weaker than the *positive semidefinite* pseudoexpectations that imply lower bounds for the SOS hierarchy, since one can show that both $\textsc{Pigeon}^{dt}$ and $(\textsc{Pigeon}^{\otimes 2})^{dt}$ are easy for the SOS hierarchy (see Appendix A). We need one more auxiliary definition before we can define our new variant of a pseudoexpectation operator.

Definition 3.4. Let $R \subseteq \{0,1\}^n \times O$ be a query total search problem, and let $\mathcal{F}$ be a family of conjunctions over $x_1, \dots, x_n$. The family $\mathcal{F}$ is *d-pairwise witnessing* for R if no conjunction in $\mathcal{F}$ witnesses a solution for R, but, for any pair of conjunctions $C_1 \ne C_2 \in \mathcal{F}$, either $C_1 C_2 \equiv 0$ or $R \restriction C_1 C_2$ has decision tree complexity at most d.

In other words, for any pair of *consistent* conjunctions C_1, C_2 in a 2-witnessing family, restricting R by $C_1 C_2$ makes the search problem efficiently solvable.

Definition 3.5. Let $R \subseteq \{0,1\}^n \times O$ be a total query search problem, and let d be a positive integer. Let $\tilde{\mathbb{E}}$ be a degree $D \geq d$ pseudoexpectation for R. The operator $\tilde{\mathbb{E}}$ is *d-collision-free* if it satisfies the following property:

– *Collision Freedom.* $\sum_{C \in \mathcal{F}} \tilde{\mathbb{E}}[C] \leq 1$ for every d-pairwise witnessing $\mathcal{F}$ of degree $\leq d$ conjunctions.

As we show next, this notion of "collision-freedom" is strong enough to prove lower bounds against PIGEON^{dt}-formulations, and therefore against PPP^{dt}.

Theorem 3.6. *Let $R \subseteq \{0,1\}^n \times O$ be a total query search problem, and let d be a positive integer. If $\tilde{\mathbb{E}}$ is a degree $D \geq 2d$, d-collision-free pseudoexpectation for R, then there is no PIGEON^{dt}-formulation of R with complexity $\leq d$.*

Proof. Suppose there is a PIGEON_N^{dt}-formulation of R with complexity at most d. For each pigeon $i = 1, 2, \ldots, N$, we have a decision tree T_i which queries at most d input bits of R and outputs some hole $T_i(x) \in [N]$ for the pigeon i. For any depth $\leq d$ decision tree T querying bits of R, let $L(T)$ denote the leaves of T, and for any leaf $\ell \in L(T)$ let C_ℓ denote the conjunction of literals on the path to ℓ.

We begin by doing some pre-processing on the PIGEON_N^{dt} formulation in order to put 0-weight on paths which map a pigeon to the forbidden hole 1. Suppose that $\ell \in L(T_i)$ is any leaf of the pigeon tree T_i labelled with 1. Let $g_{i,1} : \{0,1\}^n \to O$ be the decision tree defined by the formulation that maps the solution $i \mapsto 1$ of PIGEON^{dt} to a solution of R. For each i and for each such leaf $\ell \in T_i$, replace the leaf ℓ with a copy of $g_{i,1}$ (removing redundant queries when necessary in the subtree), and label each leaf of the new subtree with 1. It is clear that this new formulation has depth at most $2d$ instead of d. Furthermore, for any leaf ℓ labelled with 1 in this new formulation, the correctness of the original formulation implies that C_ℓ witnesses a solution to R, and therefore $\tilde{\mathbb{E}}[C_\ell] = 0$.

We first claim that

$$\sum_{\ell \in L(T)} C_\ell = 1,$$

where the equation is between polynomials, for any decision tree T. This can be seen by an easy induction on the depth of T. If the depth of T is 0, then the conjunction C_ℓ is empty, and thus $C_\ell = 1$. Inductively, consider the root node u of T. Suppose that u queries the variable x_i, and when x_i is 0 it proceeds to the subtree T_0, and when x_i is 1 it proceeds to the subtree T_1. The claim follows by a straightforward calculation:

$$\begin{aligned}
\sum_{\ell \in L(T)} C_\ell &= \sum_{\ell \in L(T_0)} (1 - x_i) C_\ell + \sum_{\ell \in L(T_1)} x_i C_\ell \\
&= (1 - x_i) \sum_{\ell \in L(T_0)} C_\ell + x_i \sum_{\ell \in L(T_1)} C_\ell \\
&= (1 - x_i) + x_i \\
&= 1,
\end{aligned}$$

where the third equality is by the inductive hypothesis. The claim immediately implies $\sum_{\ell \in L(T)} \tilde{\mathbb{E}}[C_\ell] = 1$ by the linearity and normalization of $\tilde{\mathbb{E}}$.

Now, for each $j = 1, 2, 3, \ldots, N$, define the set of conjunctions

$$\mathcal{H}_j := \bigcup_{i=1}^{N} \{C : C = C_\ell \text{ for some } j\text{-labelled leaf } \ell \in L(T_i)\}.$$

First, observe that

$$\sum_{i=1}^{N}\sum_{\ell\in L(T_i)}\tilde{\mathbb{E}}[C_\ell]=\sum_{j=1}^{N}\sum_{C\in\mathcal{H}_j}\tilde{\mathbb{E}}[C],$$

since each conjunction on a root-to-leaf path in T_i is mapped to exactly one set $\mathcal{H}_j$. Second, by the preprocessing step above, we know that for every conjunction $C \in \mathcal{H}_1$, we have $\tilde{\mathbb{E}}[C] = 0$, since each such C must be witnessing for R. Finally, we claim that for $j = 2, 3, \ldots, N$, the family $\mathcal{H}_j$ is d-pairwise witnessing. To see this, consider any two conjunctions $C_1 \neq C_2 \in \mathcal{H}_j$ that are consistent. These two conjunctions must have come from different decision trees T_i, T_j, since all conjunctions coming from the same decision tree are inconsistent. But then, by the correctness of the PIGEON^{dt}-formulation, it follows that pigeons i and j are mapped to the same hole under any input x consistent with C_1C_2, and therefore we can recover a solution to R using at most d more queries. Therefore $\tilde{\mathbb{E}}[\mathcal{H}_j] := \sum_{C\in\mathcal{H}_j}\tilde{\mathbb{E}}[C] \leq 1$, since $\tilde{\mathbb{E}}$ is collision-free. By combining these three facts together, along with the fact that $\sum_{\ell\in L(T)}\tilde{\mathbb{E}}[C_\ell] = 1$, we have

$$N=\sum_{i=1}^{N}\sum_{\ell\in L(T_i)}\tilde{\mathbb{E}}[C_\ell]=\sum_{j=1}^{N}\sum_{C\in\mathcal{H}_j}\tilde{\mathbb{E}}[C]=\sum_{j=2}^{N}\sum_{C\in\mathcal{H}_j}\tilde{\mathbb{E}}[C]\leq N-1,$$

which is a contradiction. □

Therefore, if we can construct a degree $\omega(\text{poly}(\log(n)))$ collision-free pseudoexpectation for $(\text{PIGEON}^{\otimes 2})^{dt}$, we will prove our main theorem. The construction of this pseudoexpectation is done in the next section.

4 Lower Bounds for Black-Box PPP

The goal of this section is to prove the following theorem.

Theorem 4.1. $(\text{PIGEON}^{\otimes 2})^{dt} \not\in \mathsf{PPP}^{dt}$.

We prove the above theorem by constructing a collision-free pseudoexpectation for the PIGEON$^{\otimes 2}$ problem. For the remainder of the paper we will now drop the "dt" superscript for notational convenience.

4.1 Constructing a Collision-Free Pseudoexpectation for PIGEON$^{\otimes 2}$

In this section, we construct a collision-free pseudoexpectation for PIGEON$^{\otimes 2}$. To do this, we must first set down some notational preliminaries.

Preliminaries. Throughout this section we will be considering PIGEON$_n^{\otimes 2}$, the input of which is two functions $f, g : [n+1] \to [n+1]$ As we discussed above, we will consider these inputs as encoded over the domain $[n+1]$ instead of a binary domain for notational convenience. Formally, the input to PIGEON$_n^{\otimes 2}$ will be two tuples $(x, y) \in [n+1]^{n+1} \times [n+1]^{n+1}$, encoding the two functions f, g, respectively. (We can always convert to the underlying binary encoding by replacing each $x_i \in [n+1]$ with $O(\log n)$ bits encoding the value pointed to by x_i.) Decision trees querying the input will have their internal nodes querying input indices (e.g. "x_i") and outputting values in $[n+1]$. Formally speaking, a decision tree querying variables taking values over $[n+1]$ is equivalent to a standard binary decision tree that has the guarantee that it always queries the entirety of a single pointer x_i.

In this multivalued framework, a "conjunction" is now a conjunction of atoms of the form $[\![x_i \mapsto j]\!]$ for some $i, j \in [n+1]$, indicating that pigeon i maps to the hole j. We can convert to the standard polynomial conjunctions discussed in the above section by employing the binary encoding. Each such conjunction over the variables of $\text{PIGEON}_n^{\otimes 2}$ therefore corresponds to an assignment of some of the pigeons to holes in the two PIGEON instances comprising the $\text{PIGEON}^{\otimes 2}$ instance.

Definition 4.2. A *matching term* M is any conjunction of variables in $\text{PIGEON}_n^{\otimes 2}$ that encodes a matching across the two underlying PIGEON instances, such that no pigeon is mapped to the hole 1 in either of the instances. Two matching terms M_1, M_2 are *consistent* if $M_1 M_2 \not\equiv 0$ — in other words, if there is no pigeon that is mapped to two different holes by $M_1 M_2$. Two matching terms M_1, M_2 are *coherent* if $M_1 M_2$ is itself a matching term (i.e. the union of matchings encoded by M_1 and M_2 form a matching).

In other words, a matching term M encodes an instance that does not witness a collision in either of the two PIGEON instances. Further note that coherency is a strictly stronger condition than consistency: if $M_1 M_2$ witnesses a collision between two pigeons, then M_1 and M_2 are *consistent* as conjunctions, but not *coherent*.

We use matchings to define our pseudoexpectation, as follows:

Definition 4.3. The *degree-d matching pseudodistribution* is defined as follows. Given two sets of pigeons P_1, P_2 from the two instances of PIGEON_n comprising $\text{PIGEON}_n^{\otimes 2}$ such that $|P_1 \cup P_2| \leq d$, the distribution $\mathcal{D}_{P_1,P_2}$ samples a uniformly random matching from the P_1 pigeons to $|P_1|$ holes and the P_2 pigeons to $|P_2|$ holes, avoiding the hole 1. Formally, given two matching terms M_1 of the P_1 pigeons to holes, and M_2 of the P_2 pigeons to holes, the corresponding pseudoexpectation is defined to be

$$\tilde{\mathbb{E}}[M_1 M_2] := \prod_{i=0}^{|P_1|-1} \frac{1}{n-i} \prod_{j=0}^{|P_2|-1} \frac{1}{n-j},$$

and extended by linearity.

We first prove that this is indeed a pseudodistribution — and thus, by Lemma 3.3, $\tilde{\mathbb{E}}$ is a pseudoexpectation — for $\text{PIGEON}_n^{\otimes 2}$.

Lemma 4.4. *For any $d \leq n-1$, the degree-d matching pseudodistribution is a pseudodistribution for $\text{PIGEON}_n^{\otimes 2}$.*

Proof. We first observe that no assignment in the support of the pseudodistribution witnesses a solution to $\text{PIGEON}_n^{\otimes 2}$, since they are matchings that avoid the hole 1 in both instances of PIGEON. So, we verify the shared marginals property. Let $S = P_1 \cup P_2$ be any subset of pigeons with $|S| \leq d-1$ and let $|P_1| = d_1, |P_2| = d_2$. Write $P_1 = \{p_1^1, p_2^1, \ldots, p_{d_1}^1\}$ and $P_2 = \{p_1^2, p_2^2, \ldots, p_{d_2}^2\}$, and let p be any pigeon not appearing in $P_1 \cup P_2$. Suppose without loss of generality that p is in the first sub-instance of PIGEON. Let $h_1^1, \ldots, h_{d_1}^1, h^1$ be any set of $d_1 + 1$ distinct holes in $[n+1] \setminus \{1\}$, and let $h_1^2, \ldots, h_{d_2}^2$ be any set of d_2 distinct holes in $[n+1] \setminus \{1\}$. Then

$$\Pr_{\mathbf{x},\mathbf{y} \sim \mathcal{D}_{S \cup p}}[(x,y) = (\mathbf{x},\mathbf{y})] = \Pr[\forall i,j : p_i^1 \mapsto h_i^1 \wedge p_j^2 \mapsto h_j^2 \wedge p \mapsto h^1] = \prod_{i=0}^{d_1} \frac{1}{n-i} \prod_{j=0}^{d_2-1} \frac{1}{n-j}.$$

Marginalizing out the pigeon p over each of the possible $n - d_1$ choices for the hole h^1 we have

$$\begin{aligned}\sum_{h^1=1}^{n-d_1} \prod_{i=0}^{d_1} \frac{1}{n-i} \prod_{j=0}^{d_2-1} \frac{1}{n-j} &= \prod_{i=0}^{d_1-1} \frac{1}{n-i} \prod_{i=0}^{d_2-1} \frac{1}{n-j} \\ &= \Pr[\forall i,j : p_i^1 \mapsto h_i^1 \wedge p_j^2 \mapsto h_j^2] \\ &= \Pr_{(\mathbf{x},\mathbf{y}) \sim \mathcal{D}_S}[(x,y) = (\mathbf{x},\mathbf{y})],\end{aligned}$$

and thus the shared marginals property is verified. □

Throughout the rest of the section, when we use the symbol $\tilde{\mathbb{E}}$ we will always be referring to the matching pseudodistribution. The rest of the section is devoted to the proof of Theorem 4.1.

Proof of Theorem 4.1.

Proof Outline. We begin by describing the overall strategy of our proof. As we have shown above, the pseudoexpectation $\tilde{\mathbb{E}}$ is indeed a pseudoexpectation for $\text{PIGEON}^{\otimes 2}$, and we must show that it is d-collision free for some suitable choice of d. Formally, the goal is to prove, for any pairwise-witnessing family $\mathcal{F}$, that

$$\tilde{\mathbb{E}}[\mathcal{F}] := \sum_{M \in \mathcal{F}} \tilde{\mathbb{E}}[M] \leq 1.$$

To do this, we must analyze pairwise-witnessing families for $\text{PIGEON}^{\otimes 2}$. First, we observe that since $\tilde{\mathbb{E}}$ places 0 weight on any conjunction that witnesses a collision across either sub-instance of $\text{PIGEON}^{\otimes 2}$, it follows that we only need to consider pairwise-witnessing families consisting entirely of matchings. Let us define a *strong* pairwise witnessing family to be a pairwise witnessing family satisfying the following stronger conclusion: if $M_1 \neq M_2 \in \mathcal{F}$ are consistent, then $M_1 M_2$ must itself witness a solution to $\text{PIGEON}^{\otimes 2}$. The first step of the argument is to reduce to the case where the family is *strongly* pairwise witnessing (cf. Lemma 4.5). This reduction is crucial for the next step.

Next, let us fix a strong pairwise witnessing family $\mathcal{F}$, and we seek to show that $\tilde{\mathbb{E}}[\mathcal{F}] \leq 1$. To prove this inequality, we will first endow $\mathcal{F}$ with "more structure", as follows: we will find a *shallow matching decision tree* $T_{\mathcal{F}}$ that "covers" the weight of the family $\mathcal{F}$, in the sense that $\tilde{\mathbb{E}}[\mathcal{F}] \leq \tilde{\mathbb{E}}[T_{\mathcal{F}}]$ (cf. Lemma 4.12). A matching decision tree is like a standard decision tree, except, it only describes matchings. The internal nodes of a matching decision tree are either *pigeon queries*, where we learn a hole that the pigeon is mapped to, or *hole queries*, where we learn which of the remaining pigeons maps to this hole. Each leaf of a matching decision tree is labelled with 0 or 1. Letting $L_1(T_{\mathcal{F}})$ denote the family of matchings corresponding to the 1-leaves of $T_{\mathcal{F}}$, we define $\tilde{\mathbb{E}}[T_{\mathcal{F}}] = \tilde{\mathbb{E}}[L_1(T_{\mathcal{F}})]$. While we cannot find a matching decision tree $T_{\mathcal{F}}$ such that $L_1(T_{\mathcal{F}}) = \mathcal{F}$ (indeed, such a strong statement is clearly false), we can settle for the weaker goal of finding a shallow matching decision tree that simply covers all of the weight of $\mathcal{F}$. Said another way, this step of the proof can be viewed as a second reduction, where now we only need to bound the weight of strong pairwise-witnessing families arising from shallow matching decision trees.

The rest of the proof is an analysis of the pseudoexpectation weight of matching decision trees that describe pairwise-witnessing families. To get an idea of how one can analyze this, it is helpful to think about how the weight of a matching decision tree evolves "marginally" as it queries pigeons and holes. Suppose that we have made some queries in our matching decision tree T, and have arrived at a node u with a partial matching M from t pigeons to t holes learned at this point. For the sake of argument, suppose that M only queries pigeons in one of the sub-instances of $\text{PIGEON}_n^{\otimes 2}$. By definition of the matching pseudoexpectation, in this case $\tilde{\mathbb{E}}[M] = (n(n-1)\cdots(n-t+1))^{-1}$. If the node u queries a pigeon p_i that can map to $n-t$ remaining holes, then the matching M is extended to $n-t$ matchings, each of weight $(n(n-1)\cdots(n-t))^{-1}$. We can see that marginally we have not increased the weight of the pairwise witnessing family, since the $(n-t)$ new matchings are balanced out by the additional decrease in weight of $(n-t)^{-1}$. Thus pigeon queries can safely be "ignored", at least at this heuristic level, since they "don't marginally contribute" to the weight of the family.

Everything changes when we make hole queries, leading us to the crux of the argument. If the node u queries a hole instead, there are now $n-t+1$ possible pigeons that can map to this hole. It follows that the partial matching M is now replaced with $n-t+1$ different matchings, each of weight $(n(n-1)\cdots(n-t))^{-1}$. We have therefore increased the total weight of the family by a $(n-t+1)/(n-t) = 1 + 1/(n-t)$ multiplicative factor. Considered by itself, this appears hopeless at first, since we have increased the weight of the family beyond 1. It is here that we must use the fact that $\mathcal{F}$ is strongly pairwise witnessing. Roughly speaking, we can show that since $\mathcal{F}$ is strongly pairwise witnessing, whenever we make a hole query, most of the leaves below the hole query must be labelled with 0, compensating for the apparent multiplicative "gain" in the total weight from the hole query (cf. Theorem 4.15). It is natural to try to prove this directly by somehow arguing about the structure of $T_{\mathcal{F}}$. However, this direct approach seems technically difficult to implement successfully. Instead, we can avoid this argument by doing a general trick that exploits the shallowness of the matching decision trees. We remark that exploiting the bounded depth via this trick is the *only* step of our argument which *fails* for the weak pigeonhole principle (i.e. where we map $2n$ pigeons to n holes), which must happen somewhere since the weak pigeonhole principle is "non-adaptively Turing closed" [Jer16], as we discussed in the introduction.

Reduction to Strong Pairwise Witnessing Families. We now begin the proof in earnest. First we exhibit the reduction to *strong* pairwise witnessing families.

Lemma 4.5. *Let n, d be positive integers with $d < n/2$. Suppose $\mathcal{F}$ is d-pairwise witnessing family of matching terms, each of degree $\leq d$, for* $\textsc{Pigeon}_n^{\otimes 2}$. *Then for any matching terms M_1, M_2 such that $M_1 M_2 \not\equiv 0$, $M_1 M_2$ witnesses a solution to* $\textsc{Pigeon}_n^{\otimes 2}$. *Equivalently, the decision tree complexity of* $\textsc{Pigeon}_n^{\otimes 2} \restriction M_1 M_2$ *is* 0.

Proof. Let $M_1, M_2 \in \mathcal{F}$ be matchings across the two $\textsc{Pigeon}_n$ instances such that $M_1 M_2 \not\equiv 0$. Since $\mathcal{F}$ is d-pairwise witnessing, it follows that we can recover a solution to $\textsc{Pigeon}_n^{\otimes 2}$ — that is, collisions in both of the sub-instances — using at most d extra queries. Suppose additionally by contradiction that $M_1 M_2$ does not witness a solution. Let $H_1 \subseteq M_1, H_2 \subseteq M_2$ be the submatchings on the first $\textsc{Pigeon}_n$ sub-instance, and suppose without loss of generality that $H_1 H_2$ does not witness a collision, and is therefore a matching. Let $p \leq d$ be the number of pigeons queried by $H_1 H_2$. It follows that $\textsc{Pigeon}_n \restriction H_1 H_2$ can recover a collision with at most d extra queries, and thus it has decision-tree complexity at most d. But $\textsc{Pigeon}_n \restriction H_1 H_2 \equiv \textsc{Pigeon}_{n-p}$, since we are just matching p pigeons to holes, and the decision tree complexity of $\textsc{Pigeon}_{n-p}$ is $\geq n - p$ by an easy adversary argument. Therefore $d \geq n - p \geq n - d$, and thus $d \geq n/2$, which is a contradiction. □

Reduction to Matching Decision Trees. We now move on to constructing a matching decision tree that covers the weight of a strong pairwise-witnessing family. In this part of the proof, it will be convenient to think of families as DNF formulas, as follows:

Definition 4.6. A *matching DNF* is a DNF F where every term M in the DNF is a matching term over the variables of $\textsc{Pigeon}_n^{\otimes 2}$. The *width* of F is the degree of the widest term in F. Given a family of matchings $\mathcal{F}$, the matching DNF corresponding to $\mathcal{F}$ is $\bigvee_{M \in \mathcal{F}} M$. We say a matching DNF is *good* if its corresponding family of matchings is strongly d-pairwise witnessing.

Given a matching DNF F we can evaluate the matching pseudoexpectation on F by defining $\tilde{\mathbb{E}}[F] := \sum_{M \in F} \tilde{\mathbb{E}}[M]$ to be the sum of the pseudoexpectation values in F. The definition of coherency leads us to the following simple observation:

Observation 4.7. *If a matching DNF F is good then every pair of matchings $M_1 \neq M_2$ in F are incoherent.*

Proof. If F is good, then every pair of matchings $M_1 \neq M_2$ are either inconsistent or witness a solution to $\textsc{Pigeon}_n^{\otimes 2}$, i.e. $M_1 M_2$ are incoherent. □

If M_1 and M_2 are two matching terms, then define the matching term

$$M_1 \restriction M_2 := \begin{cases} 0 & \text{if } M_1 \text{ and } M_2 \text{ are incoherent,} \\ 1 & \text{if } M_1 \subseteq M_2, \\ M_1 \setminus M_2 & \text{otherwise.} \end{cases}$$

We extend the restriction operator to matching DNFs F by defining

$$F \restriction M := \begin{cases} 0 & \text{if } N \restriction M = 0 \text{ for all } N \in F \\ 1 & \text{if there is an } N \in F \text{ such that } N \restriction M = 1 \\ \bigvee_{N \in F} N \restriction M & \text{otherwise} \end{cases}$$

Implicitly, in the third case above, we remove any term $N \restriction M$ of the DNF if $N \restriction M = 0$.

It is crucial to note here that the restriction operator outputs different "truth values" than it would under normal boolean assignments, due to the substitution of "coherency" for "consistency". However, these "nonstandard" truth values are consistent with the values of the pseudoexpectation operator $\tilde{\mathbb{E}}$, which is why they are useful for us. Indeed, this switch to non-standard logic and "coherency" is one of the key steps that make the proof possible.

We can now introduce the central notion of a *matching decision tree*.

Definition 4.8. Consider a $\text{PIGEON}_n^{\otimes 2}$ instance. A *matching decision tree* T is a rooted tree defined as follows. Each internal node of the tree is labelled by either a pigeon p or a hole h occurring in the $\text{PIGEON}_n^{\otimes 2}$ instance, and each leaf of the tree is labelled with 0 or 1. If a node is labelled with a pigeon p, then the outgoing edges are labelled with pairs of the form $p \mapsto h$, where h is an available hole that the pigeon p can map to. Similarly, if a node is labelled with a hole h, then the outgoing edges are labelled with pairs of the form $p \mapsto h$, where p is an unmapped pigeon that can map to h. Furthermore, no node or edge label can be repeated on any path of T, and if u is a node of T then the edge labels on the path from the root to u should determine a matching, denoted $\pi(u)$, from pigeons p to holes h. The matching tree is *full* if every internal node has the maximum number of children.

Given a matching decision tree T and a bit $b \in \{0, 1\}$, let

$$L_b(T) := \{\pi(\ell) : \ell \text{ is a leaf of } T \text{ labelled with } b\}$$

and $M(T) = L_0(T) \cup L_1(T)$. We can evaluate the matching pseudoexpectation on a matching decision tree by defining

$$\tilde{\mathbb{E}}[T] := \sum_{M \in L_1(T)} \tilde{\mathbb{E}}[M]$$

to be the sum over the matchings at the 1-leaves of T.

Next, we discuss how to relate matching decision trees and matching DNFs. If T is a matching decision tree and F is a matching DNF, then T *represents* F if for every b-labelled leaf ℓ of T, $F \restriction \pi(\ell) = b$. Furthermore, we say that T *strongly represents* F if T represents F and, furthermore, for every 1-leaf ℓ of T there is a *unique* matching $\sigma \in F$ that is coherent with $\pi(\ell)$. An obvious decision tree that represents any matching DNF is the so-called *canonical* decision tree associated with F, which we define now (and will be familiar to any reader acquainted with the switching lemma). First we need the definition of a *full* matching tree for a set of nodes U.

Definition 4.9. Let U be a subset of nodes (either pigeons or holes) in $\text{PIGEON}_n^{\otimes 2}$. The *full matching decision tree T_U covering U* is defined recursively as follows. We begin with an unlabelled root node.

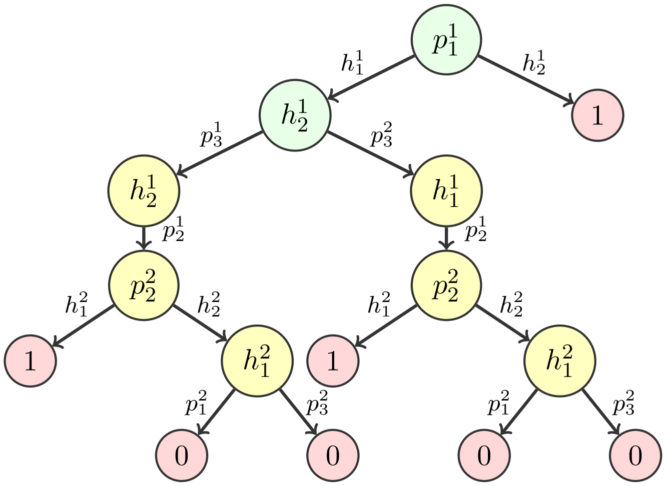

Figure 2: A canonical matching decision tree for $\llbracket p_1^1 \mapsto h_2^1 \rrbracket \vee (\llbracket p_1^1 \mapsto h_1^1 \rrbracket \wedge \llbracket p_2^2 \mapsto h_1^2 \rrbracket)$ of $\textsc{Pigeon}_2^{\otimes 2}$. The nodes in green correspond to the full matching tree for $\llbracket p_1^1 \mapsto h_2^1 \rrbracket$, while the nodes in yellow come from the full matching tree for $(\llbracket p_1^1 \mapsto h_1^1 \rrbracket \wedge \llbracket p_2^2 \mapsto h_1^2 \rrbracket) \restriction \pi(\ell)$.

While there is an unlabelled node $x \in T_U$, choose the first uncovered pigeon or hole $u \in U$ and label x with u. The children of x have edges connected to x labelled with all edges e containing u such that $\pi(u) \cup \{e\}$ is a matching.

Definition 4.10. Let $F = M_1 \vee M_2 \vee \cdots \vee M_m$ be a matching disjunction over $\textsc{Pigeon}_n^{\otimes 2}$. The *canonical matching decision tree* of F, denoted $\mathsf{Can}(F)$, is the matching decision tree defined as follows.

- If $F \equiv b$ for $b \in \{0, 1\}$, then $\mathsf{Can}(F)$ is a single leaf node labelled with b.
- If F is not constant, then consider the first i such that $M_i \not\equiv 0$. The tree $\mathsf{Can}(F)$ is constructed as follows:
  - Construct the full matching decision tree T_U covering the set of nodes U appearing in M_i.
  - Recursively replace each leaf ℓ of T_U with the canonical tree $\mathsf{Can}(F \restriction \pi(\ell))$.

It is obvious from the definition that $\mathsf{Can}(F)$ *represents* any matching disjunction F. However, it is not true that $\mathsf{Can}(F)$ will always *strongly* represent F. To see this, consider the matching disjunction $F = \llbracket p_1 \mapsto h_1 \rrbracket \vee \llbracket p_2 \mapsto h_2 \rrbracket$. The tree $\mathsf{Can}(F)$ will start by querying the pigeon p_1, and when it learns that $p_1 \mapsto h_1$ it will stop and output 1. However, the matching $\llbracket p_1 \mapsto h_1 \rrbracket$ is coherent with the matching $\llbracket p_2 \mapsto h_2 \rrbracket$, and so $\mathsf{Can}(F)$ will not strongly represent F. Thus $\mathsf{Can}(F)$ does not strongly represent F whenever F contains two coherent matchings. In fact, we can show that this is the *only* time this happens:

Lemma 4.11. *If F is any matching disjunction then $\mathsf{Can}(F)$ represents F. Furthermore, if for all matching terms $M_1 \neq M_2$ in F, M_1 and M_2 are incoherent, then $\mathsf{Can}(F)$ strongly represents F.*

Proof. The fact that $\mathsf{Can}(F)$ represents F is clear from the definition of $\mathsf{Can}(F)$. So, we focus on showing the claim about strong representation. Write $F = M_1 \vee M_2 \vee \cdots \vee M_m$, and consider any 1-leaf ℓ of $\mathsf{Can}(F)$. By construction, $\mathsf{Can}(F)$ represents F, and so $F \restriction \pi(\ell) \equiv 1$ and thus there is a matching M_i such that $M_i \subseteq \pi(\ell)$. Suppose by way of contradiction that there is a $j \neq i$ such that M_j is also coherent with $\pi(\ell)$. Then M_i and M_j are also coherent, contradicting the assumption. □

The importance of strong representation owes to the fact that it conserves pseudoexpectation weight. We prove this next:

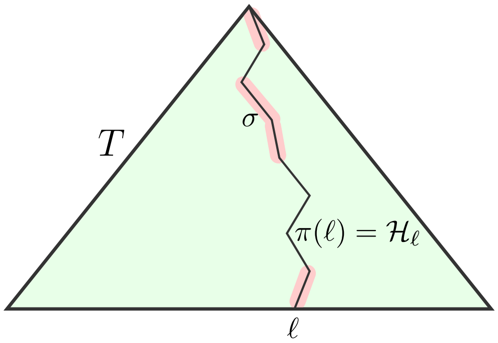

Figure 3: The base case of Lemma 4.12. The black path is $\pi(\ell)$ and the red segments are the edges in σ.

Lemma 4.12. *Consider* $\text{PIGEON}_n^{\otimes 2}$*, let* $d \leq n-1$*, and let* F *be any width-*d *matching DNF. If* T *is a full matching decision tree that strongly represents* F*, then* $\tilde{\mathbb{E}}[F] \leq \tilde{\mathbb{E}}[T]$*.*

Proof. Letting $\mathcal{H}$ denote the family of matchings associated with F, we show that $\tilde{\mathbb{E}}[T] = \tilde{\mathbb{E}}[L_1(T)] \geq \tilde{\mathbb{E}}[\mathcal{H}] = \tilde{\mathbb{E}}[F]$. Let u be any node in the tree T and let T_u be the subtree of T rooted at u. Define the sets of matching terms

$$\begin{aligned} \mathcal{H}_u &= \{M\pi(u) : M \in \mathcal{H} \text{ and } M\pi(u) \text{ is coherent}\} \\ \mathcal{M}_u &= \{N\pi(u) : N \in L_1(T_u)\}\,. \end{aligned}$$

We argue by induction over T that $\tilde{\mathbb{E}}[\mathcal{M}_u] \geq \tilde{\mathbb{E}}[\mathcal{H}_u]$. When $u = r$ is the root node, we have that $\mathcal{H}_r = \mathcal{H}$ and $\mathcal{M}_r = L_1(T)$, which completes the proof.

For the base case, consider any leaf node ℓ of the tree T. Clearly $\mathcal{M}_\ell \subseteq \{\pi(\ell)\}$ and, since T strongly represents F, $\pi(\ell)$ is coherent with at most one element of $\mathcal{H}$ which it must extend. If ℓ is labelled with 0, then by the definition of strong representation $\pi(\ell) \cup \sigma$ is incoherent for all $\sigma \in \mathcal{H}$. This means that $\mathcal{H}_\ell = \emptyset$ and the induction hypothesis is satisfied. Otherwise, let σ be the unique element of $\mathcal{H}$ such that $\pi(\ell) \supseteq \sigma$, then we have $\mathcal{H}_\ell = \mathcal{M}_\ell$, as $\mathcal{H}_\ell$ consists of exactly the extension of σ by the path $\pi(\ell)$, and the claim is true. (We remark that it is the fact that σ is unique is precisely where we need *strong* representation in order to prove the lemma.)

Now, consider any internal node u of the tree T and let $v_1, \ldots, v_c$ denote the child nodes of u in T. Let $\rho_1, \rho_2, \ldots, \rho_c$ be the edge labels on the edges connecting u to v_i in T. By definition $\bigcup_{i=1}^c \mathcal{M}_{v_i}$ is a partition of $\mathcal{M}_u$, and thus $\tilde{\mathbb{E}}[\mathcal{M}_u] = \sum_i \tilde{\mathbb{E}}[\mathcal{M}_{v_i}]$. By induction, $\tilde{\mathbb{E}}[\mathcal{H}_{v_i}] \leq \tilde{\mathbb{E}}[\mathcal{M}_{v_i}]$ for each i, and thus we show that $\tilde{\mathbb{E}}[\mathcal{H}_u] \leq \sum_i \tilde{\mathbb{E}}[\mathcal{H}_{v_i}]$. Since T is full, if $M \in \mathcal{H}$ is coherent with $\pi(u)$, then M will also be coherent with at least one child of $\pi(u)$, and thus all $N \in \mathcal{H}_u$ will be coherent with at least one child of u. For each i, we can write

$$\mathcal{H}_{v_i} = \{N\rho_i : N \in \mathcal{H}_u\}$$

where ρ_i is the edge label on the edge connecting u to v_i in T.

Consider any $N \in \mathcal{H}_u$, and recall that our goal is to prove that $\tilde{\mathbb{E}}[\mathcal{H}_u] \leq \sum_{i=1}^c \tilde{\mathbb{E}}[\mathcal{H}_{v_i}]$. There are exactly two possibilities:

1. There is a unique v_i such that $N \in \mathcal{H}_{v_i}$, or
2. N does not occur in any $\mathcal{H}_{v_i}$.

The first case occurs if the pigeon or hole queried by u occurs in N, and otherwise the second case occurs. We will prove that the weight of N is covered in either case. In the first case, the weight $\tilde{\mathbb{E}}[N]$ in $\mathcal{H}_u$

is covered exactly by the weight $\tilde{\mathbb{E}}[N]$ in $\mathcal{H}_{v_i}$, since there is a unique outgoing edge for N to follow in the decision tree. In the second case, let $t \leq c$ be the number of edges of N occurring in the PIGEON instance containing the query labelled on the node u. There are exactly $c - t$ edges among the labels $\rho_1, \rho_2, \ldots, \rho_c$ that are coherent with N, so suppose without loss of generality that the first $c - t$ labels are the coherent ones. Then the set $\mathcal{H}_{v_i}$ for $1 \leq i \leq c - t$ contains a copy of the matching $N\rho_i$, which has weight $\tilde{\mathbb{E}}[N] \cdot (c-t-1)^{-1} \geq \tilde{\mathbb{E}}[N] \cdot (c-t)^{-1}$ if u was a hole query, and weight $\tilde{\mathbb{E}}[N] \cdot (c-t)^{-1}$ if u was a pigeon query. Since there are $(c-t)$ of these matchings, in either case we have that

$$\sum_{i=1}^{c-t} \tilde{\mathbb{E}}[N\rho_i] \geq \left(\frac{c-t}{c-t}\right) \tilde{\mathbb{E}}[N] = \tilde{\mathbb{E}}[N].$$

Thus in the second case the weight of $\tilde{\mathbb{E}}[N]$ is covered as well. This means that $\tilde{\mathbb{E}}[\mathcal{H}_u] \leq \sum_{i=1}^{c} \tilde{\mathbb{E}}[\mathcal{H}_{v_i}]$, completing the induction step and the proof. □

Corollary 4.13. *For any good matching disjunction F, $\tilde{\mathbb{E}}[F] \leq \tilde{\mathbb{E}}[\mathsf{Can}(F)]$.*

Bounding the Weight of Matching Decision Trees. We are now on the final steps of the proof. First, we will repeatedly use the following useful lemma that bounds the depth of canonical decision trees.

Lemma 4.14. *Let F be any width-d matching DNF such that for all $M_1 \neq M_2 \in F$, $M_1 \cup M_2$ is incoherent. Then the depth of $\mathsf{Can}(F)$ is $O(d^2)$.*

Proof. The canonical decision tree $\mathsf{Can}(F)$ is constructed in rounds, where in each round we choose the next matching term M_i that is not set to a constant and construct the full matching decision tree over that term. We prove the following claim by induction:

Claim. Let ℓ_i be any leaf in the tree produced after i rounds of construction of $\mathsf{Can}(F)$, let M_i be any matching term in F such that $M_i \upharpoonright \pi(\ell)$ is not a constant. Then $|M_i \setminus \pi(\ell_i)| \leq |M_i| - i$.

Proof of Claim. When $i = 0$ the claim is vacuously true. So, by way of induction, suppose we have just completed the i^{th} round of the construction of $\mathsf{Can}(F)$. Let ℓ_{i-1} be the leaf of the tree chosen at the beginning of the i^{th} round, and let M_{i-1} be the matching from F chosen to be queried during the i^{th} round. By induction, we know that $|M_i \setminus \pi(\ell_{i-1})| \leq |M_i| - (i-1)$. So, we need to argue that during the i^{th} round we queried at least one more edge in M_i.

To see this, we observe that in the i^{th} round we construct the full matching decision tree over $M_{i-1} \upharpoonright \pi(\ell_{i-1})$, and thus query every node appearing in $M_{i-1} \upharpoonright \pi(\ell_{i-1})$ in this sub-tree. Since M_i and M_{i-1} are incoherent matching terms, there is a node u appearing in M_i such that u participates in a different edge in both M_i and M_{i-1}. Since the leaf ℓ_i is coherent with M_i, it cannot be coherent with M_{i-1}, and thus this node must have been queried along the path from ℓ_{i-1} to ℓ_i and the query must be coherent with M_i. Therefore, we have made one more query to M_i, and so the proof of the claim is complete. □

Due to the claim, in each round of construction the width of all coherent matching terms that remains decreases by 1. Since all terms begin with width $\leq d$, this means that the construction of $\mathsf{Can}(F)$ can continue for at most d rounds, and in each round we query at most $2d$ edges. Thus the depth of any path in $\mathsf{Can}(F)$ is at most $O(d^2)$. □

We are now ready to combine these ingredients together and prove our final weight bound. The next theorem, when combined with Theorem 3.6, immediately implies the main result (Theorem 4.1) as a corollary.

Theorem 4.15. *If $d = o(n^{1/8})$ and F is a good matching DNF of width at most d, then $\tilde{\mathbb{E}}[F] \leq 1$. In particular, letting $D = n/2$, the degree-D matching pseudoexpectation for $\mathrm{PIGEON}_n^{\otimes 2}$ is d-collision free.*

Proof. We begin by showing that the "in particular" statement holds, assuming the first part of the theorem. Let $\tilde{\mathbb{E}}$ be the degree-D matching pseudoexpectation, and let $\mathcal{F}$ be a d-pairwise-witnessing family of matching terms for $\mathrm{PIGEON}_n^{\otimes 2}$. By Lemma 4.5, we may assume that $\mathcal{F}$ is a strong pairwise-witnessing family. Let F be the good matching DNF corresponding to $\mathcal{F}$. We note that since $\mathcal{F}$ is strongly pairwise-witnessing, it follows that F is good, and in particular that every pair of matchings $M_1 \neq M_2$ in F are incoherent by Observation 4.7. Hence, $\tilde{\mathbb{E}}[\mathcal{F}] = \tilde{\mathbb{E}}[F] \leq 1$ by assumption.

We now prove the first part of the theorem. Let F be a width $\leq d$ good matching DNF chosen such that $\tilde{\mathbb{E}}[F]$ is maximized, and let $\mathcal{H}$ be the set of matchings associated with F. We will show that $\tilde{\mathbb{E}}[F] \leq 1$. In the proof of this theorem, a *weighted* matching decision tree T is a matching decision tree where each leaf ℓ is labelled with a real weight $w(\ell) \geq 0$. Given such a decision tree, we define $w(T) := \sum_{\ell \in L(T)} w(\ell)$. Our overall goal is to construct a weighted matching decision tree T such that

1. T is composed only of pigeon queries.
2. For each leaf ℓ of T, $w(\ell) \leq \tilde{\mathbb{E}}[\pi(\ell)]$.
3. $\tilde{\mathbb{E}}[F] \leq w(T)$.

The proof is complete once we have such a tree, since properties (1) and (2) imply that $w(T) \leq 1$ by an easy induction over the depth of the tree, and combining this with property (3) yields the theorem.

We first describe the construction of the tree. At a high level, we follow the construction of $\mathsf{Can}(F)$, except we "skip" the hole queries and "freeze" the matchings that would have participated in them instead, removing them from consideration in future queries of the algorithm. Formally, the construction of the tree T proceeds by the following query algorithm. Throughout the construction of T, we maintain the following data.

- A node u that is currently visited.
- A set $\mathcal{L} \subseteq \mathcal{H}$ of *live* matchings.
- A set $\mathcal{R} \subseteq \mathcal{H}$ of *frozen* matchings.

For any node u in the decision tree, let $\pi(u)$ denote the matching obtained by following the path in the tree from the root to u. Just as in the proof of Lemma 4.12, for any set of matchings $\mathcal{S}$ and any node u in the decision tree, define

$$\mathcal{S}_u := \{M\pi(u) : M \in \mathcal{S} \text{ and } M, \pi(u) \text{ are coherent}\}.$$

Initially, $\mathcal{L} = \mathcal{H}$ is the set of matchings associated with F, $\mathcal{R} = \emptyset$, and $u = r$ is the root node of the decision tree. Throughout, we maintain the important invariant that $\mathcal{L} \cap \mathcal{R}$ are disjoint, and that all matchings in $\mathcal{L} \cup \mathcal{R}$ are coherent with the matching $\pi(u)$ throughout the construction. This invariant is clearly satisfied at the beginning of the construction.

The construction proceeds in stages. At the beginning of a stage, we have arrived at node u with sets of matchings $\mathcal{L}, \mathcal{R}$. If $\mathcal{L} = \emptyset$ or if $\mathcal{L} = \{M\}$ such that $M \subseteq \pi(u)$, then we halt and label the leaf with the weight $\tilde{\mathbb{E}}[\mathcal{L}_u] + \tilde{\mathbb{E}}[\mathcal{R}_u]$. Otherwise, we pick any matching $M \in \mathcal{L}$, and write $M = \{p_1 \mapsto h_1, \ldots, p_t \mapsto h_t\}$, noting that $t \leq d$. Let $\mathcal{I} \subseteq [t]$ be the set of indices of pigeons in M that have not yet been queried. Next, we repeat the following for each index $i \in \mathcal{I}$. We query p_i, receiving some hole h in response, and remove all matchings from $\mathcal{L} \cup \mathcal{R}$ that are incoherent with this query. Afterwards, if $h \neq h_i$, then we take all matchings $N \in \mathcal{L}$ that contain the hole h_i and *freeze* them, deleting them from $\mathcal{L}$ and placing them in $\mathcal{R}$. We call these matchings *newly frozen* at this query. Once all indices have been queried in this way, we finish this stage, and continue the construction recursively. This completes the description of the weighted tree T.

By definition, the tree T is composed only of pigeon queries. Therefore it suffices to prove the following claims.

Claim 1. $w(T) \geq \tilde{\mathbb{E}}[F]$.

Claim 2. For each leaf ℓ of T, $w(\ell) \leq \tilde{\mathbb{E}}[\pi(\ell)] \leq 1$.

Assuming these claims, we complete the proof of Theorem 4.15. By induction on the size of the subtree T_u, we argue that $w(T_u) \leq 1$ for all nodes $u \in T$. This will imply that $w(T) \leq 1$, and hence Claim 1 completes the proof of the theorem.

For the base case, if ℓ is a leaf of T, then $w(T_\ell) = w(\ell) \leq 1$ by Claim 2. Suppose that u is non-leaf node of T, and let p be the pigeon queried at u. Let c be the number of holes mentioned on the root-to-u path $\pi(u)$ which come from the same PIGEON instance as p, and let $v_1, \ldots, v_{n-c}$ be the children of u. By induction, $w(T_{v_i}) \leq 1$ for all $i \in [n-c]$. Hence

$$\tilde{\mathbb{E}}[T_u] = \sum_{i=1}^{n-c} \tilde{\mathbb{E}}[(p \mapsto h_i) \circ T_{v_i}] = \frac{1}{n-c} \sum_{i=1}^{n-c} \tilde{\mathbb{E}}[T_{v_i}] \leq \frac{n-c}{n-c} = 1,$$

which completes the induction. □

We prove these two claims by induction, but an interesting feature of the proof is that the first fact will be proven by induction from the bottom-up, while the second fact will be proven by induction from the top-down. Before launching into proving these facts, we make two simple observations. First, we observe that an easy induction over the construction of the tree proves that $\mathcal{L}_u \cup \mathcal{R}_u = \mathcal{H}_u$ for each node u. Second, we observe that the depth of T is at most the depth of $\mathsf{Can}(F)$, since we are simulating the execution of $\mathsf{Can}(F)$ except that we skip the hole queries and remove any matchings participating in those hole queries from further consideration, which only decreases the depth of T relative to $\mathsf{Can}(F)$. Therefore, by Lemma 4.14, the depth of T is at most $O(d^2)$.

Now, we begin by proving the first, and easier, claim. The proof of this claim is very similar to the proof of Lemma 4.12, and so we will only sketch some of the details for the sake of brevity.

Proof of Claim 1. For any node u in the tree, we prove by induction that $\tilde{\mathbb{E}}[\mathcal{H}_u] \leq w(T_u)$, where T_u is the subtree of T rooted at u. This implies the claim, since $\mathcal{H}_r = \mathcal{H}$ and $w(T_r) = w(T)$, where r is the root node.

As a base case, consider any leaf node ℓ of the tree. By the construction of T, the weight of $w(T_\ell)$ is defined to be $\tilde{\mathbb{E}}[\mathcal{L}_\ell] + \tilde{\mathbb{E}}[\mathcal{R}_\ell]$, where $\mathcal{L}$ and $\mathcal{R}$ are the sets of matchings when the leaf ℓ is visited. However, $\mathcal{L}_\ell \cup \mathcal{R}_\ell = \mathcal{H}_\ell$, and the proof of the inductive claim is complete.

Now, by induction, consider any internal node u of the tree. Suppose that p was the pigeon queried at this node, and let $v_1, v_2, \ldots, v_c$ be the child nodes of u with the edges labelled by $h_1, h_2, \ldots, h_c$. The inductive hypothesis implies that $w(T_{v_i}) \geq \tilde{\mathbb{E}}[L_{v_i}] + \tilde{\mathbb{E}}[R_{v_i}] = \tilde{\mathbb{E}}[\mathcal{H}_{v_i}]$. By a padding argument identical to the argument at the end of Lemma 4.12, we have $\tilde{\mathbb{E}}[\mathcal{H}_u] \leq \sum_{i=1}^{c} \tilde{\mathbb{E}}[\mathcal{H}_{v_i}] \leq \sum_{i=1}^{c} w(T_{v_i}) = w(T_u)$. This completes the proof of the claim. □

Now we proceed to the proof of the second, and more difficult, claim.

Proof of Claim 2. As we have observed above, at each node u we have $\mathcal{H}_u = \mathcal{L}_u \cup \mathcal{R}_u$. In particular, since $\mathcal{H}$ is a good family, $\mathcal{H}_u$ is also a good family, and so $\mathcal{L}_u \cup \mathcal{R}_u$ is a good family. Now, consider the leaf node ℓ. By the construction of T, since ℓ is a leaf we have that either $\mathcal{L}_\ell = \emptyset$ or $\mathcal{L}_\ell = \{\pi(\ell)\}$. If $\mathcal{L}_\ell = \{\pi(\ell)\}$, then $\mathcal{R}_\ell = \emptyset$, since $\mathcal{H}_\ell$ is a good family and every matching in $\mathcal{R}_\ell$ must extend $\pi(\ell)$ by definition. Thus, in this case,

$$w(\ell) = \tilde{\mathbb{E}}[\mathcal{L}_\ell] + \tilde{\mathbb{E}}[\mathcal{R}_\ell] = \tilde{\mathbb{E}}[\pi(\ell)] + 0 = \tilde{\mathbb{E}}[\pi(\ell)],$$

completing the proof.

Moving forward, we therefore assume that $\mathcal{L}_\ell = \emptyset$ and $\mathcal{R}_\ell \neq \emptyset$. This means that $w(\ell) = \tilde{\mathbb{E}}[\mathcal{R}_\ell]$, and thus we must show that $\tilde{\mathbb{E}}[\mathcal{R}_\ell] \leq \tilde{\mathbb{E}}[\pi(\ell)]$. To show this, we prove by induction that for each node u the inequality

$$\tilde{\mathbb{E}}[\mathcal{R}_u] \leq O\left(\frac{d^2}{n}\right) \mathrm{depth}(u) \cdot \tilde{\mathbb{E}}[\pi(u)] \tag{1}$$

holds. Let us briefly assume that the inequality holds. Then, since we have observed above that for any leaf ℓ, $\mathrm{depth}(\ell) = O(d^2)$, and we have assumed that $d^2 = o(n^{1/4})$, this inequality implies that $\tilde{\mathbb{E}}[\mathcal{R}_\ell] < \tilde{\mathbb{E}}[\pi(\ell)]$, which completes the proof of Claim 2 and the theorem.

As a base case, when $u = r$ we have $\mathcal{R}_r = \emptyset$, and thus $\tilde{\mathbb{E}}[\mathcal{R}_r] = 0$. Since $\mathrm{depth}(r) = 0$ the inequality trivially holds. Now, by induction, consider an internal node u and suppose that Equation (1) holds for $\mathcal{R}_u$. Let p be the pigeon queried at the node u, and suppose that u has c children denoted $v_1, \ldots, v_c$, corresponding to the holes $h_1, h_2, \ldots, h_c$. Let v_i be any child of u, and consider the set $\mathcal{R}_{v_i}$. We prove that

$$\tilde{\mathbb{E}}[\mathcal{R}_{v_i}] \leq \frac{\tilde{\mathbb{E}}[\mathcal{R}_u]}{c} + O\left(\frac{d^2}{n}\right) \tilde{\mathbb{E}}[\pi(v_i)],$$

and Equation (1) follows for v_i by some simple algebra.

First, we assume without loss of generality that every matching $M \in \mathcal{R}_u$ contains the pigeon p. If not, letting k denote the number of edges in M in the instance of $\textsc{Pigeon}_n$ containing p, we can replace M with $n-k$ new matchings $M_1, \ldots, M_{n-k}$, which are identical to M except we add an edge in M_i from p to the ith available hole. After replacement, the new family still has weight $\mathcal{R}_u$ and is good, and moreover the families $\mathcal{R}_{v_i}$ remain unchanged.

With this simplification in mind, observe that every matching $M \in \mathcal{R}_{v_i}$ either already lies in $\mathcal{R}_u$, or, it is a *newly frozen* matching. Let $\mathcal{R}'_{v_i} = \mathcal{R}_u \cap \mathcal{R}_{v_i}$ denote the set of matchings that were inherited from $\mathcal{R}_u$, and let $\mathcal{N}_{v_i}$ denote the set of newly frozen matchings. Using this notation, we can write $\mathcal{R}_{v_i} = \mathcal{R}'_{v_i} \cup \mathcal{N}_{v_i}$, and hence

$$\tilde{\mathbb{E}}[\mathcal{R}_{v_i}] = \tilde{\mathbb{E}}[\mathcal{R}'_{v_i}] + \tilde{\mathbb{E}}[\mathcal{N}_{v_i}].$$

We prove that $\tilde{\mathbb{E}}[\mathcal{R}'_{v_i}] \leq \tilde{\mathbb{E}}[\mathcal{R}_{v_i}]/c$ and $\tilde{\mathbb{E}}[\mathcal{N}_{v_i}] \leq O(d^2/n)\tilde{\mathbb{E}}[\pi(v_i)]$, which completes the proof of Equation (1).

Let us first prove that $\tilde{\mathbb{E}}[\mathcal{R}'_{v_i}] \leq \tilde{\mathbb{E}}[\mathcal{R}_u]/c$. Since every matching $M \in \mathcal{R}_u$ contains the pigeon p, it follows that $\mathcal{R}'_{v_1}, \mathcal{R}'_{v_2}, \ldots, \mathcal{R}'_{v_c}$ is a partition of $\mathcal{R}_u$. We claim that for each $a \neq b$, $\tilde{\mathbb{E}}[\mathcal{R}'_a] = \tilde{\mathbb{E}}[\mathcal{R}'_b]$. If this is true, then we are done, since then all c sets $\mathcal{R}'_{v_i}$ comprising the partition of $\tilde{\mathbb{E}}[\mathcal{R}_u]$ have equal weight, and so by averaging they must have weight $\tilde{\mathbb{E}}[\mathcal{R}_u]/c$. So, to see the claim, suppose by contradiction that there is a pair $a \neq b$ such that $\tilde{\mathbb{E}}[\mathcal{R}'_a] > \tilde{\mathbb{E}}[\mathcal{R}'_b]$. In this case, we will show how to modify $\mathcal{R}_u$ (and thus $\mathcal{H}$) to obtain another good family with even larger weight, contradicting the assumption that F is a good DNF with the maximum possible weight.

Given a matching π containing the edge $p \mapsto h_i$ and another hole $h_j \neq h_i$, define the new matching $\mathsf{swap}(\pi, p, h_i, h_j)$ by taking π and either removing the edge $p \mapsto h_i$ and adding $p \mapsto h_j$, if h_j has no pigeon mapping to it in π, or by swapping the pigeons mapping to holes h_i and h_j if h_j does have a pigeon mapping to it. With this, define the family

$$\mathcal{R}''_b = \left\{\mathsf{swap}(\pi, p, h_a, h_b) \mid \pi \in \mathcal{R}'_a\right\}$$

to be all swaps of matchings in $\mathcal{R}'_a$. Since every matching in $\mathcal{R}'_a$ contains the edge $p \mapsto h_a$, it follows that all matchings in $\mathcal{R}''_b$ are well defined, they all contain $p \mapsto h_b$, and also $\tilde{\mathbb{E}}[\mathcal{R}''_b] = \tilde{\mathbb{E}}[\mathcal{R}'_a] > \tilde{\mathbb{E}}[\mathcal{R}'_b]$. With this in mind, consider the set $\mathcal{R}'_u = (\mathcal{R}_u \setminus \mathcal{R}'_b) \cup \mathcal{R}''_b$. We immediately have that $\tilde{\mathbb{E}}[\mathcal{R}'_u] > \tilde{\mathbb{E}}[\mathcal{R}_u]$, and thus the weight has increased. Thus, if we prove that $\mathcal{R}'_u$ is also a good family, we will have contradicted the maximality of $\tilde{\mathbb{E}}[F]$, since it implies the maximality of $\tilde{\mathbb{E}}[\mathcal{R}_u]$.

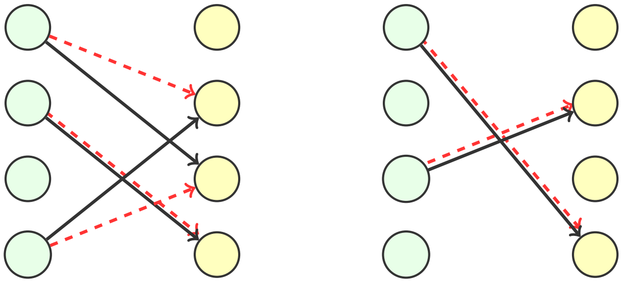

Figure 4: The swap operation applied to $(\pi, p_1^1, h_2^1, h_3^1)$. The original matching π is given in red, while the new matching $\mathsf{swap}(\pi, p_1^1, h_2^1, h_3^1)$ is given in black.

Consider any two distinct matchings $\pi, \sigma \in \mathcal{R}'_u$, and we prove that π and σ are either inconsistent or witnessing. We only need to consider the case where $\pi, \sigma \in \mathcal{R}''_b$, since if one of π or σ is not in $\mathcal{R}''_b$ then either they are inconsistent on the pigeon p, or, they are both contained in a set of the form $\mathcal{R}'_i$ for some $i \neq b$, which we already know to be a good family. So, suppose that $\pi = \mathsf{swap}(\pi', p, h_a, h_b)$ and $\sigma = \mathsf{swap}(\sigma', p, h_a, h_b)$ for some $\pi', \sigma' \in \mathcal{R}'_a$. Since $\pi \neq \sigma$ we immediately have that $\pi' \neq \sigma'$, and thus they must either be inconsistent or witness. Since π and σ are obtained by swapping $p \mapsto h_a$ with $p \mapsto h_b$, it follows by a case analysis that π and σ must be inconsistent or witness as well. This completes the proof that $\mathcal{R}'_u$ is a good family, and we obtain our final contradiction. Therefore, $\tilde{\mathbb{E}}[\mathcal{R}'_{v_i}] \leq \tilde{\mathbb{E}}[\mathcal{R}_u]/c$.

Finally, we prove that $\tilde{\mathbb{E}}[\mathcal{N}_{v_i}] \leq O(d^2/n)\tilde{\mathbb{E}}[\pi(v_i)]$. This is implied immediately by the next lemma, setting $\mathcal{M} = \mathcal{N}_{v_i}$, $\pi = \pi(v_i)$, and h is the hole used that witnesses the newly-frozen matchings.

Lemma 4.16. *Let n be any sufficiently large integer, and let $k \leq d = o(n^{1/8})$. Let π be a matching of size k on $\text{PIGEON}_n^{\otimes 2}$, and let h be a hole not appearing in π. Let $\mathcal{M}$ be any good family of size $\leq d$ matchings such that each matching in $\mathcal{M}$ contains both π and h. Then*

$$\tilde{\mathbb{E}}[\mathcal{M}] = O\left(\frac{d^2 \cdot \tilde{\mathbb{E}}[\pi]}{n}\right).$$

As discussed above, this lemma implies $\tilde{\mathbb{E}}[\mathcal{N}_{v_i}] \leq O(d^2/n)\tilde{\mathbb{E}}[\pi(v_i)]$, which completes the proof of Equation (1) and hence Claim 2. □

Proof of Lemma 4.16. Let $T = \mathsf{Can}(\mathcal{M})$, although, with the added assumption that the tree begins by first querying all of π, and then immediately afterwards it queries the hole h. Since every matching $M \in \mathcal{M}$ contains π, any leaf of T not consistent with π is set to 0. Let D be the depth of T, and note that $D = O(d^2)$. Let u be the first node in T consistent with π, and we note that the hole h is queried at u. Let $v_1, v_2, \ldots, v_c$ be the children of u, corresponding to the pigeons $p_1, p_2, \ldots, p_c$, and note that $n+1-k \leq c \leq n+1$. Observe that $\mathcal{M} = \mathcal{M}_u$, and since every matching in $\mathcal{M}$ contains the hole h, it follows that $\mathcal{M}_{v_1}, \mathcal{M}_{v_2}, \ldots, \mathcal{M}_{v_c}$ is a partition of $\mathcal{M}$. For each $i = 1, \ldots, c$, define $\mathcal{M}'_i = \{\pi(\ell) : \ell \in L_1(T_{v_i})\}$ to be the set of matchings obtained by taking root-to-leaf paths in the subtree rooted at v_i, and let $\mathcal{M}' = \bigcup_{i=1}^c \mathcal{M}'_i$.

Before we bound the weight of $\mathcal{M}$, we make a key observation about the family $\mathcal{M}'$.

Key Observation. If $\tau \in \mathcal{M}'_i$ is any matching, then there is a set of $t > c - D + 1$ indices $I_\tau \subseteq \{1, 2, \ldots, c\}$ such that for each $j \in I_\tau$ and each $\tau' \in \mathcal{M}'_j$, the matchings τ and τ' are incoherent.

To prove this key observation, consider any $\sigma \in \mathcal{M}_{v_i}$ and any $\sigma' \in \mathcal{M}_{v_j}$. Since $\mathcal{M}$ is good, σ and σ' must be either inconsistent or witnessing. Write $\sigma = \pi(p_i \mapsto h)\tau$ and $\sigma' = \pi(p_j \mapsto h)\tau'$ (noting that

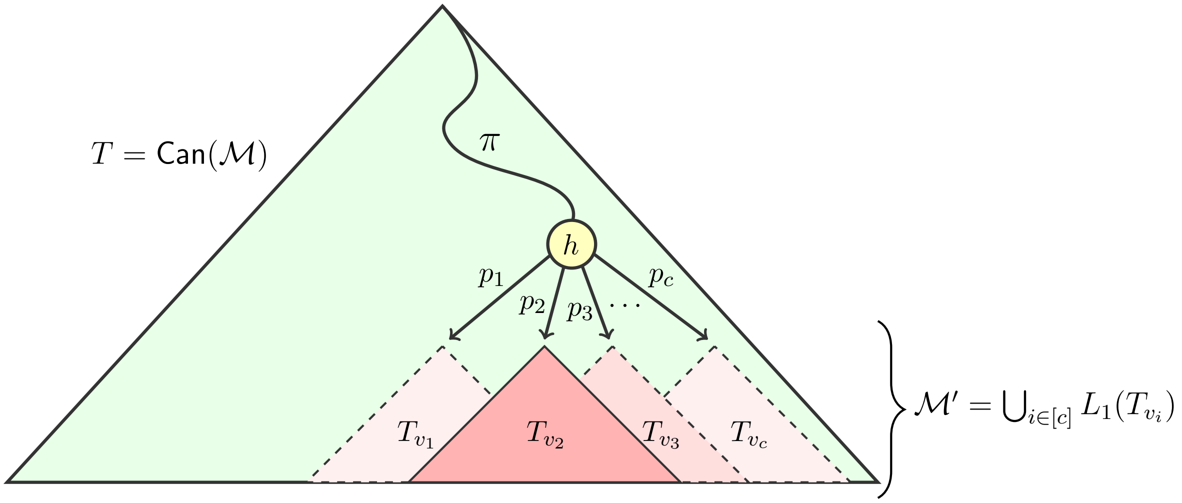
$T = \mathsf{Can}(\mathcal{M})$
π
h
p_1
p_2
p_3
$\cdots$
p_c
T_{v_1}
T_{v_2}
T_{v_3}
T_{v_c}
$\mathcal{M}' = \bigcup_{i\in[c]} L_1(T_{v_i})$

Figure 5: The family $\mathcal{M}'$ formed by taking the 1-paths of the sub-trees trees $T_{v_1}, \dots, T_{v_c}$ rooted at the children of a leaf of a path π in $\mathsf{Can}(\mathcal{M}')$.

$\tau \in \mathcal{M}_i'$ and $\tau' \in \mathcal{M}_j'$), and suppose that τ and τ' are coherent matchings. Then the only possibility is that either p_i occurs in τ' or p_j occurs in τ, since otherwise σ and σ' would not be inconsistent or witness. Since τ has at most $D-1$ edges in it, this can only occur for at most $c-D+1$ of the subtrees of u, proving the observation.

We now continue with the proof. Consider the following useful notation. If π and σ are coherent matchings, then we write

$$\tilde{\mathbb{E}}[\pi|\sigma] := \frac{\tilde{\mathbb{E}}[\pi\sigma]}{\tilde{\mathbb{E}}[\sigma]},$$

and refer to $\tilde{\mathbb{E}}[\pi|\sigma]$ as the *conditional* pseudoexpectation of π given σ. We extend the conditional notation to sets in the natural way: if $\mathcal{U}$ is a set of matchings coherent with π, then $\tilde{\mathbb{E}}[\mathcal{U}|\pi] := \sum_{\sigma\in\mathcal{U}} \tilde{\mathbb{E}}[\sigma|\pi]$. With this notation in hand, we can write

$$\begin{aligned}\tilde{\mathbb{E}}[\mathcal{M}] &= \tilde{\mathbb{E}}[\mathcal{M}_u] \\ &= \tilde{\mathbb{E}}[\pi] \sum_{i=1}^{c} \tilde{\mathbb{E}}[p_i \mapsto h|\pi] \tilde{\mathbb{E}}[\mathcal{M}_i'|\pi \circ (p_i \mapsto h)] \\ &= \frac{\tilde{\mathbb{E}}[\pi]}{c} \sum_{i=1}^{c} \tilde{\mathbb{E}}[\mathcal{M}_i'|\pi \circ (p_i \mapsto h)].\end{aligned}$$

We now prove the bound

$$\sum_{i=1}^{c} \tilde{\mathbb{E}}[\mathcal{M}_i'|\pi \circ (p_i \mapsto h)] \le D = O(d^2). \tag{2}$$

Once we have this bound, the proof is complete, since $c \ge n-k \ge n-d \ge 0.99n$ for sufficiently large n, assuming $d = o(n^{1/8})$. Plugging both of these bounds in yields

$$\tilde{\mathbb{E}}[\mathcal{M}] \le O\left(\frac{\tilde{\mathbb{E}}[\pi] d^2}{n}\right)$$

as desired.

To establish Equation (2) we build another weighted matching decision tree, T', which will bound the weight of $\sum_{i=1}^{c} \tilde{\mathbb{E}}[\mathcal{M}'_i | \pi \circ (p_i \mapsto h)]$. In particular, we will construct T' so that the following holds:

$$D \geq w(T') \geq \sum_{i=1}^{c} \tilde{\mathbb{E}}\big[\mathcal{M}'_i | \pi \circ (p_i \mapsto h)\big].$$

We build T' by constructing the canonical matching decision tree for $\mathcal{M}'$ (with some slight modifications described below). This will cover the weight of $\mathcal{M}'$. To instead cover the weight of $\sum_{i=1}^{c} \tilde{\mathbb{E}}[\mathcal{M}'_i | \pi \circ (p_i \mapsto h)]$, we re-weight the leaves. There is tension here: we would like to assign sufficiently large weight to the leaves so that we upper-bound $\sum_{i=1}^{c} \tilde{\mathbb{E}}[\mathcal{M}'_i | \pi \circ (p_i \mapsto h)]$, however we cannot exceed a total weight of D. By the Key Observation, we will show that at most $(D-1)$-many matchings can be coherent with any leaf ℓ of T', and so we should weight ℓ by $(D-1)$ times the weight of the matching $\pi(\ell)$. However, this is a huge overkill, as we haven't accounted for the conditional expectation — the expectation of each matching in $\mathcal{M}'_i$ is conditioned on the fact that we have queried the $(k+1)$-many holes and pigeons $\pi \circ (p_i \mapsto h)$. Therefore, we should actually weight ℓ by $(D-1)$ times the weight of $\pi(\ell)$ conditioned on $\pi \circ (p_i \mapsto h)$. However, the at most $(D-1)$-many matchings that are coherent with $\pi(\ell)$ may come from different $\mathcal{M}'_i$ and hence may be conditioned on different p_i. The key is that we only care about covering the weight of this family, and hence it only matters that we condition on a matching of size $k+1$. Hence, it suffices to weight each leaf by $(D-1) \cdot \tilde{\mathbb{E}}[\pi(\ell) | \pi \circ (p^* \mapsto h^*)]$ for some arbitrary p^*, h^* not queried in $\pi(\ell)$. We now formally describe T'.

Let $\text{PIGEON}^{\otimes 2} \restriction \pi$ be obtained by fixing the variables of $\text{PIGEON}^{\otimes 2}$ according to π. This decision tree will query pigeons and holes from $\text{PIGEON}^{\otimes 2} \restriction \pi$; that is, we will build T' assuming that π has already been queried. Note that the matchings in $\mathcal{M}'$ do not mention any pigeons or holes mentioned by π. Let $F_{\mathcal{M}'}$ be the matching DNF corresponding to $\mathcal{M}'$. We construct the weighted matching decision tree T' covering $F_{\mathcal{M}'}$ by the following modification of the construction of the canonical decision tree:

1. At each *recursive round*, in which we extend a leaf ℓ constructed in the tree so far by choosing a term in $F_{\mathcal{M}'} \restriction \pi(\ell)$ and query every pigeon and hole mentioned within it, the term of $F_{\mathcal{M}'} \restriction \pi(\ell)$ which is the chosen is the one with the largest width.
2. If we reach a leaf ℓ where $F_{\mathcal{M}'} \restriction \pi(\ell) = b \in \{0,1\}$, then if $b = 1$ we label the weight of this leaf ℓ with $w(\ell) := (D-1) \cdot \tilde{\mathbb{E}}[\pi(\ell) | \pi \circ (p^* \mapsto h^*)]$, where p^* and h^* are a pigeon and a hole which have not been queried by $\pi(\ell)$; this is to ensure that T' correctly covers the weight and so the particular names of p^* and h^* are immaterial. Otherwise, if $b = 0$ then we label it by $w(\ell) := 0$.

We stress that only pigeons and holes from $\text{PIGEON}^{\otimes 2} \restriction \pi$ are queried during this process.

The $(D-1)$ factor in the weight comes from the fact that Key Observation only guarantees that $\tau \in \mathcal{M}'_i$ is incoherent with matchings from at least $c - (D-1)$-many $\mathcal{M}'_j$. This differs from previous cases where we constructed canonical trees from good families (in which every pair of matchings is incoherent). Even so, the Key Observation still suffices to bound the depth.

Lemma 4.17. *T' satisfies the following:*

(i) T' has depth at most D^3

(ii) For each leaf node ℓ of T', there are at most $(D-1)$-many indices i such that $\mathcal{M}'_{i,\ell}$ is non-empty, and for each such i, $|\mathcal{M}'_{i,\ell}| = 1$.

Proof. The proof of (i) is similar to the proof of Lemma 4.14, however in that proof we relied on every pair of matchings in $\mathcal{M}'$ being incoherent. Now, $\mathcal{M}'$ only satisfies the following two weaker properties: (1) The Key Observation holds for every $\tau \in \mathcal{M}'$, and (2) for every $i \in [c]$, every pair of matchings in $\mathcal{M}'_i$

is incoherent. (2) holds because every pair of paths in a matching decision tree are incoherent, and each $\mathcal{M}'_i := \{\pi(\ell) : \ell \in L_1(T_{v_i})\}$. As we will see, the same general idea can still be carried out, however at the cost of an additional multiplicative factor of D.

Say that a matching $\tau \in \mathcal{M}'$ is *constant* under a matching γ if its corresponding matching term becomes constant after restricting by γ. That is, if γ and τ are incoherent or $\tau \subseteq \gamma$. The key to the proof (i) is the following technical claim.

Claim. Let ℓ_i be any leaf in the tree produced after i recursive rounds in the construction of T', let τ be the matching that is queried at the $(i+1)$-st recursive round, where $\tau \in \mathcal{M}'_t$ for some $t \in [c]$, and let ℓ_{i+1} be any leaf that is reached after querying τ. Then, for any $j \in I_\tau \cup \{t\}$ and $\tau' \in \mathcal{M}'_j$ such that $\tau \upharpoonright \pi(\ell)$ is not a constant, $|\tau' \setminus \pi(\ell_{i+1})| \leq |\tau' \setminus \pi(\ell_i)| - 1$.

Proof of Claim. In the $(i+1)$-st recursive round we construct the full matching decision tree over the nodes in $\tau \setminus \pi(\ell_i)$. Let $I_\tau \subseteq [c]$ be the set of indices guaranteed for τ by the Key Observation, and pick some $j \in I_\tau \cup \{t\}$ and any $\tau' \in \mathcal{M}'_j$ such that $\tau' \upharpoonright \pi(\ell)$ is not constant. We argue that after querying τ, the width of $\tau \upharpoonright \pi(\ell)$ drops by at least 1. To do so, it suffices to show that in the $(i+1)$-st round we query at least one edge of $\tau \setminus \rho(\ell_i)$.

Since $j \in I_\tau \cup \{t\}$, τ and τ' are incoherent, and since $|\tau' \upharpoonright \pi(\ell)| \neq 0$, τ' and $\pi(\ell)$ are coherent. Hence, $\tau \upharpoonright \pi(\ell)$ and $\tau' \upharpoonright \pi(\ell)$ are incoherent, and so they have at least one pigeon or hole in common. Therefore, after querying the nodes of $\tau \upharpoonright \pi(\ell)$ in the $(i+1)$-st round, $|\tau' \setminus \pi(\ell_{i+1})| \leq |\tau' \setminus \pi(\ell_i)| - 1$. □

Using this claim, we prove (i) by induction: for $i = 1, \ldots, D$, we argue that if we have reached a node ℓ after iD-many recursive rounds in the construction of T', the width of every matching in $\mathcal{M}'$ has dropped by at least i. That is, for every $\tau' \in \mathcal{M}'$, either $\tau' \upharpoonright \pi(\ell)$ is constant or $|\tau \setminus \pi(\ell)| \leq D - i$. This suffices to prove (i) as each matching has size at most D. For $y \in [D^2]$ we will denote by ℓ_y any leaf node which we have arrived at after y recursive rounds in the construction of T'.

Supposing that the inductive hypothesis holds at the end of the the $D(i-1)$-st recursive round, we prove that it holds at the end of the Di^{th} recursive round. Let $\tau \in \mathcal{M}'_t$ be the matching whose nodes were queried in the $(D(i-1)+1)$-st recursive round. By the Claim, for every $j \in I_\tau \cup \{t\}$ and every $\tau' \in \mathcal{M}'_j$ such that $\tau' \upharpoonright \pi(\ell_{(D(i-1)+1)})$ is not constant, we have that $|\tau' \setminus \pi(\ell_{(D-i)+1})| \leq D - i$. That is, the width of all of the matchings in the families indexed by $I_\tau \cup \{t\}$ has dropped by at least 1.

It remains to argue that the width of all matchings in $\mathcal{M}'_k$ for $k \in [c] \setminus (I_\tau \cup \{t\})$ drops by at least 1 after $D - 1$ additional rounds. To see that this happens, note that construction of T' always chooses the matching in $\mathcal{M}'$ with the largest width (under the current restriction) to query. Therefore, if there are any matchings of width $D - i + 1$, they will be queried next. The Claim ensures that we only need to query at most $D - 1$ matchings before they all have width at most $D - i$. Indeed, By the claim, whenever we query some $\tau \in \mathcal{M}'_z$ for some $z \in [c] \setminus (\tau \cup \{t\})$, the width of all non-constant matchings in $\mathcal{M}'_z$ drops by at least 1. Hence, the width of all non-constant matchings in $\mathcal{M}'$ drops to $\leq D - i$ after $|[c] \setminus I_\tau| = c - (c - (D+1)) = D - 1$ additional rounds, completing the induction and the proof of (i).

To prove (ii), let ℓ be a leaf of T' and recall that

$$\mathcal{M}'_{i,\ell} := \{\tau\pi(\ell) : \tau \in \mathcal{M}'_i \text{ and } \tau, \pi(\ell) \text{ are coherent}\}.$$

By the construction of T', every $\tau \in \mathcal{M}$ must either be contained in, or incoherent with $\pi(\ell)$, as otherwise we would have queried τ, contradicting that ℓ is a leaf. Let $I := \{i \in [s] : \mathcal{M}'_{i,\ell} \neq \emptyset\}$ and consider any τ in some $\mathcal{M}'_i$ such that $\tau \subseteq \pi(\ell)$. By the Key Observation, there is a set $I_\tau \subseteq [c]$ with $|I_\tau| \geq c - D + 1$ such that τ is incoherent with every $\tau' \in \mathcal{M}'_j$ for every $j \in I_\tau$. Hence, as $\tau \subseteq \pi(\ell)$ we have $I_\tau \cap I = \emptyset$, and so $|I| = c - |I_\tau| \leq D + 1$.

Finally, suppose that there is some $i \in I$ such that $|\mathcal{M}'_{i,\ell}| > 1$. Then there are $\tau, \tau' \in \mathcal{M}'$ such that $\tau, \tau' \subseteq \pi(\ell)$. This contradicts that every pair of matchings in $\mathcal{M}'_i$ are incoherent. □

In the remainder we will argue that

$$D \geq w(T') \geq \sum_{i=1}^{c} \tilde{\mathbb{E}}\big[\mathcal{M}'_i | \pi \circ (p_i \mapsto h)\big],$$

which completes the proof of Equation (2), and hence Claim 2.

First, we show that $w(T') \geq \sum_{i=1}^{c} \tilde{\mathbb{E}}[\mathcal{M}'_i | \pi \circ (p_i \mapsto h)]$. This follows from an almost identical induction to what we have done before (e.g. in Lemma 4.12), albeit with a slight twist in the base case. Let v be any node in the tree T', and we prove by induction that

$$w(T'_v) \geq \sum_{i=1}^{c} \tilde{\mathbb{E}}\big[\mathcal{M}'_{i,v} | \pi \circ (p_i \mapsto h)\big].$$

For the base case, let $v = \ell$ be any leaf node of the tree T' and let $w(\ell) = (D-1) \cdot \tilde{\mathbb{E}}[\pi(\ell) | \pi \circ (p^* \mapsto h^*)]$ be its weight, for some p^*, h^* not queried in $\pi(\ell)$ or π. By Lemma 4.17, there are at most $D-1$ indices i such that $\mathcal{M}'_{i,\ell}$ is non-empty. For each of these indices, there will be a unique matching $M \in \mathcal{M}'_{i,\ell}$, and $M \subseteq \pi(\ell)$. Let $M_1, \ldots, M_z$ be these matchings, for $z \leq D-1$, then

$$\begin{aligned}
\sum_{i=1}^{c} \tilde{\mathbb{E}}\big[\mathcal{M}'_{i,\ell} | \pi \circ (p_i \mapsto h)\big] &= \sum_{i=1}^{z} \tilde{\mathbb{E}}\big[M_i | \pi \circ (p_i \mapsto h)\big] \\
&= \sum_{i=1}^{z} \tilde{\mathbb{E}}\big[M_i | \pi \circ (p^* \mapsto h^*)\big] && (\text{As } M_i \subseteq \pi(\ell) \text{ and } p^*, h^* \notin \pi(\ell)) \\
&\leq (D-1)\tilde{\mathbb{E}}\big[\pi(\ell) | \pi \circ (p^* \mapsto h^*)\big] \\
&= w(\ell) = w(T_\ell).
\end{aligned}$$

The inductive case follows identically to the inductive case in the proof of Claim 1 above and in the proof of Lemma 4.12. Hence, $w(T') \geq \sum_{i=1}^{c} \tilde{\mathbb{E}}[\mathcal{M}'_i | \pi \circ (p_i \mapsto h)]$.

Next, we argue that $D \geq w(T')$. We will assume that T' satisfies the following properties:

1. It is composed entirely of hole queries,
2. All of the hole queries in T' and π come from the same PIGEON instance,
3. Every leaf of T has depth exactly D^3.

We claim that this assumption is without loss of generality, as it can only increase the weight of T'. As we have made use of similar properties throughout the paper, we only sketch the proof. Consider a leaf ℓ in the construction of T^*, and let k be the number of queries made to, say, the first PIGEON instance. To see that (1) holds, observe that hole queries are the only queries which increase the weight of the tree. Indeed, if we extend $\pi(\ell)$ by a pigeon query to the first PIGEON instance then the total weight of the children of ℓ becomes $w(\pi(\ell)) \cdot \sum_{i=1}^{n-t} \frac{1}{n-t} = w(\pi(\ell))$, whereas if we extended it by a hole query the weight becomes $w(\pi(\ell)) \cdot \sum_{i=1}^{n-t+1} \frac{1}{n-t} = w(\pi(\ell)) \cdot (1 + \frac{1}{n-t})$. To see (2), observe that the weight-gain $1 + \frac{1}{n-t}$ of a hole query increases with t, the number of queries made to the same PIGEON-instance. Finally, (3) holds as any leaf of depth $< D^3$ can be extended by additional hole queries to increase the weight of T^*.

Hence we assume (1) – (3). We have already occupied k pigeon to holes due to π and every path in T' has depth exactly D^3. This means that there are at most $\prod_{i=0}^{D^3-1}((n+1)-(k+i))$ root-to-leaf paths in T'. Since each path has depth exactly D^3, it follows that the weight of each leaf is bounded above by $(D-1)\prod_{i=0}^{D^3-1}(n-(k+1+i))^{-1}$, since each leaf must match the same number of pigeons to holes, and each leaf is labelled with with $(D-1)\tilde{\mathbb{E}}\big[\pi(\ell)|\pi \circ (p^* \mapsto h^*)\big]$. Note that this "$+1$" comes from the additional condition that $(p^* \mapsto h^*)$. Therefore, the total weight of this tree is

$$\begin{aligned}
W(T') &\le (D-1)\prod_{i=0}^{D^3-1}\frac{n+1-(k+i)}{n-(k+1+i)}\\
&= (D-1)\left(\frac{n+1-k}{n-(k+D^3-1)}\right)\left(\frac{n-k}{n-(k+D^3)}\right)\\
&= (D-1)\left(1+\frac{D^3}{n+1-k-D^3}\right)\left(1+\frac{D^3}{n-k-D^3}\right)\\
&\le (D-1)\left(1+\frac{D^3}{n-k-D^3}\right)^2
\end{aligned}$$

As $k \le d$, $D = O(d^2)$, and $d = o(n^{1/8})$, $D^3/(n-k-D^3) \le 2/n$, for sufficiently large n. Hence,

$$w(T') \le (D-1)\left(1+\frac{D^3}{n-k-D^3}\right)^2 \le (D-1)\left(1+\frac{2}{n}\right)^2 \le (D-1)+1 = D.$$

This completes the proof of Lemma 4.16. □

We note that Lemma 4.16 is where our technique would break down if one tried to use it to "prove" $\textsc{Pigeon}_n^{2n} \otimes \textsc{Pigeon}_n^{2n}$ does not reduce to $\textsc{Pigeon}_n^{2n}$. (In particular, the upper bound in Equation (2) fails, as in the case of hole queries the number of leaves will grow very quickly relative to the weight of the conditional pseudoexpectation.)

5 Lower Bounds for $\textsc{Pigeon}^{\otimes 2}$ Generalizations

In this section we extend the argument from the previous section in order to prove Theorem 1.6.

Theorem 1.6. *For all constant k,* $\textsc{Pigeon}^{\otimes k}$ *is not black-box reducible to* $\textsc{Pigeon}^{\otimes k-1}$.

To prove this, one might hope to directly generalize the proof of Theorem 4.1. However, it is not clear how to construct a low-depth matching decision tree which represents the natural extension of d-pairwise witnessing families to the setting where we have k instances of $\textsc{Pigeon}$. Instead, we will give an inductive argument. We would like to argue that if we have a reduction from $\textsc{Pigeon}^{\otimes k+1}$ to $\textsc{Pigeon}^{\otimes k}$, then this implies that $\textsc{Pigeon}^{\otimes k}$ reduces to $\textsc{Pigeon}^{\otimes k-1}$, and hence we would contradict our main theorem that $\textsc{Pigeon}^{\otimes 2}$ does not reduce to $\textsc{Pigeon}$. A natural approach is to find a restriction which we could apply to the $\textsc{Pigeon}^{\otimes k}$-formulation of $\textsc{Pigeon}^{\otimes k+1}$ that would leave us with a $\textsc{Pigeon}^{\otimes k-1}$-formulation of $\textsc{Pigeon}^{\otimes k}$. To do so, it suffices to pick one of the k instances of $\textsc{Pigeon}$ from $\textsc{Pigeon}^{\otimes k}$-formulation and try to find a pair of pigeons i, j and paths π_1, π_2 in their decision trees such that under $\pi_1\pi_2$, the pigeons i and j fly to the same hole (thus fixing one of the collisions of $\textsc{Pigeon}^{\otimes k}$, reducing it to $\textsc{Pigeon}^{\otimes k-1}$) and such that $\pi_1\pi_2$ witnesses at most one collision in $\textsc{Pigeon}^{\otimes k+1}$. It turns out that we can guarantee the existence of such a pair $\pi_1\pi_2$ by proving a slightly strengthened version of our $\textsc{Pigeon}^{\otimes 2}$ lower-bound. We now introduce this strengthened version.

Definition 5.1. The $\textsc{Pigeon}^{(\otimes k,2)}$ problem is defined as follows. The input is k functions $f_1, \ldots, f_k : [n+1] \to [n+1]$. The output is a solution to $\textsc{Pigeon}$ on *any two* of the k sub-instances of $\textsc{Pigeon}$.

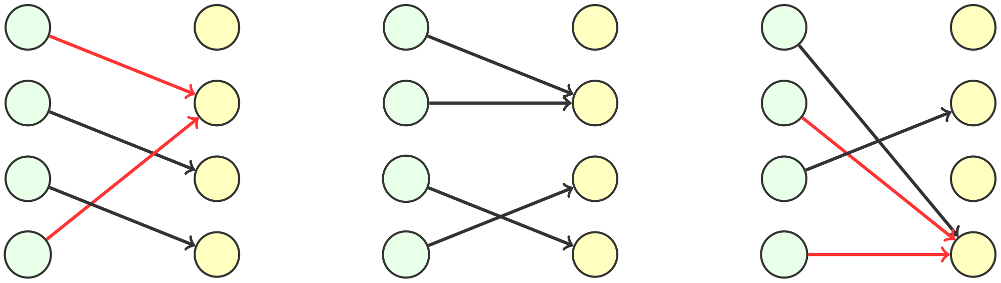

Figure 6: An instance of $\text{PIGEON}^{(\otimes 3,2)}$. A solution $(1(2,4), 3(2,4))$ is indicated by the red edges.

Note that $\text{PIGEON}^{\otimes 2} = \text{PIGEON}^{(\otimes 2,2)}$, in the above notation. Next we define our matching pseudoexpectation for k copies of PIGEON, generalizing our earlier definition of pseudoexpectation for just two copies of PIGEON. Recall that a *matching term* is just a conjunction of the variables such that no collision between pigeons is witnessed and no pigeon is mapped to hole 1 in any of the instances.

Definition 5.2. The *degree-d, k-matching pseudodistribution* is defined as follows. Given k sets of pigeons $P_1, P_2, \dots, P_k$ from the k subinstances comprising $\text{PIGEON}_n^{\otimes k}$ such that $|\bigcup_{i=1}^k P_i| \le d$, the distribution $\mathcal{D}_{P_1,\dots,P_k}$ samples a uniformly random matching from P_i pigeons to $|P_i|$ holes in $[n+1] \setminus \{1\}$ for each $i \in [k]$. Formally, given k matching terms M_i of the P_i pigeons to holes, $i \in [k]$, the corresponding pseudoexpectation is defined to be

$$\tilde{\mathbb{E}}\left[\prod_{i\in[k]} M_i\right] := \prod_{i_1=0}^{|P_1|-1} \frac{1}{n-i_1} \prod_{i_2=0}^{|P_2|-1} \frac{1}{n-i_2} \cdots \prod_{i_k=0}^{|P_k|-1} \frac{1}{n-i_k},$$

and extended to all matching terms by linearity.

By an analogous argument to the proof of Lemma 4.4, this is a pseudo-expectation for $\text{PIGEON}^{\otimes k}$ and $\text{PIGEON}^{(\otimes k,2)}$. By following the proof of Theorem 4.15, we can show the following generalization (we defer the argument to the end of the section).

Theorem 5.3. *Let k be any constant, let $D = n/2$ and $d = o(n^{1/4})$. The degree-D, k-matching pseudoexpectation $\tilde{\mathbb{E}}$ is d-collision free for $\text{PIGEON}_n^{(\otimes k,2)}$.*

Using this theorem we now prove Theorem 1.6.

Proof of Theorem 1.6. We will proceed by induction on k, using our separation of $\text{PIGEON}^{\otimes 2}$ from PIGEON (Theorem 4.1) as the base case. Throughout this proof, we will write $A \le B$ to denote when there is a $\text{poly}(\log(n))$-complexity B-formulation of problem A. Our inductive hypothesis is the following:

Inductive Hypothesis. For any m and any $M = m^{\text{poly}(\log(m))}$, $\text{PIGEON}_m^{\otimes k} \not\le \text{PIGEON}_M^{\otimes k-1}$.

Assume the inductive hypothesis for k, and suppose by way of contradiction that there is a $\text{PIGEON}_N^{\otimes k}$-formulation of $\text{PIGEON}_n^{\otimes k+1}$ for some $N = n^{\text{poly}(\log(n))}$ and decision trees of depth $d = \text{poly}(\log(n))$. We write this assumption as

$$\text{PIGEON}_n^{\otimes k+1} \le \text{PIGEON}_N^{\otimes k}. \tag{3}$$

Let $T_1^1, \dots, T_{N+1}^1$ be the decision trees defining the first PIGEON_N-instance in $\text{PIGEON}_N^{\otimes k}$, where T_i^1 specifies to which of the holes the i^{th} pigeon flies.

For any decision tree T, let $L(T)$ denote the leaves of T, and for any $\ell \in L(T)$, let C_ℓ denote the conjunction of literals on the path to ℓ. For any hole $h \in [N+1]$, define the family

$$\mathcal{H}_h := \bigcup_{i=1}^{N+1} \{C : C = C_\ell \text{ for some } h\text{-leaf } \ell \in L(T_i^1)\}.$$

As in the proof of Theorem 3.6, we have

$$N+1 = \sum_{i=1}^{N+1} \sum_{\ell \in L(T_i^1)} \tilde{\mathbb{E}}[C_\ell] = \sum_{h=1}^{N+1} \sum_{C \in \mathcal{H}_h} \tilde{\mathbb{E}}[C],$$

since we are summing over all of the paths in each decision tree. Thus, by averaging, there must be some hole $h \in [N+1]$ with $\tilde{\mathbb{E}}[\mathcal{H}_h] := \sum_{C \in \mathcal{H}_h} \tilde{\mathbb{E}}[C] \geq 1 + 1/N$. We now break into cases, depending on the hole h and the structure of $\mathcal{H}_h$, extracting a $\textsc{Pigeon}^{\otimes k-1}$-formulation of $\textsc{Pigeon}^{\otimes k}$ in each of them. We will refer to the $\textsc{Pigeon}$ instances comprising $\textsc{Pigeon}^{\otimes k}$ as *sub-instances*.

Case 1. $\tilde{\mathbb{E}}[\mathcal{H}_h] \geq 1 + 1/N$ for $h = 1$. Since $h = 1$, restricting along any matching term $M \in \mathcal{H}_1$ forces some pigeon i in the first sub-instance of $\textsc{Pigeon}_N^{\otimes k}$ to fly to hole 1. This means that

$$\textsc{Pigeon}_N^{\otimes k} \restriction M \leq \textsc{Pigeon}_N^{\otimes k-1},$$

since we can just map pigeon i to hole 1 in the first instance, and ignore the other pigeons in $\textsc{Pigeon}_N^{\otimes k}$ that were mapped under the restriction M. On the other hand, as M is a matching of at most $d = \mathrm{poly}(\log(n))$ pigeons of $\textsc{Pigeon}_n^{\otimes k+1}$ to holes, restricting by M leaves us with an instance of $\textsc{Pigeon}_n^{\otimes k+1}$ that is no easier than $\textsc{Pigeon}_{0.99n}^{\otimes k+1}$, and therefore $\textsc{Pigeon}_n^{\otimes k+1} \restriction M \geq \textsc{Pigeon}_{0.99n}^{\otimes k+1}$. Since $\textsc{Pigeon}_{0.99n}^{\otimes k} \leq \textsc{Pigeon}_{0.99n}^{\otimes k+1}$ trivially, we therefore have

$$\textsc{Pigeon}_{0.99n}^{\otimes k} \leq \textsc{Pigeon}_{0.99n}^{\otimes k+1} \leq \textsc{Pigeon}_n^{\otimes k+1} \restriction M.$$

Composing these reductions together with Equation (3), we have

$$\textsc{Pigeon}_{0.99n}^{\otimes k} \leq \textsc{Pigeon}_N^{\otimes k-1},$$

contradicting the inductive hypothesis.

Case 2. $\tilde{\mathbb{E}}[\mathcal{H}_h] \geq 1 + 1/N$ **for some** $h \neq 1$**.** We first assume that there are matching terms $M_1, M_2 \in \mathcal{H}_h$ such that $M_1 M_2 \neq 0$, and they witness a collision in *at most one* of the $k+1$ sub-instances comprising $\textsc{Pigeon}_n^{\otimes k+1}$. As $M_1 M_2 \neq 0$, M_1 and M_2 must have come from different trees T_i^1 and T_j^1 for $i \neq j$, and therefore restricting by $M_1 M_2$ maps the i^{th} and j^{th} pigeon of the first sub-instance of $\textsc{Pigeon}_N^{\otimes k}$ to hole h. Again, this means that the resulting instance $\textsc{Pigeon}_N^{\otimes k} \restriction M_1 M_2$ reduces efficiently to $\textsc{Pigeon}_N^{\otimes k-1}$, and we therefore have

$$\textsc{Pigeon}_N^{\otimes k} \restriction M_1 M_2 \leq \textsc{Pigeon}_N^{\otimes k-1}.$$

Our goal now is to prove that

$$\textsc{Pigeon}_{0.99n}^{\otimes k} \leq \textsc{Pigeon}_n^{\otimes k+1} \restriction M_1 M_2. \tag{4}$$

Combining the above two reductions with Equation (3) yields $\textsc{Pigeon}_{0.99n}^{\otimes k} \leq \textsc{Pigeon}_N^{\otimes k-1}$, a contradiction to the inductive hypothesis.

Suppose first that M_1 and M_2 are coherent, meaning that M_1M_2 is a matching. In this case, if we restrict $\text{PIGEON}_n^{\otimes k+1}$ by M_1M_2, we do not witness any collision, but, we have mapped $2d = O(\text{poly}(\log(n)))$ pigeons to holes. The resulting instance is therefore no easier than $\text{PIGEON}_{0.99n}^{\otimes k+1}$, and $\text{PIGEON}_{0.99n}^{\otimes k}$ trivially reduces to $\text{PIGEON}_{0.99n}^{\otimes k+1}$. This means that we have again constructed a reduction from $\text{PIGEON}_{0.99n}^{\otimes k}$ to $\text{PIGEON}_N^{\otimes k-1}$, proving Equation (4) and contradicting the induction hypothesis.

Next, suppose that M_1 and M_2 witness *exactly one* collision in $\text{PIGEON}_n^{\otimes k+1}$. Now, if we restrict $\text{PIGEON}_n^{\otimes k+1}$ by M_1M_2 we have a collision in one sub-instance, along with mapping $d = \text{poly}(\log(n))$ other pigeons to holes. The resulting instance is therefore no easier than $\text{PIGEON}_{0.99n}^{\otimes k}$. Therefore, we have shown $\text{PIGEON}_{0.99n}^{\otimes k} \le \text{PIGEON}_n^{\otimes k+1} \restriction M_1M_2$, again proving Equation (4) and finishing the proof of this case.

Finally, let us suppose that we are not in either of the above two subcases. This means that for all pairs $M_1, M_2 \in \mathcal{H}_h$, either M_1M_2 is inconsistent (meaning $M_1M_2 \equiv 0$), or, M_1M_2 witnesses *at least two* collisions among the $k+1$ sub-instances of $\text{PIGEON}_n^{\otimes k+1}$. In other words, $\mathcal{H}_h$ is a d-pairwise-witnessing family for $\text{PIGEON}^{(\otimes k+1,2)}$ with $\tilde{\mathbb{E}}[\mathcal{H}_h] \ge 1 + 1/N$. But Theorem 5.3 states that the k-matching pseudoexpectation $\tilde{\mathbb{E}}$ is d-collision-free (and collision-freeness by definition implies that the pseudoexpectation is at most 1), which is a contradiction. This completes the proof. □

It remains to prove Theorem 5.3; we only sketch the proof as it is an immediate generalization of the proof of Theorem 4.15.

Proof Sketch of Theorem 5.3. We will argue that $\tilde{\mathbb{E}}$ is d-collision free for $\text{PIGEON}_n^{(\otimes k,2)}$. The proof is identical to that of Theorem 3.6 and in what remains, we verify that the intermediate lemmas do indeed withstand the generalization to $\text{PIGEON}^{\otimes k}$. Intuitively this should hold as the only thing which has changed is that we are working with k instances of PIGEON_n, rather than two, however we are still only searching for a solution to two instances.

The definition of a matching term is identical to before, but now the conjunctions read variables from any of the k PIGEON sub-instances instead of just two. As before, a family $\mathcal{F}$ of matching terms is *d-pairwise witnessing* if for any $M_1, M_2 \in \mathcal{F}$, $M_1M_2 = 0$ or M_1M_2 witnesses a solution in at least two of the k sub-instances of $\text{PIGEON}^{(\otimes k,2)}$. It suffices to show that for any pairwise-witnessing family $\mathcal{F}$ that $\tilde{\mathbb{E}}[\mathcal{F}] \le 1$.

Lemma 4.5 shows that d-pairwise witnessing families are in fact *strong* pairwise witnessing. In our setting, what changes is that instead of knowing that for two matchings $M_1, M_2 \in \mathcal{F}$, that they pairwise witness two (out of two) PIGEON sub-instances after d additional queries, now M_1, M_2 will pairwise witness *some* pair of PIGEON sub-instances (out of the k sub-instances). If we restrict to that pair of sub-instances, the proof proceeds identically.

Next, we reduce any strong pairwise witnessing family $\mathcal{F}$, represented by a matching DNF F, to a matching decision tree which strongly represents F. The definition of matching decision trees, and the construction of the canonical matching decision tree for F readily generalizes to our setting. Indeed, each pigeon still has only $[n]$ holes to which it can fly and each hole has $[n+1]$ potential pigeons, the only difference is that there are k instances of PIGEON, rather than two. For Lemma 4.12, the induction argument depends only on the width of the matching DNF F instead of the number of PIGEON instances, and no change to the argument is necessary. Similarly, Lemma 4.14 works for any incoherent matching DNF.

Finally, Theorem 4.15, Lemma 4.16 and Lemma 4.17, and their subclaims also do not require any modification, as their proofs rely on properties that do not change in the $\text{PIGEON}^{(\otimes k,\,2)}$ setting: the definitions of conjunction width and incoherency, each pigeon can fly to at most n holes, and each hole can receive at most $n+1$ pigeons. □

6 Acknowledgements

Noah Fleming, Stefan Grosser, and Robert Robere were supported by NSERC. Toni Pitassi was supported by NSF AF:Medium 2212136. Part of this work was completed while the authors were attending the Simons Institute for the Theory of Computing in the Winter 2023 programs on "Meta-Complexity" and "Extended Reunion: Satisfiability". The authors thank Mika Göös and Susanna de Rezende for some helpful discussions.

References

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 166–175. IEEE Computer Society, 2004.

[Ajt88] Miklos Ajtai. The complexity of the pigeonhole principle. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 346–355, 1988.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108. ACM, 1996.

[BCE$^+$98] Paul Beame, Stephen Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. *Journal of Computer and System Sciences*, 57(1):3–19, 1998.

[BFH$^+$23] Romain Bourneuf, Lukáš Folwarczný, Pavel Hubácek, Alon Rosen, and Nikolaj I. Schwartzbach. Ppp-completeness and extremal combinatorics. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 22:1–22:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[BFI23] Sam Buss, Noah Fleming, and Russell Impagliazzo. TFNP characterizations of proof systems and monotone circuits. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 30:1–30:40. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[BIK$^+$92] Paul Beame, Russell Impagliazzo, Jan Krajícek, Toniann Pitassi, Pavel Pudlák, and Alan R. Woods. Exponential lower bounds for the pigeonhole principle. In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 200–220. ACM, 1992.

[BIK$^+$94] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. In *Proceedings of the 35th Symposium on Foundations of Computer Science (FOCS)*, pages 794–806, 1994.

[BJ12] Samuel R. Buss and Alan S. Johnson. Propositional proofs and reductions between NP search problems. *Annals of Pure and Applied Logic*, 163(9):1163–1182, 2012.

[BJP+19] Frank Ban, Kamal Jain, Christos H. Papadimitriou, Christos-Alexandros Psomas, and Aviad Rubinstein. Reductions in PPP. *Inf. Process. Lett.*, 145:48–52, 2019.

[BK94] Sam Buss and Jan Krajíček. An application of boolean complexity to separation problems in bounded arithmetic. In *Proceedings of the London Mathematical Society*, volume 69, pages 1–21, 1994.

[Bli14] H. F. Blichfeldt. A new principle in the geometry of numbers, with some applications. *Transactions of the American Mathematical Society*, 15(3):227–235, 1914.

[BM04] Joshua Buresh-Oppenheim and Tsuyoshi Morioka. Relativized NP search problems and propositional proof systems. In *Proceedings of the 19th IEEE Conference on Computational Complexity (CCC)*, pages 54–67, 2004.

[CDT09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56(3):14:1–14:57, 2009.

[CLRS13] Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large LP relaxations. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 350–359. IEEE Computer Society, 2013.

[Das19] Constantinos Daskalakis. Equilibria, fixed points, and computational complexity. In *Proceedings of the International Congress of Mathematicians (ICM)*. World Scientific, 2019.

[DGP09] Constantinos Daskalakis, Paul Goldberg, and Christos Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[DR23] Ben Davis and Robert Robere. Colourful TFNP and propositional proofs. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference, CCC 2023, July 17-20, 2023, Warwick, UK*, volume 264 of *LIPIcs*, pages 36:1–36:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[ER60] Paul Erdös and Richard Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society, Second Series*, 35:85–90, 1960.

[FKP19] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends in Theoretical Computer Science*, 14(1-2):1–221, 2019.

[GHJ+22] Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Separations in proof complexity and TFNP. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1150–1161. IEEE, 2022.

[GIPS21] Shafi Goldwasser, Russell Impagliazzo, Toniann Pitassi, and Rahul Santhanam. On the pseudo-deterministic query complexity of NP search problems. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 36:1–36:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[GKRS18] Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In *Proceedings of the 10th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 124, pages 38:1–38:19, 2018.

[GM08] Konstantinos Georgiou and Avner Magen. Expansion fools the Sherali-Adams system: Compromising local and global arguments. Technical report, University of Toronto, 2008.

[Hak85] Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.

[HV21] Pavel Hubácek and Jan Václavek. On search complexity of discrete logarithm. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, August 23-27, 2021, Tallinn, Estonia*, volume 202 of *LIPIcs*, pages 60:1–60:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[Jer16] Emil Jerábek. Integer factoring and modular square roots. *J. Comput. Syst. Sci.*, 82(2):380–394, 2016.

[JLRX23] Siddhartha Jain, Jiawei Li, Robert Robere, and Zhiyang Xun. Pigeonhole principle and Ramsey in TFNP. Manuscript in preparation., 2023.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 60–73. ACM, 2021.

[JPY88] David Johnson, Christos Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.

[KMR17] Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of csps. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 590–603. ACM, 2017.

[KNY19] Ilan Komargodski, Moni Naor, and Eylon Yogev. White-box vs. black-box complexity of search problems: Ramsey and graph property testing. *J. ACM*, 66(5):34:1–34:28, 2019.

[LRS15] James R. Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 567–576. ACM, 2015.

[Mor01] Tsuyoshi Morioka. Classification of search problems and their definability in bounded arithmetic. Master's thesis, University of Toronto, 2001.

[MP91] Nimrod Megiddo and Christos Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.

[Pap94] Christos Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.

[PPY23] Amol Pasarkar, Christos H. Papadimitriou, and Mihalis Yannakakis. Extremal combinatorics, iterated pigeonhole arguments and generalizations of PPP. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 88:1–88:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[Pud15] Pavel Pudlák. On the complexity of finding falsifying assignments for Herbrand disjunctions. *Archive for Mathematical Logic*, 54(7-8):769–783, 2015.

[Raz98] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Comput. Complex.*, 7(4):291–324, 1998.

[Raz01] Alexander A. Razborov. Proof complexity of pigeonhole principles. In Werner Kuich, Grzegorz Rozenberg, and Arto Salomaa, editors, *Developments in Language Theory, 5th International Conference, DLT 2001, Vienna, Austria, July 16-21, 2001, Revised Papers*, volume 2295 of *Lecture Notes in Computer Science*, pages 100–116. Springer, 2001.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93. ACM, 2005.

[SZZ18] Katerina Sotiraki, Manolis Zampetakis, and Giorgos Zirdelis. PPP-completeness with connections to cryptography. In *Proceedings of the 59th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 148–158, 2018.

A SOS Bounds for Nonadaptive Pigeon

For a clause $C = \bigvee_{i \in I} x_i \vee \bigvee_{j \in J} \neg x_j$, the conjunction $\overline{C}$ is $\prod_{i \in I}(1 - x_i) \prod_{j \in J} x_j$.

Definition A.1. Given an unsatisfiable CNF formula $F = C_1 \wedge \ldots \wedge C_m$, a *Sum-of-Squares* derivation of a polynomial p is given by a list of polynomials $p_1, \ldots, p_m, q_1, \ldots, q_k$ such that

$$\sum_{i \in [m]} \overline{C}_i p_i + \sum_{j \in [k]} q_j^2 = p,$$

where all operations are done in multilinear polynomial arithmetic (so $x_i^2 = x_i$) over $\mathbb{R}$. A Sum-of-Squares refutation of F is a derivation of the polynomial $-t$ for some $t > 0$. The *degree* of the proof is the maximum degree among the $\overline{C}_i p_i$ and q_j^2, and the degree required by Sum-of-Squares to refute F is the minimum degree of any Sum-of-Squares proof of F.

Let $[n]_0 = \{0, \ldots, n\}$. For ease of exposition we will work with n-ary variables $p_i^1, p_i^2 \in [n-1]_0$ encoding the holes to which the i^{th} pigeon of the first instance and the i^{th} pigeon of the second instance are sent to. These will each be represented by $\log n$ boolean variables $p_{i,\ell}^1, p_{i,\ell}^2$ for $\ell \in [\log n]$ representing the ℓ^{th} bit of p_i^1 and p_i^2 respectively. We will denote by $[\![p_i^1 \neq h]\!]$ the disjunction $(p_{i,1}^1)^{h_1} \vee \ldots \vee (p_{i,\log n}^1)^{h_{\log n}}$ stating that pigeon i of the first instance does not fly to hole h, where $h_1 \ldots h_{\log n}$ is the binary encoding of h, and $(p_i^1)^{h_i} = p_i^1$ if $h_i = 1$ and $\neg p_i^1$ if $h_i = 0$. As well, $[\![p_i^1 = h]\!]$ is the conjunction $\neg [\![p_i^1 \neq h]\!]$.

PIGEON$_n^{\otimes 2}$ is represented by the unsatisfiable CNF formula formed by the conjunction of the following clauses,

$$\begin{array}{lr} [\![p_i^1 \neq h]\!] \vee [\![p_j^1 \neq h]\!] \vee [\![p_k^2 \neq h']\!] \vee [\![p_\ell^2 \neq h']\!] & \forall i \neq j \in [n-1]_0, k \neq \ell \in [n-1]_0, h, h' \in [n-1]_0, \\ [\![p_i^\alpha \neq 0]\!] & \forall \alpha \in \{1,2\}, i \in [n-1]_0. \end{array}$$

Theorem A.2. *There is a $O(\log n)$-degree Sum-of-Squares proof of* PIGEON$_n^{\otimes 2}$.

We make some preliminary observations about our n-ary variables.

(a) $[\![p_i^\alpha = h]\!]^2 = [\![p_i^\alpha = h]\!]$, as $[\![p_i^\alpha = h]\!]$ is a conjunction and we are working in multilinear arithmetic.

(b) $\sum_{h\in[n]}[\![p_i^\alpha = h]\!] = 1$, as we are summing over all conjunctions over the variables $p_{i,j}^\alpha$ for $j \in [\log n]$.

The following claim will allow us to transform the clauses of $\textsc{Pigeon}_n^{\otimes 2}$ into a more useful form.

Claim A.3. There is a $O(\log n)$-degree Sum-of-Squares derivation of

$$1 - [\![p_i^1 = h]\!][\![p_j^1 = h]\!] - [\![p_k^2 = h']\!][\![p_\ell^2 = h']\!]$$

from $\textsc{Pigeon}_n^{\otimes 2}$ for every $h, h', i \neq j, k \neq \ell \in [n]$.

Proof. Consider the square polynomial

$$\begin{aligned}
&(1 - [\![p_i^1 = h]\!][\![p_j^1 = h]\!])^2(1 - [\![p_k^2 = h']\!][\![p_\ell^2 = h']\!])^2 \\
=&(1 - [\![p_i^1 = h]\!][\![p_j^1 = h]\!])(1 - [\![p_k^2 = h']\!][\![p_\ell^2 = h']\!]) && \text{(By (a))} \\
=&1 - [\![p_i^1 = h]\!][\![p_j^1 = h]\!] - [\![p_k^2 = h']\!][\![p_\ell^2 = h']\!] + [\![p_i^1 = h]\!][\![p_j^1 = h]\!][\![p_k^2 = h']\!][\![p_\ell^2 = h']\!] \\
=&1 - [\![p_i^1 = h]\!][\![p_j^1 = h]\!] - [\![p_k^2 = h']\!][\![p_\ell^2 = h']\!] && \text{(By the first axiom of }\textsc{Pigeon}_n^{\otimes 2}\text{)}
\end{aligned}$$

□

We now prove the theorem.

Proof of Theorem A.2. First, we derive that each pair of holes receives at most one pigeon. Next, we will derive that each pigeon goes to at least one hole. Summing over the $2n$ pigeons and $2(n-1)$ non-zero holes will complete the proof.

For a fixed $h, h' \in [n-1]$,

$$\begin{aligned}
&\Big(1 - \sum_{i\in[n-1]_0} [\![p_i^1 = h]\!]\Big)^2 + \Big(1 - \sum_{i\in[n-1]_0} [\![p_i^2 = h']\!]\Big)^2 \\
=&2 - \sum_{i\in[n-1]_0} [\![p_i^1 = h]\!] + \sum_{i\neq j}[\![p_i^1 = h]\!][\![p_j^1 = h]\!] - \sum_{i\in[n-1]_0} [\![p_i^1 = h']\!] + \sum_{i\neq j}[\![p_i^2 = h']\!][\![p_j^2 = h']\!] \\
= &1 - \sum_{i\in[n-1]_0} [\![p_i^1 = h]\!] - \sum_{i\in[n-1]_0} [\![p_i^2 = h']\!]. && \text{(By Claim A.3)}
\end{aligned}$$

Summing over all $h, h' \in [n-1]$, we get

$$\begin{aligned}
&2(n-1) - \sum_{h\in[n-1]}\sum_{i\in[n-1]_0} [\![p_i^1 = h]\!] - \sum_{h'\in[n-1]}\sum_{i\in[n-1]_0} [\![p_i^2 = h']\!] \\
=&2(n-1) - \sum_{h\in[n-1]_0}\sum_{i\in[n-1]_0} [\![p_i^1 = h]\!] - \sum_{h'\in[n-1]_0}\sum_{i\in[n-1]_0} [\![p_i^2 = h']\!], \qquad (5)
\end{aligned}$$

where the second line follows by the second axiom of $\textsc{Pigeon}^{\otimes 2}$.

On the other hand, by (a) $\sum_{h\in[n-1]_0}[\![p_i^\alpha = h]\!] - 1 = 0$, and so summing over all of the pigeons we obtain

$$\begin{aligned}
&\sum_{i\in[n-1]_0}\Big(\sum_{h\in[n-1]_0} [\![p_i^1 = h]\!] - 1\Big)^2 + \sum_{i\in[n-1]_0}\Big(\sum_{h\in[n-1]_0} [\![p_i^2 = h]\!] - 1\Big)^2 \\
= &\sum_{i\in[n-1]_0}\Big(\sum_{h\in[n-1]_0} [\![p_i^1 = h]\!] - 1\Big) + \sum_{i\in[n-1]_0}\Big(\sum_{h\in[n-1]_0} [\![p_i^2 = h]\!] - 1\Big) \\
= &\sum_{i\in[n-1]_0}\sum_{h\in[n-1]_0} [\![p_i^1 = h]\!] + \sum_{i\in[n+1]_0}\sum_{h\in[n-1]_0} [\![p_i^2 = h]\!] - 2n.
\end{aligned}$$

Adding this to (5) derives -2. □

We remark that Sum-of-Squares is also capable of proving more general *adaptive* problems in $\mathsf{FP}^{\mathsf{PPP}}$ such as the following: we are given k layers of instances of $\textsc{Pigeon}_n$, with one instance on the first layer, $\binom{n}{2}$ instance on the second, and so on. A solution (a pair of colliding pigeons) to an instance on the i^{th} layer points to a PIGEON instance on the $(i+1)^{st}$ layer to solve. The final solution consists of a sequence of k solutions, for the k PIGEON instances, one per layer, that are associated with the "path" from layer 1 to the last layer.