Electronic Colloquium on Computational Complexity, Report No. 17 (2024)

On Pigeonhole Principles and Ramsey in TFNP

Siddhartha Jain
UT Austin

Jiawei Li
UT Austin

Robert Robere
McGill

Zhiyang Xun
UT Austin

January 23, 2024

Abstract

The generalized pigeonhole principle says that if $tN + 1$ pigeons are put into N holes then there must be a hole containing at least $t + 1$ pigeons. Let t-PPP denote the class of all total NP-search problems reducible to finding such a t-collision of pigeons. We introduce a new hierarchy of classes defined by the problems t-PPP. In addition to being natural problems in TFNP, we show that classes in and above the hierarchy are related to the notion of multi-collision resistance in cryptography, and contain the problem underlying the breakthrough average-case quantum advantage result shown by Yamakawa & Zhandry (FOCS 2022).

Finally, we give lower bound techniques for the black-box versions of t-PPP for any t. In particular, we prove that RAMSEY is not in t-PPP, for any t that is sub-polynomial in $\log(N)$, in the black-box setting. Goldberg and Papadimitriou conjectured that RAMSEY reduces to 2-PPP, we thus refute it and more in the black-box setting. We also provide an ensemble of black-box separations which resolve the relative complexity of the t-PPP classes with other well-known TFNP classes.

1 Introduction

The theory of TFNP is the study of NP search problems which are *guaranteed* to have solutions. In most problems studied in the literature, this guarantee is usually due to a non-constructive combinatorial lemma [JPY88, Pap94]. Perhaps the most famous example is the Pigeonhole Principle, which defines the search problem PIGEON: given a mapping from $N + 1$ pigeons to N holes, find two pigeons that are in the same hole.

One perspective on the study of TFNP is as an attempt to understand how constructive these combinatorial principles are in relation to each other. For example, a natural problem that has so far resisted classification is the RAMSEY problem: given a graph on $N = 2^n$ vertices, find a clique or independent set of size $n/2$. Since the underlying combinatorial principle here comes from extremal combinatorics, it is natural to ask if there is a reduction to or from the Pigeonhole Principle, since it is the prototypical example of extremal reasoning. One way to formalize this question is to ask whether RAMSEY is in PPP, the TFNP subclass defined by polynomial-time reductions to the canonical problem PIGEON. This was formally conjectured by Goldberg and Papadimitriou [GP17].

Conjecture 1 (Goldberg and Papadimitriou [GP17])**.** RAMSEY is in PPP.

Obtaining a reduction from RAMSEY to PPP has been quite elusive. Recently, there has been much attention given to placing RAMSEY in any natural subclass of TFNP [KoT22, PPY23, BFH+23]. In particular, Pasarkar, Papadimitriou, and Yannakakis [PPY23] defined a novel new TFNP subclass

ISSN 1433-8092

called PLC, capturing the *iterative* use of the pigeonhole principle, and showed that RAMSEY was contained in it. They also showed that $\mathsf{PPP} \subseteq \mathsf{PLC}$, but it was unclear how much extra power PLC possesses over PPP.

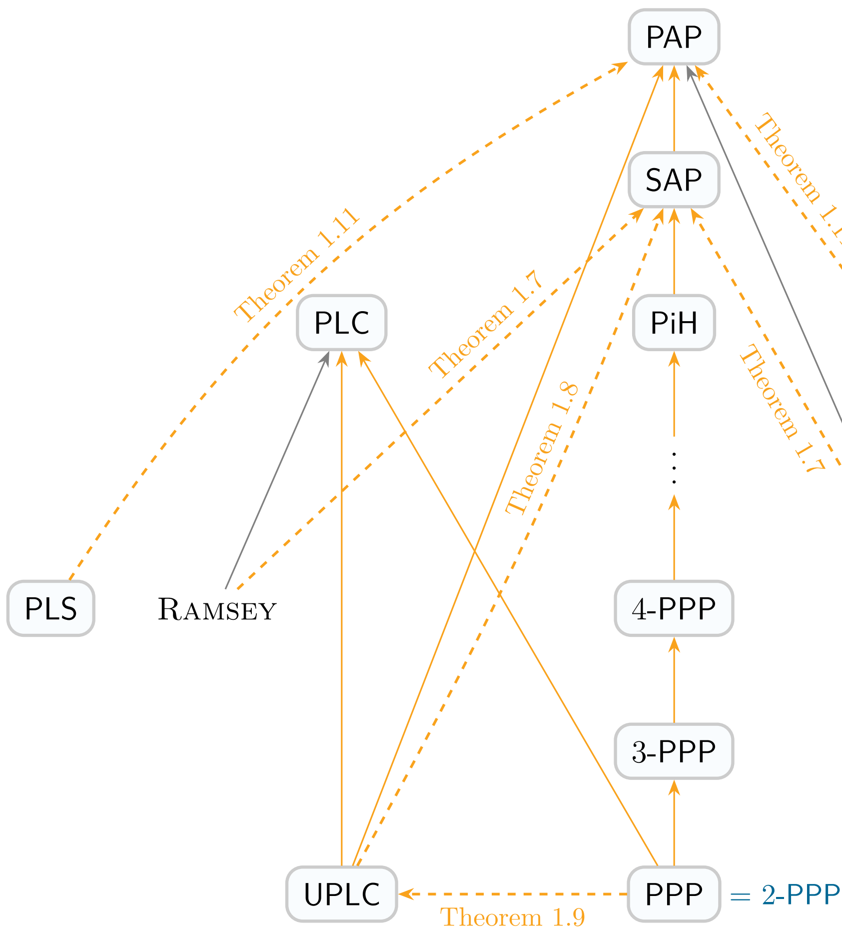

Figure 1: Complexity classes and problems defined by Generalized Pigeonhole Principles and Ramsey. An arrow $\mathsf{A} \to \mathsf{B}$ means $\mathsf{A} \subseteq \mathsf{B}$. An orange arrow $\mathsf{A} \to \mathsf{B}$ means $\mathsf{A} \subseteq \mathsf{B}$ and $\mathsf{B} \not\subseteq \mathsf{A}$ in the black-box setting. A dashed orange arrow $\mathsf{A} \dashrightarrow \mathsf{B}$ means $\mathsf{A} \not\subseteq \mathsf{B}$ in the black-box setting. All black-box separations in this figure are contributions of this work, labelled by the corresponding theorem.
We refer to the partial order of t-PPP and t-PWPP classes as the Pecking Order, while the union of t-PPP for all constant t is referred to as the Pigeon Hierarchy (PiH). We note that while our result BIRAMSEY $\notin \mathsf{SAP}$ (in the black-box setting) applies for the standard parameter regime, BIRAMSEY $\in \mathsf{PAP}$ is for a slightly smaller parameter.

1.1 Our Results

In this work, we initiate a systematic study on the computational complexity of the *generalized pigeonhole principles* in TFNP, and by which we gain more insights on the complexity of RAMSEY and related classes. The generalized pigeonhole principle corresponds to the following problem t-PIGEON$_N^M$: given a succinct encoding of a mapping from M pigeons to N holes, where $M > (t-1)N$, find t pigeons that are in the same hole. By generalizing the techniques from Komargodski, Naor and Yogev [KNY19], we draw deep connection between Ramsey (and its variant BiRamsey) with t-Pigeon$_N^M$. In particular, we rule out Conjecture 1 in the black-box setting, which is a direct corollary of our structural results on t-Pigeon$_N^M$ with different parameter settings. Note that any unconditional (white-box) separation within TFNP would imply $\mathsf{P} \neq \mathsf{NP}$, thus, one can only hope to prove separations in the *black-box* setting, where the input are given by query access to an oracle.

We now state our results more formally, starting with the definition of the generalized Pigeon problem. A high-level overview of our results is depicted in Figure 1.

Definition 1.1. (t-Pigeon$_N^M$) For parameters $t(n), M(n)$, and $N(n)$ satisfying $M > (t-1)N$, the problem is as follows.

> **Input** (n, h), where n is given in unary and h is a mapping from M pigeons to N holes encoded by a poly(n) size circuit.
>
> **Solutions** a *t-collision* in h, that is a set of t pigeons that are each assigned to the same hole by the circuit.

Moreover, we regularly consider the *tight* versions of these problems. For that purpose, we use t-Pigeon to indicate t-Pigeon$_N^M$ with the tight parameter setting $M = (t-1)N + 1$. For instance, 2-Pigeon is the original Pigeon problem. In a typical setting of interest, $N = 2^n$, and so we will often parametrize by $t(\log N)$ instead of $t(n)$ in order to simplify parameters.

For any function $t(n)$ depending on $n = \log N$, we can now define t-PPP to be all problems in TFNP which reduce to t-Pigeon. It is a simple exercise to see that these classes form a "Pecking Order" — a tower of complexity classes where each one contains the previous one (Lemma 2.9). For constant values of t, we define a hierarchy which we call the Pigeon Hierarchy. We denote this hierarchy by $\mathsf{PiH} = \bigcup_{t=2}^{\infty} t\text{-}\mathsf{PPP}$.

One can also consider the t-PPP class when t is a slowly growing function of the input size, as the $t(\log N)$-Pigeon$_N^M$ problem remains in TFNP whenever t is a polynomial function. With this in mind, we define the class PAP — for *polynomial averaging principle* — to contain all problems that are black-box reducible to $\log(N)$-Pigeon. In fact, we can show that the class PAP is *robust* to the choice of t in its definition. To state this formally, we introduce a definition:

Definition 1.2. A function $b(n)$ is *polynomially close* to $a(n)$ if there exists a polynomial $p(n)$ such that for any n, $b(n) \leq a(p(n))$, and $a(n) \leq b(p(n))$.

In Theorem 2.11, we show that if $a(n)$ and $b(n)$ are polynomially close functions then a-Pigeon and b-Pigeon define the same complexity class. This means that we can equivalently define PAP as all problems reducible to poly$(\log(N))$-Pigeon. To state our results in the strongest possible way, it will also be convenient to define the class SAP — for *subpolynomial averaging principle* — to be all problems reducible to t-Pigeon for some $t(n) = t(\log N) = (\log N)^{o(1)}$. Intuitively, SAP contains all problems defined by t-Pigeon, but excludes the hardest problems in PAP. For convenience, we use the general term "Pecking Order" to refer to the entire array of classes from PPP to PAP.

We are now ready to state our main results. In the black-box setting, instead of providing the inputs succinctly via polynomial-size boolean circuits, we instead think of the inputs as being provided as black-box oracles. For example, in the black-box version of the Pigeon problem, the map from pigeons to holes is provided by an oracle f which, given the name of the pigeon, outputs the hole that the pigeon maps to. In general, if A is a total search problem defining a TFNP subclass A via black-box reducibility, we will use A^{dt} (A^{dt}, resp.) to denote the black-box versions of this problem and class (we refer to Section 2 for formal definitions). It is important to note that in the theory of TFNP, *all* currently known class inclusions also hold in the black-box setting. This means that black-box separations are quite significant, since they rule out all currently existing techniques for proving inclusions between TFNP subclasses.

Our first result is that the Pigeon Hierarchy (PiH) forms a strict hierarchy in the black-box setting. Indeed, we can prove very strong black-box separations between t-$\textsc{Pigeon}_N^M$ and $(t+1)$-$\textsc{Pigeon}_{N'}^{M'}$.

Theorem 1.3. Let t be a constant, and function M, N, M', N' are parameters chosen so that $N' = \text{poly}(N)$, $M \geq tN + 1$, $M' \geq (t-1)N' + 1$. Then $(t+1)$-$\textsc{Pigeon}_N^M$ does not have an efficient black-box reduction to t-$\textsc{Pigeon}_{N'}^{M'}$.

This immediately implies the following corollary:

Corollary 1.4. For any constant t, $(t+1)$-$\mathsf{PPP}^{dt} \not\subseteq t$-$\mathsf{PPP}^{dt}$. In particular, PiH^{dt} forms a strict hierarchy in the black-box setting.

We also get structural results for non-constant t. Indeed, we have a complete characterization of the structure of the Pecking Order in the black-box setting.

Theorem 1.5. For any two functions $a(n), b(n)$, $a(n)$-PPP is polytime equivalent to $b(n)$-PPP in the black-box setting if and only if $a(n)$ and $b(n)$ are polynomially close.

The Pecking Order and Ramsey. For certain parameters t, M, N, we show how to reduce t-$\textsc{Pigeon}_N^M$ to $\textsc{Ramsey}$. In fact, this reduction even holds for the $\textsc{BiRamsey}$ problem, where we are given a bipartite graph on (N, N) vertices, and have to output either a $(\log N)/2$-biclique or a $(\log N)/2$-bi-independent set.

Theorem 1.6. When $2t(2\log N - 1) \leq \log M$, t-$\textsc{Pigeon}_N^M$ can be black-box reduced to $\textsc{Ramsey}$ and $\textsc{BiRamsey}$.

When combined with Theorem 1.3, we therefore can prove $\textsc{Ramsey} \notin \mathsf{PPP}$ in the black-box setting, resolving Conjecture 1. This is by noting that we can reduce 3-$\textsc{Pigeon}_N^{N^7}$ to $\textsc{Ramsey}$. This also shows that $\textsc{BiRamsey} \notin \mathsf{PPP}$ in the black-box setting. Further, even $\log N$-$\textsc{Pigeon}_N^{N^{5\log N}}$ can be reduced to $\textsc{Ramsey}$, which Theorem 3.10 shows does not lie in SAP. In other words, $\textsc{Ramsey}^{dt}$ and $\textsc{BiRamsey}^{dt}$ do not reduce to t-$\textsc{Pigeon}$ for any $t(\log N) = (\log N)^{o(1)}$.

Theorem 1.7. $\textsc{Ramsey}^{dt}, \textsc{BiRamsey}^{dt} \notin \mathsf{SAP}^{dt}$. In particular, $\textsc{Ramsey}^{dt}, \textsc{BiRamsey}^{dt} \notin \mathsf{PPP}^{dt}$.

As discussed before, $\textsc{Ramsey}$ has long been considered a "rogue" problem in TFNP and efforts have been made to put it in a natural complexity class. Progress was made on this question recently by Pasarkar, Papadimitriou and Yannakakis [PPY23]. They defined a subclass of TFNP they call Polynomial Long Choice (PLC) which guarantees totality by an *iterated* application of the pigeonhole principle. Their problem can be seen as a game played between an explorer and an adversary on a finite universe U where the explorer picks an element and the adversary tries to restrict the choices by forbidding subsets of U. The game ends when the explorer has found $n+1$ elements not forbidden by the adversary. The problem is total since the explorer can always succeed if they manage to pick the partition with the majority of vertices. Crucially, the adversary is *adaptive*: the forbidden sets depend on the choices made by the explorer.

One of the main results of [PPY23] was to show that $\textsc{Ramsey} \in \mathsf{PLC}$. They also defined Unary PLC (UPLC) which captures the same game but with a non-adaptive adversary. Complementing this result, we show in Lemma 5.17 that $\mathsf{UPLC} \subseteq \mathsf{PAP}$, but, it is not contained in any lower level of the Pecking Order in the black-box setting. In particular, $\mathsf{UPLC}^{dt} \not\subseteq \mathsf{PPP}^{dt}$, resolving an open question of Pasarkar, Papadimitriou, and Yannakakis.

Theorem 1.8. $\mathsf{UPLC}^{dt} \not\subseteq \mathsf{SAP}^{dt}$.

Conversely, we can also show that $\mathsf{PPP}^{dt} \not\subseteq \mathsf{UPLC}^{dt}$, resolving another open question of Pasarkar, Papadimitriou, and Yannakakis.

Theorem 1.9. $\mathsf{PPP}^{dt} \not\subseteq \mathsf{UPLC}^{dt}$. Consequently, $\mathsf{PLC}^{dt} \not\subseteq \mathsf{UPLC}^{dt}$.

Taken together, our results provide a nearly complete picture of the relative complexities of the Pecking Order, PLC, UPLC, and RAMSEY in the black-box setting.

The Pecking Order and Quantum Complexity. It has remained an open problem to get a separation between BQP and BPP with respect to a *random oracle*, explicitly posed by Aaronson and Ambainis [AA14]. They even identified a technical barrier to proving such a separation, called the Aaronson-Ambainis conjecture, which has evaded attacks by experts in Boolean function analysis for almost a decade. Most natural candidates for getting such a separation are provably false [BBBV97, AS04, Yue14, Zha15]. Such a separation is implied by an *average-case* separation between quantum and randomized query complexity.

Recently, a breakthrough result was shown in this area by Yamakawa-Zhandry [YZ22] who got such a separation for *search* problems, evading the technical barrier which existed for decision problems. They also noted that their problem can be modified to show a separation between quantum and randomized query complexity in TFNP. We show that their problem lies in the Pecking Order, giving the first inclusion in a natural subclass of TFNP.

Theorem 1.10. Yamakawa-Zhandry's Problem is contained in PAP.

Moreover, the proof of Theorem 5.18 suggests that the Yamakawa-Zhandry's problem necessarily corresponds to finding a $\mathrm{poly}(n)$-collision. By our separation results in Pecking Order (Theorem 3.10), one may suspect that Yamakawa-Zhandry's problem is not contained in any lower level of the Pecking Order, like PPP.

On the other hand, PAP is a rather loose upper-bound for the Yamakawa-Zhandry's problem. The *structure* of the Yamakawa-Zhandry's problem, imposed by a specific choice of *error-correcting code* (ECC) in its definition, is lost during our reduction. The structure of the ECC is essential for its quantum speed-up, considering the fact that even PWPP cannot be solved efficiently by quantum algorithm in the black-box setting [AS04].

The Pecking Order, PLS, and PPA. It is natural to relate the Pecking Order to other well-studied and important TFNP subclasses. The seminal work of [BCE+98] has shown that the important TFNP subclass PPA, embodying the *handshaking lemma*, is not contained in PPP in the black-box setting. Moreover, the recent breakthrough of [GHJ+22] showed that the TFNP subclass PLS, embodying problems with (not necessarily efficient) local search heuristics, is *also* not contained in PPP in the black-box setting. These are the two of the three strongest TFNP subclasses out of the "original five" TFNP subclasses defined by Papadimitriou [Pap94], the third being PPP. Using our techniques, we are able to improve these results, showing that neither of these classes are even contained in PAP — the highest level of the Pecking Order:

Theorem 1.11. $\mathsf{PLS}^{dt} \not\subseteq \mathsf{PAP}^{dt}$ and $\mathsf{PPA}^{dt} \not\subseteq \mathsf{PAP}^{dt}$.

The Pecking Order and Cryptography. The search problems corresponding to the pigeonhole principles also naturally shows up in cryptography: given a hash function which compresses the input, how hard is it to find a collision? This is known as collision resistance. Recently there has been work investigating a generalization of this to *multi-collision resistance* [BDRV18, BKP18, KNY18, RV22], which essentially corresponds to problems defined in the Pecking Order. It has remained open to show that there is no black-box reduction from breaking multi-collision resistance to collision resistance[1]. Our separation results in the Pecking Order (Theorem 3.10) make the first step towards showing a fully black-box separation between MCRH and CRH.

In addition, we separate PPP from n-PWPP for any randomized black-box reduction.

Theorem 1.12. There is no randomized black-box reduction from Pigeon to n-Pigeon_N^M with $M \geq (n-1+c)N$ for a constant $c > 0$.

Berman et al. [BDRV18] and Komargodski et al. [KNY18] show that there is no fully black-box construction from one-way permutation (corresponds to PPP) to MCRH (corresponds to t-PWPP). Besides the similarities to our result, they also differ in the following aspects.

- Both [BDRV18] and [KNY18] use a very indirect approach to rule out fully black-box construction. Specifically, they present a fully black-box construction of *constant-round* statistically hiding commitment schemes using MCRH, while a lemma from Haitner et al. [HHRS15] shows that there is no fully black-box construction of constant-round statistically hiding commitment schemes from one-way permutation. In contrast, our proof is a direct one.
- The reduction (i.e., the construction) in the fully black-box construction uses a (uniform) polynomial-time Turing machine; while our result also rules out (non-uniform) decision tree reductions. On the other hand, in the cryptography setting, it could be a Turing reduction (multi-query), while we only consider many-one reductions (single query) in this work.

1.2 Our Techniques

Our main results all follow from our black-box separations between $(t+1)$-Pigeon_N^M and t-$\text{Pigeon}_{N'}^{M'}$ for a rich parameter set of M', N', M, N (Theorem 1.3). Our black-box separations follow from developing tools in *propositional proof complexity*. The connection between logic and TFNP has long been acknowledged [BK94, MPW00, Mor01, Tha02, BM04, Jeřo9, KoNT11, BJ12, ST11, BB14, Jeř16, KoT22], and the first paper to prove oracle separations for TFNP [BCE+98] also invoked a Nullstellensatz lower bound for their result. Recently, equivalences have been established between complexity measures of certain proof systems and the complexity of black-box reductions to corresponding subclasses of TFNP. The first example of this was Göös, Kamath, Robere and Sokolov [GKRS19], followed by Göös et al. [GHJ+22] which proved equivalences for the prior natural-studied subclasses of TFNP and used this characterizations to provide the final remaining black-box separations between these subclasses. This was followed by Buss, Fleming and Impagliazzo [BFI23] who showed that this equivalence holds for every TFNP problem — in a certain sense — although the corresponding proof system they construct is somewhat artificial, and hard to analyze directly.

Our techniques follow in this vein, as our lower-bound tools are directly inspired from propositional proof complexity. However, a major place where we depart is that there is currently no known natural proof system characterizing *any* of the classes in the Pecking Order, including PPP! This

[1]To the best of our knowledge, this is still an open problem. We were notified by the authors of [KNY18] that there was an error in their proof that claimed to give a fully black-box separation between MCRH and CRH.

means that we do not have any lower-bound tools from proof complexity to borrow directly, and must instead develop new ones.

The tools we develop to prove our lower bounds are generalizations of *pseudoexpectation operators*, which have been instrumental in proving lower bounds for both the Sherali-Adams and the Sum-of-Squares proof system [FKP19]. Concurrent work of Fleming, Grosser, Pitassi, and Robere [FGPR23] introduced a new type of pseudoexpectation operator called a *collision-free pseudoexpectation operator* that was especially tailored for black-box PPP lower bounds. In this work, we generalize their notion of a collision-free pseudoexpectation operator so that they provide lower bounds against all the classes comprising the Pecking Order. All of our main lower bound results follow from our refinements of collision-free pseudoexpectations, when combined with reductions placing various other problems inside the Pecking Order. We refer to Section 3.1 for further details on our pseudoexpectation technique.

Relationship with the work of Fleming, Grosser, Pitassi, and Robere [FGPR23]. In Section 3.1 we introduce a new type of pseudoexpectation operator called a (d, t, ε)-*collision-free pseudoexpectation* (cf. Definition 3.6), and show that the construction of such operators implies lower bounds for t-PPP^{dt}. This definition is a generalization of the notion of collision-free pseudoexpectation operators introduced in the concurrent work of Fleming, Grosser, Pitassi, and Robere. (To be precise, the original notion of a collision-free pseudoexpectation corresponds to the parameter setting $(d, 2, 1)$ in the above definition). The lower bounds presented in this paper are orthogonal to the results of [FGPR23]. The main result in the concurrent work is to give a black-box separation between PPP^{dt} and its Turing closure $\mathsf{FP}^{\mathsf{PPP}^{dt}}$, a result which is not implied by the results in this paper.

1.3 Open problems

In our opinion, a fine-grained study of extremal combinatorics problems with respect to the two generalizations of PPP (PLC and Pecking Order) merits further research. The connections to cryptography, and quantum complexity also raise several intriguing questions. Here we list some open problems.

1. Can we show a black-box separations between PAP and PLC? This would conceptually separate the power of the iterated pigeonhole argument from the generalized pigeonhole argument.
2. Does Ramsey lie in PAP? With a slight loss in parameter BiRamsey lies in PAP, as shown in Lemma 4.2.
3. The definition of PWPP is robust to different compression rate [Jeř16], e.g., both 2-$\textsc{Pigeon}_N^{2N}$ and 2-$\textsc{Pigeon}_N^{N^2}$ are PWPP-complete. However, this is not known for t-PWPP (cf. [BKP18]) for any $t > 2$. One could either try to find reductions showing the equivalence, or give black-box separations for different compression rates.
4. Is $\mathsf{UPLC} = n$-PWPP? Given Lemmas 5.16 and 5.17, this would be true if n-PWPP is robust for a certain range of compression rates.
5. Can we extend our separation results in Pecking Order to a fully black-box separation between the multi-collision-resistant hash function (MCRH) and the standard collision-resistant hash function (CRH)?
6. Is there a natural TFNP subclass — simpler than the Yamakawa-Zhandry's problem — in the Pecking Order which is contained in Total Function BQP?

Paper Organization In Section 2 we introduce necessary preliminaries for the paper, including definitions of the main TFNP search problems under consideration, as well as the necessarily definitions for black-box TFNP. In Section 3, we introduce and develop our lower-bound technique of (d, t, ε)-collision-free pseudoexpectations, and use it to prove lower bounds against the Pecking Order. We also directly prove randomized separations between PPP and problems in Pecking Order that have non-trivial compression. Then, in Section 4, we prove our results relating RAMSEY and BIRAMSEY to the Pecking Order. Finally, Section 5 proves our other inclusions in and separations from the Pecking Order, including proving that $\mathsf{PLS}^{dt}, \mathsf{PPA}^{dt} \not\subseteq \mathsf{PAP}^{dt}$, our results relating to PLC and UPLC, and also our results about the Yamakawa-Zhandry's problem.

2 Preliminaries

Unless stated otherwise, $N = 2^n$. We also assume all the integer-valued functions defined in this paper are monotone and can be calculated efficiently.

2.1 Ramsey Theory

Ramsey theory is most generally defined to be a branch of combinatorics that studies how large some structure must be such that a property holds, which is usually the presence of a substructure. This branch was started by the study of the *Ramsey number*, which we define below.

Definition 2.1. $R(s, t)$ is defined to be the smallest integer n such that for any graph on n vertices, there is an independent set of size s or a clique of size t.

Theorem 2.2 (Ramsey's theorem [Ram29])**.** $R(s, t)$ is finite for every pair of integers s, t.

Finding upper and lower bounds on $R(s, t)$ has been a big program in the combinatorics community. But recently there has been interest in building *explicit* graphs with no clique or independent set of size K (which witness lower bounds for $R(K, K)$). These are motivated by connections to pseudorandomness. Constructing a pseudorandom object called a two-source *disperser* is essentially equivalent to constructing explicit bipartite Ramsey graphs, and getting better parameters has seen a long line of work [CG88, Coh16, CZ19, Li23]. This program makes progress on answering an old question of Erdős [Erd47]: can we construct explicit $O(\log n)$-Ramsey graphs on n vertices? We formally define this object below.

Definition 2.3 (***K***-Ramsey)**.** A graph on N vertices is said to be K-Ramsey if it does not contain a clique or independent set of size K.

Definition 2.4 (Bipartite ***K***-Ramsey)**.** A bipartite graph on (N, N) vertices is said to be K-BiRamsey if it does not contain a biclique or independent set of size (K, K).

We recall a recent result of Li [Li23].

Theorem 2.5 ([Li23])**.** There exists a constant $c > 1$ such that for every integer N there exists a (strongly) explicit construction of a bipartite K-Ramsey graph on N vertices with $K = \log^c N$.

We now define TFNP problems corresponding to these extremal combinatorics results.

Definition 2.6 (***K***-RAMSEY)**.** For this problem, the input specifies a graph on $N = 2^n$ vertices; the parameter $K(n)$ is a function satisfying $R(K, K) \leq N$.

Input A pair (n, C), where $C : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$ is a circuit of poly(n) size.

Solutions We have two kinds of solutions. One is a certificate that C is not a valid **encoding** of a simple undirected graph: u for which $C(u,u) = 1$ (self-loops) or u, v for which $C(u,v) \neq C(v,u)$ (directed edge). Else, we want to find K indices $V = \{v_1, v_2 \dots v_K\}$ which form a **clique** or **independent set**.

Note that we must choose K such that $R(K,K) \leq N$ in order for the problem to be total. An analogous condition holds for the bipartite analogue defined below below. We use $R_b(s,t)$ to denote the bipartite analogue of the Ramsey number.

Definition 2.7 (***K*-BiRamsey**)**.** For this problem, the input specifies a bipartite graph with N vertices on each side, where $N = 2^n$; the parameter $K(n)$ is a function satisfying $R_b(K,K) \leq N$.

Input A pair (n, C), where $C : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$ is a circuit of poly(n) size.

Solutions (K,K) indices $U = \{u_1, u_2 \dots u_K\}$ and $V = \{v_1, v_2 \dots v_K\}$ such that (U,V) forms either a **biclique** or an **independent set**.

We drop the parameter K when we mean $K = n/2$. This is the existence result (up to logarithmic factors) guaranteed by the original theorem by Ramsey [Ram29]. Note that recently the first constant factor improvement in this existential result was obtained by Campos et al. [CGMS23], which corresponds to an exponential improvement in the Ramsey number. We also omit the size parameter N when it is clear in the context.

2.2 Pecking Order

We define a set of classes which correspond to the *generalized pigeonhole principle*, which says that for $t \geq 2$, if $(t-1)N + 1$ pigeons map to N holes, then there exists a collision of t pigeons in a hole. The *pigeonhole principle* is a special case with $t = 2$. Similarly, we can define a generalization of the computational problem as follows.

Definition 1.1. (t-Pigeon$_N^M$) For parameters $t(n), M(n)$, and $N(n)$ satisfying $M > (t-1)N$, the problem is as follows.

Input (n, h), where n is given in unary and h is a mapping from M pigeons to N holes encoded by a poly(n) size circuit.

Solutions a *t-collision* in h, that is a set of t pigeons that are each assigned to the same hole by the circuit.

Note that for this problem to be in TFNP, $t(n)$ is at most a polynomial function, otherwise a solution will be too long; $t(n)$ could be a constant function. As an important special case, we use t-Pigeon to indicate t-Pigeon$_N^M$ with the tight parameter setting $M = (t-1)N + 1$.

Definition 2.8 (***t*-PPP and *t*-PWPP**)**.** For any function $t(n) \geq 2$, we define the classes

***t*-PPP** which includes any problem that is black-box reducible to t-Pigeon.

***t*-PWPP** which includes any problem that is black-box reducible to t-Pigeon$_N^M$ with $M = (t-1+c)N$ for a constant $c > 0$.

We get PPP and PWPP by taking $t = 2$ in Definition 2.8. Trivially, we have t-PWPP $\subseteq$ t-PPP for any $t \geq 2$. A simple *padding* argument shows that those classes form a hierarchy.

Lemma 2.9. t-PPP $\subseteq$ $(t+1)$-PPP. More generally, a-PPP $\subseteq$ b-PPP if $a(n) \leq b(n)$ for all n.

Proof. Given a t-Pigeon instance (n, h) ($(t-1)N+1$ pigeons and N holes), we reduce it to a $(t+1)$-Pigeon instance (n, h') by adding N dummy pigeons. The i-th dummy pigeon is mapped to the i-th hole for any $i \in [N]$; the mapping for other pigeons are left unchanged. It is easy to verify that any $(t+1)$-collision in h' is a t-collision of h.

The general case follows from the same trick of adding dummy pigeons. $\square$

This proof can be generalized to show that a-PWPP $\subseteq b$-PWPP when $a(n) \leq b(n)$. We call the hierarchy of classes for *constant* values of t the *Pigeon Hierarchy*, denoted by PiH. We call the entire collection of problems in TFNP (which correspond to $t(n) = O(\text{poly}(n))$) the *Pecking Order*.

Definition 2.10 (Pigeon Hierarchy)**.**

$$\mathsf{PiH} = \bigcup_{t=2}^{\infty} t\text{-}\mathsf{PPP}$$

Now we show that a partial converse of Lemma 2.9 also holds, which states that if two functions $a(n)$, $b(n)$ are "relatively close", then a-PPP is equivalent to b-PPP. This illustrates that for non-constant values of t the problems do not form a strict hierarchy.

Theorem 2.11. If function $b(n)$ is polynomially close to $a(n)$, then a-PPP $= b$-PPP.

Proof. Let $p(n)$ be a polynomial such that $b(n) \leq a(p(n))$ for all n. Let $c(n) := a(p(n))$. Lemma 2.9 implies that b-PPP $\subseteq c$-PPP. By the symmetry of the statement, it suffices to show that c-PPP $\subseteq a$-PPP.

Take a c-Pigeon instance $x := (n, h)$. For technical convenience, we use an alternate encoding where $M = N \cdot (c(n) - 1)$, and the goal is to find either $(c(n) - 1)$ pigeons mapped to 1 or a $c(n)$-collision. It is easy to show the equivalence between this alternate version and the original version.

We now construct an a-Pigeon instance $x' := (h', n')$, where $n' = p(n)$. Let $N' := 2^{n'}$, $M' := a(n') \cdot N'$ be the number of holes and pigeons that are supposed to be in x'. x' is simply chosen as N'/N disjoint copies of x in parallel. It is easy to verify that the number of holes in x' is $N \cdot (N'/N) = N'$, and the number of pigeons in x' is

$$M \cdot (N'/N) = (c(n) - 1) \cdot N' = (a(n') - 1) \cdot N' = M'.$$

Formally, we index each of hole in x' by a pair $(i, l) \in [N] \times [N'/N]$; each pigeon in x' is also index by a pair $(j, l) \in [M] \times [N'/N]$. Now, $h' : [M] \times [N'/N] \mapsto [N] \times [N'/N]$ is defined as

$$h'(j, l) := (h(j), l), \forall (j, l) \in [M] \times [N'/N].$$

Note that any collision in x' must come from a single copy of x, because different copies of x are disjoint. Also given that $a(n') = c(n)$, any solution of x' immediately corresponds to a solution of x. This concludes the correctness of our reduction.

$\square$

Again, one can easily generalize the proof of Theorem 2.11 to show that a-PWPP $= b$-PWPP when $a(n), b(n)$ are polynomially close.

Note that all polynomial functions are polynomially close to each other. Therefore, t-PPP is equivalent for any choice of $t(n) = \Theta(\text{poly}(n))$. We call this class PAP for "Polynomial Averaging Principle".

Definition 2.12 (Polynomial Averaging Principle)**.** PAP is the set of all relations which are reducible to n-PPP.

Note that for t-Pigeon to be in TFNP, $t(n)$ must be at most a polynomial in n. Therefore, PAP locates at the top of the Pecking Order. Similar to this definition, we define SAP.

Definition 2.13 (Subpolynomial Averaging Principle)**.** SAP is the set of all relations which are reducible to $t(n)$-PPP for some $t(n)$ sub-polynomial in n.

2.3 Decision Tree TFNP

Definition 2.14. A *query total search problem* is a sequence of relations $\mathrm{R} = \{R_n \subseteq \{0,1\}^n \times O_n\}$, where O_n are finite sets, such that for all $x \in \{0,1\}^n$ there is an $o \in O_n$ such that $(x, o) \in R_n$. A total search problem R is in TFNP^{dt} if for each $o \in O_n$ there is a decision tree T_o with depth $\mathrm{poly}(\log n)$ such that for every $x \in \{0,1\}^n$, $T_o(x) = 1$ iff $(x, o) \in \mathrm{R}$.

In general, we use $DT(\mathrm{R})$ to denote the query complexity of a search problem R. To simplify the presentation and the relationship between the black-box model and the white-box model, we adhere to several conventions:

- For problems with a non-binary input alphabet, we simulate it with the usual binary encoding. For instance, in the pigeonhole principle, a mapping of a pigeon to a hole (i.e. a pointer in the range $[n]$) can be simulated by a $\log n$ bit binary encoding. All problems discussed in this paper have $\mathrm{poly}(n)$ alphabet size, and can therefore be simulated with $O(\log n)$ overheads.
- The problem R_n is permitted to have input bits on the order of $\mathrm{poly}(n)$.
- We sometimes abuse notation by calling an individual relation R_n a search problem, rather than a whole sequence $\mathrm{R} = (R_n)$.
- The superscript "dt" is omitted when referring to TFNP^{dt} problems if the context makes it clear.

For instance, when t-Pigeon_N^M problem is defined in the black-box model, the polynomial-size circuit h encoding the mapping of pigeons to holes is replaced by an oracle that queries the pigeons and maps them to holes. In the black-box model we also have an appropriate version of reducibility between search problems, where the reductions are computed by low-depth decision trees. We introduce this next.

Definition 2.15 (Decision Tree Reduction)**.** A *decision tree reduction* from relation $R \subseteq \{0,1\}^n \times O$ to $Q \subseteq \{0,1\}^m \times O'$ is a set of depth-d decision trees $f_i : \{0,1\}^n \to \{0,1\}$ for each $i \in [m]$ and $g_o : \{0,1\}^n \to O$ for each $o \in O'$ such that for any $x \in \{0,1\}^n$,

$$(x, g_o(x)) \in R \Leftarrow (f(x), o) \in Q,$$

where $f(x) \in \{0,1\}^m$ has $f_i(x)$ as the i-th bit. The *depth* of the reduction is d, and the *size* of the reduction is $\log m$. The *complexity* of the reduction is $d + \log m$, and we write $Q^{dt}(R)$ to denote the minimum complexity of a decision-tree reduction from R to Q, or ∞ if one does not exist.

We extend these notations to sequences in the natural way. If R is a single search problem and $\mathrm{Q} = (Q_m)$ is a sequence of search problems, then we denote by $\mathrm{Q}^{dt}(R)$ the minimum of $Q_m^{dt}(R)$ over all m. If $\mathrm{R} = (R_n)$ is also a sequence, then we denote by $\mathrm{Q}^{dt}(\mathrm{R})$ the function $n \mapsto \mathrm{Q}^{dt}(R_n)$. A TFNP^{dt} problem R can be *black-box reduced* to TFNP^{dt} problem Q, written $\mathrm{R} \leq_m \mathrm{Q}$, if there is a $\mathrm{poly}(\log(n))$-complexity decision-tree reduction from R to Q.

In general, if a syntactic TFNP subclass A is defined by polynomial-time black-box reductions to a complete problem A, we will use A^{dt} to denote the class of query total search problems that can be efficiently black-box reduced to A. The next theorem motivates the decision-tree setting: constructing separations in the decision tree setting implies the existence of generic oracles separating the standard complexity classes.

Theorem 2.16 ([BCE+98], Informal)**.** For two syntactical TFNP subclasses A, B, $\mathsf{A}^{dt} \nsubseteq \mathsf{B}^{dt}$ *implies* the existence of a (generic) oracle O separating A from B, i.e., there is no black-box reduction from A to B.

We will also deal with *randomized* reductions in TFNP. We define these formally below.

Definition 2.17 (Randomized Decision Tree Reduction)**.** A *randomized decision tree reduction* from relation $R \subseteq \{0,1\}^n \times O$ to $Q \subseteq \{0,1\}^m \times O'$ is a distribution D of depth-d decision tree reductions from R to Q, such that for any $x \in \{0,1\}^n$,

$$\Pr_{((f_i),(g_o))\sim D}[(x, g_o(x)) \in R \Leftarrow (f(x), o) \in Q] \geq \frac{1}{2}.$$

3 Structure of the Pecking Order

In this section, we study the structure of the Pecking Order with respect to two major factors — the number of collisions and the compression rate.

3.1 Separations via the Number of Collisions

In this section we prove Theorem 1.3, our separation of the Pigeon Hierarchy, restated here for convenience.

Theorem 1.3. Let t be a constant, and function M, N, M', N' are parameters chosen so that $N' = \mathrm{poly}(N)$, $M \geq tN + 1$, $M' \geq (t-1)N' + 1$. Then $(t+1)$-$\textsc{Pigeon}_N^M$ does not have an efficient black-box reduction to t-$\textsc{Pigeon}_{N'}^{M'}$.

As we discussed in the introduction, our lower bounds are proved using (generalizations of) tools from propositional proof complexity, and in particular the theory of *pseudoexpectation operators*. Our main theorem is proved by generalizing the notion of *collision-free pseudoexpectation operators*, designed by Fleming, Grosser, Pitassi, and Robere to give a black-box separation between PPP^{dt} from its Turing-closure, to the entire Pecking Order [FGPR23].

Pseudoexpectation Operators. First we introduce the notion of a *pseudoexpectation operator*, and for this we need to recall some basic results about multilinear polynomials. Let $x_1, x_2, \ldots, x_n$ be a family of $\{0,1\}$-valued variables. We consider real-coefficient polynomials $p \in \mathbb{R}[x_1, \ldots, x_n]$ over these variables. All polynomials that we consider are *multilinear*, meaning the individual degree of any variable is at most 1. The algebra of multilinear polynomials is described by the quotient ring $\mathbb{R}[x_1, \ldots, x_n]/\langle x_i^2 - x_i \rangle_{i=1}^n$. Formally, the addition of two multilinear polynomials is still a multilinear polynomial, but, if we multiply two multilinear polynomials then, after multiplication, we drop the exponents of all variables to 1. For example, $(x+y)(x+y) = x + 2xy + y$, as multilinear polynomials.

For any $S, T \subseteq [n]$ with $S \cap T = \emptyset$, define the polynomial

$$C_{S,T} := \prod_{i \in S} x_i \prod_{j \in T} (1 - x_j).$$

Note that for $\{0,1\}$-assignments the polynomial $C_{S,T}$ encodes the truth value of the conjunction $\bigwedge_{i \in S} x_i \wedge \bigwedge_{j \in T} \overline{x}_j$, and thus we will refer to $C_{S,T}$ as a "conjunction" in an abuse of notation. If $R_n \subseteq \{0,1\}^n \times O$ is a query total search problem, and $o \in O$, then a conjunction C *witnesses* the solution o if $C(x) = 1 \Rightarrow (x, o) \in R_n$ for every $x \in \{0,1\}^n$. Similarly, we say a conjunction C *witnesses* R_n if it witnesses some solution to R_n. Each conjunction $C = C_{S,T}$ is naturally associated with a partial restriction $\rho(C) \in \{0,1,*\}$ defined by setting

$$\rho(C)_i = \begin{cases} 1 & i \in S, \\ 0 & i \in T, \\ * & \text{otherwise.} \end{cases}$$

Definition 3.1. Let $n \geq d$ be positive integers. Let $\mathcal{P}_{n,d}$ be the collection of all degree $\leq d$ multilinear polynomials over variables $x_1, \ldots, x_n$. An operator $\widetilde{\mathbb{E}} : \mathcal{P}_{n,d} \to \mathbb{R}$ is a *degree-d pseudoexpectation operator* if it satisfies the following three properties:

- **Linearity.** $\widetilde{\mathbb{E}}$ is linear.
- **Normalized.** $\widetilde{\mathbb{E}}[1] = 1$.
- **Nonnegativity.** $\widetilde{\mathbb{E}}[C] \geq 0$ for all degree $\leq d$ conjunctions C.

Furthermore, if $R \subseteq \{0,1\}^n \times O$ is a query total search problem, then $\widetilde{\mathbb{E}}$ is *R-Nonwitnessing* if it additionally satisfies the following property:

- ***R*-Nonwitnessing.** $\widetilde{\mathbb{E}}[C] = 0$ for any conjunction C witnessing R.

If $\mathcal{F}$ is any sequence of degree $\leq d$ multilinear polynomials, then we write $\widetilde{\mathbb{E}}[\mathcal{F}] := \sum_{p \in \mathcal{F}} \widetilde{\mathbb{E}}[p]$.

Pseudoexpectation operators were originally introduced to prove lower bounds on the degree of Sherali-Adams refutations for unsatisfiable CNF formulas [FKP19]. Often the easiest way to construct a pseudoexpectation is to construct an object called a *pseudodistribution* instead. We introduce pseudodistributions next:

Definition 3.2. Let $x_1, \ldots, x_n$ be a set of $\{0,1\}$-valued variables, and let $d \leq n$. A *degree-d pseudodistribution* over these variables is a family of probability distributions

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

such that the following properties hold:

- For each set $S \subseteq [n]$, $|S| \leq d$, $\mathcal{D}_S$ is supported on $\{0,1\}^S$, interpreted as boolean assignments to variables in $\{x_i : i \in S\}$.
- For each $S, T \subseteq [n]$, $|S|, |T| \leq d$, 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 variables in B with respect to $\mathcal{D}_A$.

The following standard lemma allows us to construct pseudoexpectations from pseudodistributions. In fact, the two objects are equivalent, but we will not need the converse construction in this paper.

Lemma 3.3 ([FKP19])**.** Let $\mathcal{D}$ be a degree-d pseudodistribution over variables $x_1, \ldots, x_n$. The operator $\widetilde{\mathbb{E}}$ defined by

$$\widetilde{\mathbb{E}}\left[\prod_{i \in S} x_i\right] = \Pr_{y \sim \mathcal{D}_S}[\forall i \in S : y_i = 1]$$

and extended to all multilinear polynomials by linearity, is a degree-d pseudoexpectation.

The following pseudodistribution, and its accompanying pseudoexpectation, is the central pillar of all lower bounds in this paper. This example is, in fact, one of the classical examples of a pseudodistribution [GM08, Lemma 2].

Definition 3.4 (Matching Pseudodistribution)**.** Consider t-$\textsc{Pigeon}_N^M$ with $M \geq (t-1)N+1$ pigeons and N holes, and let $d \leq N/2$. The *degree-d matching pseudodistribution* for this instance is the following pseudodistribution. For any set of input variables S in t-$\textsc{Pigeon}_N^M$, let $p(S)$ denote the set of pigeons mentioned among variables in S. For each subset S of variables with $|S| \leq d$, define the distribution $\mathcal{D}_S$ by sampling a uniformly random matching from the pigeons in $p(S)$ to $|p(S)|$ holes, and assigning the variables in S according to this matching. Further define $\mathcal{D} = \{\mathcal{D}_S : |S| \leq d\}$ to be the collection of all such distributions.

The above construction indeed defines a pseudodistribution, as shown in [GM08]. To put it simply, if we sample a random matching from t pigeons to t holes and then marginalize one pigeon out, then the result is a random matching from $t-1$ pigeons to $t-1$ holes.

Lower Bounds for the Pecking Order. We are now ready to prove the main lower bound result of this paper. As we have mentioned above, degree-d pseudoexpectation operators were originally introduced to prove lower bounds for Sherali-Adams refutations. In order to do this, we must construct high-degree pseudoexpectation operators which are additionally *nonwitnessing*, meaning that $\widetilde{\mathbb{E}}[C] = 0$ whenever C is a conjunction witnessing a solution to our search problem R. A recent work [GHJ+22] has shown that a query total search problem R is in PPADS^{dt} if and only if an unsatisfiable CNF formula $\neg\mathrm{Total}(R)$ expressing the totality of R has low-degree unary Sherali-Adams refutations. This means that constructing a nonwitnessing pseudoexpectation for R automatically implies that $R \notin \mathsf{PPADS}^{dt}$. However, R can admit a nonwitnessing pseudoexpectation but still lie in PPP^{dt} or higher up in the Pecking Order — for example, the matching pseudoexpectation is a nonwitnessing pseudoexpectation for 2-$\textsc{Pigeon}$, which is the complete problem for PPP.

Therefore, to prove lower bounds for higher levels of the Pecking Order, we must strengthen the definition of a pseudoexpectation. To introduce this strengthening, we need the following auxiliary definition.

Definition 3.5. Let $R \subseteq \{0,1\}^n \times O$ be a query total search problem, and let $\mathcal{F}$ be a family of conjunctions over input variables of R. For $t \geq 2$, the family $\mathcal{F}$ is said to be *t-witnessing for R* if for any subset $\mathcal{S} \subseteq \mathcal{F}$ with $|\mathcal{S}| = t$, either $\prod_{C \in \mathcal{S}} C \equiv 0$, or $DT(R \restriction \rho) \leq \log^{O(1)} n$, where ρ is the concatenation of $\rho(C)$ for all $C \in \mathcal{S}$. In other words, either every subset of t conjunctions is inconsistent, or, there is a shallow decision tree solving the restricted problem $DT(R \restriction \rho)$, where ρ is the union of partial assignments corresponding to the t conjunctions. We say that $\mathcal{F}$ is *t-witnessing* if the problem R is clear from context.

Definition 3.6. Let $R \subseteq \{0,1\}^n \times O$ be a total query search problem, let d, t be positive integers, and let $\varepsilon > 0$ be a real parameter. Let $\widetilde{\mathbb{E}}$ be a degree $D \geq d$ pseudoexpectation operator. Then $\widetilde{\mathbb{E}}$ is *(d,t,ε)-collision-free for R* if it satisfies the following property:

- ***t*-Collision-Freedom.** $\widetilde{\mathbb{E}}[\mathcal{F}] \leq \varepsilon$, for every t-witnessing family $\mathcal{F}$ of degree $\leq d$ conjunctions.

The notion of a collision-free pseudoexpectation was introduced by Fleming, Grosser, Pitassi, and Robere [FGPR23] in the special case where $t = 2, \varepsilon = 1$, in order to separate PPP from its Turing closure in the black-box setting. The above definition generalizes this notion to arbitrary size-t collisions. As we will see, t-Collision Freedom is the additional property that is required of a pseudoexpectation in order to rule out membership of problems in t-PPP. The following theorem generalizes the same theorem for $M = N+1, t = 2, \varepsilon = 1$, proved by [FGPR23].

Theorem 3.7. Let $R \subseteq \{0,1\}^n \times O$ be a query total search problem. Let M, N, t be positive integers with $M \geq (t-1)N + 1$, and let $0 \leq \varepsilon < M/N$ be any real parameter. If there is a (d, t, ε)-collision-free pseudoexpectation operator for R then there is no depth-d decision-tree reduction from R to t-$\textsc{Pigeon}_N^M$.

Proof. For the sake of intuition, we first prove this for the case of $(d, 2, \varepsilon)$-pseudoexpectation operators, corresponding to the classical TFNP class PPP, but the argument easily generalizes to arbitrary t, N, and $M \geq (t-1)N + 1$. Let us assume by way of contradiction that there is a degree-d decision-tree reduction from R to 2-$\textsc{Pigeon}_N^{N+1}$ for some N, and let $\widetilde{\mathbb{E}}$ denote the $(d, 2, \varepsilon)$-collision-free pseudoexpectation for R. Let $T_1, T_2, \ldots, T_{N+1}$ denote the depth-d decision trees in the 2-$\textsc{Pigeon}_N^{N+1}$ instance produced by the reduction mapping the pigeons to holes. First, for any decision tree T_i let $L(T_i)$ denote the leaves of T_i, and, for any leaf ℓ of T_i, let C_ℓ denote the conjunction obtained by multiplying the literals along the path to ℓ. An easy induction on the depth of the tree shows that for every tree T_i,

$$\sum_{\ell \in L(T_i)} \widetilde{\mathbb{E}}[C_\ell] = 1.$$

Now for any hole $h \in [N]$, let $\mathcal{F}_h$ denote the set of all conjunctions C_ℓ that correspond to paths of any decision tree among $T_1, \ldots, T_{N+1}$ such that the leaf ℓ of that path is labelled with h. Since this instance of 2-$\textsc{Pigeon}_N^{N+1}$ is obtained via a depth-d reduction from R, for any pair of conjunctions $C, D \in \mathcal{F}_h$, either C and D are inconsistent, or, $DT(R \restriction \rho(CD)) \leq d$ since a collision of two pigeons implies that we can recover a solution of R after at most d more queries. It follows that the family $\mathcal{F}_h$ is 2-witnessing, in the language of Definition 3.5.

Since $\mathcal{F}_h$ is 2-witnessing for each hole, it follows that $\widetilde{\mathbb{E}}[\mathcal{F}_h] \leq \varepsilon$ for every $h \in [N]$ since $\widetilde{\mathbb{E}}$ is $(d, 2, \varepsilon)$-collision-free. Now, observe that

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

since each of the $N+1$ pigeons are mapped to exactly one hole under a total assignment to the variables. This means that

$$N + 1 = \sum_{i=1}^{N+1} \sum_{\ell \in L(T_i)} \widetilde{\mathbb{E}}[C_\ell] = \sum_{h=1}^{N} \widetilde{\mathbb{E}}[\mathcal{F}_h] \leq \varepsilon N < N + 1,$$

which is a contradiction.

To generalize this for arbitrary t, we instead consider reductions to t-$\textsc{Pigeon}_N^M$ and substitute $(d, 2, \varepsilon)$-witnessing with (d, t, ε)-witnessing in the above proof. Since the instance of t-$\textsc{Pigeon}_N^M$ is obtained by depth-d reduction from R, it follows now that for every set of t distinct conjunctions $C_1, C_2, \ldots, C_t \in \mathcal{F}_h$, either $\prod_{i=1}^t C_i$ is inconsistent, or, $DT(R \restriction \rho(C_1 C_2 \cdots C_t)) \leq d$, since a collision of t pigeons implies that we can recover a solution of R after at most d more queries. Therefore, $\mathcal{F}_h$ is now t-witnessing, and so $\widetilde{\mathbb{E}}[\mathcal{F}_h] \leq \varepsilon$. We now have

$$M = \sum_{i=1}^{M} \sum_{\ell \in L(T_i)} \widetilde{\mathbb{E}}[C_\ell] = \sum_{i=1}^{N} \widetilde{\mathbb{E}}[\mathcal{F}_h] \leq \varepsilon N < M,$$

a contradiction. $\square$

The previous theorem gives us a powerful method to prove lower bounds against levels of the Pecking Order. In particular, we will be able to show that the *matching pseudoexpectation* is t-collision-free against $(t+1)$-$\textsc{Pigeon}_N^M$ for *all* t. The main observation here is that the union of t matchings from M to N has a collision of size at most t, and therefore cannot witness a $(t+1)$-collision of pigeons. This means that any t-subset of conjunctions from a t-witnessing family of matchings is inconsistent. The next technical lemma shows how to upper-bound the weight of such inconsistent families.

Lemma 3.8. Let $x_1, \ldots, x_n$ be a set of variables, and let t, d be chosen so that $(t-1)d^2 \leq n$. Let $\mathcal{F}$ be any family of degree $\leq d$ conjunctions over these variables, such that for every subset $\mathcal{S} \subseteq \mathcal{F}$ with $|\mathcal{S}| = t$, $\prod_{C \in \mathcal{S}} C \equiv 0$. If $\widetilde{\mathbb{E}}$ is a degree $D \geq (t-1)d^2$ pseudoexpectation operator, then $\widetilde{\mathbb{E}}[\mathcal{F}] \leq t-1$.

Proof. Let us first observe that the statement of the lemma would obviously be true if $\widetilde{\mathbb{E}}$ were the expectation over a *true* probability distribution. This is because at most $t-1$ distinct conjunctions in $\mathcal{F}$ are consistent with any total assignment, and thus no set of t conjunctions can be simultaneously activated under a sample from the true probability distribution.

To prove this claim for $\widetilde{\mathbb{E}}$ we first need to introduce some notation. Let T be any decision tree querying the variables $x_1, \ldots, x_n$ and outputting 0 or 1. Let $L_b(T)$ denote the leaves of T labelled with $b \in \{0,1\}$, and let $L(T) = L_0(T) \cup L_1(T)$. For any leaf $\ell \in L(T)$ let C_ℓ denote the conjunction of literals on the path from the root to ℓ, and let $\rho_\ell = \rho(C_\ell)$ denote the partial assignment corresponding to this path. If the depth of T is at most d, define

$$\widetilde{\mathbb{E}}[T] := \sum_{\ell \in L_1(T)} \widetilde{\mathbb{E}}[C_\ell].$$

An easy induction on the depth of T shows that $\widetilde{\mathbb{E}}[T] \leq 1$.

Starting with the family $\mathcal{F}$, we create a depth $\leq (t-1)d^2$ decision tree T such that

$$\widetilde{\mathbb{E}}[\mathcal{F}] \leq (t-1)\widetilde{\mathbb{E}}[T] \leq t-1.$$

If ρ is a partial assignment, then let $\mathcal{F} \restriction \rho = \{C \restriction \rho : C \in \mathcal{F}\}$, where it is understood that we remove any conjunctions that are set to 0 or 1 under the restriction ρ.

We construct the decision tree T by the following recursive algorithm. The decision tree maintains a partial assignment to the above variables. Initially, $\rho = \emptyset$. The algorithm proceeds in rounds. If $\mathcal{F} \restriction \rho = \emptyset$, we halt and output 1, if there are any conjunctions in $\mathcal{F}$ consistent with ρ, or 0 otherwise. We proceed assuming $\mathcal{F} \restriction \rho \neq \emptyset$. In this case, we choose $t-1$ conjunctions $C_1, C_2, \ldots, C_{t-1}$ in $\mathcal{F} \restriction \rho$ and query all unqueried variables among these conjunctions — if there are less than $t-1$, then we simply query all the variables among all remaining conjunctions. After this querying stage, we have learned a partial assignment σ to the newly queried variables, and we then recurse on the family $\mathcal{F} \restriction \rho\sigma$.

The construction above plainly terminates on every branch, since we are reducing the length of each conjunction after every round of querying. We argue that the depth of the tree is at most $(t-1)d^2$ and that $\widetilde{\mathbb{E}}[\mathcal{F}] \leq (t-1)\widetilde{\mathbb{E}}[T]$, which completes the proof of the theorem.

First, we argue that the depth of the tree is at most D. To see this, we observe that in each round we query at most $(t-1)d$ variables, and we argue that on every branch the algorithm terminates after at most d rounds. To see this, consider the i-th round, where we query conjunctions $C_1, \ldots, C_{t-1}$. Since $\mathcal{F}$ is t-witnessing, it follows that every conjunction C remaining in $\mathcal{F}$ must conflict with at least one literal contained in $C_1, \ldots, C_{t-1}$. Therefore, after querying all unqueried variables in $C_1, \ldots, C_{t-1}$, we must query at least one variable from every remaining conjunction in $\mathcal{F}$. This means that at the end of the ith round, we must reduce the length of every remaining conjunction by at least one. Since each conjunction has at most d variables to begin with, it follows that the entire process can proceed for at most d rounds. Thus, the depth of the tree is at most D.

Let us now see that $\widetilde{\mathbb{E}}[\mathcal{F}] \leq (t-1)\widetilde{\mathbb{E}}[T]$. First, we observe that since the depth of T is at most $(t-1)d^2$, it follows that every for every leaf ℓ of T the conjunction C_ℓ has degree at most D. This means that $\widetilde{\mathbb{E}}[T]$ is well-defined since $\widetilde{\mathbb{E}}$ is a degree-D pseudoexpectation.

So, it remains to show that $\widetilde{\mathbb{E}}[\mathcal{F}] \leq (t-1)\widetilde{\mathbb{E}}[T]$, noting that $\widetilde{\mathbb{E}}[T] \leq 1$ for any decision tree T. For any node u in T let T_u denote the subtree rooted at u, and let C_u denote the conjunction of the literals along the path to u. We prove by induction on the height of u that

$$\sum_{C \in \mathcal{F}} \widetilde{\mathbb{E}}[C_u C] \leq (t-1) \sum_{\ell \in L_1(T_u)} \widetilde{\mathbb{E}}[C_u C_\ell]. \tag{1}$$

Once we have this equation, setting u to be the root node r yields

$$\sum_{C \in \mathcal{F}} \widetilde{\mathbb{E}}[C] \leq (t-1) \sum_{\ell \in L_1(T)} \widetilde{\mathbb{E}}[C_\ell] = (t-1)\widetilde{\mathbb{E}}[T],$$

as desired.

If u is a 1-leaf node of T_u, then $L_1(T_u) = \{u\}$ and so

$$\sum_{\ell \in L_1(T_u)} \widetilde{\mathbb{E}}[C_u C_\ell] = \widetilde{\mathbb{E}}[C_u].$$

On the other hand, for any leaf u, by construction of the tree if $C_u C \not\equiv 0$ then $C_u C = C_u$. Since the family $\mathcal{F}$ is t-witnessing, there are at most $t-1$ possible choices of $C \in \mathcal{F}$ such that $C_u C \not\equiv 0$ since every set of t conjunctions in $\mathcal{F}$ are inconsistent. Therefore

$$\sum_{C \in \mathcal{F}} \widetilde{\mathbb{E}}[C_u C] \leq (t-1)\widetilde{\mathbb{E}}[C_u] = (t-1) \sum_{\ell \in L_1(T_u)} \widetilde{\mathbb{E}}[C_u C_\ell],$$

proving the base case of (1).

For the inductive step, consider a node u in T querying a variable x_i with children v_0, v_1 corresponding to the two outcomes of the query. Then

$$\begin{aligned}
\sum_{C \in \mathcal{F}} \widetilde{\mathbb{E}}[C_u C] &= \sum_{C \in \mathcal{F}} \widetilde{\mathbb{E}}[C_u x_i C + C_u(1-x_i)C] \\
&= \sum_{C \in \mathcal{F}} \widetilde{\mathbb{E}}[C_u x_i C] + \widetilde{\mathbb{E}}[C_u(1-x_i)C] \\
&= \sum_{C \in \mathcal{F}} \widetilde{\mathbb{E}}[C_{v_1} C] + \sum_{C \in \mathcal{F}} \widetilde{\mathbb{E}}[C_{v_0} C] \\
&\leq (t-1) \sum_{\ell \in L_1(T_{v_1})} \widetilde{\mathbb{E}}[C_{v_1} C_\ell] + (t-1) \sum_{\ell \in L_1(T_{v_0})} \widetilde{\mathbb{E}}[C_{v_0} C_\ell] \\
&= (t-1) \left(\sum_{\ell \in L_1(T_{v_1})} \widetilde{\mathbb{E}}[C_u x_i C_\ell] + \sum_{\ell \in L_1(T_{v_0})} \widetilde{\mathbb{E}}[C_u(1-x_i)C_\ell] \right) \\
&= (t-1) \sum_{\ell \in L_1(T_u)} \widetilde{\mathbb{E}}[C_u C_\ell],
\end{aligned}$$

where the inequality follows by the induction hypothesis, and the last equality follows since the leaves of T_u are exactly the union of leaves of T_{v_0} and T_{v_1}. This proves (1) for all nodes u of T, completing the proof. □

With this technical lemma in hand, we are now ready to prove the main theorem of this section.

Theorem 3.9. Let t, d, M, N, M', N' be positive integers chosen so that $M \geq tN+1$, $M' \geq (t-1)N'+1$, and $(t-1)d^2 \leq N/2$. Then the $(t+1)$-$\textsc{Pigeon}_N^M$ problem does not have a depth-d decision-tree reduction to t-$\textsc{Pigeon}_{N'}^{M'}$.

Proof. Let $D = (t-1)d^2 \leq N/2$, and let $\widetilde{\mathbb{E}}$ be the degree-D pseudoexpectation obtained from the degree-D matching pseudodistribution (Definition 3.4) for $(t+1)$-$\textsc{Pigeon}_N^M$. We show that $\widetilde{\mathbb{E}}$ is $(d, t, t-1)$-collision-free for $(t+1)$-$\textsc{Pigeon}_N^M$. By Theorem 3.7, this implies that $(t+1)$-$\textsc{Pigeon}_N^M$ does not depth-d reduce to t-$\textsc{Pigeon}_{N'}^{M'}$ for any M', N' with $M' \geq (t-1)N' + 1$.

Consider any t-witnessing family $\mathcal{F}$ for $(t+1)$-$\textsc{Pigeon}_N^M$ in which every conjunction has degree $\leq d$. Without loss of generality, we can remove any conjunction C from $\mathcal{F}$ such that $\widetilde{\mathbb{E}}[C] = 0$. Thus we can assume that $\rho(C)$ encodes a partial matching for all $C \in \mathcal{F}$. Let $\mathcal{S} \subseteq \mathcal{F}$ be any collection of t conjunctions in $\mathcal{F}$, let $C_{\mathcal{S}} = \prod_{C \in \mathcal{S}} C$ denote the conjunction obtained by multiplying all conjunctions in $\mathcal{S}$.

Suppose that $C_{\mathcal{S}} \not\equiv 0$, and let $\rho = \rho(C_{\mathcal{S}})$ denote the corresponding partial assignment. Since each constituent conjunction of $C_{\mathcal{S}}$ is a partial matching, and there are only t conjunctions, it follows that ρ cannot witness a solution to $(t+1)$-$\textsc{Pigeon}_N^M$ since it can only contain a collision of at most t pigeons in any hole. A simple adversary strategy then implies that $DT((t+1)\text{-}\textsc{Pigeon}_N^M \restriction \rho) = \Omega(N)$, since we can respond to any unqueried pigeon by placing that pigeon into any hole with $\leq t-1$ pigeons. Since the family is t-witnessing, it therefore follows that $C_{\mathcal{S}} \equiv 0$, i.e., $\rho(C_{\mathcal{S}})$ is an inconsistent partial assignment that tries to place at least one pigeon in two different holes. By Lemma 3.8, this means that $\widetilde{\mathbb{E}}[\mathcal{F}] \leq t-1$, and therefore $\widetilde{\mathbb{E}}$ is a $(d, t, t-1)$-collision-free pseudoexpectation operator for $(t+1)$-$\textsc{Pigeon}_{N'}^{M'}$. Applying Theorem 3.7 completes the proof. $\square$

Our main result separating the Pecking Order is now an immediate corollary of the previous theorem. We recall it here for convenience.

Theorem 1.3. Let t be a constant, and function M, N, M', N' are parameters chosen so that $N' = \mathrm{poly}(N)$, $M \geq tN + 1$, $M' \geq (t-1)N' + 1$. Then $(t+1)$-$\textsc{Pigeon}_N^M$ does not have an efficient black-box reduction to t-$\textsc{Pigeon}_{N'}^{M'}$.

Further, we note that we can prove a generalization of Theorem 3.9, separating the problem with parameter $a(n)$ from $b(n)$ whenever $a(n)$ is not polynomially close to gbn). Also note that since $(t+1)$ is not polynomially close to t for any constant t, this is strict generalization. We prove this using the same technique as the theorem above, so we only provide a proof sketch here.

Theorem 3.10. Let $a(n), b(n)$ be functions such that $a(n)$ is larger than $b(n)$ and is not polynomially close to $b(n)$. Let t, d, M, N, M', N' be positive integers chosen so that $N' = \mathrm{poly}(N)$, $M \geq tN+1$, $M' \geq (t-1)N'+1$, and $t \leq \mathrm{poly}(\log(N))$. Then $a(n)$-$\textsc{Pigeon}_N^M$ does not have an efficient black-box reduction to $b(n)$-$\textsc{Pigeon}_{N'}^{M'}$.

Proof Sketch. Similar to the proof of Theorem 3.9, we construct a $(d, b(n), b(n)-1)$-Collision-free pseudoexpectation operator for $a(n)$-$\textsc{Pigeon}_M^N$ using the Matching pseudodistribution (Definition 3.4) combined with Lemma 3.8. Here, we used that since $a(n)$ is not polynomially close to $b(n)$, no $b(n)$-collection of partial assignments can be witnessing for $a(n)$-$\textsc{Pigeon}_N^M$. Applying Theorem 3.7 completes the proof. $\square$

Note that the statement of the above theorem separates all problems which do not have reductions guaranteed by Theorem 2.11. Combining the two, we can conclude Theorem 1.5.

Theorem 1.5. For any two functions $a(n), b(n)$, $a(n)$-PPP is polytime equivalent to $b(n)$-PPP in the black-box setting if and only if $a(n)$ and $b(n)$ are polynomially close.

3.2 Separations via the Compression Rate

In this subsection, we present another type of separation within the Pecking Order, which is due to the difference on compression rate.

Theorem 1.12. There is no randomized black-box reduction from PIGEON to n-PIGEON$_N^M$ with $M \geq (n-1+c)N$ for a constant $c > 0$.

Corollary 3.11. $\mathsf{PPP}^{dt} \not\subseteq n\text{-}\mathsf{PWPP}^{dt}$.

Notation. We consider PIGEON instances with $N+1$ pigeons and N holes. With a little abuse of notation, we denote a PIGEON instance by a string $h \in [N]^{N+1}$, where h_i is the hole that pigeon i gets mapped to. Assume that the n-PIGEON$_N^M$ instance $f(h)$ reduced from h has $M(n')$ pigeons, $N(n')$ holes, and the solution is any n'-collision. We have $M' > N' \cdot (n'-1+c)$ for a constant $c > 0$ from the theorem statement. Formally, for any $i \in [M']$, pigeon i from $f(h)$ is mapped to hole $f_i(h)$. Without loss of generality, we assume all decision trees $(f_i, g_o)_{i\in[M'],o\in[M']^{n'}}$ have the same depth $d = \mathrm{poly}(\log N)$.

We then proceed in four steps:

1. Finding an appropriate distribution D_N of hard PIGEON instances.
2. Showing that with high probability, there exists a "non-witnessing" solution in the n-PIGEON$_N^M$ instance $f(h)$ when h is drawn from D_N.
3. Arguing that the error probability is high if all decision trees (g_o) are depth-0.
4. Generalizing the error analysis to depth-d.

A hard distribution. By Yao's Minimax principle, it suffices to consider a family of distributions (D_N) of PIGEON instances, and then show that any deterministic low-depth black-box reduction (f_i, g_o) must be wrong with high probability. A natural choice of D_N is taking $h_1, \ldots, h_N$ to be a random permutation over $[N]$, while h_{N+1} is always set to 1. Now, the only possible solution is the index $i^* \in [N]$ such that $h_{i^*} = 1$. In the rest of this proof, we use i^* rather than the collision pair $(i^*, N+1)$ to indicate the solution of h; we also ignore h_{N+1} and assume h is permutation on $[N]$.

Find a non-witnessing solution. For any $i \in [M'], h \in D_N$, we say pigeon i in $f(h)$ is *non-witnessing* if the decision tree path in f_i realized by h is not witnessing, i.e., f_i does not query i^* when evaluating on h; otherwise, we call pigeon i *witnessing*.

Let $o = (i_1, \ldots, i_{n'})$ be any solution (i.e., an n'-collision) of $f(h)$. By taking the union of all decision tree paths in $(f_{i_j})_{j\in[n']}$ realized by h, we get a partial assignment h_o of size at most $d \cdot n' = \mathrm{poly}(\log N)$ in h. Without loss of generality, we assume this partial assignment h_o is also returned by the reduction as part of the solution.

We say o is a *non-witnessing* solution if for all $j \in [n']$, the pigeon i_j is non-witnessing, i.e., the partial assignment h_o does not witness the location of i^*; otherwise, o is a *witnessing* solution. Intuitively, a non-witnessing solution reveals almost no information about the key location i^*.

We now prove the following key lemma.

Lemma 3.12. When h is drawn from D_N, $f(h)$ has a non-witnessing solution with probability $1 - \mathrm{negl}(\log N)$.

Proof. Since h is a random permutation, and f_i only has d levels, for any pigeon $i \in [M']$ in $f(h)$, we have

$$\Pr_{h \sim D_N}[i \text{ is witnessing}] \leq \frac{d}{N} = \mathrm{negl}(\log N).$$

Using Markov's inequality, the probability that $f(h)$ has at least cN' witnessing pigeons is $\mathrm{negl}(\log N)$. In other words, with probability $1 - \mathrm{negl}(\log N)$, $f(h)$ has more than $(n'-1) \cdot N'$ non-witnessing pigeon.

Therefore, with probability $1 - \mathrm{negl}(\log N)$, $f(h)$ has a n'-collision with only non-witnessing pigeons, i.e., a non-witnessing solution. □

Success probability for depth-0. We say a reduction is depth-k ($k \leq d$) if all the decision trees (g_o) are depth-k, while (f_i) are still depth-d. For a fixed family of depth-d decision trees (f_i), define p_k as the maximal success probability of any depth-k ($k \leq d$) reduction. We first consider the success probability of depth-0 reduction, i.e., $g_o \in [N]$ is a fixed location.

Let R, Q be the short-hand for Pigeon and n-Pigeon_N^M. Formally, our goal in this step is to show the following inequality.

Lemma 3.13.

$$p_0 := \max_{f,g} \Pr_{h \sim D_N}[(h, g_o) \in R \Leftarrow (f(h), o) \in Q] < \mathrm{negl}(\log N).$$

Let $h^{(1)}$ be an instance of Pigeon with solution i^*. The key technical trick here is to roll a second dice, which helps us estimate the error probability. Specifically, we take a uniformly random index $j^* \in [N]$, and then generate $h^{(2)}$ from $h^{(1)}$ by swapping the solution from location i^* to j^*. Formally,

$$h^{(2)}_{j^*} = 1, h^{(2)}_{i^*} = h^{(1)}_{j^*}; \quad h^{(2)}_i = h^{(1)}_i, \forall i \in [N] \backslash \{i^*, j^*\}.$$

By symmetry, we know the marginal distribution of $h^{(2)}$ is also a random permutation, so we can calculate the success probability on $h^{(2)}$ instead.

Let $\mathsf{NonWit}(h, o)$ indicate the event that o is a non-witnessing solution for $f(h)$, and denote the uniform distribution over $[N]$ by U_N. We have the following observation regarding our second dice.

Lemma 3.14. For any $h^{(1)}$ and o,

$$\Pr_{j^* \sim U_N}[\mathsf{NonWit}(h^{(2)}, o) \mid \mathsf{NonWit}(h^{(1)}, o)] = 1 - \mathrm{negl}(\log N).$$

Proof. Since o is non-witnessing, the partial assignment h_o does not contain location i^*. Recall that $h^{(2)}$ is different to $h^{(1)}$ only on location i^* and j^*. Therefore, with $1 - \mathrm{negl}(\log N)$ probability on j^*, h_o is also a partial assignment of $h^{(2)}$, which implies that o is a non-witnessing solution of $f(h^{(2)})$. □

Now we are ready to prove Lemma 3.13.

Proof of Lemma 3.13. Fix an arbitrary depth-0 reduction (f, g). We give an overview before diving into the calculations. We first randomly draw $h^{(1)}$ from D_N, and there are two possible cases: either $f(h^{(1)})$ has a non-witnessing solution, or $f(h^{(1)})$ does not have any non-witnessing solution. The second case will happen with very low probability (Lemma 3.12), so we can assume $f(h^{(1)})$ has a non-witnessing solution o^*. We then roll a second dice $j \sim [N]$ and generate $h^{(2)}$ accordingly.

We know that o^* is also a non-witnessing solution of $f(h^{(2)})$ with high probability (Lemma 3.14); among these $h^{(2)}$, the reduction could possibly be correct only when $j = g_{o^*}$.

Formally, let o^* be the first non-witnessing solution of $f(h^{(1)})$ in the lexicographical order, if $f(h^{(1)})$ has a non-witnessing solution; otherwise, let o^* to be the lexicographically first solution of $f(h^{(1)})$. We first consider whether $h^{(1)}$ has a non-witnessing solution.

$$\begin{aligned}
&\Pr_{h^{(2)}\sim D_N}[(h^{(2)}, g_o) \in R \Leftarrow (f(h^{(2)}), o) \in Q] \\
&= \Pr_{h^{(1)}\sim D_N, j^*\sim U_N}[(h^{(2)}, g_o) \in R \Leftarrow (f(h^{(2)}), o) \in Q] \\
&\leq \Pr_{h^{(1)}, j^*}[(h^{(2)}, g_o) \in R \Leftarrow (f(h^{(2)}), o) \in Q \mid \mathsf{NonWit}(h^{(1)}, o^*)] + \Pr_{h^{(1)}}[\neg\mathsf{NonWit}(h^{(1)}, o^*)] \qquad (2)
\end{aligned}$$

Note that the term $\Pr_{h^{(1)}}[\neg\mathsf{NonWit}(h^{(1)}, o^*)]$ is at most $\mathrm{negl}(\log N)$ from Lemma 3.12. So it remains to bound the first item in inequality (2), which is the success probability on $h^{(2)}$ condition on $h^{(1)}$ has a non-witnessing solution o^*.

$$\begin{aligned}
&\Pr_{h^{(1)}, j^*}[(h^{(2)}, g_o) \in R \Leftarrow (f(h^{(2)}), o) \in Q \mid \mathsf{NonWit}(h^{(1)}, o^*)] \\
&\leq \Pr_{h^{(1)}, j^*}[(h^{(2)}, g_{o^*}) \in R \wedge \mathsf{NonWit}(h^{(2)}, o^*) \mid \mathsf{NonWit}(h^{(1)}, o^*)] \\
&\quad + \Pr_{h^{(1)}, j^*}[\neg\mathsf{NonWit}(h^{(2)}, o^*) \mid \mathsf{NonWit}(h^{(1)}, o^*)] \\
&\leq \Pr_{h^{(1)}, j^*}[(h^{(2)}, g_{o^*}) \in R \wedge \mathsf{NonWit}(h^{(2)}, o^*) \mid \mathsf{NonWit}(h^{(1)}, o^*)] + \mathrm{negl}(\log N) \qquad (3) \\
&= \Pr_{h^{(1)}, j^*}[(j^* = g_{o^*}) \wedge \mathsf{NonWit}(h^{(2)}, o^*) \mid \mathsf{NonWit}(h^{(1)}, o^*)] + \mathrm{negl}(\log N) \\
&\leq \Pr_{h^{(1)}, j^*}[(j^* = g_{o^*}) \mid \mathsf{NonWit}(h^{(1)}, o^*)] + \mathrm{negl}(\log N) \\
&= \frac{1}{N} + \mathrm{negl}(\log N) \\
&= \mathrm{negl}(\log N). \qquad (4)
\end{aligned}$$

Note that inequality (3) comes from Lemma 3.14.

□

Success probability for depth-*d*. We now consider depth-k reduction for $k \leq d$, i.e., all (g_o) are depth-k. Define event $\mathsf{Hit\text{-}i}(h^{(1)}, o)$ to be true if o is a witnessing solution of $f(h^{(1)})$, or i^* is queried when evaluating g_o on $h^{(1)}$. Intuitively, $g_o(h^{(1)})$ has to *guess* a location if $\mathsf{Hit\text{-}i}(h^{(1)}, o)$ is not true,

The following lemma effectively reduces the problem to the case of depth-$(k-1)$.

Lemma 3.15. For a depth-k reduction (f_i, g_o),

$$\Pr_{h^{(1)}\sim D_N}[\mathsf{Hit\text{-}i}(h^{(1)}, o) \Leftarrow (f(h^{(1)}), o) \in Q] \leq p_{k-1}$$

Proof. We construct a depth-$(k-1)$ reduction (f_i, g'_o): For each possible $o \in [M']^{n'}$, the structure of g'_o is same as the first $k-1$ levels of g_o. If the solution i^* is witnessed by o itself or the first $k-1$

queries of g'_o, then g'_o will output i^*; otherwise, g'_o will output the location that is going be queried in g_o in the k-th level, if g_o is evaluated on the same input. Therefore, we have

$$\Pr_{h^{(1)}\sim D_N}[\mathsf{Hit\text{-}i}(h^{(1)},o) \Leftarrow (f(h^{(1)}),o)\in Q] = \Pr_{h^{(1)}\sim D_N}[(h^{(1)},g'_o)\in R \Leftarrow (f(h^{(1)}),o)\in Q] \le p_{k-1}.$$

□

Define event $\mathsf{Hit\text{-}j}(h^{(1)},o,j^*)$ to be true if the partial assignment h_o contains the location j^*, or j^* is queried when g_o is evaluated on $h^{(1)}$. Note that when both $\mathsf{Hit\text{-}i}(h^{(1)},o)$ and $\mathsf{Hit\text{-}j}(h^{(1)},o,j^*)$ are false, we have $g_o(h^{(1)}) = g_o(h^{(2)})$, because they only differ in location i^* and j^*. We have the following lemma with the same argument in Lemma 3.14.

Lemma 3.16. For any $h^{(1)}$ and o,

$$\Pr_{j^*\sim U_N}[\neg\mathsf{Hit\text{-}j}(h^{(1)},o,j^*) \mid \neg\mathsf{Hit\text{-}i}(h^{(1)},o)] = 1-\mathrm{negl}(\log N).$$

Now, let us wrap everything up.

Proof of Theorem 1.12. We follow a similar strategy as in the proof of Lemma 3.13. For any depth-k reduction (f,g), there are two possibilities regarding to $h^{(1)}$: either $f(h^{(1)})$ has a (non-witnessing) solution o^* that $\mathsf{Hit\text{-}i}(h^{(1)},o^*)$ is false, or $\mathsf{Hit\text{-}i}(h^{(1)},o)$ is true for any solution o of $f(h^{(1)})$. The second case will happen with probability at most p_k by Lemma 3.15, and we also roll a second dice j^* to analyze the first case.

Formally, let o^* to be the lexicographically first solution of $f(h^{(1)})$ that $\mathsf{Hit\text{-}i}(h^{(1)},o)$ is false, if such a solution exists; otherwise, let o^* to be the lexicographically first solution of $f(h^{(1)})$. We have

$$\begin{aligned}
&\Pr_{h^{(1)}\sim D_N, j^*\sim U_N}[(h^{(2)},g_o(h^{(2)}))\in R \Leftarrow (f(h^{(2)}),o)\in Q] \\
&\le \Pr_{h^{(1)},j^*}[(h^{(2)},g_o(h^{(2)}))\in R \Leftarrow (f(h^{(2)}),o)\in Q \mid \neg\mathsf{Hit\text{-}i}(h^{(1)},o^*)] + \Pr_{h^{(1)}}[\mathsf{Hit\text{-}i}(h^{(1)},o^*)] \\
&\le \Pr_{h^{(1)},j^*}[(h^{(2)},g_o(h^{(2)}))\in R \Leftarrow (f(h^{(2)}),o)\in Q \mid \neg\mathsf{Hit\text{-}i}(h^{(1)},o^*)] + p_{k-1} && (5)\\
&\le \Pr_{h^{(1)},j^*}[(j^* = g_{o^*}(h^{(2)})) \wedge \neg\mathsf{Hit\text{-}j}(h^{(1)},o^*,j^*) \mid \neg\mathsf{Hit\text{-}i}(h^{(1)},o^*)] \\
&\quad + \Pr_{h^{(1)},j^*}[\mathsf{Hit\text{-}j}(h^{(1)},o^*,j^*) \mid \neg\mathsf{Hit\text{-}i}(h^{(1)},o^*)] + p_{k-1} \\
&\le \Pr_{h^{(1)},j^*}[(j^* = g_{o^*}(h^{(2)})) \wedge \neg\mathsf{Hit\text{-}j}(h^{(1)},o^*,j^*) \mid \neg\mathsf{Hit\text{-}i}(h^{(1)},o^*)] + p_{k-1} + \mathrm{negl}(\log N) && (6)\\
&\le \Pr_{h^{(1)},j^*}[(j^* = g_{o^*}(h^{(1)})) \mid \neg\mathsf{Hit\text{-}i}(h^{(1)},o^*)] + p_{k-1} + \mathrm{negl}(\log N) \\
&= \frac{1}{N} + p_{k-1} + \mathrm{negl}(\log N) \\
&= p_{k-1} + \mathrm{negl}(\log N). && (7)
\end{aligned}$$

Inequality (5) uses Lemma 3.15, and inequality (6) comes from Lemma 3.16. From inequality (7), we have the success probability of any depth-d reduction

$$p_d \le d\cdot\mathrm{negl}(\log N) + p_0 = \mathrm{negl}(\log N).$$

□

The proof of Theorem 1.12 leverages the fact that PPP instance could have few or unique solutions, while t-PWPP instances always have *abundant* solutions; moreover, the totality of t-PWPP is *robust*, i.e., t-PWPP is total even when a small fraction of pigeons are removed. This idea is generalized in a concurrent work by Li [Li24], which provides a general framework for showing randomized black-box separation between TFNP problems with very few or abundant solutions. A similar idea also appears in an earlier work of Müller [Mül21] in the language of bounded arithmetic.

4 Ramsey and the Pecking Order

In this section, we show that problems in the Pecking Order can be reduced to Ramsey and BiRamsey (Lemma 4.1). As a consequence of the generality of parameters in the structural theorem of the Pecking Order (Theorem 3.10), we immediately get that Ramsey^{dt} is not in PPP^{dt} (Theorem 1.6). In the reverse direction, we show that BiRamsey with a slightly weaker parameter is included in PAP (Lemma 4.2); combined with Theorem 1.6, we get an almost tight characterization of BiRamsey using the Pecking Order.

We start with reducing problems in the Pecking Order to Ramsey, which directly generalize a reduction that appeared in [KNY19]. For ease of discussion, we only state the following result in terms of Ramsey, yet the same argument also holds true for BiRamsey.

Lemma 4.1 (Generalization of [KNY19, Theorem 3.1])**.** Suppose there exists a graph on N vertices with no K clique or independent set. Then t-Pigeon_N^M can be black-box reduced to $t(K-1)$-Ramsey when $2t(K-1) \leq \log M$.

Proof. Given a graph $G_0(V_0, E_0)$ on N vertices such that it does not contain a K clique or independent set, we will build an instance G given by $(\log M, E)$ of $t(K-1)$-Ramsey from an instance C of t-Pigeon_N^M.

Let $h : [M] \mapsto [N]$ be an instance of t-Pigeon_N^M. We will define a $t(K-1)$-Ramsey instance G using the *graph hash product* from [KNY19]. Let $G = G_0 \otimes h = (V, E)$ be a graph on $V = [M]$ such that

$$(u, v) \in E \iff h(u) = h(v) \text{ or } (h(u), h(v)) \in E_0.$$

Now we prove that the solution S returned by $t(K-1)$-Ramsey witnesses a t-collision in h. Let S' be set $\{h(u) \mid u \in S\} \subseteq V(G_0)$. By the definition of G, S' is a clique if S is a clique, and S' is an independent set if S is an independent set. Therefore, S' is either a clique or an independent set. Since G_0 does not contain a K clique or independent set, we have $|S'| < K$. Given $|S| = t(K-1)$, by an averaging argument S must witness a $\frac{t(K-1)}{K-1} = t$ collision in h. □

Using the probabilistic method, Erdös [Erd47] shows that there exists a graph on N vertices with no clique or independent set of size $K = 2\log N$. This gives us the following theorem as a corollary:

Theorem 1.6. When $2t(2\log N - 1) \leq \log M$, t-Pigeon_N^M can be black-box reduced to Ramsey and BiRamsey .

We also note that by Theorem 2.5, we have that for some $c > 1$, we have an explicit efficient black-box reduction from t-Pigeon_N^M to Ramsey whenever $2t(\log^c N - 1) \leq \log M$. As a consequence of these reductions, we get two corollaries of Theorem 3.10 regarding the place of K-Ramsey in TFNP.

Theorem 1.7. $\text{Ramsey}^{dt}, \text{BiRamsey}^{dt} \notin \mathsf{SAP}^{dt}$. In particular, $\text{Ramsey}^{dt}, \text{BiRamsey}^{dt} \notin \mathsf{PPP}^{dt}$.

Proof. Theorem 1.6 shows that there exists some polynomial $p(n)$ such that $p(n)$-$\textsc{Pigeon}_N^M$ reduces to $\textsc{Ramsey}_M$ for large enough M. However, by Theorem 3.10, we know that there's no black-box reduction from $p(n)$-$\textsc{Pigeon}_N^M$ to $t(n)$-$\textsc{Pigeon}_{N'}^{M'}$ for any subpolynomial $t(n)$. This implies that $\textsc{Ramsey}$ is not in $t(n)$-PPP^{dt} for any subpolynomial $t(n)$. □

Actually, by being a little more careful with the parameters, we can prove a stronger version of the theorem following the same argument, which shows that even $\log^c(N)$-$\textsc{Ramsey}^{dt}$ is not in SAP^{dt}.

Finally, we show that with a slight loss in the parameter, $\textsc{BiRamsey}$ fits into the Pecking Order.

Lemma 4.2 (Generalized of Theorem 3.10 from [KNY19])**.** $\frac{n-\log n}{2}$-$\textsc{BiRamsey} \in \mathsf{PAP}$.

Proof. Given a bipartite graph $G = ([N], [N], E)$, we say $E(x, y) = 1$ if $(x, y) \in E$ and $E(x, y) = 0$ otherwise. We construct an n-$\textsc{Pigeon}_{N/n}^N$ instance using function $h : [N] \mapsto \{0, 1\}^{n-\log n}$ defined by

$$h(y) := (E(x, y))_{x\in[n-\log n]}.$$

Our goal is to prove that we can efficiently find a clique or independent set of size $(n - \log n)/2$ from an n-collision in h. Let $y_1, \ldots, y_n \in [N]$ be an n-collision in h. Then by the definition of h, we have for each $x \in [n - \log n]$, $E(x, y_1) = E(x, y_2) = \cdots = E(x, y_n)$.

At least half of the x's in $[n - \log n]$ will have the same value for $E(x, y_1)$, and we let these indices be $x_1 < \ldots < x_{(n-\log n)/2}$. This gives us that for all $i, j \in [(n - \log n)/2]$, $E(x_i, y_j) = E(x_1, y_j) = E(x_1, y_1)$. Therefore, $(\{x_1, \ldots, x_{(n-\log n)/2}\}, \{y_1, \ldots, y_{(n-\log n)/2}\})$ is either a biclique or an independent set.

□

Query complexity of Ramsey. Besides the relative complexity of $\textsc{Ramsey}$, one might also ask questions about its query complexity in various models (deterministic, randomized, quantum). To find a clique or independent set of size k, it certainly suffices to query all the vertices of an arbitrary subgraph of size $R(k, k)$. This gives us an upper bound on the deterministic query complexity of $\textsc{Ramsey}$ of $\binom{R(n/2,n/2)}{2}$. Plugging in the best known upper bound on the diagonal Ramsey number [CGMS23], we get an upper bound on $N^{2-\epsilon}$ for a small constant ϵ. On the lower bound front, we can infer that the quantum query complexity of $\textsc{Ramsey}$ is at least $N^{1-o(1)}$ due to the reduction in Lemma 4.1 combined with the quantum query lower bound by Liu and Zhandry [LZ19]. Since the best lower bound we have on $R(t, t)$ is $2^{\widetilde{O}(t/2)}$ [Erd47], improving even the deterministic lower bound significantly beyond $\Omega(N)$ would give a *new combinatorics result*! The best deterministic lower bound known is due to Conlon et al. [CFGH19]. We note this as an exciting approach to getting better lower bounds for the Ramsey numbers.

5 Relationship to Other Classes

5.1 Polynomial Local Search (PLS) and Polynomial Parity Argument (PPA)

Prior work in the literature has shown that $\mathsf{PPA}^{dt} \not\subseteq \mathsf{PPP}^{dt}$ [BCE+98] and $\mathsf{PLS}^{dt} \not\subseteq \mathsf{PPP}^{dt}$ [GHJ+22]. In this section, we prove that neither of these classes are contained in PAP^{dt}, recalled here.

Theorem 1.11. $\mathsf{PLS}^{dt} \not\subseteq \mathsf{PAP}^{dt}$ and $\mathsf{PPA}^{dt} \not\subseteq \mathsf{PAP}^{dt}$.

Both of these separations use the shared concept of *gluability*, which we first define.

Definition 5.1. Let C be a conjunction and let T be a decision tree. A *completion* of C by T is any conjunction of the form CC_ℓ where ℓ is any leaf of T.

Definition 5.2. A query total search problem $R \subseteq \{0,1\}^n \times O$ is (d,t)-*gluable* if for every conjunction C of degree at most d there is a depth $O(d)$ decision tree T_C such that the following holds. Let $C_1, C_2, \ldots, C_t$ be any sequence of t consistent conjunctions, and let $C'_1, \ldots, C'_t$ be any sequence of t conjunctions chosen so that C'_i is a completion of C_i by T_{C_i}. If C'_i is non-witnessing for each $i = 1, 2, \ldots, t$ and $C'_1 C'_2 \cdots C'_t$ is consistent, then $DT(R \restriction \rho(C'_1 C'_2 \cdots C'_t)) = \omega(\text{poly}(\log(n)))$.

A (slightly weaker) notion of gluability was introduced by Göös et al. [GHJ+22] in order to prove $\mathsf{PLS}^{dt} \not\subseteq \mathsf{PPP}^{dt}$, although, the idea underlying gluability already appeared implicitly in [BCE+98]. The reason for introducing gluability is that if a problem R is gluable *and* admits a non-witnessing pseudoexpectation operator $\widetilde{\mathbb{E}}$, then one can show automatically that $\widetilde{\mathbb{E}}$ is also collision-free, and Theorem 3.7 could be applied. We prove this now in a strong form.

Definition 5.3. Let $R \subseteq \{0,1\}^n \times O$ be a query total search problem, and let $\widetilde{\mathbb{E}}$ be a degree-d pseudoexpectation operator. Then $\widetilde{\mathbb{E}}$ is ε-*nonwitnessing for* R if the following property holds:

- **ε-Nonwitnessing.** $\widetilde{\mathbb{E}}[C] \leq \varepsilon$ for any degree-d conjunction C witnessing R.

Note that 0-nonwitnessing is synonymous with R-nonwitnessing (cf. Definition 3.1).

We now show that ε-weak pseudoexpectation operators for gluable problems are automatically collision-free for a suitable choice of parameters.

Lemma 5.4. Let $R \subseteq \{0,1\}^n \times O$ be any (d,t)-gluable query total search problem, let $0 \leq \varepsilon < 1$, and let $D \geq td^2$. Any degree-D, ε-nonwitnessing pseudoexpectation operator $\widetilde{\mathbb{E}}$ for R is also $(d, t, t-1+\varepsilon n^{O(d)})$-collision-free.

Proof. Let $\widetilde{\mathbb{E}}$ be the ε-nonwitnessing pseudoexpectation operator for R, and let $\mathcal{F}$ be any t-witnessing family of degree $\leq d$ conjunctions. Let $\mathcal{C}(\mathcal{F})$ be the family obtained by replacing each conjunction C in $\mathcal{F}$ with all of its completions CC_ℓ for $\ell \in L(T_C)$. We first observe that $\widetilde{\mathbb{E}}[\mathcal{C}(\mathcal{F})] = \widetilde{\mathbb{E}}[\mathcal{F}]$. To see this, consider any conjunction $C \in \mathcal{F}$. If T_C is the decision tree completing C in the definition of gluability, then an easy induction on the depth of the decision tree shows that

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

where the equality is between multilinear polynomials. This implies that $C = \sum_{\ell \in L(T_C)} CC_\ell$ and thus

$$\widetilde{\mathbb{E}}[C] = \sum_{\ell \in L(T_C)} \widetilde{\mathbb{E}}[CC_\ell].$$

It immediately follows that $\widetilde{\mathbb{E}}[\mathcal{F}] = \widetilde{\mathbb{E}}[\mathcal{C}(F)]$, since $\mathcal{C}(\mathcal{F})$ is obtained by replacing each conjunction C with its completions.

So, it suffices to consider $\mathcal{C}(\mathcal{F})$ instead. Since $\mathcal{F}$ was t-witnessing, $\mathcal{C}(\mathcal{F})$ is also t-witnessing. Let $W \subseteq \mathcal{C}(\mathcal{F})$ be the collection of all conjunctions in $\mathcal{C}(\mathcal{F})$ that are themselves witnessing, and let $X = \mathcal{C}(\mathcal{F}) \setminus W$ be the remaining conjunctions. We bound the weight of W and X separately.

We first bound the weight of W, as it is easier. Since every conjunction $C' \in W$ is witnessing, it follows that $\widetilde{\mathbb{E}}[C'] \leq \varepsilon$ since $\widetilde{\mathbb{E}}$ is ε-nonwitnessing for R. This means that $\widetilde{\mathbb{E}}[W] \leq |W|\varepsilon \leq \varepsilon n^{O(d)}$, since every conjunction in W has width at most $O(d)$.

We now bound the weight of X. As R is t-gluable, it follows that if we choose any sequence $C'_1, C'_2, \ldots, C'_t$ of conjunctions in X, either $C' = C'_1 C'_2 \cdots C'_t$ is itself inconsistent or $DT(R \upharpoonright \rho(C')) = \omega(\text{poly}(\log(n)))$. However, since $\mathcal{F}$ is t-witnessing, if C' is consistent then $DT(R \upharpoonright \rho(C')) \leq \text{poly}(\log(n))$. Therefore C' is inconsistent, and so $\widetilde{\mathbb{E}}[X] \leq t-1$ by Lemma 3.8. Combining the two weight bounds yields $\widetilde{\mathbb{E}}[\mathcal{C}(\mathcal{F})] = \widetilde{\mathbb{E}}[W] + \widetilde{\mathbb{E}}[X] = t - 1 + \varepsilon n^{O(d)}$, and the proof is complete. □

We now use gluability to prove lower bounds for PLS^{dt} and PPA^{dt}. To prove these lower bounds, we must briefly deviate through proof complexity, and in particular the connections between black-box TFNP classes and propositional proof systems. In particular, we will use the above lemma to prove the following theorem.

Theorem 5.5. Let $R = \{R_n \subseteq \{0,1\}^n \times O_n\}$ be a query total search problem, let $t = O(\text{poly}(\log(n)))$. If R is $(p(\log(n)), t)$-gluable for any polynomial function p and $R \notin \mathsf{PPADS}^{dt}$, then $R \notin t\text{-}\mathsf{PPP}^{dt}$.

To prove this theorem, we will need to define the *Sherali-Adams* proof system. To avoid introducing proof-complexity preliminaries, we will state the definition of the proof system directly in terms of the total search problems R, although we refer the interested reader to [GHJ+22] for technical details. If $R \subseteq \{0,1\}^n \times O$ is a query total search problem in TFNP, then we define a related unsatisfiable CNF formula $\neg\text{Total}(R)$ that encodes the (false) statement "R is not total". Formally,

$$\neg\text{Total}(R) := \bigwedge_{o \in O} \bigwedge_{\ell \in L_1(T_o)} \neg C_\ell$$

where $\{T_o\}_{o \in O}$ is the family of poly$(\log(n))$-depth decision trees witnessing solutions of R, and $L_1(T_o)$ is the set of 1-leaves of the decision tree T_o. We will be interested in *refutations* of the formula $\neg\text{Total}(R)$ — in other words, proofs of the tautology "R is total". Recall that a *conical junta* is any non-negative linear combination of conjunctions, that is an expression of the form $J = \sum_D \lambda_D D$, where each D is a conjunction and $\lambda_D > 0$. The *degree* of J is the maximum degree of any conjunction D appearing in the sum. The *magnitude* of J, denoted $||J||$, is $\max_D \lambda_D$.

Definition 5.6. Let $R \in \mathsf{TFNP}^{dt}$ be a total query search problem. A *$\mathbb{Z}$-Sherali-Adams proof of totality for* R is a sequence of conical juntas $\Pi = (J, J_\ell)_{o \in O, \ell \in L_1(T_o)}$ over the variables of R such that

$$-1 = \sum_{o \in O} \sum_{\ell \in L_1(T_o)} -J_\ell C_\ell + J$$

where we are working in multilinear polynomial algebra. The *degree* of the refutation Π is $\deg(\Pi) := \max\{\deg J_o C_o\}_o \cup \{\deg J\}$. The *magnitude* of the refutation Π is $||\Pi|| := \max\{||J_o||\}_o \cup \{J\}$.

Göös et al. [GHJ+22] proved that that $R \in \mathsf{PPADS}^{dt}$ if and only if it admits Sherali-Adams proofs with low degree and magnitude.

Theorem 5.7 ([GHJ+22])**.** Let $R \in \mathsf{TFNP}^{dt}$ be a total query search problem. Then $R \in \mathsf{PPADS}^{dt}$ if and only if for some constant c, R admits a $\log^c(n)$-degree, $n^{\log^c(n)}$-magnitude $\mathbb{Z}$-Sherali-Adams proof of totality.

One can show that *any* lower bound against degree-d Sherali-Adams with small magnitude implies the existence of a weak pseudoexpectation operator (in other words, ε-nonwitnessing operators are *complete* for unary Sherali-Adams lower bounds). The proof of this follows the usual proof of completeness for Sherali-Adams proofs via convex duality (see e.g. [FKP19]), and was first observed by Hubáček, Khaniki, and Thapen [HKT24].

Theorem 5.8 ([HKT24])**.** If there is no degree-d $\mathbb{Z}$-Sherali-Adams proof of totality for R of magnitude $\leq k$, then there is a degree-d, $1/k$-nonwitnessing pseudoexpectation operator for R.

We can now prove Theorem 5.5.

Proof of Theorem 5.5. We assemble all the ingredients that were introduced in this section. First, assume by contradiction that $R \in t\text{-}\mathsf{PPP}^{dt}$. This implies that for some constant C, there is a depth-$\log^C n$ decision-tree reduction from R to t-$\textsc{Pigeon}_N^M$ for $N = n^{\log^C n}$. However, since $R \notin \mathsf{PPADS}^{dt}$, it follows from the above theorem that for *every* constant d_0 there is no degree-$\log^{d_0} n$, $\mathbb{Z}$-Sherali-Adams proof of totality for R with magnitude $n^{\log^{d_0} n}$. Set $d = \log^{2C} n$, $D = td^2 = t\log^{4C} n$, and $\varepsilon = 1/n^D$. By Theorem 5.8, the non-existence of a $\mathbb{Z}$-Sherali-Adams proof implies that there is a degree-D, ε-nonwitnessing pseudoexpectation operator $\widetilde{\mathbb{E}}$ for R. Finally, since R is $(p(\log n), t)$-gluable for any polynomial function p, it is in particular $(\log^{2C} n, t) = (d, t)$-gluable.

Our choice of parameters implies that

$$\varepsilon n^{O(d)} = \frac{n^{O(\log^{2C} n)}}{n^{t\log^{4C} n}} = o\left(\frac{1}{N}\right),$$

where the last equality follows since $N = n^{\log^C n}$. By Lemma 5.4, the last two statements in the previous paragraph imply that the pseudoexpectation $\widetilde{\mathbb{E}}$ for R is a $(\log^{2C} n, t, t-1+o(1/N))$-collision-free pseudoexpectation operator. Applying Theorem 3.7, we have that R does not have a $\log^{2C} n$-depth reduction from R to t-$\textsc{Pigeon}_N^M$ for any $M \geq (t-1)N+1$ (note that when $M \geq (t-1)N+1$ it follows that $M/N \geq (t-1)+1/N \geq (t-1)+o(1/N)$ for sufficiently large n). This is a contradiction, and it follows that $R \notin t\text{-}\mathsf{PPP}^{dt}$. ☐

In the remainder of this section, we prove the required gluability results for PPA and PLS.

PPA^{dt} is gluable. We first introduce the defining problem for the class PPA^{dt}, called $\textsc{Leaf}$.

Definition 5.9 ($\textsc{Leaf}_n$)**.** This problem is defined on a set of n nodes, denoted by $[n]$, where the node 1 is "distinguished". For input, we are given a neighbourhood $N_u \subseteq [n]$ of size $|N(u)| \leq 2$ for each node $u \in [n]$. Given this list of neighbourhoods, we create an undirected graph G where we add an edge uv if and only if $u \in N(v)$ and $v \in N(u)$. We say $u \in [n]$ is a *leaf* if it has in-degree 1 and out-degree 0. The goal of the search problem is to output either

1. 1, if 1 is not a leaf in G, or *(no distinguished leaf)*
2. $u \neq 1$, if u is a leaf in G. *(proper leaf)*

The class PPA^{dt} contains all query total search problems with $\text{poly}(\log(n))$-complexity reductions to $\textsc{Leaf}$.

The seminal work by Beame et al. [BCE+98] proved that $\textsc{Leaf} \notin \mathsf{PPP}^{dt}$, which implies that $\textsc{Leaf} \notin \mathsf{PPADS}^{dt}$. Therefore, by Theorem 5.5, to prove $\mathsf{PPA}^{dt} \not\subseteq \mathsf{PAP}^{dt}$, we need to show that $\textsc{Leaf}_n$ is $(\text{poly}(\log n), \text{poly}(\log n))$-gluable.

Lemma 5.10. $\textsc{Leaf}_n$ is $(p(\log(n)), p(\log(n)))$-gluable for any polynomial p.

Proof. Let $d, t = \text{poly}(\log(n))$. Let C be any conjunction of degree d over the variables of $\textsc{Leaf}_n$. The decision tree T_C completing C does the following: if C ever queries a node $u \in [n]$, receiving a neighbourhood $N(u)$, T_C queries the nodes in $N(u)$ as well. In total, this requires $O(d)$ more queries. Now, let $C_1, C_2, \ldots, C_t$ be any sequence of t consistent conjunctions, and let $C'_1, C'_2, \ldots, C'_t$

be any sequence of t conjunctions chosen so that C_i' is a completion of C_i by T_{C_i}. Assume that these completions are all consistent and non-witnessing, and let $C' = C_1'C_2' \cdots C_t'$. First suppose by way of contradiction that C' witnesses a $\textsc{Leaf}_n$ solution u. But in either case, the fact that u is a solution must have been witnessed by some C_i' in the sequence, since we have explicitly queried all the neighbourhoods of nodes in each C_i. This is a contradiction, and thus C' must be non-witnessing.

Let us now show $\textsc{Leaf}_n \restriction \rho(C')$ has large decision tree depth. We can give a simple adversary argument as follows. Let U be the set of all nodes currently queried — initially, U contains all nodes queried by C', and thus $|U| = \text{poly}(\log(n))$. Say a node u is on the *boundary* of U if it has not been queried, but appears as a neighbour of a queried node. Consider a decision tree T querying nodes in $\textsc{Leaf}_n$. If T queries a node v not in the boundary of U, then output $N(v) = \emptyset$. Otherwise, suppose T queries a node v in the boundary of U. If two nodes $\{u_1, u_2\} \subseteq U$ have v in their neighbour set, then set $N(v) = \{u_1, u_2\}$. Otherwise, if just one node $u \in U$ has v in its neighbour set, then set $N(v) = \{u, w\}$, where w is any node not in U nor the boundary of U. Since the degree of every node is ≤ 2 and $|U| = \text{poly}(\log(n))$, we can clearly continue this adversary strategy for $\Omega(n)$ queries. Therefore $DT(\textsc{Leaf}_n \restriction \rho(C')) = \Omega(n)$, and thus $\textsc{Leaf}_n$ is $(\text{poly}(\log(n)), \text{poly}(\log(n)))$-gluable. $\square$

Corollary 5.11. $\mathsf{PPA}^{dt} \not\subseteq \mathsf{PAP}^{dt}$.

PLS^{dt} is gluable. Let us first recall the $\textsc{Sink-of-Dag}$ problem, also denoted $\textsc{SoD}$, which is the defining problem for PLS. Our definition follows that of Göös et al. [GHJ+22].

Definition 5.12 ($\textsc{Sink-of-Dag}$)**.** The $\textsc{SoD}_n$ problem is defined on the $[n] \times [n]$ grid, where the node $(1,1)$ is "distinguished". As input, for each grid node $u = (i,j) \in [n] \times [n]$, we are given an "active bit" $a_u \in \{0,1\}$. Furhter, we are given a *successor* $s_u \in [n]$, interpreted as naming a node $(i+1, s_u)$ on the next row. We say a node u is *active* if $a_u = 1$, otherwise it is *inactive*. A node u is a *proper sink* if u is inactive but some active node has u as a successor. The goal of the search problem is to output either

1. $(1,1)$, if $(1,1)$ is inactive *(inactive distinguished source)*
2. (n,j), if (n,j) is active, *(active sink)*
3. (i,j) for $i \leq n-1$, if (i,j) is active and its successor is a proper sink. *(proper sink)*

We define the class PLS^{dt} to be all total query search problems with $\text{poly}(\log(n))$-complexity $\textsc{Sink-of-Dag}$-formulations.

One of the main results of Göös et al. is the following.

Theorem 5.13 ([GHJ+22], Corollary 1.)**.** $\mathsf{PLS}^{dt} \not\subseteq \mathsf{PPADS}^{dt}$.

Göös et al. already showed that the $\textsc{SoD}_n$ problem is $(\text{poly}(\log(n)), 2)$-gluable, but it is not hard to see that their proof generalizes to show that it is actually $(\text{poly}(\log(n)), t)$-gluable for any $t = \text{poly}(\log n)$. The proof of the following lemma closely follows the argument of [GHJ+22, Lemma 13].

Lemma 5.14. $\textsc{SoD}$ is $(p(\log(n)), p(\log(n)))$-gluable for any polynomial p.

Proof. Let C be any conjunction of degree $d = \text{poly}(\log(n))$, defined over the variables of $\textsc{SoD}_n$. The decision tree T_C completing C starts by checking whether C queries any active node below row $n-d-1$. If yes, T picks any one such active node and follows the successor path until a sink is found, making the completion witnessing. Note that this step incurs at most $O(d)$ queries. Finally,

T ensures that any query to a successor variable in C is followed by a query to the active bit of the successor. This also costs $O(d)$ further queries. Thus the depth of T_C is $O(d) = \text{poly}(\log(n))$.

As in the definition of gluability, let $C_1, C_2, \ldots, C_t$ be any sequence of t consistent conjunctions, and let $C'_1, C'_2, \ldots, C'_t$ be any sequence of completions of those conjunctions by their respective decision trees. Suppose that C'_i is non-witnessing for each i and that $C'_1, C'_2, \ldots, C'_t$ are consistent, and suppose by way of contradiction that the conjunction $C' = C'_1 C'_2 \cdots C'_t$ is witnessing. If it reveals a SoD solution u of type (1) or (2), then it must be that for some i, C'_i queries an active bit of u: a contradiction with the fact that C'_i is non-witnessing. On the other hand, if C' reveals a solution u of type (3), then it must be that for some i, C'_i checks for the successor s_u of u, but the completion T_{C_i} forces this check to be followed by a query to the active bit of s_u, making one of the initial partial assignments witnessing as well. Hence C' is non-witnessing.

We finally argue that $R_n \restriction \rho(C')$ has query complexity greater than $\ell = \text{poly}(\log(n))$ by describing an adversary that can fool any further ℓ queries to p without witnessing a solution. Recall that p makes no queries to nodes below row $n-d-1$. The adversary answers queries as follows. If the successor pointer of an active node is queried, then we answer with a pointer to any unqueried node on the next row and make it active (there always exists one as $d \ll n$). If a node u is queried that is not the successor of any node, we make u inactive ($a_u = 0$ and s_u is arbitrary). This ensures that a solution can only lie on the very last row n, which is not reachable in ℓ queries starting from row $n-d-1$. □

We can thus infer Theorem 1.11, which we restate here for convenience.

Theorem 1.11. $\mathsf{PLS}^{dt} \not\subseteq \mathsf{PAP}^{dt}$ and $\mathsf{PPA}^{dt} \not\subseteq \mathsf{PAP}^{dt}$.

5.2 Iterated Pigeonhole

In this section, we discuss the relationship between the Pecking Order and classes PLC and UPLC defined by [PPY23]. We give an alternate definition of UPLC which is more convenient for our reductions. The class UPLC is defined by its complete problem with the same name, specified below[2].

Definition 5.15 (UPLC)**.** A universe of elements $\{0,1\}^n$ is considered.

Input A circuit $P : \{0,1\}^n \mapsto \{0,1\}^{n-1}$.

Solutions A set of n distinct strings $a_1 \ldots a_n$ such that for every j the strings $P(a_j), P(a_{j+1}) \ldots P(a_n)$ agree on the prefix of length j.

We get the above definition of UPLC by concatenating the circuits in the original definition [PPY23]. We refer to the type of solution reported by UPLC as the "lower-triangular condition". This is illustrated in Figure 2a.

Lemma 5.16. $n/2$-PIGEON$^{N}_{\sqrt{N}}$ is in UPLC.

Proof. Given an instance of $n/2$-PIGEON$^{N}_{\sqrt{N}}$, say $h : \{0,1\}^n \mapsto \{0,1\}^{n/2}$, we construct an instance P of UPLC by simply appending 0s to the circuit.

$$P(x_1 x_2 \ldots x_{n-1}) = h(x_1 \ldots x_{n/2}) 0^{n/2-1}.$$

The solution to UPLC gives us the lower triangular condition for some set of strings $\{a_0, a_1 \ldots a_n\}$ which in particular gives us an $n/2$-collisions for h, since $a_{\frac{n}{2}} \ldots a_n$ agree on the first $n/2$ bits. This proof is illustrated in Figure 2b. □

[2]The complete problem of UPLC is called UNARY LONG CHOICE in the original paper [PPY23].

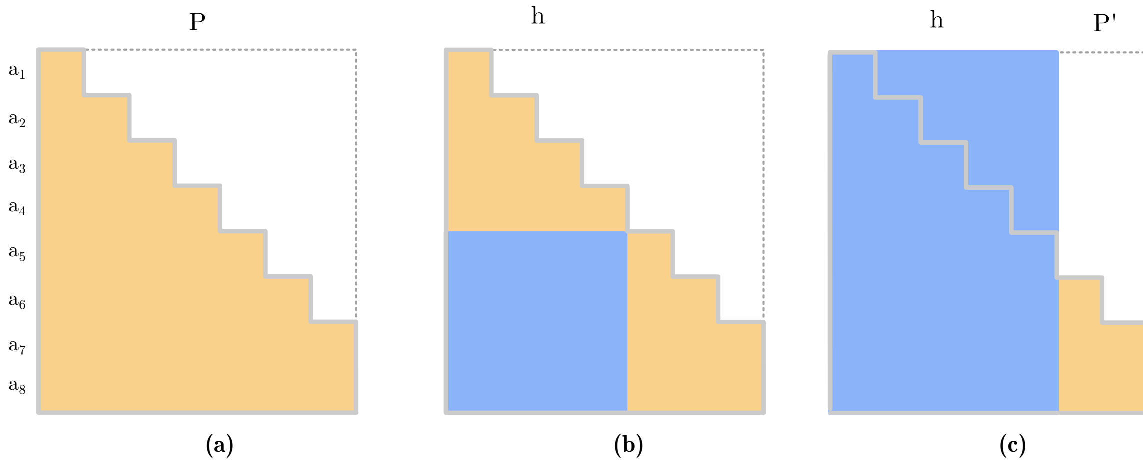

Figure 2: UPLC solutions and reductions. **(a)** illustrates the lower-triangular condition of solutions to UPLC. **(b)** is an illustration of Lemma 5.16. The blue shaded region is a valid solution to $n/2$-PWPP which is a subset of the yellow shaded region, the lower-triangular collision returned by UPLC. **(c)** is an illustration of Lemma 5.17. The blue shaded region is the collision returned by n-PIGEON, and the yellow shaded region is computed by solving a $\log n$ size UPLC instance by brute force.

Theorem 1.8. $\mathsf{UPLC}^{dt} \not\subseteq \mathsf{SAP}^{dt}$.

Proof. This is by combining Lemma 5.16 with Theorem 3.9. □

Further, we complete a fine-grained understanding of the position of UPLC in the Pecking Order by placing it in n-PWPP.

Lemma 5.17. UPLC $\subseteq n$-PWPP.

Proof. Without loss of generality, we assume n is a power of 2. Given a circuit $P : \{0,1\}^n \mapsto \{0,1\}^{n-1}$ of the UPLC instance, we "split" the output into two parts: the first $n - \log n$ bits are considered as an instance of n-$\text{PIGEON}^N_{N/n}$ and the remaining $\log n - 1$ bits are considered an instance of UPLC in $\log n$ scale.

More specifically, we first solve the n-$\text{PIGEON}^N_{N/n}$ instance defined by a mapping $h : \{0,1\}^n \mapsto \{0,1\}^{n-\log n}$, where h outputs the first $n - \log n$ bits of P. Note that n-$\text{PIGEON}^N_{N/n} \in n$-PWPP. Suppose we get an n-collision $(a_1, \ldots, a_n)$ of h in the first step.

We then consider the UPLC instance P' on the universe $U = \{a_1, \ldots, a_n\}$, where $P' : U \mapsto \{0,1\}^{\log n - 1}$ is defined by the remaining $\log n - 1$ bits of P. We can solve this much smaller UPLC instance in polynomial time, and get a set of solution $(a_{i_1}, \ldots, a_{i_{\log n}})$.

Finally, we get the solution to the original UPLC instance by rearranging the elements in U: we put $(a_{i_1}, \ldots, a_{i_{\log n}})$ in the end with the same order, and put everything else in before with arbitrary order. The first $n - \log(n)$ columns of the lower-triangular condition are satisfied by finding the n-collision of h, and the remaining $\log n - 1$ columns are fulfilled by solving the small UPLC instance P'.

This proof is illustrated in Figure 2c. □

Combining Theorem 1.12, Lemma 5.17, and the fact that PPP $\subseteq$ PLC [PPY23], we get two more black-box separations regarding to UPLC.

Theorem 1.9. $\mathsf{PPP}^{dt} \not\subseteq \mathsf{UPLC}^{dt}$. Consequently, $\mathsf{PLC}^{dt} \not\subseteq \mathsf{UPLC}^{dt}$.

Remark. Note that UPLC is total even if the universe has size $2^{n-1}+1$. We start with the universe $U_0 = [2^{n-1}+1]$. In the i-th step, the i-th bit of P divides the current set U_{i-1} into two subsets. We set U_i to be the larger one of them (and break ties arbitrarily). In this way, we have $|U_{n-1}| \geq 2$. Note that we do not pick an element in each step like we did in the case of PLC. Instead, once we have $U_1, \ldots, U_{n-1}$ ready, we can easily pick the solution sequence in the reverse order from U_{n-1} to U_1.

5.3 Total Function BQP

In this section, we put the problem defined by Yamakawa-Zhandry [YZ22] for their breakthrough result in the Pecking Order. Yamakawa-Zhandry's problem was first defined relative to a random oracle. It was later adapted to constitute a total problem ([YZ22], Section 6), which is the version we considered in this paper.

Theorem 5.18. Yamakawa-Zhandry's Problem is contained in PAP.

We start with formally defining the Yamakawa-Zhandry's problem.

Definition 5.19 (Yamakawa-Zhandry's Problem [YZ22], Simplified)**.** Fix an error correcting code $C \subseteq \Sigma^n$ on alphabet Σ. Let (h_k) be a family of λ-wise independent functions from C to $\{0,1\}^n$.

Input The input encodes a mapping $f : \Sigma \to \{0,1\}$. We abuse notation and define $f : \Sigma^n \to \{0,1\}^n$ as $f(a_1 a_2 \ldots a_n) = f(a_1) \cdot f(a_2) \cdot \ldots \cdot f(a_n)$.

Solutions The goal is to find a key k and t codewords $c^{(1)} \ldots, c^{(t)} \in C$ such that $f(c^{(i)}) \oplus h_k(c^{(i)}) = 0^n, \forall i \in [t]$.

The parameters satisfy $n < \lambda \ll t = \mathrm{poly}(n)$, $|C| \geq 2^{2n}$, and $|\Sigma| = 2^{\mathrm{poly}(n)}$.

This definition is slightly different to the one appeared in [YZ22][3], which is simpler to analyze. A *folded Reed-Solomon* code (see [GRS23], Section 17.1) with a certain parameter setting is chosen for the ECC C in [YZ22]. The structure of the ECC is crucial for the exponential quantum speed-up. It is straightforward to verify that the problem as stated here preserves the quantum upper bound and the classical lower bound. We assume the λ-wise independent functions family (h_k) is implemented by the well-known low-degree polynomial construction.

Definition 5.20 ([Vad12], Section 3.5.5)**.** Let $\mathbb{F}$ be a finite field. Define the family of functions $\mathcal{H} = \{h_{a_0, a_1 \ldots a_{\lambda-1}} : \mathbb{F} \mapsto \mathbb{F}\}$, where each $h_{a_0, a_1 \ldots a_{\lambda-1}} = a_0 + a_1 x + a_2 x^2 \ldots a_{\lambda-1} x^{\lambda-1}$ for $a_0, a_1 \ldots a_{\lambda-1} \in \mathbb{F}$.

It is noted in [Vad12] that this family forms a λ-wise independent set.

A further simplification. To show Theorem 5.18, we consider a simplification of the Yamakawa-Zhandry's Problem, in which we assume all the codewords have n distinct letters, and no letter appears in two different codewords. We call this new problem AllZeroColumn. The rationale behind the naming is that we can now imagine we are given an *arbitrary* 0/1 matrix F of size $n \times |C|$ representing the mapping f, where the i-th column of F corresponds to the result of applying f to the i-th codeword.

Definition 5.21 (AllZeroColumn)**.** Fix a 0-1 matrix family $\mathcal{H}$ of size $n \times m$, where the entries are λ-wise independent, implemented by the low-degree polynomial construction.

[3] In [YZ22], the input contains t mappings $f_1, \ldots, f_t$, and the goal is find a key k and t codewords $c^{(1)} \ldots, c^{(t)} \in C$ such that $f_i(c^{(i)}) \oplus h_k(c^{(i)}) = 0^n, \forall i \in [t]$.

Input The input is a 0-1 matrix F of size $n \times m$.

Solutions The goal is to find t indices $j_1, \ldots, j_t \in [m]$ and a matrix $H_k \in \mathcal{H}$ (which could be succinctly represented by a key k), such that for any $i \in [t]$, the j_i-th column of matrix $F \oplus H_k$ is $\mathbf{0}^n$.

The parameters satisfy $n < \lambda \ll t = \text{poly}(n)$, and $2^n \cdot t \le m = 2^{\text{poly}(n)}$.

As such, the Yamakawa-Zhandry's problem can be seen as AllZeroColumn with a *promise*: F but has the structure imposed by the ECC C in the definition of the Yamakawa-Zhandry's problem. Theorem 5.18 is then implied by the following lemma.

Lemma 5.22. AllZeroColumn is contained in PAP.

Proof. Given an AllZeroColumn instance specified by a 6-tuple $(n, m, \lambda, t, F, \mathcal{H})$, we construct the following t-Pigeon$_N^M$ instance for $M = m \cdot \frac{|\mathcal{H}|}{2^n}$ and $N = |\mathcal{H}|$.

Pigeons Each pigeon is a pair (j, k). Here j is in $[m]$, and k is a key for the family $\mathcal{H}$ such that the j-th column of matrix $F \oplus H_k$ is $\mathbf{0}^n$.

Holes Each hole is a key k.

Pigeon (j, k) is now mapped to hole k. Recall that we have $n < \lambda$, thus, each index j could pair with exactly 2^{-n} fractions of keys to form a valid pigeon. Hence, there are $M = m \cdot \frac{|\mathcal{H}|}{2^n}$ pigeons. Since $m \ge 2^n \cdot t$ in the AllZeroColumn instance, we can verify that $M \ge t \cdot N$. Also, any t-collision in the t-Pigeon$_N^M$ instance directly corresponds to a solution of the AllZeroColumn instance.

Since we implemented $\mathcal{H}$ using the low-degree polynomial construction Definition 5.20, this reduction can be implemented in poly(n) time in the white-box setting using polynomial interpolation. □

Acknowledgements

We thank Beach House for providing topical procrastination music [Hou15]. We thank Mika Göös, Scott Aaronson, David Zuckerman, Alexandors Hollender, Gilbert Maystre, Ninad Rajgopal, Chetan Kamath for discussions about the Pigeonhole principle. We also thank Scott Aaronson for suggesting the name Pecking Order. SJ thanks Mika Göös for countless insightful discussions about TFNP and complexity theory more generally.

SJ and JL are supported by Scott Aaronson's Vannevar Bush Fellowship from the US Department of Defense, the Berkeley NSF-QLCI CIQC Center, a Simons Investigator Award, and the Simons "It from Qubit" collaboration. ZX is supported by NSF award CCF-2008868 and the NSF AI Institute for Foundations of Machine Learning (IFML).

References

[AA14] Scott Aaronson and Andris Ambainis. The need for structure in quantum speedups. *Theory Comput.*, 10:133–166, 2014. doi:10.4086/toc.2014.v010a006.

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004. doi:10.1145/1008731.1008735.

[BB14] Arnold Beckmann and Samuel R. Buss. Improved witnessing and local improvement principles for second-order bounded arithmetic. *ACM Trans. Comput. Log.*, 15(1):Art. 2, 35, 2014. doi:10.1145/2559950.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. doi:10.1137/S0097539796300933.

[BCE+98] Paul Beame, Stephen Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. volume 57, pages 3–19. 1998. 27th Annual ACM Symposium on the Theory of Computing (STOC'95) (Las Vegas, NV). doi:10.1006/jcss.1998.1575.

[BDRV18] Itay Berman, Akshay Degwekar, Ron D. Rothblum, and Prashant Nalini Vasudevan. Multi-collision resistant hash functions and their applications. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 133–161. Springer, 2018. doi:10.1007/978-3-319-78375-8_5.

[BFH+23] Romain Bourneuf, Lukáš Folwarczný, Pavel Hubáček, Alon Rosen, and Nikolaj I. Schwartzbach. PPP-completeness and extremal combinatorics. In *14th Innovations in Theoretical Computer Science Conference*, volume 251 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 22, 20. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2023. doi:10.4230/lipics.itcs.2023.22.

[BFI23] Sam Buss, Noah Fleming, and Russell Impagliazzo. TFNP characterizations of proof systems and monotone circuits. In *14th Innovations in Theoretical Computer Science Conference*, volume 251 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 30, 40. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2023. doi:10.4230/lipics.itcs.2023.30.

[BJ12] Samuel R. Buss and Alan S. Johnson. Propositional proofs and reductions between NP search problems. *Ann. Pure Appl. Logic*, 163(9):1163–1182, 2012. doi:10.1016/j.apal.2012.01.015.

[BK94] Samuel R. Buss and Jan Krajíček. An application of Boolean complexity to separation problems in bounded arithmetic. *Proc. London Math. Soc. (3)*, 69(1):1–21, 1994. doi:10.1112/plms/s3-69.1.1.

[BKP18] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In *STOC'18—Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 671–684. ACM, New York, 2018. doi:10.1145/3188745.3188870.

[BM04] Josh Buresh-Oppenheim and Tsuyoshi Morioka. Relativized NP search problems and propositional proof systems. In *19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21-24 June 2004, Amherst, MA, USA*, pages 54–67. IEEE Computer Society, 2004. doi:10.1109/CCC.2004.1313795.

[CFGH19] David Conlon, Jacob Fox, Andrey Grinshpun, and Xiaoyu He. Online Ramsey numbers and the subgraph query problem. In *Building bridges II—mathematics of László Lovász*, volume 28 of *Bolyai Soc. Math. Stud.*, pages 159–194. Springer, Berlin, [2019] ©2019.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. volume 17, pages 230–261. 1988. Special issue on cryptography. doi:10.1137/0217015.

[CGMS23] Marcelo Campos, Simon Griffiths, Robert Morris, and Julian Sahasrabudhe. An exponential improvement for diagonal ramsey, 2023. arXiv:2303.09521.

[Coh16] Gil Cohen. Two-source dispersers for polylogarithmic entropy and improved Ramsey graphs. In *STOC'16—Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 278–284. ACM, New York, 2016. doi:10.1145/2897518.2897530.

[CZ19] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. *Ann. of Math. (2)*, 189(3):653–705, 2019. doi:10.4007/annals.2019.189.3.1.

[Erd47] Paul Erdös. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53:292–294, 1947. URL: https://api.semanticscholar.org/CorpusID:14215209.

[FGPR23] Noah Fleming, Stefan Grosser, Toniann Pitassi, and Robert Robere. Black-box PPP is not Turing-closed. In submission, 2023.

[FKP19] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Found. Trends Theor. Comput. Sci.*, 14(1-2):front matter, 1–221, 2019. doi:10.1561/0400000086.

[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. doi:10.1109/FOCS54457.2022.00111.

[GKRS19] Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In *10th Innovations in Theoretical Computer Science*, volume 124 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 38, 19. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019.

[GM08] Konstantinos Georgiou and Avner Magen. Limitations of the sherali-adams lift and project system: Compromising local and global arguments. Technical Report 587, University of Toronto, 2008.

[GP17] Paul W. Goldberg and Christos H. Papadimitriou. TFNP: an update. In Dimitris Fotakis, Aris Pagourtzis, and Vangelis Th. Paschos, editors, *Algorithms and Complexity - 10th International Conference, CIAC 2017*, volume 10236 of *Lecture Notes in Computer Science*, pages 3–9, 2017. doi:10.1007/978-3-319-57586-5_1.

[GRS23] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory (Draft)*. 2023. URL: https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/.

[HHRS15] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - tight lower bounds on the round and communication complexities of statistically hiding commitments. *SIAM J. Comput.*, 44(1):193–242, 2015. doi:10.1137/130938438.

[HKT24] Pavel Hubáček, Erfan Khaniki, and Neil Thapen. TFNP intersections through the lens of feasible disjunction. To appear at ITCS 2024, 2024.

[Hou15] Beach House. *PPP*. Aug 2015. URL: https://en.wikipedia.org/wiki/PPP_(song).

[Jeř09] Emil Jeřábek. Approximate counting by hashing in bounded arithmetic. *J. Symbolic Logic*, 74(3):829–860, 2009. doi:10.2178/jsl/1245158087.

[Jeř16] Emil Jeřábek. Integer factoring and modular square roots. *J. Comput. System Sci.*, 82(2):380–394, 2016. doi:10.1016/j.jcss.2015.08.001.

[JPY88] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? volume 37, pages 79–100. 1988. 26th IEEE Conference on Foundations of Computer Science (Portland, OR, 1985). doi:10.1016/0022-0000(88)90046-3.

[KNY18] Ilan Komargodski, Moni Naor, and Eylon Yogev. Collision resistant hashing for paranoids: dealing with multiple collisions. In *Advances in cryptology—EUROCRYPT 2018. Part II*, volume 10821 of *Lecture Notes in Comput. Sci.*, pages 162–194. Springer, Cham, 2018. URL: https://doi.org/10.1007/978-3-319-78375-8_6, doi:10.1007/978-3-319-78375-8_6.

[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), jul 2019. doi:10.1145/3341106.

[KoNT11] Leszek Aleksander Koł odziejczyk, Phuong Nguyen, and Neil Thapen. The provably total NP search problems of weak second order bounded arithmetic. *Ann. Pure Appl. Logic*, 162(6):419–446, 2011. doi:10.1016/j.apal.2010.12.002.

[KoT22] Leszek Aleksander Koł odziejczyk and Neil Thapen. Approximate counting and NP search problems. *J. Math. Log.*, 22(3):Paper No. 2250012, 31, 2022. doi:10.1142/S021906132250012X.

[Li23] Xin Li. Two source extractors for asymptotically optimal entropy, and (many) more. *CoRR*, abs/2303.06802, 2023. arXiv:2303.06802, doi:10.48550/arXiv.2303.06802.

[Li24] Jiawei Li. Total search problems with abundant solutions. To appear at ITCS 2024, 2024.

[LZ19] Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In *Advances in cryptology—EUROCRYPT 2019. Part III*, volume 11478 of *Lecture Notes in Comput. Sci.*, pages 189–218. Springer, Cham, 2019. doi:10.1007/978-3-030-17659-4.

[Mor01] Tsuyoshi Morioka. Classification of search problems and their definability in bounded arithmetic. *Electron. Colloquium Comput. Complex.*, TR01-082, 2001. URL: https://eccc.weizmann.ac.il/eccc-reports/2001/TR01-082/index.html, arXiv:TR01-082.

[MPW00] Alexis Maciel, Toniann Pitassi, and Alan R. Woods. A new proof of the weak pigeonhole principle. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 368–377. ACM, New York, 2000. doi:10.1145/335305.335348.

[Mül21] Moritz Müller. Typical forcings, NP search problems and an extension of a theorem of riis. *Ann. Pure Appl. Log.*, 172(4):102930, 2021. URL: https://doi.org/10.1016/j.apal.2020.102930, doi:10.1016/J.APAL.2020.102930.

[Pap94] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. volume 48, pages 498–532. 1994. 31st Annual Symposium on Foundations of Computer Science (FOCS) (St. Louis, MO, 1990). doi:10.1016/S0022-0000(05)80063-7.

[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*, volume 251 of *LIPIcs*, pages 88:1–88:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.88.

[Ram29] F. P. Ramsey. On a Problem of Formal Logic. *Proc. London Math. Soc. (2)*, 30(4):264–286, 1929. doi:10.1112/plms/s2-30.1.264.

[RV22] Ron D. Rothblum and Prashant Nalini Vasudevan. Collision-resistance from multi-collision-resistance. In *Advances in cryptology—CRYPTO 2022. Part III*, volume 13509 of *Lecture Notes in Comput. Sci.*, pages 503–529. Springer, Cham, [2022] ©2022. URL: https://doi.org/10.1007/978-3-031-15982-4_17, doi:10.1007/978-3-031-15982-4_17.

[ST11] Alan Skelley and Neil Thapen. The provably total search problems of bounded arithmetic. *Proc. Lond. Math. Soc. (3)*, 103(1):106–138, 2011. doi:10.1112/plms/pdq044.

[Tha02] Neil Thapen. A model-theoretic characterization of the weak pigeonhole principle. *Ann. Pure Appl. Logic*, 118(1-2):175–195, 2002. doi:10.1016/S0168-0072(02)00038-6.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Found. Trends Theor. Comput. Sci.*, 7(1-3):1–336, 2012. doi:10.1561/0400000010.

[Yue14] Henry Yuen. A quantum lower bound for distinguishing random functions from random permutations. *Quantum Inf. Comput.*, 14(13-14):1089–1097, 2014.

[YZ22] Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 69–74. IEEE, 2022. doi:10.1109/FOCS54457.2022.00014.

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Inf. Comput.*, 15(7-8):557–567, 2015.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il