Proposal Summary

This project investigates novel constructions of error-correcting codes supporting sublinear-time error-detection, sublinear-time error-correction and efficient list-decoding, as well as their applications in the areas of complexity theory and pseudorandomness. The project builds upon several recent successes of the PI, such as the construction of new high rate error-correcting codes allowing, for the first time, sublinear-time error-correction.

The classical theory of error-correcting codes by Shannon and Hamming has developed into a flourishing subject, and has been hugely influential in the design of communication and storage systems. However the more modern aspects of error-correcting codes, including those relevant to complexity theory and pseudorandomness, have lagged behind, and there is a lot here that is very poorly understood. This project aims to remedy this situation by systematically exploring what can be achieved in the realm of local-decoding, local-testing, list-decoding and local list-decoding, and by exploring the implications of this in complexity theory and pseudorandomness.

Specific goals of this research project include:

- Locally testable and locally decodable codes of high rate with polylogarithmic query complexity,
- Probabilistically checkable proofs of linear length, checkable in sublinear time,
- Hardness amplification converting worst-case hard functions to average-case hard functions with negligible change in the input size,
- Strong average case circuit lower bounds against the circuit class $\mathrm{AC}^0[\oplus]$,
- Improved constructions of randomness extractors.

Intellectual Merit: Error-correcting codes, complexity theory and pseudorandomness have had very productive interaction, which underlies some of the basic results in all these fields. Motivated by recent developments in coding theory by the PI, this project will revisit this interaction, develop new error-correcting codes and coding-theoretic tools, to use them to address fundamental problems in complexity theory and pseudorandomness.

Broader Impact: One aspect of the broader impact of the project will be the error-correcting codes and algorithms developed. These have the potential to be applied to real-world data storage applications, which is very relevant to current technology. The educational component of this project will involve the mentoring and education of junior researchers who intend to pursue their own careers in research, including hopefully some women and minorities. This project will also develop courses, and make the course materials publicly available. The PI and junior researchers will actively seek out broad dissemination of the progress in research, by presenting the research and its outcomes in seminars at leading conferences, workshops, and academic and industrial research institutions. Finally, this project will strengthen connections between Computer Science, Electrical Engineering and Mathematics (in particular between complexity theory, coding theory, information theory, algebra and additive combinatorics).

Keywords: Error-Correcting Codes, Sublinear-time Algorithms, List-Decoding, Complexity Theory, Proof Systems, Pseudorandomness, Randomness Extraction.

CAREER: Title omitted

Name Omitted

1 Introduction

This project investigates novel constructions of error-correcting codes supporting sublinear-time error-detection, sublinear-time error-correction and efficient list-decoding, as well as their applications in the areas of complexity theory and pseudorandomness.

The classical subject of error-correcting codes was created in the 1940s by Shannon and Hamming, anticipating the need for greater reliability for the looming digital revolution. It has since developed into a flourishing discipline playing a vital part in the design of communication and storage systems. In modern times, this subject has been reinvigorated through interaction with computer science. One the one hand, the theory and practice of error-correcting codes has benefited greatly from the algorithmic insights and methods of computer science; on the other hand, key notions involving error-correcting codes have played a vital role in modern results of complexity theory and pseudorandomness. This two-way interaction is founded on some amazing coding-theoretic phenomena, such as the ability to detect and correct errors in sublinear time, as well as some deep interconnections, such as the close relationship between error-correction and the conversion of approximate solutions to exact solutions.

Recent developments by the PI in the theory of error-correcting codes related to sublinear-time error-correction and list-decoding have given a glimpse of some unexpected phenomena. For example, a recent result of the PI showed, for the first time, that there could be error-correcting codes of high rate which support sublinear time error-correction. These phenomena have the potential to translate into powerful applications in complexity theory and pseudorandomness. This project will pursue these and other applications, while at the same time further exploring the basic questions in error-correcting codes related to sublinear-time algorithms and list-decoding. Specific research goals of this project include:

1. Systematically investigating what can be algorithmically done in sublinear-time with high-rate error-correcting codes. This includes developing new codes and algorithms, as well as understanding the limitations.

2. Exploring the applicability of such error-correcting codes to real-world computer systems, and developing the necessary codes and algorithms to enable this.

3. Understanding the power of such high-rate codes for complexity theoretic applications. Specifically,

(a) We will work towards developing a theory of high-rate (or low-redundancy) probabilistic proof systems, and studying the amount of redundancy needed in Probabilistically Checkable Proofs and Interactive Proofs.

(b) We will investigate what kinds of hardness amplification and random self-reductions are possible in the high-rate regime.

4. Searching for new methods for proving correlation bounds against low-degree polynomials, of the kind relevant for average-case $\mathrm{AC}^0[\oplus]$ circuit lower bounds.

5. Investigating new approaches to constructing randomness extractors, perhaps via their strong relationship to list-decodable error-correcting codes.

6. Constructing/disproving the existence of very high rate locally testable codes, of the kind relevant for the Small Set Hypothesis and the Unique Games Conjecture.

As I will describe below, it is now especially apt to conduct such a research program, given the current state of knowledge in the area, and given recent exciting and promising developments. The other main aspect of this proposal is the educational aspect, which will involve mentoring and advising young researchers, as well as developing course materials for new courses.

The rest of this proposal is organized as follows. In the next section we explain the context for this research. In Section 3 we give a detailed description of the questions investigated by the proposed research. In Section 4 we outline our educational plan, and how it integrates with our research proposal. In Section 5 we describe the broader impact of the research and educational aspects of our proposal. Finally we summarize some of the previous research accomplishments of the PI.

2 The Context

Before we can describe the concrete questions addressed by this proposal, we explain some of the context motivating these questions.

An error-correcting code is given by an *encoding map* $E : \{0,1\}^k \to \{0,1\}^n$, which "encodes" strings of length k into strings of length n (everything can also be done with $\{0,1\}$ replaced by any finite set Σ; we stick to $\{0,1\}$ for this discussion). The image of this map is called the code, which we will denote by $\mathcal{C}$, and its elements are called codewords. The main measures of the quality of an error-correcting code are its rate R and its minimum distance δ. The rate R is defined to be k/n, which measures the redundancy/wastage introduced in the encoding. The minimum distance $\delta \in (0,1)$ is defined to the be the smallest (fractional) Hamming distance ($\Delta(\cdot,\cdot)$) between two distinct elements of $\mathcal{C}$. The trivial but key observation that underlies the theory is that if one is give $r \in \{0,1\}^n$ which is at Hamming distance at most $\delta/2$ of a codeword $c \in \mathcal{C}$, then c is *uniquely* determined. Thus if one designs an error-correcting code with large rate R and large minimum distance δ, one can use it to encode data with very little wastage of space, while at the same time, protect it against a large fraction of errors.

One of the fundamental research programs undertaken by classical coding theory, and completed by the 1970s, was to construct efficiently encodable and efficiently decodable error-correcting codes which have both rate and minimum distance being $\Omega(1)$. Not only were the so-developed codes useful for actual error-correcting applications, these codes were based on using algebraic and combinatorial objects such as polynomial, finite fields and randomization for constructing discrete structures, a paradigm which held great promise for further applicability.

Since the late 1980s, error-correcting codes and the paradigms for constructing them found great impact in theoretical computer science. In particular, error-correcting codes based on polynomials played a central role in the development of Interactive Proofs, Probabilistically Checkable Proofs (PCPs), cryptographic hard-core bits, hardness amplifiers and pseudorandom generators. The centerpiece of all these developments was the fact that a multivariate low-degree polynomial over a finite field could be locally interpolated at a point x by looking at the values taken by that polynomial on all other points of any line passing through x. This endowed the evaluation table of a low-degree multivariate polynomial with some local robustness; errors can be corrected by only looking at a few other entries of the table.

Motivated by this, one can define a *locally decodable code* as an error-correcting code equipped with a (randomized) decoding algorithm, which when given as input a received word $r \in \{0,1\}^n$ which is with distance $0 < \delta_0 < \frac{\delta}{2}$ of a codeword, and a message coordinate $i \in [k]$, the algorithm looks only at $o(k)$ entries of r and returns the "correct" message bit m_i with high probability (i.e., if m is the unique codeword such that $\Delta(E(m), r) < \delta_0$, then the algorithm returns m_i with high probability). Similarly one can define *locally testable codes*, which come with a testing algorithm that with high probability distinguishes, using few queries, between a given received word being within distance ϵ_1 of some codeword, and being further than distance ϵ_2 of every codeword. This ability to work with error-correcting codes in *sublinear-time* formed the conceptual heart of the various developments in theoretical computer science mentioned above.

Another concept from coding theory that has played a key role in recent developments has been *list-decoding*. Here one wants to recover from as large a fraction of errors as possible. An immediate obstacle is that once we go beyond $\delta/2$ fraction errors, the original message m need not be uniquely determined given the received word r. Thus we need to settle for a *list* of candidate messages. As long as this list has bounded size, we may hope to find this list in a reasonable amount of time. Actually doing this is a highly nontrivial algorithmic task, and until recently, it was not known how to do this efficiently for any nontrivial code. The first nontrivial list-decoding algorithm was given by Sudan [41], and it was for error-correcting codes based on univariate polynomials (Reed-Solomon codes). This algorithm eventually played an important role in the construction of hardness amplifiers [42], pseudorandom generators [43, 38] and randomness extractors [38]. This is an example of a fundamental contribution made to coding theory by theoretical computer science, which eventually fed back to have a big impact in theoretical computer science itself.

2.1 Recent Developments

In recent years, the notions of local-decoding, local-testing and list-decoding have been studied quite a bit.

For local decoding, a large amount of work has gone into codes which are locally decodable with a constant number of queries [25, 3, 46, 16], and today we know codes with subexponential length achieving this. Very recent work of the PI and coauthors [31] has explored the other end of the spectrum, that of codes with linear length (and hence constant rate), and showed that there exist codes of rate approaching 1 which are locally decodable with k^ϵ queries for every $\epsilon > 0$. The codes constructed in [31] achieving this are called multiplicity codes, and they are closely related to the ubiquitous polynomial codes. A large part of this research project is motivated by the natural conjecture that using multiplicity codes can significantly improve many of the known applications of polynomial codes across theoretical computer science.

Despite their superficial similarity to locally decodable codes, the theory of locally testable codes has evolved in a significantly different way. In fact, progress on locally testable codes has come from corresponding advances in the theory of PCPs. After a large number of works [23, 8, 4, 35, 7, 12], we now know that there are locally testable codes, testable with a constant number of queries, which have only slightly subconstant rate. However the only known locally testable codes of constant rate require $\Omega(k^{\epsilon})$ queries for testing. Some of the questions in this research project hope to advance on this state of affairs. In a different regime of parameters, the PI and coauthors [9] showed that there were codes of very high rate $(1 - o(1))$ which could be locally tested for just a constant number of errors, using ϵk queries to the received word (for any $\epsilon > 0$). These codes played an important role in some recent advances [2] on inapproximability and the Unique Games Conjecture.

List-decoding has also been intensely studied in past few years. Fundamental breakthroughs [18, 36] in list-decoding of algebraic codes culminated in the basic result of Guruswami and Rudra [22] giving explicit constructions of error-correcting codes achieving "list-decoding capacity": they have rate R and can be list-decoded from (the maximum possible) $1 - R - \epsilon$ fraction errors in polynomial time (for any $\epsilon > 0$). Recently, the PI showed [28] that multiplicity codes can *also* achieve list-decoding capacity. Another result of the PI [21] showed that random linear codes achieve list-decoding capacity with optimal list-size.

These and other developments suggest some exciting and promising research directions which we describe in the following section.

3 Proposed Research

3.1 High Rate Locally Decodable Codes and Locally Testable Codes

While the original reason for considering locally decodable codes and locally testable codes was for applications in complexity theory (such as PCPs and interactive proofs), the most natural application for such error-correcting codes ought to be for error-correction itself! Although we have known such codes for a long time, they never made an impact on actual error-correction systems because all these codes had very poor rates (there was no known locally decodable code with rate $> 1/2$). Indeed, codes used in the real-world for error-correction all have rate very close to 1. It was also widely believed that locally decodable codes could not have rate close to 1 (for example, --- [14] conjectured that locally decodable codes of rate $> 1 - \epsilon$ could not exist, even if they had to correct only a subconstant fraction of errors).

Recently I (along with my coauthors) [31] made significant progress on this problem (and in particular refuted ---'s conjecture). We constructed a new family of error-correcting codes called **Multiplicity Codes**. These codes are very natural and are closely related to polynomial codes. Multiplicity codes are based on evaluations of high-degree polynomials and their derivatives; the inclusion of the derivatives allows one to consider polynomials of total degree larger than the size of the underlying finite field, thus going beyond the range handled by the Schwartz-Zippel lemma.

Formally, the order-s multiplicity code of degree-d m-variate polynomials over $\mathbb{F}_q$ is the code defined as follows: for each m-variate polynomial over $\mathbb{F}_q$ of degree at most d, there is a codeword, whose symbols consist of the evaluations of that polynomial and all its partial derivatives up to order s at each point of $\mathbb{F}_q^m$.

The main property achieved by multiplicity codes is the following: for every $\alpha, \epsilon > 0$ there is a multiplicity code of rate $1 - \alpha$, which allows for local decoding from some constant fraction of errors in sublinear time $O(k^\epsilon)$. The key points are that (1) α could be arbitrarily small, and (2) there is absolutely no relationship between ϵ, α. For previously known codes, to get anything nontrivial whatsoever ($\epsilon < 1$), we needed the rate to be $< 1/2$, and furthermore, once we do have rate $R < 1/2$, the query complexity of decoding had to be at least $\Omega(k^{\epsilon_R})$. Thus multiplicity codes even achieved, for the first time, rate $\Omega(1)$ while allowing for local decoding in time k^ϵ for arbitrary $\epsilon > 0$ (independent of the rate).

This raises many exciting possibilities. Now that we know that it is possible to have rate very close to 1 while having nontrivial local decodability, it makes sense to ask how small we can make the decoding query complexity/time. In particular, I hope to investigate the following fundamental question.

Question 1 *Do there exist locally decodable codes of rate $\Omega(1)$ (or even $1 - \epsilon$, for every $\epsilon > 0$) with polylogarithmic query complexity?*

By results of Katz and Trevisan [25], it is known that the query complexity for a constant rate code cannot be sublogarithmic. Multiplicity codes show that there are codes of rate $\Omega(1)$ with query complexity $O(k^\epsilon)$ for every ϵ (independent of the rate). It is also known that there are codes of polynomially-small rate $\Omega(k^{-\epsilon})$ which are locally decodable with polylogarithmic query complexity (these codes are the classical polynomial-based Reed-Muller codes). The question above is tantalizingly placed between them. A positive answer to this question (with codes of rate close to 1) could have tremendous applications to real-world error correction.

Another closely related question asks something similar for locally testable codes.

Question 2 *Do there exist locally testable codes of rate $\Omega(1)$ with constant query complexity?*

This question is intimately related to the existence of linear size PCPs, which we discuss in a later section.

The state of the art for locally testable codes is noticeably different from the state of the art for locally decodable codes. Using technology from the theory of PCPs, it was shown by Ben-Sasson and Sudan [7] and Dinur [12] (see also Meir [34]) that there exist locally testable codes of rate $\frac{1}{\text{poly}\log k}$ which are locally testable with *constantly* many queries. This implies that the query complexity of local testing is provably better than that for local decoding. However, for codes of rate $\Omega(1)$, nothing better than the $O(k^\epsilon)$-query decoding was known. Recently, Viderman [45] showed that one could get locally testable codes of rate $1 - \alpha$ testable with k^ϵ queries, for arbitrary $\alpha, \epsilon > 0$ (in that work, multiplicity codes are cited as inspiration for exploring this regime of parameters, although the techniques are unrelated).

One technique that is available in the case of locally testable codes and not for locally decodable codes is proof-composition, a tool from the theory of PCPs. However, all known methods for proof composition end up losing some superconstant factors in the rate. I believe that this intermediate goal, of developing a composition method that does not hurt the rate, is a worthwhile pursuit that may have applications to other settings.

A closely related, and more approachable, question is whether multiplicity codes are locally testable.

Question 3 *For every $\alpha, \epsilon > 0$, are there multiplicity codes of rate $1 - \alpha$ locally testable with $O(k^\epsilon)$ queries?*

This would be especially relevant for applications, since it would provide a *single* code of high-rate which is both locally testable and locally decodable.

Due to their strong relationship to polynomial codes, one would suspect that the answer is yes (and there is a natural candidate local test for this), but there seem to be some interesting challenges to proving this. In particular, the naive adaptations of the classical local testers for polynomial codes, in the setting of order-s multiplicity codes of m-variate degree-d polynomial codes over $\mathbb{F}_q$ require $q > \frac{d}{s} \cdot m$, which immediately makes the rate bounded strictly below 1.

3.2 Using Multiplicity Codes for Error-Correction

One of the main potential uses of high rate locally decodable codes is for real computer systems, for storing data to protect against errors. Ideally, error-correcting schemes are completely transparent to other aspects of the computer system; when data is needed for use, the scheme should be able to retrieve that data essentially as fast as it takes to read that data in an error-free setting.

Classical error-correcting codes encode k bits of data into $O(k)$ codeword bits, and to retrieve any one bit of the original (corrected) message, the entire codeword is processed and decoded, resulting in a processing time that is at least $\Omega(k)$. Thus more having more data increases the access time proportional to the length of the data, which would not do for real applications at all. Instead, practitioners today divide the data up into small blocks, and encode each block separately. Now the access time for bits is reduced back to a constant, but the resilience to error has drastically reduced from a constant fraction of all the bits to a constant *number* of bits.

Locally decodable codes are ideal for this situation. They allow one to encode the entire data, thus keeping the resilience against a constant fraction of errors, while retaining easy access to bits of the original data.

The simultaneously high rate and local decodability of multiplicity codes makes them a very suitable candidate for such applications (in fact, it is the only known code achieving these properties). Thus it is natural to explore all the favorable properties of multiplicity codes with respect to error-correction. We pose some of these questions below.

Question 4 *Can multiplicity codes of minimum distance δ be deterministically decoded from $\delta/2$ fraction errors in near-linear time?*

Such a deterministic decoder for the case of univariate multiplicity codes could be used to speed up the local decoder for general multiplicity codes (since decoding univariate multiplicity codes shows up as a subroutine in the local decoding algorithm).

Apart from quick decoding from up to half the minimum distance, it is useful to have the ability to recover from even more errors, perhaps at the cost of a slightly slower decoding algorithm.

Question 5 *Can multiplicity codes of minimum distance δ be list-decoded from $(\delta - \epsilon)$ fraction errors in polynomial time?*

In [28], I studied some aspects of the list-decodability of multiplicity codes. One result that I showed there was a positive answer to the above question for univariate multiplicity codes over prime fields. This has the following strong consequence: *multiplicity codes can achieve list-decoding capacity!* Thus the one family of error-correcting codes which has the best known high-rate local decodability also has the largest possible list-decoding radius. I find the fact that both these fundamental properties appear in one family of error-correcting codes the strongest evidence that multiplicity codes have a lot of potential in error-correcting applications.

Going beyond this, one could even ask if multiplicity codes achieve list-decoding capacity with optimal list-size.

Question 6 *Can multiplicity codes of minimum distance δ be list-decoded from $(\delta - \epsilon)$ fraction errors with constant list-size?*

Answering this question about list-decoding would automatically translate into a corresponding algorithm for "local list-decoding", which would have applications for hardness amplification (described in a later section).

3.2.1 Practical Issues

While this is a theoretical research project, I think it is extremely important to also keep the models, questions and directions grounded to reflect qualitative aspects of the underlying practical issues. That is why I also hope to address the following question.

Question 7 (Informal) *Explore the algorithmic questions encountered by practitioners in implementing error-correction systems based on multiplicity codes.*

Now we mention one example of a concrete question that arises from this direction. After discussions with a number of data storage practitioners, I found that one potential application of locally decodable codes is to data centers, where there are large array's of hard disks storing data, and the major kind of failure to combat is an entire hard disk failing. The ideal mode for error correction here is to treat each hard disk as one symbol from a (gigantic) alphabet, and whenever a hard disk fails, we use the other hard disks to recover the contents of that disk. Two key aspects of this model which differ from the locally decodable error-correction model are (1) the kinds of failures are erasures (not errors), and (2) it is much easier to check if an erasure happened in a given coordinate than to read the symbol in that coordinate. In this setting, how can we locally decode multiplicity codes? Here we know all the erased locations of the codeword, and the algorithmic question here is to decide which of the remaining symbols to read in order to recover the value of a given location. Maybe in this setting a larger number of erasures can be handled? Undoubtedly there are many more theoretically-meaningful algorithmic questions that arise while looking at the practical aspects, and I hope to formulate them concretely and tackle them over the course of this project.

3.3 High Rate PCPs and Interactive Proofs

A prime place where high rate codes supporting local algorithms could be useful is in probabilistic proof systems, such as PCPs and Interactive Proofs. Probabilistic proof systems are based on encoding statements to be verified in a certain format, so that any cheating in the proof can be caught easily, either interactively or non-interactively. Invariably, the encodings employ error-correcting codes (to see at least why error-correcting codes are relevant, they map distinct messages to codewords that are far apart in Hamming distance, so that their distinctness can be noticed easily).

The PCP Theorem states (in one of its many equivalent forms) that for every language $L \in \mathsf{NP}$, there is a format for writing a polynomial-length proof of membership for that language, such that membership of a given input x in L can be verified by a probabilistic verifier by making only $O(1)$ queries into the proof. The PCP theorem has had a tremendous impact on a wide array of topics across theoretical computer science, in particular on the theory of approximation algorithms and inapproximability.

The PCP theorem directly implies that, unless $\mathsf{3SAT}$ has a $2^{o(n)}$ time algorithm, one cannot get a $(1 - \epsilon_0)$ approximation to $\mathsf{MAX3SAT}$ in $2^{n^{\epsilon_1}}$ time, for some absolute constants $\epsilon_0, \epsilon_1 > 0$. In recent years, there has been much research on the improving length of PCPs as a function of the length of the witness of the original NP language (this can be viewed as improving the "rate" of the PCP). In the application to inapproximability of $\mathsf{MAX3SAT}$, this corresponds to increasing the absolute constant ϵ_1.

Given that we now know that even codes of very high rate can support sublinear time decoding and testing, it seems natural to ask if we can translate this into its PCP analogue. To this end, we would like to investigate the following well-known open question.

Question 8 *Do there exist PCPs of linear length, checkable with $O(1)$ query complexity?*

This would translate into obtaining the optimal $\epsilon_1 = 1$ in the above mentioned inapproximability of $\mathsf{MAX3SAT}$, and would imply sharp time complexity lower bounds for a number of other fundamental optimization problems.

While answering this question will likely require a number of new ideas, the following question seems to be much more directly related to the recent advances on codes.

Question 9 *Do there exist PCPs of linear length with $O(n^\epsilon)$ query complexity?*

A positive answer to this question would rule out the possibility of a $2^{o(n)}$-time $(1-\epsilon_0)$-approximation algorithm for $\mathsf{MAX}n^\epsilon\mathsf{CSP}$. This would be the first essentially tight time-complexity lower bound (albeit conditional) for a natural approximation problem.

Traditional PCPs based on polynomial codes use more than just their local-testability and local-decodability; they take advantage of the inherent algebra of the code to perform an *arithmetization* of the NP-complete problem at hand. The fact that multiplicity codes are also based on polynomials is very encouraging; it suggests that arithmetization may be possible with multiplicity codes too. However, in order to obtain a high rate PCP or even a constant rate PCP, the traditional versions of arithmetization do not suffice, and some sort of high rate version of arithmetization seems to be needed. This seems like an exciting possibility to explore, and I believe that such a tool could be useful in other contexts too.

Another setting where we will explore the possibility of using high-rate codes is in the theory of interactive proofs. The fundamental IP = PSPACE theorem of [33, 39] states that for every language L in PSPACE, membership in L can be proved by a prover P to a verifier V by an interactive proof involving a polynomial amount of communication between P and V.

Here too, encoding via polynomial codes, local-decoding of these codes and arithmetization, are all important tools for the known proof of this theorem. Thus we believe that using multiplicity codes might be a way to make progress on the following basic question.

Question 10 *How much communication and randomness is required to conduct an interactive proof for a language in PSPACE? In particular, can the total communication and randomness of an interactive proof for a Totally Quantified Boolean Formula (TQBF) φ be made at most $O(|\varphi|)$?*

A beautiful recent result [19] gave some very strong results on the resources required for interactive proofs of languages in uniform NC. The techniques there were also based on polynomial encodings, and perhaps they may also admit some improvements via multiplicity codes.

3.4 High rate hardness amplification

The classical Impagliazzo-Wigderson theorem [24], part of the hardness vs. randomness paradigm, shows that if certain complexity lower bounds held, then randomized algorithms are no more powerful than deterministic algorithm. A key component of the proof is hardness amplification: given a function $f : \{0,1\}^n \to \{0,1\}$ which is hard in the worst-case, one can produce from it a function $f' : \{0,1\}^{n'} \to \{0,1\}$ which is hard in the average-case (i.e., hard to compute on a uniformly random input). Apart from being useful for hardness vs. randomness, that fact that hardness amplification can be done is a basic result about the complexity of computation.

We now know that hardness amplification is intimately related to error-correcting codes via the notion of "local list-decoding" (the ability to list-decode from a large fraction of errors in sublinear time) [42]. The existence of an efficiently encodable error-correcting code with a good local list-decoding algorithm immediately gives a hardness-amplifying transformation. The rate of the error-correcting code directly translates into the number of input bits of the hard on average function (the relationship between n' and n in the above notation), the efficiency of the encoding translates into the complexity of the hard on average function, and the fraction of errors recoverable from translates into the hardness of the hard on average function.

We hope to make progress on the following basic question.

Question 11 *Let $\rho > 0$. What is the smallest n' (as a function of n) for which there is a hardness amplifying transformation taking a worst-case hard function on n input bits to an average case hard function on n' bits which cannot be computed correctly on more than $\frac{1}{2} + \rho$ fraction of the inputs?*

Equivalently, what is the smallest rate binary locally list-decodable code which can be efficiently encoded and also locally list-decoded from $1/2 - \rho$ fraction errors using polylogarithmic time and queries? This question asks the list-decoding analogue of Question 1, and perhaps lower bounds here will be easier than for Question 1.

A more immediate goal, is to understand what kind of hardness amplification follows from multiplicity codes. We have some preliminary results in this direction [28], which show that high rate multiplicity codes have good local-list-decoding algorithms (up to the so-called Johnson bound). The full scope of the implications of this for Question 11 remains to be explored.

I would like to highlight one particular question here that looks especially approachable. The goal is to get a good hardness amplification of the permanent. Indeed the permanent was the very first function to which hardness amplification was applied [32], where it was shown that the permanent of a random matrix over large fields is hard to compute on average. I think it is will be a worthwhile goal to try to make progress on the following question using the existing knowledge around multiplicity codes.

Question 12 *For $q = O(1)$, is the permanent of a uniformly random matrix over $\mathbb{F}_q$ hard to compute with probability greater than $1 - \epsilon$ for some $\epsilon > 0$?*

The fact that permanents are polynomials, that derivatives of permanents are permanents, and that multiplicity codes are based on polynomials and derivatives, all suggest that multiplicity codes should have a lot to say about the average case hardness of the permanent. Earlier, Fortnow and Feigenbaum [17] showed that under a certain non-uniform distribution, the permanent over $\mathbb{F}_q$ is mildly-hard on average.

3.5 Error-correcting codes and average case lower bounds against $\mathrm{AC}^0[\oplus]$

Another question that this project hopes to make progress on is the question of getting better average case lower bounds against the circuit class $\mathrm{AC}^0[\oplus]$ (which is the class of constant-depth polynomial-size circuits composed of unbounded fan-in AND, OR, PARITY and NOT gates). Strong enough average-case lower bounds for $\mathrm{AC}^0[\oplus]$ would give (via the hardness vs. randomness paradigm) pseudorandom generators against the class $\mathrm{AC}^0[\oplus]$, thus showing that randomized uniform $\mathrm{AC}^0[\oplus]$ circuits cannot be much more powerful than deterministic uniform $\mathrm{AC}^0[\oplus]$ circuits. This has been a fundamental open question for a long time.

Classical work of Razborov [37] and Smolensky [40] gave an approach to proving average-case lower bounds for $\mathrm{AC}^0[\oplus]$: a function is average-case hard for $\mathrm{AC}^0[\oplus]$ if it is average-case hard for $\mathbb{F}_2$-polynomials of degree at most $d = \mathrm{poly}\log n$. Motivated by this, we will try to develop techniques to address the following question.

Question 13 *Find an explicit function $f : \{0,1\}^n \to \{0,1\}$ whose distance from all $\mathbb{F}_2$ polynomials of degree at most $d = \mathrm{poly}\log n$ is at least $(\frac{1}{2} - n^{-\omega(1)})$.*

In coding theoretic terms, this corresponds to finding a received word which is very far from all codewords of the Reed-Muller code of degree d polynomials. This would have non-trivial implications for average case lower bounds for $\mathrm{AC}^0[\oplus]$ circuits, and also for pseudorandom generation against these circuits.

One possible approach to this problem is via list-decoding *algorithms*. The approach requires an error-correcting code $\mathcal{C}$, a codeword $c \in \mathcal{C}$, and a good list-decoding algorithm $\mathcal{A}$ for $\mathcal{C}$. Now consider an arbitrary low-degree polynomial p, and consider it as a received word for the error-correcting code $\mathcal{C}$. If we could analyze the behavior of the list-decoding algorithm $\mathcal{A}$ when it is given input p, and show that the returned list of codewords is does not contain c, then this would imply that c is not close to p. Thus we conclude that c is far from low-degree polynomials; which is what we wanted.

Implementing this program requires the choice of a good error-correcting codes with a list-decoding algorithm that is simple enough to analyze on specific inputs. I have had some preliminary success with this approach, in giving an alternate proof of the (previously known) fact that certain codewords of the dual-BCH code are exponentially uncorrelated with $\mathbb{F}_2$-polynomials of degree at most

$d = 1$. I believe that this approach is quite promising, and in particular I think certain codewords of the dual-BCH code will have the desired property for d as large as $\mathrm{poly}\log n$.

I have also executed this strategy on another closely related code, which is based on cubic (and higher) residuosity in finite fields [27]. In this case I was able to handle polynomials of degree as large as n^ϵ, but the correlation bound I could show was only polynomially small, and thus it failed to answer Question 13. Nevertheless, this did show a complexity theoretic lower bound, that computing cubic residuosity and cube roots over finite fields $\mathbb{F}_{2^n}$ is hard on average for $\mathrm{AC}^0[\oplus]$ circuits. I view this as a kind of validation that this approach has the potential to lead to interesting results.

3.6 Improved randomness extractors

In recent years, error-correcting codes have also been instrumental for progress in the area of randomness extraction. A few key fundamental questions about randomness extraction remain, and this research project hopes to make progress on them.

A randomness extractor is a deterministic function which takes in as input some weakly random variables, and outputs a random variable whose distribution is very close to purely random (i.e., uniformly distributed over the co-domain of the function). Such randomness extractors could be used for generating random bits (for use in algorithms and cryptography) from natural phenomena exhibiting some randomness. They also have extensive uses in pseudorandomness, cryptography and complexity theory.

Below we will talk about seeded randomness extractors (which is the most widely-studied kind of randomness extractor). A seeded randomness extractor takes in two random variables, an n-bit "weakly-random" variable and an $O(\log n)$ bit purely random seed, and is supposed to output an m-bit long string whose distribution is very close to purely random. How big m can be depends on the *min-entropy* of the weakly random input.

Recent work of the PI [15] (see also [44]) gave constructions with the best known dependence of m on the min-entropy k of the weakly-random input, namely $m = k(1 - o(1))$. We hope to build on this and get a completely optimal extractor in this regard:

Question 14 *Construct seeded randomness extractors with optimal output length (i.e., $m = k - O(1)$).*

A related question, which is equally fundamental but has seen much less progress, is to construct the so-called "linear-degree" extractors:

Question 15 *Construct seeded randomness extractors with optimal seed length (i.e., $\log n + O(1)$).*

Recently Zuckerman showed [47] how to construct such extractors as long as $k = \Omega(n)$.

An interesting variant of the above questions is to ask if optimal extractors can be constructed non-uniformly. A specific approach is to analyze if random linear extractors are optimal (a random linear extractor is one where the output is random linear function of the weak random source, a different random linear function being chosen for each possible value of the seed): this would show the existence of small circuits computing optimal randomness extractors.

Question 16 *Are random linear extractors optimal?*

In recent work [21], I and my coauthors showed that random linear codes are optimal list-decodable codes. Given the strong relationship that list-decodable codes and extractors are known to have, I believe that the techniques in [21] will give some insight as to how to proceed with the above question.

3.7 Local testability of very high rate codes and the Unique Games Conjecture

The final research topic of this proposal has to do with very high rate locally testable codes. I and my coauthors proved [9] that there exist error-correcting codes encoding k-bit messages which have very high rate $(1 - o(1))$, and are locally testable from constantly many errors with ϵk query complexity. While being of interest for error-correction, it also turned out to have applications to the study of the Unique Games Conjecture and the Small Set Expansion hypothesis [2]. In particular, the codes were used to show that there exist instances on which the Arora-Barak-Steurer algorithm for Unique Games and Small Set Expansion provably does not perform well.

The following question on the existence of very high rate codes is motivated by these developments:

Question 17 *For every $\epsilon > 0$, do there exist locally testable codes encoding k bits into $k + O(\log k)$ bits, and which can be locally tested from a constant number of errors using ϵk queries?*

I believe that such codes should exist, and it would be very interesting to develop tools to find/construct such codes. Furthermore, a positive answer would be interesting also because [2] it would give expander graphs with many large eigenvalues, and would rule out a certain approach to disproving the Unique Games Conjecture. A negative answer would show that the Arora-Barak-Steurer algorithm can detect small-set expanders, thus refuting the Small-Set Expansion hypothesis, and this would be *extremely* interesting.

Finally, this could be potentially useful as an error-correcting code in practice, where codes of constant distance are quite commonly used.

4 Educational Plan

A major part of this project involves educational activities. This will include course and course material development, as well as the mentoring of young researchers, at both the undergraduate level and the graduate level.

Being part of both a Computer Science and a Mathematics department, I have a unique opportunity to teach and advise students from both these departments. This is particularly relevant for my research program since its scope includes a broad range of topics from these areas.

4.1 Course Development

I am planning to develop two new courses: one specialized one at the graduate level focusing on the areas studied by this proposal, and another general one at the undergraduate level emphasizing the interconnections and symbiosis between computer science and mathematics. Apart from these courses, I also will be regularly teaching courses in discrete mathematics and computer science at all levels.

I view teaching as one of the privileges of being at an academic institution, and I take it very seriously (while also enjoying it very much). In the 2011-2012 year, I taught 3 courses (at various levels). Overall I believe the students enjoyed and got a lot out of these courses (my teaching evaluation scores were 4.83, 4.78 and 4.43, on a scale of 0 to 5).

I will now describe the courses that I plan to develop.

Course – Error-Correcting Codes in Algorithms and Complexity Theory: In Spring 2013, I am going to teach a new course on error-correcting codes and their applications across theoretical computer science. This will be aimed at theoretically inclined graduated students. It will cover both basic results and more advanced research-level material. The main thrust will be to illustrate how the body of ideas and techniques coming from the theory of error-correcting codes forms an integral part of many modern algorithms and in our present understanding of complexity theory and pseudorandomness.

In addition to the fundamentals of error-correcting codes, I plan to discuss many applications such as hashing, interactive proofs, PCPs, belief propagation, randomness extractors, hardness amplification, data structures and secure multiparty computation. I will also discuss topics at the forefront of modern research, including many of the topics which will be studied in the research component of the proposal. At a higher level level, I believe that such a course would equip students with (1) the ability to use combinatorial and algebraic tools to design discrete combinatorial structures with interesting properties, and more importantly (2) the ability to design solutions to algorithmic and complexity theoretic problems by using such tailor-made discrete combinatorial structures.

I plan to develop lecture materials, make them publicly available, and reuse them and polish them in future incarnations of the course.

Course – The Basic Mathematics of Theoretical Computer Science: I am planning to develop a new undergraduate course called "The Basic Mathematics of Theoretical Computer Science", generally aimed at junior/senior undergraduate students in computer science and mathematics.

The main goal of this course is to show students in both computer science and mathematics some of the beautiful results of theoretical computer science which make ingenious use of simple tools from mathematics. In particular, I will demonstrate how easily accessible and well motivated the mathematical questions studied in theoretical computer science are. Underlying every clever algorithm there is a solid mathematical theorem; and the search for nontrivial algorithms motivates the search for interesting theorems. The main part of this course will be to give many, many instances of this phenomenon (for example, the success story of random walks as a tool in algorithm design, and the mathematics behind it).

I believe that the more practice-oriented students will get excited by the kinds of fundamental computational problems that can get solved using, for example, elementary linear algebra and probability, that were hitherto dry, formal subjects for them. I also believe that the more theory-oriented students will get excited by the kinds of mathematical problems and techniques whose study arises from natural and well-motivated computer science problems.

One consequence that I hope will come from this course is to have students understand that theoretical computer science is an area where CS and mathematics can interact well and make each other more exciting. I also hope to encourage more interaction between students of computer science and mathematics through this course.

I plan to write lecture notes for this course and put them up on the internet for the general public to read. I feel that there is not much expository material on this theme, and I hope to partially remedy this.

4.2 Mentoring

This project will involve the mentoring of graduate students in their Ph.D. research, and also some advanced undergraduate students working on their theses.

My approach to mentoring graduate students is to initially give them my own problems that I think are approachable, and to then encourage them to come out with their ideas and directions for tackling the problems. For more senior graduate students, my philosophy is to inculcate in them a few basic principles, such as appreciation for good questions and the art of formulating them, and then to encourage them to read and discuss papers in areas which I feel have a lot of research potential. My role will be as much that of a sounding board as it will be that of an active collaborator.

The topic of this proposal involves problems of widely varying depths and and will be able to engage graduate students at various stages of their career.

The proposed research program also has a lot of scope for involving undergraduates in various more approachable research projects. This could range from the more practice-oriented implementations and deployment of error-correcting schemes for real systems, to theoretical questions such as exploring the efficiency of various decoding algorithms.

5 Broader Impact

The broader impact of this proposal will be through the research results, dissemination of research results and materials, as well as outreach activities.

Research: The most direct broad impact emanating from the research of this project would be through the error-correcting codes and related algorithms developed. This proposal in a large part deals with error-correcting codes of high-rate, which are precisely the kinds of error-correcting codes that are used/needed in real-world computer systems. As part of this project, we will pursue the practical aspects of such error-correcting codes in enabling new kinds of applications that were not conceivable previously. Furthermore, the applications in complexity theory and pseudorandomness could be applied to efficient proof systems and pseudorandom objects, which could have broader impact in cryptography and network design.

Dissemination: The PI, along with the young researchers participating in this project, will further contribute to the broad dissemination of the research by presenting the results of the research in conferences, in seminars at academic institutions across the world, and at industrial research labs. Furthermore, all research papers and course materials developed will be made publicly available at the PI's website. Finally, the PI will also write surveys and expository material to keep the advanced research material organized in a coherent fashion where it can be accessible to beginning young researchers.

Outreach: The PI strongly believes in exposing high-school students and undergraduates to exciting developments in modern research. Such exposure goes a long way in inspiring students to pursue careers in science, technology and academia. In this spirit, the PI will give introductory lectures to various undergraduate organizations and high-schools. The PI will also integrate his outreach efforts with those of the nearby centers, DIMACS and The Center for Computational Intractability. The PI will give lectures to high-school students, undergraduates and general audiences through programs organized by these centers, and also participate in organizing such outreach programs. For example, the PI has given/is scheduled to give expository lectures at the DIMACS REU program (for undergraduates), the HEROES conference at Rutgers (for high-school students) and the Computer Science Summer Program organized by Rajiv Gandhi and the CCI (for a wide variety of students). Through such lectures, and other fora, the PI hopes to instill excitement for higher education in computer science (and theoretical computer Science in particular) in students.

6 Previous Research Accomplishments

Below I outline some of my significant research results.

Multiplicity Codes: My most significant result was the introduction and development of the theory of multiplicity codes [31, 28]. Their main feature is that they combine local decodability with high-rate for the first time, thus raising the very exciting possibility that local-decoding could be applicable in the real world. Furthermore, the existence of such codes was completely contrary to widely held intuitions in this area, which strongly suggests that they could be used in other surprising contexts in complexity theory and pseudorandomness. This work is also the motivation for many of the aspects of this research proposal.

Coding Theory: In coding theory, I have proved a number of results establishing optimal and near-optimal results on the local testability, local decodability, list-decodability and local list-decodability for a number of classical error-correcting codes. These works cover various algebraic codes (Reed-Solomon codes [6], Reed-Muller codes [9, 11], Hadamard codes [20, 13], dual-BCH codes [30, 29]) as well as their complete opposite, random codes [30, 29, 21]. In these works I have developed a wide variety of general techniques for working with error-correcting codes in sublinear-time and in the presence of many errors.

Complexity Theory and Pseudorandomness: In complexity theory and pseudorandomness, I have focussed on various questions in circuit complexity and explicit constructions of pseudorandom objects. In [27], I showed that certain elementary finite field operations are hard for the circuit class $\mathrm{AC}^0[\oplus]$. This work led to a new kind exponential sum bound, of interest in number theory. In [26], I proved explicit lower bounds for, and proved a law explaining the behavior of, the logic $\mathrm{FO}[\oplus]$ on random graphs. This work answered a long-standing question in logic, regarding generalizations of the classical zero-one law for random graphs, to the case of logics with counting quantifiers. In [5], I constructed the best-known explicit "affine dispersers": these are two-colorings of $\mathbb{F}_2^n$ which do not contain any monochromatic subspace of large dimension, improving results of Barak et. al. [1] and Bourgain [10].