On Approximating the Entropy of Polynomial Mappings

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Citable Link</td>
<td><a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:33896773">http://nrs.harvard.edu/urn-3:HUL.InstRepos:33896773</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at <a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP">http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP</a></td>
</tr>
</tbody>
</table>

(Article begins on next page)
On Approximating the Entropy of Polynomial Mappings

Zeev Dvir¹  Dan Gutfreund²  Guy N.Rothblum¹  Salil Vadhan³
¹Center for Computational Intractability and Department of Computer Science, Princeton University, 35 Olden Street, Princeton, NJ, USA.
²IBM Research, Haifa, Israel.
³School of Engineering and Applied Sciences & Center for Research on Computation and Society, Harvard University, 33 Oxford Street, Cambridge, MA, USA.
zeev.dvir@gmail.com  danny.gutfreund@gmail.com  rothblum@alum.mit.edu  salil@seas.harvard.edu

Abstract: We investigate the complexity of the following computational problem:

POLYNOMIAL ENTROPY APPROXIMATION (PEA): Given a low-degree polynomial mapping \( p : \mathbb{F}^n \rightarrow \mathbb{F}^m \), where \( \mathbb{F} \) is a finite field, approximate the output entropy \( H(p(U_n)) \), where \( U_n \) is the uniform distribution on \( \mathbb{F}^n \) and \( H \) may be any of several entropy measures.

We show:

- Approximating the Shannon entropy of degree 3 polynomials \( p : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m \) over \( \mathbb{F}_2 \) to within an additive constant (or even \( n^3 \)) is complete for \( \text{SZKP}_L \), the class of problems having statistical zero-knowledge proofs where the honest verifier and its simulator are computable in logarithmic space. (\( \text{SZKP}_L \) contains most of the natural problems known to be in the full class \( \text{SZKP} \).
- For prime fields \( \mathbb{F} \neq \mathbb{F}_2 \) and homogeneous quadratic polynomials \( p : \mathbb{F}^n \rightarrow \mathbb{F}^m \), there is a probabilistic polynomial-time algorithm that distinguishes the case that \( p(U_n) \) has min-entropy (or even Renyi entropy) greater than \( (2 + o(1))k \) from the case that \( p(U_n) \) has min-entropy \( k \).
- For degree \( d \) polynomials \( p : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m \), there is a polynomial-time algorithm that distinguishes the case that \( p(U_n) \) has max-entropy \( k \) (where the max-entropy of a random variable is the logarithm of its support size) from the case that \( p(U_n) \) has max-entropy at least \( (1 + o(1)) \cdot k^d \) (for fixed \( d \) and large \( k \)).

Keywords: cryptography; computational complexity; algebra; entropy; statistical zero knowledge; randomized encodings

1 Introduction

We consider the following computational problem:

POLYNOMIAL ENTROPY APPROXIMATION (PEA): Given a low-degree polynomial mapping \( p : \mathbb{F}^n \rightarrow \mathbb{F}^m \), where \( \mathbb{F} \) is a finite field, approximate the output entropy \( H(p(U_n)) \), where \( U_n \) is the uniform distribution on \( \mathbb{F}^n \).

In this paper, we present some basic results on the complexity of PEA, and suggest that a better understanding might have significant impact in computational complexity and the foundations of cryptography.

Note that PEA has a number of parameters that can be varied: the degree \( d \) of the polynomial mapping, the size of the finite field \( \mathbb{F} \), the quality of approximation (eg multiplicative or additive), and the measure of entropy (eg Shannon entropy or min-entropy). Here we are primarily interested in the case where the degree \( d \) is bounded by a fixed constant (such as 2 or 3), and the main growing parameters are \( n \) and \( m \). Note that in this case, the polynomial can be specified explicitly by \( m \cdot \text{poly}(n) \) coefficients, and thus “polynomial time” means \( \text{poly}(m, n, \log |\mathbb{F}|) \).

Previous results yield polynomial-time algorithms for PEA in two special cases:

Exact Computation for Degree 1: For polynomials \( p : \mathbb{F}^n \rightarrow \mathbb{F}^m \) of degree at most 1, we can write \( p(x) = Ax + b \) for \( A \in \mathbb{F}^{m \times n} \) and \( b \in \mathbb{F}^m \). Then \( p(U_n) \) is uniformly distributed on the affine subspace \( \text{Image}(A) + b \), and thus has entropy exactly \( \log |\text{Image}(A)| = \text{rank}(A) \cdot \log |\mathbb{F}| \).

Multiplicative Approximation over Large Fields: In their work on randomness extractors for polynomial sources, Dvir, Gabizon, and Wigderson [DGW] related the entropy of \( p(U_n) \) to the rank of the Jacobian matrix \( J(p) \), whose \( (i, j) \)’th entry is the partial derivative \( \partial p_i / \partial x_j \), where \( p_i \) is the \( i \)’th component of \( p \). Specifically, they showed that the min-entropy of \( p(U_n) \) is essentially within a \( (1 + o(1)) \cdot \text{multiplicative factor of} \text{rank}(J(p)) \cdot \log |\mathbb{F}| \), where the rank is computed over the polynomial ring \( \mathbb{F}[x_1, \ldots, x_n] \).

This tight approximation holds over prime fields.
of size exponential in $n$. Over fields that are only mildly large (say, polynomial in $n$) the rank of the Jacobian still gives a one-sided approximation to the entropy.

In this paper, we study PEA for polynomials of low degree (namely 2 and 3) over small fields (especially the field $\mathbb{F}_2$ of two elements). Our first result characterizes the complexity of achieving good additive approximation:

**Theorem 1.1 (informal).** The problem $\text{PEA}_{\mathbb{F}_2,3}^+$ of approximating the Shannon entropy of degree 3 polynomials $p : \mathbb{F}_2^n \to \mathbb{F}_2^n$ to within an additive constant (or even $n^{1/3}$) is complete for $\text{SZKP}_L$, the class of problems having statistical zero-knowledge proofs where the honest verifier and its simulator are computable in logarithmic space (with two-way access to the input, coin tosses, and transcript).\(^1\)

In particular, the output entropy approximation problem is at least as hard as **Graph Isomorphism**, **Quadratic Residuosity**, the **Discrete Logarithm**, and the approximate **Closest Vector Problem**, as the known statistical zero-knowledge proofs for these problems [GMR, GMW, GK, GG] have verifiers and simulators that can be computed in logarithmic space.

Theorem 1.1 is proven by combining the reductions for known $\text{SZKP}$-complete problems [SV, GV] with the randomized reductions developed by Applebaum, Ishai, and Kushilevitz in their work on cryptography in NC\(^0\) [IK, AIK]. Moreover, the techniques in the proof can also be applied to the specific natural complete problems mentioned above, and most of these can be solved by polynomial-time reductions to special cases of $\text{PEA}_{\mathbb{F}_2,3}^+$ that may be easier to solve (e.g. ones where the output distribution is uniform on its support, and hence all entropy measures coincide).

The completeness of $\text{PEA}_{\mathbb{F}_2,3}^+$ raises several intriguing (albeit speculative) possibilities:

**Combinatorial or Number-Theoretic Complete Problems for $\text{SZKP}_L$:**

Ever since the first identification of complete problems for $\text{SZKP}$ (standard statistical zero knowledge, with verifiers and simulators that run in polynomial time rather than logarithmic space) [SV], it has been an open problem to find combinatorial or number-theoretic complete problems. Previously, all of the complete problems for $\text{SZKP}$ and other zero-knowledge classes (e.g. [SV, DDPY, GV, GSV2, BG, Vad, Mal, CCKV]) refer to estimating statistical properties of arbitrary efficiently samplable distributions (namely, distributions sampled by boolean circuits). Moving from a general model of computation (boolean circuits) to a simpler, more structured model (degree 3 polynomials) is a natural first step to finding other complete problems, similarly to how the reduction from CIRCUIT-SAT to 3-SAT is the first step towards obtaining the wide array of known NP-completeness results. (In fact we can also obtain a complete problem for $\text{SZKP}_L$ where each output bit of $p : \mathbb{F}_2^n \to \mathbb{F}_2^n$ depends on at most 4 input bits, making the analogy to 3-SAT even stronger.)

**Cryptography Based on the Worst-case Hardness of $\text{SZKP}_L$:**

It is a long-standing open problem whether cryptography can be based on the worst-case hardness of NP. That is, can we show that NP $\not\subset$ BPP\(^2\) implies the existence of one-way functions? A positive answer would yield cryptographic protocols for which we can have much greater confidence in their security than any schemes in use today, as efficient algorithms for all of NP seems much more unlikely than an efficient algorithm for any of the specific problems underlying present-day cryptographic protocols (such as FACTORING). Some hope was given in the breakthrough work of Ajtai [Ajt], who showed that the worst-case hardness of an approximate version of the **Shortest Vector Problem** implies the existence of one-way functions (and in fact, collision-resistant hash functions). Unfortunately, it was shown that this problem is unlikely to be NP-hard [GG, AR, MX]. In fact, there are more general results, showing that there cannot be (nonadaptive, black-box) reductions from breaking a one-way function to solving any NP-complete problem (assuming NP $\not\subset$ coAM) [FF, BT, AGGM].

We observe that these obstacles for NP do not apply to $\text{SZKP}$ or $\text{SZKP}_L$, as these classes are already contained in AM $\cap$ coAM [For, AH]. Moreover, being able to base cryptography on the hardness of $\text{SZKP}$ or $\text{SZKP}_L$ would also provide cryptographic protocols with a much stronger basis for security than we have at present — these protocols would be secure if any of the variety of natural problems in $\text{SZKP}_L$ are worst-case hard (e.g. **Quadratic Residuosity**, **Graph Isomorphism**, **Discrete Logarithm**, the approximate **Shortest Vector Problem**).

Our new complete problem for $\text{SZKP}_L$ provides natural approaches to basing cryptography on $\text{SZKP}$-hardness. First, we can try to reduce $\text{PEA}_{\mathbb{F}_2,3}^+$ to the approximate **Shortest Vector Problem**. This would suffice by the aforementioned result of Ajtai [Ajt]. Alternatively, we can try to exploit the algebraic structure in $\text{PEA}_{\mathbb{F}_2,3}^+$ to give a worst-case/average-case reduction for it (i.e. reduce arbitrary instances to random ones). This would show that if $\text{SZKP}_L$ is worst-case hard, then it is also average-case hard. Un-

---

\(^1\) See Sections 3 and 4 for the formal definitions of the notions involved and the formal statement of the theorem.

\(^2\) Or NP $\not\subset$ i.o.–BPP, where i.o.–BPP is the class of problems that can be solved in probabilistic polynomial time for infinitely many input lengths.
like NP, the average-case hardness of SZKP is known to imply the existence of one-way functions by a result of Ostrovsky [Ost], and in fact yields even stronger cryptographic primitives such as constant-round statistically hiding commitment schemes [OV, RV].

New Algorithms for SZKP\textsubscript{L} Problems:

On the flip side, the new complete problem may be used to show that problems in SZKP\textsubscript{L} are easier than previously believed, by designing new algorithms for PEA. As mentioned above, nontrivial polynomial-time algorithms have been given in some cases via algebraic characterizations of the entropy of low-degree polynomials (namely, the Jacobian rank) [DGW]. This motivates the search for tighter and more general algebraic characterizations of the output entropy, which could be exploited for algorithms or for worst-case/average-case connections. In particular, this would be a very different way of trying to solve problems like GRAPH ISOMORPHISM and QUADRATIC RESIDUOUSITY than previous attempts. One may also try to exploit the complete problem to give a quantum algorithm for SZKP\textsubscript{L}. Aharonov and Ta-Shma [AT] showed that all of SZKP would have polynomial-time quantum algorithms if we could solve the QUANTUM STATE GENERATION (QSG) problem: given a boolean circuit $C : \{0,1\}^m \rightarrow \{0,1\}^n$, construct the quantum state $\sum_x |C(x)\rangle$. Using our new complete problem, if we can solve QSG even in the special case that $C$ is a degree 3 polynomial over $\mathbb{F}_2$, we would get quantum algorithms for all of SZKP\textsubscript{L} (including GRAPH ISOMORPHISM and the approximate SHORTEST VECTOR PROBLEM, which are well-known challenges for quantum computing).

While each of these potential applications may be remote possibilities, we feel that they are important enough that any plausible approach is worth examining.

Our Algorithmic Results.

Motivated by the above, we initiate a search for algorithms and algebraic characterizations of the entropy of low-degree polynomials over small finite fields (such as $\mathbb{F}_2$), and provide partial results:

- For degree $d$ (multilinear) polynomials $p : \mathbb{F}_2^d \rightarrow \mathbb{F}_2^m$, the rank of the Jacobian $J(p)$ (over $\mathbb{F}_2[x_1,\ldots,x_n]$) does not provide better than a $2^{d-1} - o(1)$ multiplicative approximation to the entropy $H(p(U_n))$. Indeed, the polynomial mapping

$$p(x_1,\ldots,x_n,y_1,\ldots,y_{d-1}) =
\quad (y_1y_2\cdots y_{d-1}) \cdot (x_1, x_2,\ldots, x_n)$$

has Jacobian rank $n$ but output entropy smaller than $n/2^{d-1} + 1$.

- For prime fields $\mathbb{F} \neq \mathbb{F}_2$ and homogeneous quadratic polynomials $p : \mathbb{F}^n \rightarrow \mathbb{F}^m$, there is a probabilistic polynomial-time algorithm that distinguishes the case that $p(U_n)$ has entropy smaller than $k$ from the case that $p(U_n)$ has min-entropy (or even Renyi entropy) greater than $(2 + o(1))k$. This algorithm is based on a new formula for the Renyi entropy of $p(U_n)$ in terms of the rank of random directional derivatives of $p$.

- For degree $d$ polynomials $p : \mathbb{F}_q^m \rightarrow \mathbb{F}_q^m$, there is a polynomial-time algorithm that distinguishes the case that $p(U_n)$ has max-entropy smaller than $k$ (where the max-entropy of a random variable is the logarithm of its support size) from the case that $p(U_n)$ has max-entropy at least $(1 + o(1)) \cdot k^d$ (for fixed $d$ and large $k$). This algorithm is based on relating the max-entropy to the dimension of the $\mathbb{F}_2$-span of the $p$’s components $p_1,\ldots,p_m \in \mathbb{F}_2[x_1,\ldots,x_n]$.

While our algorithms involve entropy measures other than Shannon entropy (which is what is used in the SZKP\textsubscript{L}-complete problem PEA\textsubscript{+}(AT), recall that many of the natural problems in SZKP\textsubscript{L} reduce to special cases where we can bound other entropy measures such as max-entropy or Renyi entropy. See Section 4.4.

2 Preliminaries and Notations

For two discrete random variables $X, Y$ taking values in $S$, their statistical difference is defined to be $\Delta(X,Y) \overset{\text{def}}{=} \max_{T \subseteq S} |\Pr[X \in S] - \Pr[Y \in S]|$. We say that $X$ and $Y$ are $\varepsilon$-close if $\Delta(X,Y) \leq \varepsilon$. The collision probability of $X$ is defined to be $\text{cp}(X) \overset{\text{def}}{=} \sum_x \Pr[X = x]^2 = \Pr[X = X']$, where $X'$ is an iid copy of $X$. The support of $X$ is $\text{Supp}(X) \overset{\text{def}}{=} \{x \in S : \Pr[X = x] > 0\}$. $X$ is flat if it is uniform on its support.

For a function $f : S^n \rightarrow T^m$, we write $f_i : S^n \rightarrow T$ for the $i$’th component of $f$. When $S$ is clear from context, we write $U_n$ to denote the uniform distribution on $S^n$, and $f(U_n)$ for the output distribution of $f$ when evaluated on a uniformly chosen element of $S^n$. The support of $f$ is defined to be $\text{Supp}(f) \overset{\text{def}}{=} \text{Im}(f)$.

For a prime power $q = p^t$, $\mathbb{F}_q$ denotes the unique finite field of size $q$. For a mapping $P : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$, we say that $P$ is a polynomial mapping if each $P_i$ is a polynomial (in $n$ variables). The degree of $P$ is $\deg(P) = \max_i \deg(P_i)$.

Notions of Entropy.

Throughout this work we consider several different notions of entropy, or the “amount of randomness” in a random variable. The standard notions of Shannon Entropy, Renyi Entropy, and Min-Entropy are three such notions. We also consider the (log) support size, or
maximum entropy, as a (relaxed) measure of randomness.

**Definition 2.1.** For a random variable $X$ taking values in a set $S$, we consider the following notions of entropy:

- **Min-entropy:** $H_{\min}(X) \overset{\text{def}}{=} \min_{x \in S} \log \frac{1}{p_X(x)}$.
- **Rényi entropy:** $H_{\text{Rényi}}(X) \overset{\text{def}}{=} \log \left( \frac{1}{E_{x \sim X} \left[ p_X(x) \right]} \right)$.
- **Shannon-entropy:** $H_{\text{Shannon}}(X) \overset{\text{def}}{=} \log \left| \text{Supp}(X) \right|$.
- **Max-entropy:** $H_{\max}(X) \overset{\text{def}}{=} \log \left| \text{Supp}(X) \right|$.

(All logarithms are base 2 except when otherwise noted.)

These notions of entropy are indeed increasingly relaxed, as shown in the following claim:

**Claim 2.2.** For every random variable $X$ it holds that

$$0 \leq H_{\min}(X) \leq H_{\text{Rényi}}(X) \leq H_{\text{Shannon}}(X) \leq H_{\max}(X).$$

Moreover, if $X$ is flat, all of the entropy measures are equal to $\log |\text{Supp}(X)|$.

### 3 Entropy Difference and Polynomial Entropy Difference

In this section we define the **Entropy Difference** and **Polynomial Entropy Difference** problems, which are the focus of this work.

**Entropy Difference Promise Problems.**

The promise problem **Entropy Difference** (ED) deals with distinguishing an entropy gap between two random variables represented as explicit mappings, computed by circuits or polynomials, and evaluated on a uniformly random input. In this work, we consider various limitations on these mappings both in terms of their computational complexity and their degree (when viewed as polynomials).

In what follows $C$ will be a concrete computational model, namely every $c \in C$ computes a function with finite domain and range. Examples relevant to this paper include:

- The class $\text{CIRCUIT}$ of all boolean circuits $C : \{0,1\}^n \to \{0,1\}^m$ (the concrete model corresponding to polynomial time).
- The class $\text{BP}$ of all branching programs $C : \{0,1\}^n \to \{0,1\}^m$ (the concrete model corresponding to logarithmic space).
- The class $\text{POLYNOMIALS}_{F,d}$ of all degree $d$ polynomials $p : \mathbb{F}^n \to \mathbb{F}^m$.

The promise problem ED [GV] is defined over pairs of random variables represented as mappings computed by boolean circuits, where the random variables are the outputs of the circuits evaluated on uniformly chosen inputs. The problem is to determine which of the two random variables has more Shannon entropy, with a promise that there is an additive gap between the two entropies of at least 1. We generalize this promise problem to deal with different notions of entropies, entropy gaps, and the complexity of the mappings which represent the random variables.

**Definition 3.1 (Generalized Entropy Difference).** The promise problem $\text{ED}_{C_{\text{ENT, ENT,GAP}}}$, is defined by the entropy measures $u_{\text{ENT}}$ and $t_{\text{ENT}}$ from the set $\{\text{MIN, RÉNYI, SHANNON, MAX}\}$, an entropy gap, which can be $+c$, $\times c$ or $\exp(c)$ for some constant $c > 0$, referring to additive, multiplicative or exponential gaps in the problem’s promise, and a concrete computational model $C$. For a pair of mappings $p, q \in C$, the random variables $P$ and $Q$ are (respectively) the evaluation of the mappings $p$ and $q$ on a uniformly random input. The Yes and No instances are, for additive gap $+c$

$$\text{YES} = \{(p, q) : H_{u_{\text{ENT}}}(P) \geq H_{t_{\text{ENT}}}(Q) + c\},$$

$$\text{NO} = \{(p, q) : H_{u_{\text{ENT}}}(P) + c \leq H_{t_{\text{ENT}}}(Q)\};$$

for multiplicative gap $\times c$ (for $c > 1$)

$$\text{YES} = \{(p, q) : H_{u_{\text{ENT}}}(P) \geq H_{t_{\text{ENT}}}(Q) \cdot c\},$$

$$\text{NO} = \{(p, q) : H_{u_{\text{ENT}}}(P) \cdot c \leq H_{t_{\text{ENT}}}(Q)\};$$

and for exponential gap $\exp(c)$

$$\text{YES} = \{(p, q) : H_{u_{\text{ENT}}}(P) \geq H_{t_{\text{ENT}}}(Q)^c\},$$

$$\text{NO} = \{(p, q) : H_{u_{\text{ENT}}}(P)^c \leq H_{t_{\text{ENT}}}(Q)\}.$$

We always require that $u_{\text{ENT}}$ is more stringent than $t_{\text{ENT}}$ in that $H_{u_{\text{ENT}}}(X) \leq H_{t_{\text{ENT}}}(X)$ for all random variables $X$. (This ensures that the YES and NO instances do not intersect.) If we do not explicitly set the different parameters, then the default entropy type for $u_{\text{ENT}}$ and $t_{\text{ENT}}$ is SHANNON, the default gap is an additive gap $= +1$, and the class $C$ is $\text{CIRCUIT}$.

Note that, by Claim 2.2, with all other parameters being equal - the more relaxed the entropy notion $u_{\text{ENT}}$ is, the easier the problem becomes. Similarly, the more stringent the entropy notion $t_{\text{ENT}}$ is, the easier the problem becomes.

**Polynomial Entropy Difference (PED).**

The main problem we focus on in this work is entropy difference for low-degree polynomial mappings.

**Definition 3.2 (Polynomial Entropy Difference).** The promise problem $\text{PED}_{F_{\text{ENT, ENT,GAP}}}$ is the entropy difference problem for degree $d$ polynomials over $F$, i.e. it is the promise problem $\text{ED}_{\text{POLYNOMIALS}_{F,d}}$ (see Definition 3.1). The default values for $u_{\text{ENT}}$, $t_{\text{ENT}}$, and GAP (if not specified explicitly) are as in Definition 3.1.
**POLYNOMIAL ENTROPY APPROXIMATION.**

Another natural algorithmic problem is that of approximating the entropy of a polynomial mapping up to a constant, multiplicative, or exponential approximation factor. We discuss this problem informally, focusing on its connection to POLYNOMIAL ENTROPY DIFFERENCE.

The POLYNOMIAL ENTROPY APPROXIMATION problem is, given a polynomial mapping $p$ of low degree, which induces a random variable $P$, to output an approximation $k$ to its entropy. For the approximation problem $\text{PEA}^\text{ENT,ENT,}^+_{c}$, we require that (w.h.p) the approximation $k$ satisfy:

$$H_{\text{ENT}}(P) - c < k < H_{\text{ENT}}(P) + c.$$

Using binary search, this approximation problem can be shown to be equivalent to deciding the following promise problem:

**YES** = \{ $(p,k) : H_{\text{ENT}}(P) \geq k + c$, \}

**NO** = \{ $(p,k) : H_{\text{ENT}}(P) \leq k - c$. \}

For notational convenience, we will also denote this promise problem by $\text{PEA}^\text{ENT,ENT,}^+_{c}$. PEA is defined analogously for multiplicative $(\times c)$ and exponential ($\exp(c)$) approximation.

We note that (as is the case for ENTROPY APPROXIMATION and ENTROPY DIFFERENCE in the statistical zero-knowledge literature [GSV2]), for fixed notions of entropy in the upper bounds and lower bounds, the PDA and PEA problems are computationally equivalent up to some loss in the approximation factor, for both additive and multiplicative approximation.

In one direction, $\text{PED}^\text{ENT,ENT,}^+_{c}$ reduces to $\text{PEA}^\text{ENT,ENT,}^+_{c/2}$. To see this, approximate the entropy of the two distributions $P$ and $Q$, get answers $k_p$ and $k_q$ (respectively), and accept if $k_p > k_q$. Otherwise reject. For a YES instance, $H_{\text{ENT}}(P) \geq H_{\text{ENT}}(Q) + c$, and so if the PEA approximation error is less than $c/2$ we get that $k_p$ must be greater than $H_{\text{ENT}}(P) - c/2$ and $k_q$ must be less than $H_{\text{ENT}}(Q) + c/2$, and so (w.h.p) $k_p > k_q$ and we accept. For NO instances, the reverse holds and w.h.p we reject.

In the other direction, we get that $\text{PEA}^\text{ENT,ENT,}^+_{c}$ reduces to $\text{PED}^\text{ENT,ENT,}^+_{c}$. If the given parameter $k$ is an integer multiple of $\log |F|$, then we can just construct $q$ so that $Q$ is a flat distribution with a support of size $2^k$ (e.g. $q$ is the identity map on $F^n$ for $n = k/\log |F|$), and then answer the to the PED instance $(p,q)$ is equal to the answer to the PEA instance $(p,k)$. In case $k$ is not an integer, then we instead apply the above reduction to the instance $(p', [tk])$ for a large enough integer $t$, where $p' = (p(x_1), \ldots, x_t) = p(x_1), p(x_2), \ldots, p(x_t)$. For a YES instance $(p,k)$ of PEA, we have

$$H_{\text{ENT}}(P') = t \cdot H_{\text{ENT}}(P) \geq t \cdot (k + c) \geq [tk] + c$$

for $t \geq 1 + 1/c$, so $(p', [tk])$ is also a YES instance of PEA. NO instances can be analyzed similarly.

For multiplicative approximation, we can reduce $\text{PEA}^\text{ENT,ENT,}^+_{c}$ to $\text{PED}^\text{ENT,ENT,}^+_{c}$ for any constant $c' < c$. For a YES instance $(p,k)$ of $\text{PEA}^\text{ENT,ENT,}^+_{c}$, we have

$$H_{\text{ENT}}(P') = t \cdot H_{\text{ENT}}(P) \geq t \cdot k \geq c' \cdot [tk],$$

provided $t \geq c'/(c - c')k$. We may assume that $k$ is bounded below by a constant, because the Schwartz–Zippel Lemma (cf. Lemma 5.10), implies that a nonconstant polynomial mapping of degree $d$ must have min-entropy at least $\log((1/|F|^{-d}))$, which is constant for fixed $|F|$ and $d$. So for a sufficiently large constant $t$, $(p', [tk])$ is a YES instance of $\text{PEA}^\text{ENT,ENT,}^+_{c'}$. NO instances can be analyzed similarly, and thus we can apply the above reduction for integer thresholds.

### 4 Hardness of POLYNOMIAL ENTROPY DIFFERENCE

In this section we present evidence that even when we restrict PED to low degree polynomial mappings, and even when we work with relaxed notions of entropy, the problem remains hard. This is done first by using the machinery of randomizing polynomials [IK, AIK] to reduce ED for rich complexity classes (such as log space) to PED (section 4.1). We then argue the hardness of ED for log-space computations, first via the problem’s completeness for a rich complexity class (a large subclass of SZKP), and then via reductions from specific well-studied hard problems.

#### 4.1 Randomized Encodings

We recall the notion of randomized encodings that was developed by Applebaum, Ishai, and Kushilevitz [IK, AIK]. Informally, randomized encoding of a function $f$ is a randomized function $g$ such that the (randomized) output $g(x)$ determines $f(x)$, but reveals no other information about $x$. We need the perfect variant of this notion, which we now formally define. (We comment that [IK, AIK] use different, more cryptographic, terminology to describe some of the properties below).

**Definition 4.1. [IK, AIK] Let $f : \{0,1\}^n \to \{0,1\}^\ell$ be a function. We say that the function $\hat{f} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^\ell$ is a perfect randomized encoding of $f$ with blowup $b$ if it is:**

- **Input independent:** for every $x, x' \in \{0,1\}^n$ such that $f(x) = f(x')$, the random variables $\hat{f}(x, U_m)$ and $\hat{f}(x', U_m)$ are identically distributed.

- **Output disjoint:** for every $x, x' \in \{0,1\}^n$ such that $f(x) \neq f(x')$, $\text{Supp}(\hat{f}(x, U_m)) \cap \text{Supp}(\hat{f}(x', U_m)) = \emptyset$. 

• Uniform: for every \( x \in \{0,1\}^n \) the random variable \( \hat{f}(x, U_m) \) is uniform over \( \text{Supp}(\hat{f}(x, U_m)) \).

• Balanced: for every \( x, x' \in \{0,1\}^n \) \( |\text{Supp}(f(x, U_m))| = |\text{Supp}(f(x', U_m))| = b \).

We now set up notations and state some simple claims about randomized encodings.

Let 
\[
f : \{0,1\}^n \rightarrow \{0,1\}^t
\]
be a function and let \( \hat{f} : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^{s} \) be a perfect randomized encoding of \( f \) with blowup \( b \).

For \( y \in \text{Supp}(f) \), define the set \( S_y \subseteq \{0,1\}^s \) to be:
\[
\{ z \in \{0,1\}^s : \exists (x, r) \in \{0,1\}^n \times \{0,1\}^m \text{ s.t. } f(x) = y \land \hat{f}(x, r) = z \}
\]

By the properties of perfect randomized encodings, the sets \( S_y \) form a balanced partition of \( \text{Supp}(f) \), indeed \( S_y = \text{Supp}(f(x, U_m)) \) for every \( x \) such that \( f(x) = y \), and hence \( |S_y| = b \). With this notation, the following claim is immediate.

Claim 4.2. \( \text{Supp}(\hat{f}) = b \cdot \text{Supp}(f) \)

For every \( z \in \text{Supp}(\hat{f}) \), we denote by \( y_z \) the unique string in \( \text{Supp}(f) \) such that \( z \in S_{y_z} \). For any \( x \in \{0,1\}^n \), \( \hat{f}(x, U_m) \) is uniformly distributed over \( S_{f(x)} \).

It follows that,

Claim 4.3. For every \( z \in \text{Supp}(\hat{f}) \),
\[
\Pr[\hat{f}(U_n, U_m) = z] = \frac{1}{b} \Pr[f(U_n) = y_z]
\]

We now state the relation between the entropy of \( \hat{f}(U_n, U_m) \) and the entropy of \( f(U_n) \) for each one of the entropy measures.

Claim 4.4. Let
\[
\text{ENT} \in \{\text{MIN}, \text{RENYI}, \text{SHANNON}, \text{MAX}\}
\]
then \( H_{\text{ENT}}(\hat{f}(U_n, U_m)) = H_{\text{ENT}}(f(U_n)) + \log b \)

Proof. For \( \text{ENT} = \text{MAX} \), the claim follows directly from Claim 4.2. For \( \text{ENT} = \text{MIN} \), the claim follows directly from Claim 4.3. For \( \text{ENT} = \text{SHANNON} \),
\[
H_{\text{SHANNON}}(\hat{f}(U_n, U_m)) = H_{\text{SHANNON}}(f(U_n)) + \log b.
\]

The first equality follows from the fact that \( \hat{f}(x, r) \) determines \( f(x) \) (follows from output disjointness). The second equality uses the chain rule for conditional entropy. The third equality follows from input independence, and the last equality follows from the fact that the perfect randomized encoding is uniform, balanced and has blowup \( b \).

By similar reasoning, for \( \text{ENT} = \text{RENYI} \), we have
\[
\text{cp}(\hat{f}(U_n, U_m)) = \Pr[\hat{f}(U_n, U_m) = \hat{f}(U'_n, U'_m)] = \Pr[f(U_n) = f(U'_n)] \cdot \Pr[\hat{f}(U_n, U_m) = \hat{f}(U'_n, U'_m) | f(U_n) = f(U'_n)] = \text{cp}(f(U_n)) \cdot (1/b).
\]

\[ \square \]

4.2 From Branching-Program Entropy Difference to Polynomial Entropy Difference

Applebaum, Ishai and Kushilevitz [IK, AIK] showed that logspace mappings (represented by the branching programs that compute the output bits) have randomized encodings which are polynomial mappings of degree three over the field with two elements.

Theorem 4.5. [IK, AIK] Given a branching program \( f : \{0,1\}^n \rightarrow \{0,1\}^t \), we can construct in polynomial time a degree 3 polynomial \( \hat{f} : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^t \) that is a perfect randomized encoding of \( f \). Moreover, the blowup \( b \) is a power of 2 and can be computed in polynomial time from \( f \).

Based on this theorem we show that the log-space entropy difference problem (for the various notions of entropy which we defined above) with additive gap reduces to the polynomial entropy difference problem with the same gap.

Theorem 4.6. The promise problem \( \text{ED}_{\text{BP}}^{\text{ENT}, \text{ENT} + c} \) for \( \text{ENT} \in \{\text{MIN}, \text{RENYI}, \text{SHANNON}, \text{MAX}\} \), Karp-reduces to the promise problem \( \text{PED}_{\mathbb{F}_2^n, 3}^{\text{ENT}, \text{ENT} + c} \).

Proof. Given an instance \( (X, Y) \) of \( \text{ED}_{\text{BP}}^{\text{ENT}, \text{ENT} + c} \) apply on each one of the branching programs \( X \) and \( Y \) the reduction from Theorem 4.5, to obtain a pair of polynomials \( \hat{X} \) and \( \hat{Y} \) of degree 3 over \( \mathbb{F}_2 \). By padding the output of \( \hat{X} \) or \( Y \) with independent uniformly distributed bits, we can ensure that \( \hat{X} \) and \( \hat{Y} \) have the same blow-up. By Claim 4.4, \( H_{\text{ENT}}(\hat{X}) - H_{\text{ENT}}(\hat{Y}) = \text{cp}(\hat{X}) - \text{cp}(\hat{Y}) \) and \( H_{\text{ENT}}(\hat{X}) - H_{\text{ENT}}(\hat{Y}) = H_{\text{ENT}}(X) - H_{\text{ENT}}(Y) \). It follows that yes (resp. no) instances of \( \text{ED}_{\text{BP}}^{\text{ENT}, \text{ENT} + c} \) are mapped to yes (resp. no) instances of \( \text{PED}_{\mathbb{F}_2^n, 3}^{\text{ENT}, \text{ENT} + c} \).

\[ \square \]

4.3 Polynomial Entropy Difference and Statistical Zero-Knowledge

Goldreich and Vadhan [GV] showed that the promise problem \( \text{ED} \) (ENTROPY DIFFERENCE problem for
Shannon entropy with additive gap and polynomial-size circuits) is complete for $\text{SZKP}$, the class of problems having statistical zero-knowledge proofs. We show a computationally restricted variant of this result, showing that $\text{PED}_{F_2,3}$ is complete for $\text{SZKP}_L$, the class of problems having statistical zero-knowledge proofs in which the honest verifier and its simulator are computable in logarithmic space (with two-way access to the input, coin tosses, and transcript).

**Theorem 4.7.** The promise problem $\text{PED}_{F_2,3}$ is complete for the class $\text{SZKP}_L$.

We start with proving that the problem is hard for the class.

**Lemma 4.8.** The promise problem $\text{PED}_{F_2,3}$ is hard for the class $\text{SZKP}_L$ under Karp-reductions.

**Proof.** We show that the promise problem $\text{EDBP}$ is hard (under Karp-reductions) for the class $\text{SZKP}_L$. The proof then follows by Theorem 4.6. The hardness of $\text{EDBP}$ follows directly from the reduction of [GV] which we now recall. Given a promise problem in $\text{SZKP}$ with a proof system $(P, V)$ and a simulator $S$, it is assumed w.l.o.g. that on instances of length $n$, $V$ tosses exactly $\ell = \ell(n)$ coins, the interaction between $P$ and $V$ consists of exactly $2r = 2r(n)$ messages each of length exactly $\ell$, the prover sends the odd messages and the last message of the verifier consists of its random coins. Furthermore, the simulator for this protocol always outputs transcripts that are consistent with $V$’s coins. For problems in $\text{SZKP}_L$, using the fact that the verifier is computable in logspace (with two-way access to the input, its coin tosses, and the transcript), we can obtain such a simulator that is computable in logspace (again with two-way access to the input and its coin tosses). On input $x$, we denote by $S(x)_i$ ($1 \leq i \leq 2r$) the distribution over the $(i \cdot \ell)$-long prefix of the output of the simulator. That is, the distribution over the simulation of the first $i$ messages in the interaction between $P$ and $V$.

The reduction maps an instance $x$ to a pair of distributions $(X_x, Y_x)$:

- $X_x$ outputs independent samples from the distributions $S(x)_2, S(x)_4, \ldots, S(x)_{2r}$.
- $Y_x$ outputs independent samples from the distributions $S(x)_1, S(x)_3, \ldots, S(x)_{2r-1}$ and $U_{\ell-2}$.

Since $S$ is computable in logarithmic space, we can efficiently construct branching programs $X_x$ and $Y_x$ that sample from the above distributions.  

To complete the proof of Theorem 4.7 we show that $\text{PED}_{F_2,3}$ is in the class $\text{SZKP}_L$. This follows easily from the proof that $\text{ED}$ is in $\text{SZKP}$ [GV]. We give here a sketch of the proof.

**Lemma 4.9.** $\text{PED}_{F_2,3}$ has a statistical zero-knowledge proof system where the verifier and the simulator are computable in logarithmic space.

**Sketch.** We use the same proof system and simulator from [GV]. We need to show that on instance $(X, Y)$ where $X$ and $Y$ are $\text{POLYNOMIALS}_{F_2,3}$-mappings, the verifier and the simulator are computable in logarithmic space. For simplicity we assume that both $X$ and $Y$ map $n$ input bits to $m$ output bits. We start with the complexity of the verifier. The protocol is public coins, so we only need to check that the verifier’s final decision can be computed in logspace. This boils down to two operations which the verifier performs a polynomial number of times: (a) evaluating the $\text{POLYNOMIALS}_{F_2,3}$-mapping of either $X$ or $Y$ on an input specified in the transcript, and (b) evaluating a function $h : \{0, 1\}^{n+m} \rightarrow \{0, 1\}^k$, from a family of 2-universal hash functions, where both the description of $h$ and the input on which to evaluate $h$ are specified in the transcript. (See [GV] for the details.) The former can be done in logspace as it involves evaluating polynomials of degree $3$ over $\mathbb{F}_2$. The latter can be done in logspace if we use standard 2-universal families of hash functions, such as affine-linear maps from $\mathbb{F}_2^{n+m}$ to $\mathbb{F}_2^3$.

Turning to the simulator, we see that its output consists of many copies of triplets taking the following form: $(h, r, x)$ where $r \in R \{0, 1\}^n$, $x$ is an output of either $X$ or $Y$ on a uniformly chosen input which is part of the simulator’s randomness, and $h : \{0, 1\}^{n+m} \rightarrow \{0, 1\}^k$ is a function uniformly chosen from the family of 2-universal hash functions subject to the constraint that $h(r, x) = 0$. As in the verifier’s case, $x$ can be computed by a logspace mapping since $X$ and $Y$ are $\text{POLYNOMIALS}_{F_2,3}$-mappings. Choosing $h$ from the family of hash functions under a constraint $h(z) = 0$ can be done efficiently if we use linear hash functions $h(z) = Az + b$. We simply choose the matrix $A$ uniformly at random and set $b = -Az$.  

### 4.3.1 Additional Remarks

We remark, without including proofs, that similar statements as in the one from Theorem 4.7 can be shown for other known complete problems in $\text{SZKP}$ and its variants [SV, DDPY, GV, GSV2, Vad, Mal, BG, CCKV]). We also mention that all the known closure and equivalence properties of $\text{SZKP}$ (e.g. closure under complement [Oka]), equivalence between honest and dishonest verifiers [GSV1], and equivalence between public and private coins [Oka]) also hold for the class $\text{SZKP}_L$.

Finally, we mention that by using the locality reduction of [AIK] we can further reduce $\text{PED}_{F_2,3}$ to $\text{ED}_{NC^0_4}$, where $NC^0_4$ is the class of functions for which every output bit depends on at most four input bits.
4.4 Hardness Results

Given the results of Section 4.3, we can conclude that POLYNOMIAL ENTROPY DIFFERENCE (with additive Shannon entropy gap) is at least as hard as problems with statistical zero-knowledge proofs with logarithmic space verifiers and simulators. This includes problems such as GRAPH ISOMORPHISM, QUADRATIC RESIDUOUSITY, DECISIONAL DIFFIE HELLMAN, and the approximate CLOSEST VECTOR PROBLEM, and also many other cryptographic problems.

For the reduction from GRAPH ISOMORPHISM, we note that the operations run by the verifier and the simulator in the statistical zero-knowledge proof of [GMW], the most complex of which is permuting a graph, can all be done in logarithmic space. Similarly, for the approximate CLOSEST VECTOR PROBLEM, the computationally intensive operations run by the simulator in the zero-knowledge proof of [GG] (and the alternate versions in [MG]) are sampling from a high-dimensional Gaussian distribution and reducing modulo the fundamental parallellepiped. These can be done in logarithmic space. (To reduce modulo the fundamental parallellepiped we need to change the noise vector from the standard basis to the given lattice basis and back. By pre-computing the change-of-basis matrices, the sampling algorithm only needs to compute matrix-vector products, which can be done in logarithmic space.)

For the QUADRATIC RESIDUOUSITY and DECISIONAL DIFFIE HELLMAN problems, we show that in fact they reduce to an easier variant of PED, note that the operations run by the verifier and the simulator in the statistical zero-knowledge proof of [GMW], the most complex of which is permuting a graph, can all be done in logarithmic space. Similarly, for the approximate CLOSEST VECTOR PROBLEM, the computationally intensive operations run by the simulator in the zero-knowledge proof of [GG] (and the alternate versions in [MG]) are sampling from a high-dimensional Gaussian distribution and reducing modulo the fundamental parallellepiped. These can be done in logarithmic space. (To reduce modulo the fundamental parallellepiped we need to change the noise vector from the standard basis to the given lattice basis and back. By pre-computing the change-of-basis matrices, the sampling algorithm only needs to compute matrix-vector products, which can be done in logarithmic space.)

For the QUADRATIC RESIDUOUSITY and DECISIONAL DIFFIE HELLMAN problems, we show that in fact they reduce to an easier variant of PED, note that the operations run by the verifier and the simulator in the statistical zero-knowledge proof of [GMW], the most complex of which is permuting a graph, can all be done in logarithmic space. Similarly, for the approximate CLOSEST VECTOR PROBLEM, the computationally intensive operations run by the simulator in the zero-knowledge proof of [GG] (and the alternate versions in [MG]) are sampling from a high-dimensional Gaussian distribution and reducing modulo the fundamental parallellepiped. These can be done in logarithmic space. (To reduce modulo the fundamental parallellepiped we need to change the noise vector from the standard basis to the given lattice basis and back. By pre-computing the change-of-basis matrices, the sampling algorithm only needs to compute matrix-vector products, which can be done in logarithmic space.)

For the QUADRATIC RESIDUOUSITY and DECISIONAL DIFFIE HELLMAN problems, we show that in fact they reduce to an easier variant of PED, note that the operations run by the verifier and the simulator in the statistical zero-knowledge proof of [GMW], the most complex of which is permuting a graph, can all be done in logarithmic space. Similarly, for the approximate CLOSEST VECTOR PROBLEM, the computationally intensive operations run by the simulator in the zero-knowledge proof of [GG] (and the alternate versions in [MG]) are sampling from a high-dimensional Gaussian distribution and reducing modulo the fundamental parallellepiped. These can be done in logarithmic space. (To reduce modulo the fundamental parallellepiped we need to change the noise vector from the standard basis to the given lattice basis and back. By pre-computing the change-of-basis matrices, the sampling algorithm only needs to compute matrix-vector products, which can be done in logarithmic space.)

For the QUADRATIC RESIDUOUSITY and DECISIONAL DIFFIE HELLMAN problems, we show that in fact they reduce to an easier variant of PED, note that the operations run by the verifier and the simulator in the statistical zero-knowledge proof of [GMW], the most complex of which is permuting a graph, can all be done in logarithmic space. Similarly, for the approximate CLOSEST VECTOR PROBLEM, the computationally intensive operations run by the simulator in the zero-knowledge proof of [GG] (and the alternate versions in [MG]) are sampling from a high-dimensional Gaussian distribution and reducing modulo the fundamental parallellepiped. These can be done in logarithmic space. (To reduce modulo the fundamental parallellepiped we need to change the noise vector from the standard basis to the given lattice basis and back. By pre-computing the change-of-basis matrices, the sampling algorithm only needs to compute matrix-vector products, which can be done in logarithmic space.)
The probability that for no \( i \in [2k] \), \( r_i \in [N] \) is at most \( 1/N^2 \). Hence in the above analysis both probabilities change by at most \( 1/N^2 \). The support of the modified mapping remains the same as the original one. It follows (by the proof above) that there is a constant gap between the max and min entropies of yes and no instances. This gap can be amplified to be more than 1 by taking two independent copies of the mapping.

### 4.4.2 Decisional Diffie-Hellman

**Definition 4.13 (Decisional Diffie Hellman).**
The promise problem Decisional Diffie Hellman is defined with respect to a family \( \mathcal{G} \) of cyclic groups of prime order. It is defined as follows:

\[
\text{DDH}_\text{YES} = \{(G, g, g^a, g^{a^b} : G \in \mathcal{G} \text{ of prime order } q, g \text{ generator of } G, a, b \in \mathbb{Z}_q\}
\]

\[
\text{DDH}_\text{NO} = \{(G, g, g^a, g^{a^b} : G \in \mathcal{G} \text{ of prime order } q, g \text{ generator of } G, a, b, c \in \mathbb{Z}_q, c \neq a \cdot b\}
\]

**Claim 4.14.** Decisional Diffie Hellman reduces to the problem \( \text{PED}_{\min, \max}^{\min, \max, +1} \).

**Proof Sketch.** We use the random self-reducibility of DDH, due to Naor and Reingold [NR]. They showed how to transform a given DDH instance \( x = (G, g, g^a, g^b, g^c) \) into a new one \( (G, g, g^{a'}, g^{b'}, g^{c'}) \), such that \( a', b' \) are uniformly random in \( \mathbb{Z}_q \) and: (i) if \( x \) is a YES instance (i.e. \( c = a \cdot b \)) then \( c' = a' \cdot b' \), so the output (in its entirety) is uniform over a set of size \( |G|^2 \). On the other hand, (ii) if \( x \) is a NO instance then \( c' \) (and also \( g^{c'} \)) is uniformly random given \( (G, g, g^{a'}, g^{b'}) \) and the output (in its entirety) is uniform over a set of size \( |G|^3 \).

The mapping computed by this reduction allows us to transform a Decisional Diffie Hellman instance in an instance of ED, where yes instances are transformed into pair of mappings or distributions \((X, Y)\), where \( X \) is uniform over a set of size \( |G|^2 \) (some fixed dummy distribution) and on YES instances \( Y \) is uniform over a set of size \( |G|^2 \) and on NO instances \( Y \) is uniform over a set of size \( |G|^3 \). I.e., it reduces Decisional Diffie Hellman to \( \text{ED}_{\min, \max}^{\min, \max, +1} \).

Finally, to reduce Decisional Diffie Hellman to PED we need to activate the randomizing polynomial machinery of Theorem 4.6. The maps \( X \) and \( Y \) as described above compute multiplication (which can be done in log-space) and exponentiation, which is not a log-space operation. However, the elements being exponentiated are all known in advance. We can thus use an idea due to Kears and Valiant [KV] and compute in advance for each of these bases, say \( g \), the powers \( g, g^2, g^4, g^8, \ldots \). Each exponentiation can then be replaced by an iterated product. By the results of Beame, Cook and Hoover [BCH] the iterated product can be computed in logarithmic depth (or space). By Theorem 4.6, we conclude that Decisional Diffie Hellman reduces to \( \text{PED}_{\min, \max}^{\min, \max, +1} \).

### 5 Algorithms for Polynomial Entropy Approximation

#### 5.1 Approximating Entropy via Directional Derivatives

In this section we give an approximation algorithm for the entropy of homogenous polynomial maps of degree two, over prime fields \( \mathbb{F}_q \) other than \( \mathbb{F}_2 \). The general strategy is to relate the entropy of a quadratic map with the entropy of a random directional derivative of the map. These derivatives are of degree one and so their entropy is easily computable.

For a polynomial mapping \( P : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n \) and a vector \( a \in \mathbb{F}_q^n \) we define the directional derivative of \( P \) in direction \( a \) as the mapping \( D_a P : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n \) given by

\[
D_a P(x) \overset{\text{def}}{=} P(x + a) - P(x).
\]

It is easy to verify that for every fixed \( a \), \( D_a P(x) \) is a polynomial mapping of degree at most \( \log(P) - 1 \).

Throughout this section, \( q \) is a prime other than 2 and \( Q : \mathbb{F}_q^n \mapsto \mathbb{F}_q^n \) denotes a homogenous quadratic mapping given by \( m \) quadratic polynomials \( Q_1(x), \ldots, Q_m(x) \) in \( n \) variables \( x = (x_1, \ldots, x_n) \). For every \( i \in [m] \) there exists an \( n \times n \) matrix \( M_i \) such that \( Q_i(x) = x^T \cdot M_i \cdot x \). If \( \text{char}(\mathbb{F}_q) \neq 2 \) then we can assume w.l.o.g that \( M_i \) is always symmetric (by replacing \( M_i \) with \( (M_i + M_i^T)/2 \) if needed).

For every fixing of \( a \), \( D_a Q(x) \) is an affine (degree at most one) mapping. We denote by \( r(a) \) the rank of \( D_a Q(x, a) \) (that is, the dimension of the affine subspace that is the image of the mapping given by \( D_a Q(x) \)). We relate the \( r(a) \)'s to entropy in the following two lemmas:

**Lemma 5.1.** For every \( a \in \mathbb{F}_q^n \) we have

\[
r(a) \leq 2 \cdot H_{\text{Shannon}}(Q(U_n))/\log q.
\]

**Lemma 5.2.**

\[
E \left[ q^{-r(a)} \right] = 2^{-H_{\text{Shannon}}(Q(U_n))}.
\]

Before proving these lemmas, we use them to obtain our algorithm:

**Theorem 5.3.** There exists a probabilistic polynomial-time algorithm \( A \) that, when given a prime \( q \neq 2 \), a
homogeneous quadratic map \( Q : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n \) (as a list of coefficients), and an integer \( 0 < k \leq m \) and outputs TRUE or FALSE such that:
- If \( H_{\text{Renyi}}(Q(U_n)) \geq 2k \cdot \log(q) + 1 \) then \( A \) outputs TRUE with probability at least \( 1/2 \).
- If \( H_{\text{Shannon}}(Q(U_n)) < k \cdot \log(q) \) then \( A \) always outputs FALSE.

Proof. The algorithm simply computes the rank \( r(\alpha) \) of the directional derivative \( D_\alpha Q \) in a random direction \( \alpha \in \mathbb{F}_q^n \). If the value of \( r(\alpha) \) is at least \( 2k \) the algorithm returns TRUE, otherwise the algorithm returns FALSE. If \( H_{\text{Shannon}}(Q(U_n)) < k \cdot \log(q) \) then, from Lemma 5.1 we have that \( r(\alpha) \) will always be smaller than \( 2k \) and so the algorithm will work with probability one. If \( H_{\text{Renyi}}(Q(U_m)) \geq 2k \cdot \log(q) + 1 \) then, using Lemma 5.2 and Markov’s inequality, we get that \( q^{-r(\alpha)} \) will be at most \( 2^{-2k \log_q} \) with probability at least \( 1/2 \). Therefore, the algorithm works as promised.

We now prove the two main lemmas.

**Proof of Lemma 5.1.** Since the output of an affine mapping is uniform on its output, we have

\[
H_{\text{Shannon}}(D_\alpha Q(U_n)) = \log(q^{r(\alpha)}).
\]

By subadditivity of Shannon entropy, we have

\[
H_{\text{Shannon}}(D_\alpha Q(U_n)) \leq H_{\text{Shannon}}(Q(U_n + a)) + H_{\text{Shannon}}(Q(U_n)) = 2H_{\text{Shannon}}(Q(U_n)).
\]

The proof of Lemma 5.2 works by expressing both coefficients of the distribution \( Q(U_n) \), which are simply given by the following biases:

**Definition 5.4.** For a prime \( q \) and a random variable \( X \) taking values in \( \mathbb{F}_q \), we define

\[
bias(X) \overset{\text{def}}{=} |E[\omega_q^X]|,
\]

where \( \omega_q = e^{2\pi i/q} \) is the complex primitive \( q \)’th root of unity. For a random variable \( Y \) taking values in \( \mathbb{F}_q^m \) and a vector \( u \in \mathbb{F}_q^m \), we define we define

\[
bias_u(Y) \overset{\text{def}}{=} bias((u, Y)) = |E[\omega_q^{u \cdot Y}]|,
\]

where \( \langle \cdot, \cdot \rangle \) is inner product modulo \( q \).

Note that if \( Y \) is uniform on \( \mathbb{F}_q^m \), then \( bias_u(Y) = 0 \) for all \( u \neq 0 \). A relation between bias and rank in the case of a single output (i.e. \( m = 1 \)) is given by the following:

**Claim 5.5.** Suppose \( \text{char}(\mathbb{F}_q) \neq 2 \). Let \( R(x_1, \ldots, x_n) = x^t M x \) be a homogeneous quadratic polynomial over \( \mathbb{F}_q \) such that \( \text{rank}(M) = k \) and \( M \) is symmetric. Then,

\[
bias(R(U_n)) = q^{-k/2}.
\]

**Proof.** As shown in [LN], \( R(x) \) is equivalent (under a linear change of variables) to a quadratic polynomial \( S(x) = \sum_{i=1}^k a_i x_i^2 \) where \( a_1, \ldots, a_k \in \mathbb{F}_q \). Then

\[
bias(R(U_n)) = \text{bias}(S(U_n)) = \frac{1}{q^m} \sum_{x \in \mathbb{F}_q^m} \omega_q^{a_i x_i^2} \prod_{i \in [k]} \frac{1}{q} \sum_{y \in \mathbb{F}_q} \omega_q^{a_i y^2} = (q^{-1/2})^k = q^{-k/2},
\]

where the last equality follows from the Gauss formula for quadratic exponential sums in one variable (see [LN]).

Next we relate biases for many output coordinates to Renyi entropy.

**Claim 5.6.** Let \( X \) be a random variable taking values in \( \mathbb{F}_q^m \). Then

\[
2^{-H_{\text{Renyi}}(X)} = \mathbb{E}_{u \in \mathbb{F}_q^m} \text{bias}_u(X)^2.
\]

**Proof.** We begin by recalling that the Renyi entropy simply measures the \( \ell_2 \) distance of a random variable from uniform:

\[
2^{-H_{\text{Renyi}}(X)} = \text{cp}(X) = \sum_x \text{Pr}[X = x]^2 = \sum_x (\text{Pr}[X = x] - 1/q^m)^2 + 1/q^m = \|X - U_m\|^2 + 1/q^m,
\]

where \( \|X - U_m\| \) denotes the \( \ell_2 \) distance between the probability mass functions of \( X \) and \( U_m \) (viewed as vectors of length \( q^m \)). By Parseval, the \( \ell_2 \) distance does not change if we switch to the Fourier basis: For \( u \in \mathbb{F}_q^m \), the \( u \)’th Fourier basis function \( \chi_u: \mathbb{F}_q^m \rightarrow \mathbb{C} \) is the function given by

\[
\chi_u(x) = \frac{1}{q^{m/2}} \cdot \omega_q^{u \cdot x}.
\]

These form an orthonormal basis for the vector space of functions from \( \mathbb{F}_q^m \) to \( \mathbb{C} \), under the standard inner product \( [f, g] = \sum_{x \in \mathbb{F}_q^m} f(x)\overline{g(x)} \).
Abusing notation, we can view a random variable $X$ taking values in $\mathbb{F}_q^n$ as a function $X : \mathbb{F}_q^n \rightarrow \mathbb{C}$ where $X(x) = \Pr[X = x]$. Then the $u$’th Fourier coefficient of $X$ is given by

$$\hat{X}_u \overset{\text{def}}{=} [X, \chi_u] = \frac{1}{q^m/2} \sum_{x \in \mathbb{F}_q^n} \Pr[X = x] \cdot \omega_q^{-u \cdot x} = \frac{1}{q^m/2} \cdot \mathbb{E} \left[ \omega_q^{-(u \cdot X)} \right],$$

so $|\hat{X}_u| = (1/q^{m/2}) \cdot \text{bias}_u(X)$.

Thus, by Parseval, we have:

$$\|X - U_m\|^2 = \sum_u |\hat{X}_u - (U_m)_u|^2 = \sum_{u \neq 0} |\hat{X}_u|^2 = \mathbb{E} \left[ \text{bias}_u(X)^2 \right] - 1/q^m.$$

Putting this together with the first sequence of equations completes the proof. □

**Proof of Lemma 5.2.** Taking $X = Q(U_m)$ in Claim 5.6, we have

$$2^{-H_{\text{max}}(Q(U_n))} = \mathbb{E}_{u \in \mathbb{F}_q^n} \left[ \text{bias}_u(Q(U_n))^2 \right].$$

By Claim 5.5, $\text{bias}_u(Q(U_n))^2 = \text{bias}(\sum_i u_i Q_i(U_n))^2 = q^{-\text{rank}(\sum_i u_i M_i)}$. Note that for a $s \times t$ matrix $M$, $q^{-\text{rank}(M)} = \Pr_{v \in \mathbb{F}_q^m} [Av = 0]$. Thus, we have

$$2^{-H_{\text{max}}(Q(U_n))} = \mathbb{E}_{u \in \mathbb{F}_q^n} \left[ q^{-\text{rank}(\sum_i u_i M_i)} \right] = \mathbb{E}_{u \in \mathbb{F}_q^n} \left[ \Pr_{v \in \mathbb{F}_q^m} \left[ \sum_i u_i M_i a = 0 \right] \right] = \mathbb{E}_{a \in \mathbb{F}_q^d} \left[ \Pr_{u \in \mathbb{F}_q^n} \left[ \sum_i u_i M_i a = 0 \right] \right] = \mathbb{E}_{a \in \mathbb{F}_q^d} \left[ q^{-\text{rank}(a)} \right],$$

where the last equality is because $\sum_i u_i M_i a = 0$ iff $u M_a = 0$ where $M_a$ is the matrix whose rows are $M_1 a, \ldots, M_n a$ (and so $r(a) = \text{rank}(M_a)$). □

### 5.2 Approximating Max-Entropy over $\mathbb{F}_2$ via Rank

In this section we deal with degree $d$ polynomials over $\mathbb{F}_2$. Since the field is $\mathbb{F}_2$ we can assume w.l.o.g. that the polynomials are multilinear (degree at most 1 in each variable). We show that, for small $d$, the rank of the set of polynomials (when viewed as vectors of coefficients) is related to the entropy of the polynomial map.

The results of this section can be extended to any field but we state them only for $\mathbb{F}_2$ since this is the case we are most interested in (and is not covered by the results of Section 5.1).

The main technical result of this section is the following theorem, which we prove in Section 5.3 below. The theorem relates the entropy of a polynomial mapping (in the weak form of support) with its rank as a set of coefficient vectors.

**Theorem 5.7.** Let $P : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ be a multilinear polynomial mapping of degree $\leq d$ such that $|\text{Supp}(P)| \leq 2^k$, for $k, d \in \mathbb{N}$. Then

$$\text{rank}(P_1, \ldots, P_m) \leq \left( \frac{k + 2d}{d} \right).$$

where the rank is understood as the dimension of the $\mathbb{F}_2$-span of $P_1, \ldots, P_m$ (equivalently, the rank of the $m \times (n + d)$ matrix over $\mathbb{F}_2$ whose rows are the coefficient-vectors of the polynomials $P_i$).

Using this theorem we get the following approximation algorithm for max-entropy over characteristic two:

**Theorem 5.8.** There exists a constant $c$ and polynomial-time algorithm $A$ such that when $A$ is given as input a degree $d$ polynomial map $P : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ and an integer $0 < k \leq n$, we have:

- If $H_{\text{max}}(P(U_m)) > \left( \frac{k + 2d}{d} \right)$, then $A$ outputs TRUE.
- If $H_{\text{max}}(P(U_m)) \leq k$, then $A$ outputs FALSE.

Proof. The algorithm computes the rank of the set of polynomials $P_1, \ldots, P_m$. If it is greater than $\left( \frac{k + 2d}{d} \right)$ then it returns TRUE, otherwise it returns FALSE. The correctness follows directly from Theorem 5.7 and from the simple fact that rank at most $k$ implies support size at most $2^k$. □

### 5.3 Proof of Theorem 5.7

The idea of the proof is to find an affine-linear subspace $V \subset \mathbb{F}_2^m$ of dimension $\approx k$ such that the restriction of the polynomials $P_1, \ldots, P_m$ to this subspace does not reduce their rank. Since the restricted polynomials are polynomials of degree $\leq d$ in $\approx k$ variables we get that their rank is at most $\approx k^d$.

It turns out that it suffices to take $V$ to be a subspace that hits a large fraction of the outputs of $P$, as given by the image of $L$ in the following lemma:

**Lemma 5.9.** Let $P : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ be some function such that $|\text{Supp}(P(U_m))| \leq 2^k$ and let $\varepsilon > 0$. Then, there exists an affine-linear mapping $L : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^m$ with $\ell = \lfloor k + \log(1/\varepsilon) \rfloor$ such that

$$\Pr_{x \in \mathbb{F}_2^n} \left[ \exists y \in \mathbb{F}_2^m, \ P(x) = P(L(y)) \right] > 1 - \varepsilon.$$
Proof. We use the probabilistic method. Let \( L : \mathbb{F}_2^d \to \mathbb{F}_2^n \) be a uniformly random affine-linear mapping. Fix \( z \in \text{Image}(P) \), and let \( \mu_z = |P^{-1}(z)|/2^n \). By the pairwise independence of the outputs of \( L \) and Chebychev’s Inequality, it follows that
\[
\Pr_{L}[\text{Image}(L) \cap P^{-1}(z) = \emptyset] \leq \frac{1 - \mu_z}{\mu_z \cdot 2\ell}.
\]
(For each point \( y \in \mathbb{F}_2^d \), let \( X_y \) be the indicator variable for \( L(y) \in P^{-1}(z) \). Then the \( X_y \)'s are pairwise independent, each with expectation \( \mu_z \) and variance \( \mu_z \cdot (1 - \mu_z) \). Thus, by Chebychev’s Inequality, \( \Pr[\sum_y X_y = 0] \leq (2^\ell \cdot \mu_z \cdot (1 - \mu_z))/2^{\ell \cdot \mu_z^2} \).

Now, let \( I_z \) be an indicator random variable for \( \text{Image}(L) \cap P^{-1}(z) = \emptyset \). Then,
\[
\mathbb{E}_{L} \left[ \Pr_{x \in \mathbb{F}_2^d} [\exists y \in \mathbb{F}_2^d \text{ s.t. } P(x) = P(L(y)) \land P(x) \neq P(L(y))] \right] = \mathbb{E}_{L} \left[ \Pr_{x \in \mathbb{F}_2^d} \left[ \text{Image}(L) \cap P^{-1}(P(x)) = \emptyset \right] \right]
\]
\[
= \mathbb{E}_{L} \left[ \sum_{z \in \text{Image}(P)} \mu_z \cdot I_z \right]
\]
\[
\leq \sum_{z \in \text{Image}(P)} \mu_z \cdot \frac{1 - \mu_z}{\mu_z \cdot 2\ell}
\]
\[
= \frac{\text{Image}(P)}{2\ell} - 1
\]
\[
< \frac{2^k - 1}{2}\epsilon
\]
for \( \ell = \lfloor k + \log(1/\epsilon) \rfloor \). By averaging, there exists a fixed \( L \) such that
\[
\Pr_{x \in \mathbb{F}_2^d} [\exists y \in \mathbb{F}_2^d \text{ s.t. } P(x) = P(L(y))] < \epsilon,
\]
as desired.

To show that the property of \( L \) given in Lemma 5.9 implies that \( P \circ L \) has the same rank as \( P \) (when \( \epsilon \) is sufficiently small), we employ the following (known) version of the Schwartz-Zippel Lemma, which bounds the number of zeros of a multilinear polynomial of degree \( d \) that is not identically zero:

Lemma 5.10. Let \( P \in \mathbb{F}_2[x_1, \ldots, x_n] \) be a degree \( d \) multilinear polynomial that is not identically zero. Then
\[
\Pr[P(x) = 0] \leq 1 - 2^{-d}.
\]

Proof. The proof is by double induction on \( d = 1, 2, \ldots \) and \( n = d, d + 1, \ldots \). If \( d = 1 \) then the claim is trivial. Suppose we proved the claim for degree \( < d \) and all \( n \) and for degree \( d \) and \( < n \) variables.

If \( n = d \) (it cannot be smaller than \( d \) since the degree is \( d \)) then the bound is trivial since there is at least one point at which \( P \) is non zero and this point has weight \( 2^{-d} \).

Suppose \( n > d \) and assume w.l.o.g that \( x_1 \) appears in \( P \). Write \( P \) as
\[
P(x_1, \ldots, x_n) = x_1 \cdot R(x_2, \ldots, x_n) + S(x_2, \ldots, x_n),
\]
where \( R \) has degree \( \leq d - 1 \) and \( S \) has degree \( \leq d \). We separate into two cases. The first case is when \( R(x_2, \ldots, x_n) + S(x_2, \ldots, x_n) \) is identically zero. In this case we have
\[
P(x) = (x_1 + 1) \cdot R(x_2, \ldots, x_n)
\]
and so, by the inductive hypothesis,
\[
\Pr[P(x) = 0] = \Pr[x_1 = 1] + \Pr[x_1 = 0] \cdot \Pr[R(x_2, \ldots, x_n) = 0] \leq 1/2 + 1/2(1 + 2^{-(d-1)}) = 1 - 2^{-d}.
\]

In the second case we have that \( R(x_2, \ldots, x_n) + S(x_2, \ldots, x_n) \) is not identically zero. Now,
\[
\Pr[P(x) = 0] = \Pr[x_1 = 0] \cdot \Pr[S(x_2, \ldots, x_n) = 0] + \Pr[x_1 = 1] \cdot \Pr[R + S = 0] \leq (1/2)(1 - 2^{-d}) + (1/2)(1 - 2^{-d}) = 2^{-d},
\]
as was required.

We now combine these two lemmas to show that there exists a linear restriction of \( P \) to a small number of variables that preserves independence of the coordinates of \( P \).

Lemma 5.11. Let \( P : \mathbb{F}_2^n \to \mathbb{F}_2^m \) be a multilinear mapping of degree \( \leq d \) such that \( |\text{Supp}(P)| \leq 2^k \) for \( k, d \in \mathbb{N} \). Denote by \( P_1, \ldots, P_m \in \mathbb{F}_2[x_1, \ldots, x_n] \) the coordinates of \( P \). Suppose that \( P_1, \ldots, P_m \) are linearly independent (in the vector space \( \mathbb{F}_2[x_1, \ldots, x_n] \)). Then, there exists an affine-linear mapping \( L : \mathbb{F}_2^d \to \mathbb{F}_2^d \) with \( \ell = k + d \) such that the restricted polynomials \( P_i(L(y_1, \ldots, y_n)) \), \( j \in [m] \) are also independent.

Proof. Apply Lemma 5.9 with \( \epsilon < 2^{-d} \) on the mapping \( P \) to find an affine-linear mapping \( L : \mathbb{F}_2^d \to \mathbb{F}_2^d \) with \( \ell = k + d \) and such that
\[
\Pr_{x \in \mathbb{F}_2^d} [\exists y \in \mathbb{F}_2^d \text{ s.t. } P(x) = P(L(y))] > 1 - 2^{-d}.
\]
Call an element \( x \in \mathbb{F}_2^d \) ‘good’ if the event above happens (so \( x \) is good w.p. \( 1 - 2^{-d} \)). For \( j \in [m] \) let \( R_j(y_1, \ldots, y_n) = P_j(L(y_1, \ldots, y_n)) \) (notice that since \( L \) is linear the polynomials \( R_j \) are
also of degree at most $d$ but are not necessarily multilinear. Suppose in contradiction that the polynomials $R_1, \ldots, R_m$ are linearly dependent. So there is a non empty set $I \subseteq [m]$ such that $R_I(y) = \sum_{i \in I} R_i(y) = 0$ for every $y \in \mathbb{F}_2^n$. Let $P_I(x) = \sum_{i \in I} P_i(x)$.

Then, if $x$ is good we have that there exists $y$ such that $P(x) = P(L(y))$ and so we get that

$$P_I(x) = P_I(L(y)) = R_I(y) = 0.$$ 

This means that $P_I(x)$, which is a multilinear polynomial of degree at most $d$, is zero on a fraction bigger than $1 - 2^{-d}$ of the inputs. Using Lemma 5.10 we conclude that $P_I(x)$ is identically zero and so the $P_i$’s are linearly dependent – a contradiction. $
$

**Corollary 5.12.** Let $P : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ be a multilinear mapping of degree $\leq d$ such that $|\text{Supp}(P)| \leq 2^k$, for $k, d \in \mathbb{N}$. Denote by $P_1, \ldots, P_m \in \mathbb{F}_2[x_1, \ldots, x_n]$ the coordinates of $P$. Suppose that the set $P_1, \ldots, P_m$ has rank $\geq r$ (in the vector space $\mathbb{F}_2[x_1, \ldots, x_n]$). Then, there exists an affine-linear mapping $L : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ with $\ell = k + d$ such that the restricted polynomials $P_j(L(y_1, \ldots, y_n)), j \in [m]$ also have rank $\geq r$.

**Proof.** W.l.o.g suppose that $P_1, \ldots, P_r$ are linearly independent and apply Lemma 5.11 on the mapping $\tilde{P} = (P_1, \ldots, P_r) : \mathbb{F}_2^r \rightarrow \mathbb{F}_2^m$. The support of $\tilde{P}$ is also at most $2^k$ and so we $L : \mathbb{F}_2^r \rightarrow \mathbb{F}_2^n$ such that the restriction $\tilde{P}(L(y))$ has rank $r$. Now, adding the $m - r$ coordinates $P_{r+1}(L(y)), \ldots, P_m(L(y))$ cannot decrease the rank.

We are now ready to prove the Theorem.

**Proof of Theorem 5.7.** Let $r$ denote the rank of the set of polynomials $\{P_1, \ldots, P_m\}$. Then, using Corollary 5.12, there exists a linear mapping $L : \mathbb{F}_2^r \rightarrow \mathbb{F}_2^n$, with $\ell = k + d$, such that the restricted polynomials $P_1(L(y)), \ldots, P_m(L(y))$ also have rank $\geq r$. Since these are polynomials of degree $\leq d$ in $\ell$ variables, their rank is bounded from above by the number of different monomials of degree at most $d$ in $\ell = k + d$ variables, which equals $\binom{\ell + d}{d}$.

### Acknowledgment

We thank Daniele Micciancio and Shrenik Shah for helpful discussions. Z. Dvir was partially supported by NSF grant CCF-0832797. Part of this research was done while D. Gutfreund was at Harvard University and supported by ONR grant N00014-04-1-0478 and NSF grant CNS-0430336. G. Rothblum was supported by NSF Grants CCF-0635297, CCF-0832797, CNS-0430336 and by a Computing Innovation Fellowship. S. Vadhan was supported by NSF grant CNS-0831289.

### References


[MG] Daniele Micciancio and Shafi Goldwasser. Complexity of Lattice Problems: a cryptographic perspective, volume 671 of The Kluwer Interna-