# On Proximity Gaps for Expander Codes

Kobi Gurkan* Andrija Novakovic† Ron D. Rothblum‡

**Abstract**

Proximity gaps describe a phenomenon concerning random linear combinations of words and their distance from a linear error-correcting code. Informally, they address the following question: if a collection of words is jointly far from the code, is a random linear combination of those words likely to also remain far from the code?

Motivated by applications to concretely efficient proof systems, in this note we study this question for expander codes, a class of low-density parity-check (LDPC) codes. Recent works of Goyal and Guruswami (STOC 2026) and Goyal, Guruswami, Sun, and Wootters (RANDOM 2026; arXiv 2026) establish proximity gaps approaching capacity for *random* LDPC ensembles, but only asymptotically and for very large degrees and field sizes, well beyond what is used in practical proof systems.

We focus on general expander codes (not necessarily random) with small degree. Building on the smoothness property of expander codes due to Dinur, Sudan, and Wigderson (RANDOM 2006), we show that such codes have proximity gaps up to nearly half of the distance guaranteed by their expansion.

## 1 Introduction

Modern hash-based SNARKs rely on error-correcting codes and proximity tests to verify large computations succinctly. Loosely speaking, a proximity test with *proximity radius* $\theta$ rejects, with high probability, every word that is $\theta$-far from the code, while making only a few queries. Since each query catches a $\theta$-far word with probability roughly $\theta$, about $\lambda/\theta$ queries suffice for soundness error $2^{-\lambda}$. Thus the larger the proximity radius, the fewer the queries, and hence the smaller the resulting proof.

For Reed–Solomon codes, which underlie many hash-based SNARKs, strong proximity results are known: proximity gaps hold up to the Johnson radius [BCI+23, BCH+26] (and, asymptotically, even beyond it [Jer26]). However, the fastest provers known today are obtained from error-correcting codes that can be encoded in linear time over arbitrary fields [BCG+17, BCG20, XZS22, RR25, GLS+23, BFK+24, NST24, BCF+25, ARR25, BCFW25, BMMS26a, BMMS26b, BFRW26, KPR+26, BFMM26, GNR26, QGZ26]. Unfortunately, for these codes the Reed–Solomon proximity results do not apply, and one is left with guarantees that hold for all linear codes [AHIV23, BGKS20, Zei26], whose radius is roughly a third of the code's relative distance (see below).

*zkSecurity. Email: kobi@zksecurity.xyz.

†Bain Capital Crypto. Email: anovakovic@baincapital.com.

‡Succinct and Technion. Email: rothblum@gmail.com. Supported in part by the European Union (ERC, FASTPROOF, 101041208). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

*Low-density parity-check* (LDPC) codes [Gal62], and in particular *expander codes* [SS96],[1] are a natural candidate for obtaining stronger proximity guarantees beyond Reed–Solomon codes: they are defined by sparse constraints, have good distance, and sparse-graph codes are a central ingredient in several of the constructions above. Bolt[2] [GNR26] makes the connection especially direct: it uses *sketched codes*, a particularly efficient family of error-correcting codes in which an LDPC code is a key component.[3] As noted in [GNR26], proximity gaps for LDPC codes, which allow folding-based proximity tests to use a large radius, can dramatically improve the proof size of Bolt.

Motivated by these applications, and by Bolt in particular, in this work we study proximity gaps for expander codes.

### 1.1 Our results

Our main result is a proximity gap at (nearly) half the distance for expander codes with sufficiently good expansion. Recall that a bipartite graph with left degree $c$ is a $(\gamma, \delta)$-expander if every set $S$ of at most $\delta n$ left vertices has at least $(1-\gamma)c|S|$ neighbors, and that the expander code it defines has distance at least $d = 2(1-\gamma)\delta n$ [SS96].

**Theorem 1** (Informal version of Theorem 8)**.** *Let $C \subseteq \mathbb{F}^n$ be an expander code with expansion parameter $\gamma < 1/3$ and check set $R$, and let $(u, v)$ be an interleaved word at distance more than $d/2$ from $C^2$. Then, for every $0 < \varepsilon < 1 - 3\gamma$, except with probability $|R|/|\mathbb{F}|$ over a uniformly random $r \in \mathbb{F}$, the fold $u + rv$ is at distance at least $(1 - 3\gamma - \varepsilon) \cdot d/2$ from $C$.*

Plugging Theorem 1 into the standard folding-and-spot-check test of Ligero [AHIV23] yields a proximity test for $C^2$ with radius $d/2$ (Section 4.1). We also give a variant in which the verifier samples parity checks rather than coordinates, which reduces the folding error from $|R|/|\mathbb{F}|$ to $1/|\mathbb{F}|$ (see Theorem 12 below).

For comparison, the best guarantees known for general linear codes [BGKS20, Zei26] reach the one-and-a-half Johnson radius

$$J_{3/2}(\rho) = 1 - (1-\rho)^{1/3},$$

where $\rho$ denotes the relative distance. Half the distance exceeds this benchmark whenever $0 < \rho < 3 - \sqrt{5} \approx 0.764$. For small $\rho$, we have $J_{3/2}(\rho) \approx \rho/3$, so moving to radius $\theta = \rho/2$ reduces the number of queries by roughly a factor of $3/2$. We note that $d$ is the lower bound on the distance implied by expansion, which may be smaller than the true distance of the code; we return to this point in Section 1.2.

A natural question is whether expander codes, or LDPC codes more generally, have proximity gaps up to the Johnson radius, or even beyond it, as Reed–Solomon codes do. As we discuss next, this is known asymptotically for *random* LDPC ensembles, but not with concrete parameters for the small-degree codes used in practice.

[1] An LDPC code is the kernel of a sparse parity-check matrix. Equivalently, it is described by a bipartite graph with the $n$ coordinates of the codeword on one side and the parity checks on the other, where each check requires a weighted sum of its few neighboring coordinates to vanish. An expander code is an LDPC code whose graph is a good *expander*: every small set of coordinates is adjacent to many checks, nearly as many as the number of edges leaving it. Sipser and Spielman [SS96] showed that expansion alone guarantees good distance (and linear-time decoding), and suitable random sparse graphs are good expanders with high probability.

[2] More precisely, Bolt is a *polynomial commitment scheme* (PCS), a major component of modern SNARKs.

[3] Similar codes were introduced, concurrently and independently, in Lightning [QGZ26], but the question of proximity gaps for LDPC codes is not raised there.

**Comparison with random LDPC ensembles.** Goyal and Guruswami [GG26] recently established proximity gaps approaching capacity for subspace-design codes, and extended them to several random ensembles, including random linear codes, Reed–Solomon codes with random evaluation points, and Gallager's ensemble of LDPC codes. Goyal, Guruswami, Sun and Wootters [GGSW26a] improved the parameters for these random ensembles, and subsequently reduced the required field size exponentially [GGSW26b]. In the LDPC case, these guarantees are asymptotic in several respects. Writing $\eta$ for the gap between the proximity radius and capacity, they apply only to codes whose check degree exceeds a threshold that is not given explicitly and depends on $\eta$, the rate, the radius and the field size (see [GG26, Theorem 5.18] and [GGSW26b, Theorem 6.4]); only for sufficiently large block lengths; and only over fields of size $\exp(\Omega(\log(1/\eta)/\eta))$, that is, exponential in $1/\eta$. In particular, even setting aside concrete constants, they do not apply to LDPC codes of a fixed small degree, such as those used in practical SNARKs and polynomial commitment schemes.

Concretely, for the parameters used in Bolt [GNR26], the analysis of [GGSW26b, Theorem 6.4] requires fields of thousands of bits; see Appendix A.

**Formalization.** We provide an experimental Lean 4 formalization of the results in this note, produced largely by autoformalization, at `https://github.com/zksecurity/proximity-expanders`.

### 1.2 Open problems

Beyond the question raised above of obtaining proximity gaps past radius $d/2$ (up to the Johnson radius or further, with concrete parameters for small-degree codes), our work leaves open the following questions.

- *Half of the true distance.* Our radius is half of the distance bound $d$ implied by expansion, and requires $\gamma < 1/3$. For codes whose distance is established by other means, such as the Gallager-style counting arguments used for the codes in Bolt [GNR26], the true distance may be considerably larger. Can one obtain proximity gaps at half of the true distance of such codes? A positive answer would directly increase the proximity radius available to Bolt, and hence considerably reduce its proof size.
- *Folding error.* Theorem 1 incurs a folding error of $|R|/|\mathbb{F}|$, due to a union bound over the violated checks. The check-sampling test of Section 4.2 avoids this loss by modifying the test. Does the standard Ligero test itself achieve error $O(1/|\mathbb{F}|)$ at radius $d/2$, or is the loss inherent?

### 1.3 AI disclosure

The main ideas in this note were biologically generated. AI tools were used extensively in the writing of the note, and also assisted with the proof of Proposition 10. The Lean formalization mentioned above was likewise produced largely by AI-assisted autoformalization.

## 2 Preliminaries

For strings $x, y \in \Sigma^n$ over a finite alphabet $\Sigma$, their Hamming distance is defined as $\Delta(x, y) := |\{i \in [n] : x_i \neq y_i\}|$. For a non-empty set $S \subseteq \Sigma^n$, let $\Delta(x, S) := \min_{c \in S} \Delta(x, c)$. For a set $I \subseteq [n]$, we denote by $x_I$ the restriction of $x$ to the coordinates in $I$. Note that $\Delta(x_I, y_I) \leq \Delta(x, y)$.

A code is an injective function $C : \Sigma^\ell \to \Sigma^n$. Its rate is defined as $\kappa := \ell/n$ and its minimum distance is $\Delta_{\min}(C) := \min_{x \neq x'} \Delta(C(x), C(x'))$. We often identify $C$ with its image in $\Sigma^n$. The code is linear if $\Sigma$ is a finite field $\mathbb{F}$ and $C$ is a linear function over the field.

For a string $x \in \mathbb{F}^n$ we denote by $\mathrm{supp}(x) := \{i \in [n] : x_i \neq 0\}$ its support and by $\mathrm{wt}(x) := |\mathrm{supp}(x)|$ its Hamming weight. Note that the minimum distance of a linear code is equal to the minimum Hamming weight of a non-zero codeword.

For a code $C$ and an integer $t$, the interleaved code $C^t$ consists of tuples of $t$ codewords of $C$. We view $C^t$ as a code over the alphabet $\mathbb{F}^t$ where we group all of the $i$-th symbols of the $t$ codewords together as a single symbol. For interleaved pairs, we use the coordinate-wise distance

$$\Delta\big((u,v),(c_1,c_2)\big) := |\{i : (u_i, v_i) \neq ((c_1)_i, (c_2)_i)\}|.$$

Accordingly,

$$\Delta((u,v), C^2) := \min_{c_1,c_2 \in C} \Delta\big((u,v),(c_1,c_2)\big).$$

If $I \subseteq [n]$, we similarly define

$$\Delta_I(x_I, y_I) := |\{i \in I : x_i \neq y_i\}|.$$

In particular, restriction cannot increase distance: $\Delta_I(x_I, y_I) \leq \Delta(x, y)$.

## 2.1 Expanders

For a bipartite graph $G = (L, R, E)$ and a vertex $v$, we denote by $\Gamma(v)$ the set of neighbors of $v$, and for a set $S$ of vertices on one side we write $\Gamma(S) := \bigcup_{v \in S} \Gamma(v)$.

**Definition 2** ($(c,k)$-regular $(\gamma,\delta)$-expander)**.** *A bipartite graph $G = (L, R, E)$ is a $(c,k)$-regular $(\gamma,\delta)$-expander if every vertex in $L$ has degree $c$, every vertex in $R$ has degree $k$, and every set $S \subseteq L$ with $|S| \leq \delta|L|$ satisfies*

$$|\Gamma(S)| \geq (1-\gamma)c|S|.$$

*By bi-regularity, $c|L| = k|R|$.*

**Proposition 3** (Unique-neighbor property)**.** *Let $G = (L, R, E)$ be a $(c,k)$-regular $(\gamma,\delta)$-expander. For every $S \subseteq L$ with $|S| \leq \delta|L|$, we have*

$$|\Gamma_1(S)| \geq (1-2\gamma)c|S|,$$

*where*

$$\Gamma_1(S) := \{j \in \Gamma(S) : |\Gamma(j) \cap S| = 1\}$$

*is the set of unique neighbors of $S$.*

*Proof.* Let $a = |\Gamma_1(S)|$. Every vertex in $\Gamma(S) \setminus \Gamma_1(S)$ receives at least two edges from $S$, whereas a unique neighbor receives exactly one. Counting the $c|S|$ edges leaving $S$ therefore gives

$$c|S| \geq a + 2\big(|\Gamma(S)| - a\big) = 2|\Gamma(S)| - a.$$

Using $|\Gamma(S)| \geq (1-\gamma)c|S|$ and rearranging, we obtain

$$a \geq 2|\Gamma(S)| - c|S| \geq (1-2\gamma)c|S|.$$

□

## 2.2 LDPC and Expander Codes

**Definition 4** (LDPC codes over finite fields)**.** *Let $\mathbb{F}$ be a finite field and let $G = (L, R, E)$ be a bipartite graph with $|L| = n$ and $|R| = m$. Each vertex $j \in R$ represents a linear constraint. For every edge $(i, j) \in E$, fix a nonzero weight $w_{i,j} \in \mathbb{F}$. The code defined by this parity-check graph is*

$$C := \left\{ x \in \mathbb{F}^n : \sum_{i \in \Gamma(j)} w_{i,j} x_i = 0 \text{ for every } j \in R \right\}.$$

The following simple observation is the only place where the field and the nonzero edge weights enter our arguments.

**Claim 5** (Unique neighbors detect errors)**.** *Let $e \in \mathbb{F}^n$ be a nonzero word with support $S$, and let $j \in R$ be a check with $|\Gamma(j) \cap S| = 1$. Then $e$ violates the check $j$. In particular, $e \notin C$.*

*Proof.* Let $\Gamma(j) \cap S = \{i\}$. Then $\sum_{i' \in \Gamma(j)} w_{i',j} e_{i'} = w_{i,j} e_i$, which is nonzero since both $w_{i,j}$ and $e_i$ are nonzero. □

**Definition 6** (Expander codes)**.** *We call a code $C$ a $(c, k, \gamma, \delta)$-expander code if it is defined by a parity-check graph that is a $(c, k)$-regular $(\gamma, \delta)$-expander.*

This is the usual parity-check realization of expander codes [SS96], extended here to weighted checks over arbitrary finite fields.

**Proposition 7** (Distance of expander codes)**.** *Let $C \subseteq \mathbb{F}^n$ be a $(c, k, \gamma, \delta)$-expander code with $\gamma < 1/2$ over a finite field $\mathbb{F}$. Then*

$$\Delta_{\min}(C) \geq 2(1 - \gamma)\delta n,$$

*where $\Delta_{\min}(C)$ is the minimum absolute Hamming distance of $C$.*

*Proof.* Let $x \in C$ be nonzero and let $T = \mathrm{supp}(x)$. If $|T| \leq \delta n$, then by Proposition 3 (recall $\gamma < 1/2$) $T$ has a unique neighbor, whose constraint is violated by $x$ by Claim 5. Hence $|T| > \delta n$. Fix a subset $S \subseteq T$ of size $\delta n$.

Every check in $\Gamma_1(S)$ is adjacent to exactly one nonzero coordinate in $S$. Since $x$ satisfies this check, it must also be adjacent to a nonzero coordinate in $T \setminus S$. Thus every check in $\Gamma_1(S)$ is adjacent to $T \setminus S$. Different checks in $\Gamma_1(S)$ may share a neighbor in $T \setminus S$, but every vertex of $T \setminus S$ has degree $c$ and hence lies in at most $c$ of them. Together with Proposition 3 applied to $S$, this gives

$$c \cdot |T \setminus S| \geq |\Gamma_1(S)| \geq (1 - 2\gamma) c \delta n.$$

It follows that

$$|T| = |S| + |T \setminus S| \geq \delta n + (1 - 2\gamma)\delta n = 2(1 - \gamma)\delta n.$$

This proves the claimed distance bound. □

Throughout the remainder of the note, we denote by $d$ the lower bound on the (absolute) minimum distance of $C$ given by Proposition 7. Note that the true minimum distance may be larger.

$$d := 2(1 - \gamma)\delta n.$$

# 3 Proximity gaps for expander codes

Our main result is the following folding guarantee for expander codes.

**Theorem 8.** *Let $C \subseteq \mathbb{F}^n$ be a $(c, k, \gamma, \delta)$-expander code over a finite field $\mathbb{F}$, with check set $R$, $\gamma < 1/3$ and $\gamma\delta n \geq 1$. Fix $0 < \varepsilon < 1 - 3\gamma$. For every fixed interleaved word $(u, v) \in (\mathbb{F}^n)^2$ with $\Delta((u, v), C^2) > d/2$, we have*

$$\Pr_{r \in \mathbb{F}}[\Delta(u + rv, C) < t] \leq \frac{|R|}{|\mathbb{F}|},$$

*where*

$$d = 2(1 - \gamma)\delta n, \qquad t = (1 - 3\gamma - \varepsilon)\frac{d}{2}.$$

The proof distinguishes between interleaved words that jointly violate few parity checks and those that violate many. The first case is the heart of the matter and relies on the smoothness of expander codes, which we define and establish in Section 3.1. In Section 3.2 we use smoothness to prove our main technical lemma, which handles the case of few violations, and in Section 3.3 we complete the proof of Theorem 8. Applications to proximity tests are given in Section 4.

## 3.1 Smoothness

We use the following notion of smoothness, adapted from Dinur, Sudan, and Wigderson [DSW06].

**Definition 9** (Smooth code)**.** *Let $C \subseteq \mathbb{F}^n$ be a linear code over a finite field $\mathbb{F}$ with a bi-regular parity-check graph $G = (L, R, E)$, where $|L| = n$. For $R_0 \subseteq R$, define the supercode*

$$C(R_0) := \{x \in \mathbb{F}^n : x \text{ satisfies every constraint in } R \setminus R_0\}.$$

*For parameters $\alpha, P, D$, we say that $C$ is $(\alpha, P, D)$-smooth if, for every $R_0 \subseteq R$ with $|R_0| \leq \alpha|R|$, there is a set $L_0 \subseteq L$ with $|L_0| \leq P$ such that*

$$C(R_0)|_{L \setminus L_0}$$

*has minimum distance at least $D$. We call $P$ the* puncturing budget *and $D$ the* surviving distance.

We emphasize that both $D$ and $P$ above are *absolute* (rather than relative) quantities. Note also that $(\alpha, P, D)$-smoothness implies $(\alpha', P, D)$-smoothness for every $\alpha' \leq \alpha$.

**Proposition 10** (Smoothness at half the distance)**.** *Let $C \subseteq \mathbb{F}^n$ be a $(c, k, \gamma, \delta)$-expander code with $\gamma < 1/3$ and $\gamma\delta n \geq 1$ over a finite field $\mathbb{F}$, and let $d = 2(1 - \gamma)\delta n$ be the distance bound of Proposition 7. Fix a threshold $\tau$ with*

$$\gamma < \tau < 1 - 2\gamma,$$

*and let*

$$\alpha := (\tau - \gamma)k\frac{d}{2n}.$$

*Then the code $C$ is*

$$\left(\alpha, \frac{d}{2}, \frac{d}{2}\right) \text{-smooth.}$$

*In particular, for $0 < \varepsilon < 1 - 3\gamma$, taking $\tau = 1 - 2\gamma - \varepsilon$ shows that $C$ is $(\alpha, d/2, d/2)$-smooth with*

$$\alpha = (1 - 3\gamma - \varepsilon)(1 - \gamma)k\delta.$$

This value of $\alpha$ is automatically positive and smaller than 1. Indeed, expansion for a set of size $\delta n$ gives $(1-\gamma)c\delta n \le |R| = cn/k$, and hence $kd/(2n) = k(1-\gamma)\delta \le 1$, while $0 < \tau - \gamma < 1$.

The proof below directly adapts the proof of expander-code smoothness in [DSW06, Lemma 2]. We replace their fixed peeling threshold $1/3$ by the parameter $\tau$ and express the puncturing budget and residual distance in terms of the distance $d$.

*Proof of Proposition 10.* Fix $R_0 \subseteq R$ with $|R_0| \le \alpha|R|$. Starting with $L_0 = \varnothing$ and $R_1 = R_0$, repeat the following operation: if some $i \in L \setminus L_0$ has more than $\tau c$ neighbors in $R_1$, add $i$ to $L_0$ and add all of $i$'s neighbors to $R_1$. The process eventually terminates because every iteration adds a new vertex to $L_0$.

We first establish the distance guarantee after the process terminates. Let $I = L \setminus L_0$. Every vertex of $I$ has at most $\tau c$ neighbors in $R_1$, and every neighbor of $L_0$ belongs to $R_1$. For any nonempty $S \subseteq I$ with $|S| \le \delta n$, expansion in the original graph guarantees at least $(1-2\gamma)c|S|$ unique neighbors. At most $\tau c|S|$ of them can lie in $R_1$, so at least

$$(1-2\gamma)c|S| - \tau c|S| = (1-2\gamma-\tau)c|S| > 0$$

unique neighbors of $S$ survive in $R \setminus R_1$. Consequently, by Claim 5, no nonzero word supported on at most $\delta n$ coordinates can satisfy all the surviving constraints.

The surviving constraints involve no coordinates in $L_0$, and $R_0 \subseteq R_1$. Hence every word in $C(R_0)|_I$ satisfies them. It follows that $C(R_0)|_I$ has minimum distance greater than $\delta n$, and therefore at least $d/2$. This is the first restriction on the threshold:

$$\tau < 1 - 2\gamma.$$

It remains to prove that we do not puncture too many coordinates, namely that $|L_0| \le d/2$. Suppose otherwise, so that the process runs for at least $s := \lfloor d/2 \rfloor + 1$ steps, and consider the state right after step $s$: let $L_0'$ be the set of the first $s$ coordinates added and $R_1'$ the set of checks collected at that point. Since $d/2 = \delta n - \gamma\delta n \le \delta n - 1$, we have $|L_0'| = s \le \delta n$, so expansion gives the lower bound

$$|\Gamma(L_0')| \ge (1-\gamma)cs.$$

On the other hand, every neighbor of a peeled vertex is added to the collected set, so $\Gamma(L_0') \subseteq R_1'$, and the peeling process gives

$$|\Gamma(L_0')| \le |R_1'| < |R_0| + (1-\tau)cs.$$

Indeed, each vertex added to $L_0'$ had more than $\tau c$ neighbors already in the current set at that time, so each step introduces fewer than $(1-\tau)c$ new checks. Comparing the two bounds and using $\tau > \gamma$, $|R_0| \le \alpha|R|$ and $cn = k|R|$, we obtain

$$s < \frac{|R_0|}{(\tau-\gamma)c} \le \frac{\alpha n}{(\tau-\gamma)k} = \frac{d}{2},$$

contradicting $s = \lfloor d/2 \rfloor + 1 > d/2$. Hence $|L_0| \le d/2$, completing the proof. This is the second restriction on the threshold: $\tau > \gamma$. □

### 3.2 Main technical lemma

**Lemma 11** (Main technical lemma)**.** *Let $C \subseteq \mathbb{F}^n$ be a $(c, k, \gamma, \delta)$-expander code over a finite field $\mathbb{F}$, with $\gamma < 1/3$ and $\gamma\delta n \geq 1$. Fix $0 < \varepsilon < 1 - 3\gamma$ and set*

$$d = 2(1-\gamma)\delta n, \qquad \alpha = (1 - 3\gamma - \varepsilon)k\frac{d}{2n}.$$

*Let $(u, v) \in (\mathbb{F}^n)^2$ be a fixed interleaved word such that $\Delta((u, v), C^2) > \frac{d}{2}$ and such that at most an $\alpha$ fraction of the constraints are violated by at least one of $u$ and $v$. Then*

$$\Pr_{r\in\mathbb{F}}\left[\Delta(u + rv, C) < \frac{d}{2}\right] \leq \frac{1}{|\mathbb{F}|}.$$

*Proof.* Let $G = (L, R, E)$ be the parity-check graph of $C$, and let $R' \subseteq R$ be the set of constraints violated by at least one of $u$ and $v$; by assumption, $|R'| \leq \alpha|R|$. Let $C(R') = \{w \in \mathbb{F}^n : w$ satisfies all constraints in $R \setminus R'\}$ be the supercode obtained by removing the constraints in $R'$. Taking $\tau = 1 - 2\gamma - \varepsilon$ in Proposition 10 shows that $C$ is $(\alpha, d/2, d/2)$-smooth. Hence there is a set $L' \subseteq L$ of at most $\frac{d}{2}$ coordinates such that, for $I = L \setminus L'$, the punctured supercode

$$C_I := C(R')|_I$$

has minimum distance at least $\frac{d}{2}$. Since $u$ and $v$ satisfy every constraint in $R \setminus R'$, both $u_I$ and $v_I$ are codewords of $C_I$.

Observe that since $C_I$ is linear, every punctured fold $u_I + rv_I$ is also in $C_I$.

Since $C(R')$ is a supercode of $C$, we have $C \subseteq C(R')$. Projecting onto $I$ gives $C|_I \subseteq C(R')|_I = C_I$. In particular, $c_I \in C_I$ for every $c \in C$.

The minimum distance of $C_I$ therefore implies that, for every $c \in C$ and $r \in \mathbb{F}$, either $u_I + rv_I = c_I$ or $\Delta_I(u_I + rv_I, c_I) \geq \frac{d}{2}$. Consequently, if $u_I + rv_I \neq c_I$ for every $c \in C$, then

$$\Delta(u + rv, c) \geq \Delta_I(u_I + rv_I, c_I) \geq \frac{d}{2}$$

for every $c \in C$. Thus the fold has distance at least $d/2$ from $C$, as required in this case. Otherwise, there is some $c \in C$ such that $u_I + rv_I = c_I$. In particular, this equality is necessary for a fold to be good, meaning that its distance from $C$ is less than $d/2$.

Suppose that there are two distinct good folds $r_1, r_2 \in \mathbb{F}$. Then there are $c_1, c_2 \in C$ such that

$$u_I + r_1 v_I = (c_1)_I \qquad \text{and} \qquad u_I + r_2 v_I = (c_2)_I.$$

Subtracting these equations and using $r_1 \neq r_2$ gives

$$v_I = \left(\frac{c_1 - c_2}{r_1 - r_2}\right)_I, \qquad u_I = \left(\frac{r_1 c_2 - r_2 c_1}{r_1 - r_2}\right)_I.$$

By the linearity of $C$, both vectors inside the parentheses belong to $C$. Hence $(u, v)$ agrees with a pair of codewords from $C^2$ on every coordinate in $I$, and so

$$\Delta((u, v), C^2) \leq |L'| \leq \frac{d}{2},$$

contradicting the hypothesis of the theorem. Thus at most one $r \in \mathbb{F}$ gives a good fold, and the probability of a good fold is at most $1/|\mathbb{F}|$.

□

### 3.3 Proof of Theorem 8

Let

$$\alpha = (1 - 3\gamma - \varepsilon)k\frac{d}{2n}$$

as in Lemma 11. Since $|R| = cn/k$, we have

$$\frac{\alpha|R|}{c} = \frac{\alpha n}{k} = (1 - 3\gamma - \varepsilon)\frac{d}{2} = t.$$

Let $W$ be the set of checks violated by at least one of $u$ and $v$. Write $w_r = u + rv$ and distinguish two cases.

**Case 1: at most $\alpha|R|$ violated checks.** Suppose $|W| \leq \alpha|R|$. Since the original interleaved word is more than $d/2$-far, Lemma 11 applies and gives

$$\Pr_{r\in\mathbb{F}}[\Delta(w_r, C) < d/2] \leq \frac{1}{|\mathbb{F}|}.$$

Except for at most one value of $r$, the fold therefore has distance at least $d/2$ from $C$. Since $t < d/2$, it also has distance at least $t$.

**Case 2: more than $\alpha|R|$ violated checks.** Suppose $|W| > \alpha|R|$. For each parity check $j \in R$, define

$$h_j(x) := \sum_{i\in\Gamma(j)} w_{i,j}x_i.$$

For each $j \in W$, linearity gives

$$h_j(w_r) = h_j(u) + r\, h_j(v).$$

At least one of $h_j(u)$ and $h_j(v)$ is nonzero, so this is a nonzero affine polynomial of degree at most one in $r$. It vanishes for at most one value of $r$. Taking a union bound over $W$ gives

$$\Pr_{r\in\mathbb{F}}[\text{some } j \in W \text{ satisfies } h_j(w_r) = 0] \leq \frac{|W|}{|\mathbb{F}|}.$$

Outside this exceptional event, every check in $W$ remains violated by the fold. For any $a \in C$, we have $h_j(a) = 0$, so every violated check must touch a coordinate where $w_r$ differs from $a$. Each such coordinate touches at most $c$ checks, giving $|W| \leq c\,\Delta(w_r, a)$ for every $a \in C$. Therefore,

$$\Delta(w_r, C) \geq \frac{|W|}{c} > \frac{\alpha|R|}{c} = t.$$

The two cases exhaust all possibilities for $W$. Since $|W| \leq |R|$, the failure probability in either case is at most $|R|/|\mathbb{F}|$.

## 4 Proximity tests

We now use the results of Section 3 to obtain proximity tests for interleaved expander codes. Throughout this section, $C$, $\gamma$, $\delta$, $\varepsilon$, $d$ and $t$ are as in Theorem 8. Both tests reject every committed interleaved word $(u, v)$ with $\Delta((u, v), C^2) > d/2$ with high probability.

### 4.1 The Ligero test

We use the standard folding-and-spot-check procedure from Ligero [AHIV23]. The prover first commits to the interleaved word $(u, v)$ using a binding commitment that supports openings at individual coordinates. The verifier then samples $r \in \mathbb{F}$ uniformly and sends it to the prover, who responds with a claimed fold $\widetilde{w}$. The verifier checks that $\widetilde{w} \in C$, then samples $Q$ independent uniform coordinates from $[n]$. At each sampled coordinate $i$, the prover opens $(u_i, v_i)$, and the verifier checks both the opening and the equality $u_i + rv_i = \widetilde{w}_i$. The claimed fold is fixed before these positions are sampled. Whenever $\Delta(u + rv, C) \geq t$, all spot checks pass with probability at most $(1 - t/n)^Q$. By Theorem 8, for $\Delta((u, v), C^2) > d/2$,

$$\Pr[\text{accept}] \leq \frac{|R|}{|\mathbb{F}|} + \left(1 - \frac{t}{n}\right)^Q,$$

where $|R|/|\mathbb{F}|$ bounds the folding error by Theorem 8, and $(1 - t/n)^Q$ bounds the probability that all $Q$ spot checks miss the coordinates where the fold and the claimed codeword differ, when their distance is at least $t$.

### 4.2 Sampling constraints

The folding error $|R|/|\mathbb{F}|$ in Theorem 8 comes from the case of many violated checks. We now describe a variant of the test in which the verifier detects this case directly, by sampling parity checks instead of coordinates. The coordinates opened for each sampled check also serve as the folding spot checks. Let

$$\alpha = (1 - 3\gamma - \varepsilon)k\frac{d}{2n}$$

as in the proof of Theorem 8. The test with $Q$ check queries proceeds as follows.

1. The prover commits to the interleaved word $(u, v)$ using a binding commitment that supports openings at individual coordinates.
2. The verifier samples $r \in \mathbb{F}$ uniformly and sends it to the prover, who responds with a claimed fold $\widetilde{w}$. The verifier rejects if $\widetilde{w} \notin C$.
3. The verifier samples $Q$ independent uniform checks $j_1, \ldots, j_Q \in R$. For each sampled check $j$, the prover opens $(u_i, v_i)$ for every $i \in \Gamma(j)$.
4. The verifier validates these openings, and checks that both $u$ and $v$ satisfy each sampled check and that $u_i + rv_i = \widetilde{w}_i$ at every opened coordinate.

**Theorem 12** (Sampling constraints)**.** *Let $C$, $\gamma$, $\delta$, $\varepsilon$ and $d$ be as in Theorem 8, and let $\alpha = (1 - 3\gamma - \varepsilon)kd/(2n)$. For every committed interleaved word $(u, v)$ with $\Delta((u, v), C^2) > d/2$ and every prover strategy, the test above with $Q$ check queries accepts with probability at most*

$$\frac{1}{|\mathbb{F}|} + (1 - \alpha)^Q.$$

*Proof.* Let $W$ be the set of checks violated by at least one of $u$ and $v$.

If $|W| > \alpha|R|$, each sampled check lies in $W$ with probability greater than $\alpha$, and in that case the verifier rejects. Hence the probability that the verifier accepts is at most $(1 - \alpha)^Q$.

Now suppose that $|W| \leq \alpha|R|$. Then Lemma 11 gives $\Delta(u + rv, C) \geq d/2$ except with probability at most $1/|\mathbb{F}|$ over $r$. Outside this event, since $\widetilde{w} \in C$, the disagreement set $S = \{i : u_i + rv_i \neq \widetilde{w}_i\}$

has size at least $\lceil d/2 \rceil$. Since $\gamma\delta n \geq 1$, we have $d/2 \leq \delta n - 1$ and hence $\lceil d/2 \rceil \leq \delta n$. Choose $S_0 \subseteq S$ of size $\lceil d/2 \rceil$. Expansion and $cn = k|R|$ give

$$\Pr_{j \in R}[\Gamma(j) \cap S \neq \varnothing] = \frac{|\Gamma(S)|}{|R|} \geq \frac{|\Gamma(S_0)|}{|R|} \geq (1-\gamma)k\frac{d}{2n} \geq \alpha.$$

The claimed fold is fixed before the checks are sampled, and the verifier rejects whenever a sampled neighborhood meets $S$. Thus the verifier accepts with probability at most $(1-\alpha)^Q$ outside the exceptional event, and at most $1/|\mathbb{F}| + (1-\alpha)^Q$ overall. □

For sampling error at most $2^{-\lambda}$, it suffices to take

$$Q = \left\lceil \frac{\lambda}{-\log_2(1-\alpha)} \right\rceil.$$

This opens at most $kQ$ interleaved coordinates, or $2kQ$ field symbols, in addition to the claimed fold and authentication data. The folding spot checks require no additional coordinate openings.

## References


[AHIV23] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. *Designs, Codes and Cryptography*, 91(11):3379–3424, 2023. Preliminary version in CCS 2017. doi:10.1007/s10623-023-01222-8.

[ARR25] Noor Athamnah, Noga Ron-Zewi, and Ron D. Rothblum. Linear prover IOPs in log star rounds. In Benny Applebaum and Huijia Lin, editors, *Theory of Cryptography – TCC 2025, Part I*, Lecture Notes in Computer Science, pages 335–368. Springer, 2025. doi:10.1007/978-3-032-12287-2_12.

[BCF+25] Martijn Brehm, Binyi Chen, Ben Fisch, Nicolas Resch, Ron D. Rothblum, and Hadas Zeilberger. Blaze: Fast SNARKs from interleaved RAA codes. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part IV*, Lecture Notes in Computer Science, pages 123–152. Springer, 2025. doi:10.1007/978-3-031-91134-7_5.

[BCFW25] Benedikt Bünz, Alessandro Chiesa, Giacomo Fenzi, and William Wang. Linear-time accumulation schemes. In Benny Applebaum and Huijia Lin, editors, *Theory of Cryptography – TCC 2025, Part I*, Lecture Notes in Computer Science, pages 369–399. Springer, 2025. doi:10.1007/978-3-032-12287-2_13.

[BCG+17] Jonathan Bootle, Andrea Cerulli, Essam Ghadafi, Jens Groth, Mohammad Hajiabadi, and Sune K. Jakobsen. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part III*, Lecture Notes in Computer Science, pages 336–365. Springer, 2017. doi:10.1007/978-3-319-70700-6_12.

[BCG20] Jonathan Bootle, Alessandro Chiesa, and Jens Groth. Linear-time arguments with sublinear verification from tensor codes. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography – TCC 2020, Part II*, Lecture Notes in Computer Science, pages 19–46. Springer, 2020. doi:10.1007/978-3-030-64378-2_2.

[BCH+26] Eli Ben-Sasson, Dan Carmon, Ulrich Haböck, Swastik Kopparty, and Shubhangi Saraf. On proximity gaps of Reed–Solomon codes. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC 2026)*, pages 1157–1167. ACM, 2026. doi:10.1145/3798129.3800827.

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for Reed–Solomon codes. *Journal of the ACM*, 70(5):1–57, 2023. Preliminary version in FOCS 2020. doi:10.1145/3614423.

[BFK+24] Alexander R. Block, Zhiyong Fang, Jonathan Katz, Justin Thaler, Hendrik Waldner, and Yupeng Zhang. Field-agnostic SNARKs from expand-accumulate codes. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part X*, Lecture Notes in Computer Science, pages 276–307. Springer, 2024. doi:10.1007/978-3-031-68403-6_9.

[BFMM26] Anubhav Baweja, Giacomo Fenzi, Pratyush Mishra, and Tushar Mopuri. Field-agnostic SNARKs with small proofs via encode-repeat-accumulate (ERA) codes. Cryptology ePrint Archive, Paper 2026/864, 2026. URL: https://eprint.iacr.org/2026/864.

[BFRW26] Benedikt Bünz, Giacomo Fenzi, Ron D. Rothblum, and William Wang. TensorSwitch: Nearly optimal polynomial commitments from tensor codes. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology – CRYPTO 2026, Part IX*, Lecture Notes in Computer Science, pages 297–329. Springer, 2026. doi:10.1007/978-3-032-35424-2_11.

[BGKS20] Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, and Shubhangi Saraf. DEEP-FRI: Sampling outside the box improves soundness. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *LIPIcs*, pages 5:1–5:32, 2020. URL: https://doi.org/10.4230/LIPIcs.ITCS.2020.5.

[BMMS26a] Anubhav Baweja, Pratyush Mishra, Tushar Mopuri, and Matan Shtepel. FICS and FACS: Fast IOPPs and accumulation via code-switching. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology – CRYPTO 2026, Part IX*, Lecture Notes in Computer Science, pages 557–591. Springer, 2026. doi:10.1007/978-3-032-35424-2_19.

[BMMS26b] Anubhav Baweja, Pratyush Mishra, Tushar Mopuri, and Matan Shtepel. Query-optimal IOPPs for linear-time encodable codes. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026, Part VII*, Lecture Notes in Computer Science, pages 95–120. Springer, 2026. doi:10.1007/978-3-032-25336-1_4.

[DSW06] Irit Dinur, Madhu Sudan, and Avi Wigderson. Robust local testability of tensor products of LDPC codes. In *Proceedings of RANDOM 2006*, pages 304–315, 2006. URL: https://www.math.ias.edu/~avi/PUBLICATIONS/MYPAPERS/DSW06/dsw06.pdf.

[Gal62] Robert G. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962. doi:10.1109/TIT.1962.1057683.

[GG26] Rohan Goyal and Venkatesan Guruswami. Optimal proximity gaps for subspace-design codes and (random) Reed–Solomon codes. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC 2026)*, pages 1558–1568. ACM, 2026. doi:10.1145/3798129.3800864.

[GGSW26a] Rohan Goyal, Venkatesan Guruswami, Yihang Sun, and Mary Wootters. locality of curve-decoding and improved proximity gaps. arXiv:2607.08516, 2026. URL: https://arxiv.org/abs/2607.08516.

[GGSW26b] Rohan Goyal, Venkatesan Guruswami, Yihang Sun, and Mary Wootters. improved LCL equivalences and curve decoding alphabet sizes. arXiv:2610.38862, 2026. URL: https://arxiv.org/abs/2610.38862.

[GLS+23] Alexander Golovnev, Jonathan Lee, Srinath Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and field-agnostic SNARKs for R1CS. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part II*, Lecture Notes in Computer Science, pages 193–226. Springer, 2023. doi:10.1007/978-3-031-38545-2_7.

[GNR26] Kobi Gurkan, Andrija Novakovic, and Ron D. Rothblum. Bolt: Faster SNARKs from sketched codes. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology – CRYPTO 2026, Part IX*, Lecture Notes in Computer Science, pages 396–427. Springer, 2026. doi:10.1007/978-3-032-35424-2_14.

[Jer26] Fernando Granha Jeronimo. Algorithmic list decoding at capacity and optimal proximity gaps for Reed–Solomon codes. arXiv:2609.05870, 2026. URL: https://arxiv.org/abs/2609.05870.

[KPR+26] Vladimir Kolesnikov, Stanislav Peceny, Rahul Rachuri, Srinivasan Raghuraman, Peter Rindal, and Harshal Shah. Block-accumulate codes: Accelerated linear codes for PCGs and ZK. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology – CRYPTO 2026, Part VIII*, Lecture Notes in Computer Science, pages 397–429. Springer, 2026. doi:10.1007/978-3-032-35418-1_13.

[NST24] Vineet Nair, Ashish Sharma, and Bhargav Thankey. BrakingBase – a linear prover, poly-logarithmic verifier, field agnostic polynomial commitment scheme. Cryptology ePrint Archive, Paper 2024/1825, 2024. URL: https://eprint.iacr.org/2024/1825.

[QGZ26] Wenjie Qu, Yanpei Guo, and Jiaheng Zhang. Lightning, field-agnostic super-efficient polynomial commitment scheme. Cryptology ePrint Archive, Paper 2026/258, 2026. URL: https://eprint.iacr.org/2026/258.

[RR25] Noga Ron-Zewi and Ron D. Rothblum. Proving as fast as computing: Succinct arguments with constant prover overhead. *Journal of the ACM*, 72(2):1–54, 2025. Preliminary version in STOC 2022. doi:10.1145/3721477.

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996. URL: https://www.cs.yale.edu/homes/spielman/PAPERS/expandersIT.pdf.

[XZS22] Tiancheng Xie, Yupeng Zhang, and Dawn Song. Orion: Zero knowledge proof with linear prover time. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part IV*, Lecture Notes in Computer Science, pages 299–328. Springer, 2022. doi:10.1007/978-3-031-15985-5_11.

[Zei26] Hadas Zeilberger. Khatam: Proximity gaps for multilinear evaluation for all linear codes. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology – CRYPTO*

*2026, Part IX*, Lecture Notes in Computer Science, pages 592–611. Springer, 2026. doi:10.1007/978-3-032-35424-2_20.

# A Field sizes required by [GGSW26b] for Bolt's parameters

The proof of the result of Goyal, Guruswami, Sun and Wootters [GGSW26b, Theorem 6.4] for LDPC codes on what they call *curve decodability*, which implies proximity gaps, requires field sizes on the order of thousands of bits. This is a result of a sufficient upper bound on the code rate

$$\kappa \leq 1 - \theta - 2\eta,$$

for proximity radius $\theta$, rate $\kappa$ and a slack parameter $\eta$, under which it is likely curve decodability will be satisfied, together with other requirements from the code. For the parameters of interest and taking the largest $\eta$ possible, we have the field size

$$|\mathbb{F}| \geq \left(\frac{4}{\eta}\right)^{28/\eta},$$

as a crude lower bound (the field size required in their analysis has more than twice as many bits). For instance, for the rate $\kappa = \frac{7}{8}$ used in Bolt [GNR26] and the relative distance $\rho = 0.013$ of bolt-max, we have $|\mathbb{F}| \geq 2^{2868}$ for a proximity radius $\theta = 0.49\rho$, just below half the distance, and $|\mathbb{F}| \geq 2^{2808}$ for $\theta = \frac{\rho}{3}$, a radius already proven for general linear codes.