Electronic Colloquium on Computational Complexity, Report No. 159 (2019)

SETH-hardness of Coding Problems

Noah Stephens-Davidowitz* noahsd@gmail.com

Vinod Vaikuntanathan* vinodv@mit.edu

Abstract

We show that assuming the strong exponential-time hypothesis (SETH), there are *no non-trivial algorithms* for the nearest codeword problem (NCP), the minimum distance problem (MDP), or the nearest codeword problem with preprocessing (NCPP) on linear codes over any finite field. More precisely, we show that there are no NCP, MDP, or NCPP algorithms running in time $q^{(1-\varepsilon)n}$ for any constant $\varepsilon > 0$ for codes with q^n codewords. (In the case of NCPP, we assume non-uniform SETH.)

We also show that there are no sub-exponential-time algorithms for γ-approximate versions of these problems for some constant $\gamma > 1$, under different versions of the exponential-time hypothesis.

*Massachusetts Institute of Technology. The authors were supported by an NSF-BSF grant number 1718161 and NSF CAREER Award number 1350619.

ISSN 1433-8092

Contents

1 Introduction

A linear code is a subspace of $\mathbb{F}_q^m$ for some prime power q. For a given full-rank generator matrix $C \in \mathbb{F}_q^{m \times n}$ with $1 \leq n \leq m$, the q-ary code generated by C is

$$C\mathbb{F}_q^n = \{C\boldsymbol{z} \ : \ \boldsymbol{z} \in \mathbb{F}_q^n\} \ .$$

We call n the *rank* of the code, m the *ambient dimension*, q the *alphabet size*, and an element $C\boldsymbol{z}$ a *codeword*.

The nearest codeword problem for linear codes, denoted NCP, asks us to compute the minimal (Hamming) distance between a given target vector and a codeword in a given linear code. The closely related minimum distance problem for linear codes, denoted MDP, asks us to compute the minimal Hamming weight of a non-zero codeword in a given linear code. NCP was proved to be NP-complete by Berlekamp, McEliece and van Tilborg [BMvT78] in 1978, and the analogous result for MDP came much later with the work of Vardy [Var97] in 1997.

While the last four decades have seen progress on approximation algorithms [BK02, APY09] and heuristic algorithms [MMT11, BLP11, BJMM12] for NCP and MDP, the best known way to solve either of these two problems exactly on *worst-case* codes is essentially exhaustive search over all codewords, taking q^n time. This seems to be the state of affairs also for NCP with preprocessing (NCPP), an offline-online variant of NCP where an offline unbounded-time algorithm may first preprocess the code in a way that helps an online algorithm to find the codeword nearest to a given target vector. This is the starting point of our work: Are there better (exponential-time) algorithms for these problems? In the absence of algorithms, can we demonstrate evidence for their hardness?

In this work, we show that there are *no non-trivial algorithms* for NCP, MDP or NCPP unless a popular conjecture in fine-grained complexity called the strong exponential-time hypothesis [IP99, IPZ01] (SETH) is false. SETH postulates that for every constant $\varepsilon > 0$, there is a sufficiently large k such that k-SAT on n variables does not have algorithms that run faster than $2^{(1-\varepsilon)n}$.

Theorem 1.1 (SETH-hardness of NCP(P) and MDP)**.** *For every $\varepsilon > 0$ and any prime power $q := q(n)$, there is no $q^{(1-\varepsilon)n}$-time algorithm for NCP or MDP over codes with rank n and alphabet size q unless SETH is false. The same conclusion holds for NCPP unless non-uniform SETH is false.*

At a high level, the inspiration for our proof techniques comes from the study of fine-grained hardness of lattice problems [BGS17, AS18, ABGS19], such as the closest vector problem (CVP) which is a lattice analogue of NCP. Our results therefore follow a long line of works (such as [DMS03]) in which ideas from the study of lattices proved useful for codes, and vice versa.

While our results might look similar to the analogous lattice results in [BGS17, AS18, ABGS19] and we use similar techniques, we interpret our results quite differently for two reasons.

First, there are highly non-trivial algorithms for CVP [MV10, ADS15], while Theorem 4.2 shows that there are *no non-trivial algorithms* for NCP unless SETH is false. We therefore interpret our results as a strong *separation* between lattices and codes. Indeed, though [BGS17] proved strong lower bounds for CVP, they still do not quite match the known upper bounds. For example, in the ℓ_2 norm, there is a 2^n-time CVP algorithm, but [BGS17, ABGS19] only show that it is impossible to do better than 2^{cn} under SETH for a small constant $c > 0$; on the other hand, in ℓ_p norms for $p \notin 2\mathbb{Z}$ [BGS17, ABGS19] show hardness for running times better than 2^n, while the best algorithm

runs in time $n^{O(n)}$. Demonstrating tight hardness results for lattice problems, similar to the ones we show in this work for codes, is an important open problem. (The situation is essentially the same for the lattice analogue of MDP, the shortest lattice vector problem.)

Second, our results use elementary techniques (as we will describe shortly), while the analogous results for lattices are significantly more difficult. It is this simplicity that allows us to extend our hardness results to NCP with preprocessing. After a preliminary version of this work appeared, [ABGS19] adapted our techniques to extend the fine-grained hardness results of CVP to the preprocessing version.

We also consider fine-grained *hardness of approximation* of NCP and MDP. NP-hardness of approximation for NCP is already known to within an almost polynomial $n^{1/\log\log n}$ approximation factor [ABSS97, DKRS03]. On the other hand, polynomial hardness of approximation for MDP to within any constant factor assuming that $NP \not\subseteq P$ and to within an almost polynomial $2^{\log^{(1-\varepsilon)} n}$ approximation factor assuming that $NP \not\subseteq QP$ (quasi-polynomial time) was shown in a sequence of works [DMS03, CW10, CW12, AK14]. Some non-trivial approximation algorithms are known for NCP and MDP. In particular, Berman and Karpinski [BK02] show a polynomial-time $O(m/\log m)$-approximation algorithm (where m is the ambient dimension of the code).

Our hardness of approximation result assumes the recently introduced gap-exponential-time hypothesis (Gap-ETH) [Din16, MR17] which, roughly speaking, postulates that there is a constant $c > 0$ such that c-approximation of Max-3-SAT on n variables cannot be computed in $2^{o(n)}$ time. Applebaum [App17] has recently shown that Gap-ETH follows from a number of other assumptions, like exponential hardness for "smooth" 3-CNFs (CNFs that have nearly as many almost-satisfying assignments as satisfying assignments) or the existence of locally computable and exponentially hard one-way functions. We show:

Theorem 1.2 (Gap-ETH-hardness of NCP and MDP)**.** *For any constant prime p, there is no $2^{o(n)}$-time algorithm for γ_p-NCP over codes with rank n and alphabet size $q = p^\kappa$ for any integer $\kappa := \kappa(n) \geq 1$ unless Gap-ETH is false, where $\gamma_p > 1$ is a constant depending only on p. The same conclusion holds for γ_2-MDP for $p = 2$ unless non-uniform Gap-ETH is false.*

1.1 Overview of Our Techniques

Our techniques are quite simple (perhaps surprisingly so). We first explain the binary case $q = 2$ and then describe how to extend our results to larger q.

The (γ-)NCP Gadget. Our reduction uses a certain gadget that is the code analogue of the gadget from [BGS17]. Recall that a k-SAT instance is a list of k-clauses over n variables, each of which specifies a forbidden assignment to k variables. E.g., the clause $x_1 \vee x_2 \vee \neg x_3$ specifies that $(x_1, x_2, x_3) \neq (0, 0, 1)$. To represent such a clause in our NCP instance, we use as a gadget a generator matrix $C \in \mathbb{F}_2^{\ell \times n}$ and target $\boldsymbol{t} \in \mathbb{F}_2^\ell$ such that the codewords $\{C\boldsymbol{x} \ : \ \boldsymbol{x} \in \{0,1\}^n, \ (x_1, x_2, x_3) \neq (0, 0, 1)\}$ make up precisely *all* the closest codewords to $\boldsymbol{t}$. I.e., the closest codewords to $\boldsymbol{t}$ correspond to the assignments that satisfy our k-clause.

Given such a gadget, the reduction is straightforward. We simply construct such a C_i and $\boldsymbol{t}_i$ for each of the m clauses ϕ_i in the input k-SAT instance and create the NCP instance with generator

matrix

$$C := \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{pmatrix} \in \mathbb{F}_2^{\ell m \times n}$$

and target $\boldsymbol{t} := (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_m) \in \mathbb{F}_2^{\ell m}$. Notice that an assignment $\boldsymbol{x} \in \{0,1\}^n$ satisfies the input SAT instance if and only if $\boldsymbol{x}$ simultaneously minimizes the distance $\|C_i\boldsymbol{x} - \boldsymbol{t}_i\|_H$ for all $1 \le i \le m$. I.e., the reduction is correct.

To construct these gadgets, we first notice that it suffices to find such a gadget $C \in \mathbb{F}_2^{\ell \times k}$ for k-clauses on k variables, since we can "lift" this to $C' \in \mathbb{F}_2^{\ell \times n}$ by padding with zeros. We then use Hadamard codes. I.e., we take $C \in \mathbb{F}_2^{2^k \times k}$ to be the matrix whose rows consist of all possible vectors in $\mathbb{F}_2^k$. Notice that all non-zero codewords in this code have Hamming weight exactly 2^{k-1}. Therefore, all non-zero codewords are at distance exactly 2^{k-1} from the all-ones target vector $\boldsymbol{u} := (1, \ldots, 1) \in \mathbb{F}_2^{2^k}$, while the codeword $\mathbf{0}$ is of course at distance 2^k from $\boldsymbol{u}$. By translating $\boldsymbol{u}$ by any codeword $\boldsymbol{c}$, we can find a target whose nearest codewords consist of any set of $2^k - 1$ codewords.

In fact, a slightly more careful analysis shows that this reduction reduces approximate Max-k-SAT to approximate NCP. So, this same reduction yields the Gap-ETH hardness of γ-NCP presented in Theorem 1.2.

A note for the reader familiar with [BGS17, ABGS19]: This construction has a rather mysterious relationship with the lattice gadgets constructed in [BGS17, ABGS19]. Their constructions are far more difficult, but they also start with the basis for a Hadamard code. Specifically, their rows are scalar multiples (with scalars in $\mathbb{R}$) of the rows of our matrix (embedded in $\mathbb{R}$). Perhaps exploring this relationship further could help to resolve some of the open questions left in [BGS17, ABGS19].

Extension to NCPP. In order to show SETH-hardness of NCP with preprocessing, we wish to modify the above reduction so that the the code depends only on n and k, and not on the clauses in the input k-SAT instance. (Of course, the target will still depend on the clauses.) To do this, we construct the code C corresponding to all $m^* := 2^k\binom{n}{k}$ possible k-clauses. For each clause, we also find a target $\boldsymbol{t}_{\mathsf{off}} \in \mathbb{F}_2^{2^k}$ that is equidistant from *all the* codewords, as opposed to just $2^k - 1$ of them. (To make this work, we actually modify the gadget code slightly. See Section 4.1.) Then, given a k-SAT instance, our target $\boldsymbol{t} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{m^*})$ encodes which clauses are actually included in the instance. In particular, if clause i is present, then $\boldsymbol{t}_i$ is chosen as we described above, but if it is not present, then we "turn off the clause" by taking $\boldsymbol{t}_i = \boldsymbol{t}_{\mathsf{off}}$. Notice that this preserves the property that $\boldsymbol{x} \in \mathbb{F}_2^n$ satisfies the input SAT instance if and only if $\boldsymbol{x}$ simultaneously minimizes $\|C_i\boldsymbol{x} - \boldsymbol{t}_i\|_H$, so that the reduction is in fact correct.

Extension to Larger q. There is nothing particularly special about $\mathbb{F}_2$ in the above reductions. Indeed, by using the Hadamard code over $\mathbb{F}_q$, the same reduction maps a certain q-ary variant of k-SAT with n variables to NCP over $\mathbb{F}_q$ with rank n. The appropriate variant of k-SAT consists of "clauses" over q-ary variables that specify a single forbidden assignment to k of the variables. E.g., $(x_1, x_2, x_6) \neq (0, 3, 74)$ is a 3-clause.

We show that there is no non-trivial algorithm for this q-ary variant of k-SAT unless SETH is false. Specifically, for every $\varepsilon > 0$, there exists a constant $k \ge 2$ such that no $q^{(1-\varepsilon)n}$-time

algorithm solves this variant of k-SAT for any $q = q(n) \geq 2$. (For more details, see Theorem 3.3.) In a different context, Traxler proved a slightly weaker result, which in our terminology corresponds to a lower bound of $2^{(1-\varepsilon)n\lfloor \log_2 q \rfloor}$ [Tra08]. This is weaker, e.g, for constant q that is not a power of two.

Combining this with the above reductions gives the $q^{(1-\varepsilon)n}$-hardness of NCP and NCPP presented in Theorem 4.2. (Again, this reduction also maps q-ary approximate Max-k-SAT to approximate NCP, which yields the hardness of γ-NCP in Theorem 1.2. However, we are unable to show hardness of γ-NCP for constant $\gamma > 1$ and superconstant characteristic since our variant of Max-k-SAT is easy to approximate for superconstant q.)

Extension to MDP. We present two different reductions that extend our result to MDP. Our first is an efficient randomized reduction from an NCP instance with rank n to an MDP instance with rank $(1+\varepsilon)n$ for any constant $\varepsilon > 0$. This reduction is essentially that of Dumer, Micciancio, and Sudan [DMS03] with a different choice of parameters. More specifically, the [DMS03] reduction uses a certain gadget (called a locally dense code). We follow [DMS03] in constructing such a gadget via Reed-Solomon codes, but we choose parameters that minimize the rank of the output instance, while [DMS03] chose parameters that yield relatively large rank $\text{poly}(n)$ but allow them to prove hardness of approximation.

Unfortunately, the [DMS03] reduction is randomized, so that the above only implies hardness under a randomized variant of SETH. (There *is* a deterministic variant of the reduction due to Cheng and Wan [CW12], which we use in Appendix A to show a lower bound of $q^{(1-\varepsilon)n/2}$ for MDP. See also [Mic12, Mic14, AK14].) However, to prove our full lower bound of $q^{(1-\varepsilon)n}$ under deterministic SETH, we exploit a major difference between our setting and that of [DMS03, CW12]. In particular, our reduction does not need to be efficient. So, we show how to use the method of conditional expectations to derandomize the [DMS03] reduction in, e.g., $q^{(1-2\varepsilon)n}$ time, which suffices for our purposes.

Hardness of γ-MDP. Finally, to prove Gap-ETH-hardness of γ-approximate MDP for some constant $\gamma > 1$, we show a reduction from γ'-NCP with rank n to γ-MDP with rank Cn for some constant $C > 0$. We are still able to use the high-level reduction from [DMS03] for this task, but we require an entirely different gadget construction in order to simultaneously achieve constant-factor approximation and linear rank simultaneously.

We construct our gadget using the remarkable codes discovered by Ashikhmin, Barg, and Vlăduţ [ABV01], which have 2^{cm} non-zero codewords with minimal Hamming weight. [AS18] showed how to use lattices with the analogous property to build a similar gadget in the lattice world, and we use the same techniques. Unfortunately, such codes are only known in characteristic two, so this result only applies in this case.

1.2 Open Problems

We outline the two major open directions that arise from our work. The first is to show SETH-hardness of approximation for coding problems. While we show hardness of approximating NCP and MDP in $2^{o(n)}$-time under Gap-ETH, proving 2^{cn}-time hardness of approximation for some reasonable explicit constant $c > 0$ (perhaps even $c = 1 - \varepsilon$) is wide open, and will likely require brand new techniques.

The second is to show 2^{cm}-time hardness of coding problems for some reasonable explicit constant $c > 0$. I.e., we would like to show hardness as a function of the ambient dimension m and not the rank n, as we do in this work. Using the sparsification lemma [IPZ01] and the fact that our code constructions in some cases have constant rate (specifically, when q is constant and $m = O(n)$), our reductions do yield $2^{c_\varepsilon m}$-hardness for some small unspecified constant $c_\varepsilon > 0$, which depends on the parameter ε and $k = k(\varepsilon)$ in the SETH assumption. (Our reductions for NCPP and MDP require larger m, respectively $m \approx n^k$ and $m \approx n^{C_\varepsilon}$.) Since there are non-trivial 2^{cm}-time heuristic and average-case algorithms for NCP, a sufficiently strong result in this direction would, in our opinion, be rather intriguing in that it would separate the rigorous worst-case regime from the heuristic and average-case settings.

2 Preliminaries

For a prime power $q \geq 2$, we write $\mathbb{F}_q$ for the unique field with q elements, $\mathbb{F}_q^*$ for the non-zero elements of $\mathbb{F}_q$, and $\mathbb{F}_q[x]$ for the set of polynomials in x over $\mathbb{F}_q$. The *characteristic* of $\mathbb{F}_q$ is the unique prime p such that $q = p^\kappa$ for some integer κ. For integer $\kappa \geq 1$, we recall that $\mathbb{F}_{q^\kappa}$ is a vector space of dimension κ over $\mathbb{F}_q$, and that there is a unique field embedding of $\mathbb{F}_q$ into $\mathbb{F}_{q^\kappa}$.

Our vectors $\boldsymbol{x} \in \mathbb{F}_q^m$ are always column vectors, though for convenience we sometimes abuse notation and write, e.g., $\boldsymbol{x} = (1, 1, 1, \ldots, 1)$ or even $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2)$ when formally we should write $\boldsymbol{x} = (1, 1, 1, \ldots, 1)^T$ or $\boldsymbol{x} = (\boldsymbol{x}_1^T, \boldsymbol{x}_2^T)^T$. For $\boldsymbol{x} \in \mathbb{F}_q^m$, we write $\|\boldsymbol{x}\|_H$ for the Hamming weight of $\boldsymbol{x}$ (i.e., the number of non-zero coordinates).

2.1 Codes and coding problems

For a generator matrix $C \in \mathbb{F}_q^{m \times n}$ and a target vector $\boldsymbol{t} \in \mathbb{F}_q^m$, we write

$$\operatorname{dist}(\boldsymbol{t}, C) := \min_{\boldsymbol{z} \in \mathbb{F}_q^n} \|C\boldsymbol{z} - \boldsymbol{t}\|_H$$

for the distance between $\boldsymbol{t}$ and the code generated by C and

$$\lambda(C) := \min_{\boldsymbol{z} \in \mathbb{F}_q^n \setminus \{\boldsymbol{0}\}} \|C\boldsymbol{z}\|_H$$

for the length of the shortest non-zero vector in the code generated by C. The *kissing number* of C is the number of codewords with Hamming weight $\lambda(C)$, i.e.,

$$\left|\{\boldsymbol{z} \in \mathbb{F}_q^n \ : \ \|C\boldsymbol{z}\|_H = \lambda(C)\}\right| .$$

Definition 2.1. *For an approximation factor $\gamma \geq 1$, the γ-Nearest Codeword Problem (γ-NCP) is defined as follows. The input is a generator matrix $C \in \mathbb{F}_q^{m \times n}$, target $\boldsymbol{t} \in \mathbb{F}_q^m$, and integer distance $0 \leq d \leq m$. The input is a YES instance if $\operatorname{dist}(\boldsymbol{t}, C) \leq d$ and a NO instance if $\operatorname{dist}(\boldsymbol{t}, C) > \gamma d$.*

Definition 2.2. *For an approximation factor $\gamma \geq 1$, the γ-Minimum Distance Problem (γ-MDP) is defined as follows. The input is a generator matrix $C \in \mathbb{F}_q^{m \times n}$ and integer distance $0 \leq d \leq m$. The input is a YES instance if $\lambda(C) \leq d$ and a NO instance if $\lambda(C) > \gamma d$.*

In both cases, we omit the parameter γ when $\gamma = 1$. For convenience, we also define γ-MDP for $\gamma < 1$ to be an unsolvable problem. In particular, a reduction to γ-MDP for $\gamma < 1$ is vacuous.

NCP with preprocessing (NCPP) is the variant of NCP in which we are allowed arbitrary preprocessing on the code C (but not the target!). I.e., formally an algorithm for NCPP consists of a pair of procedures P and Q. The algorithm solves an NCPP $(C, \boldsymbol{t}, d)$ if on input C, $\boldsymbol{t}$, d and the preprocessing $P(C)$, the procedure Q returns a valid answer to the corresponding NCP problem. The running time of such an algorithm is simply the running time of Q. (Our lower bound holds even if P is an arbitrary function that, e.g., might not even be computable. We always assume that the running time of Q must be at least the size of the preprocessing $P(C)$.)

Claim 2.3. *For any generator matrix $C \in \mathbb{F}_q^{m \times n}$, target $\boldsymbol{t} \in \mathbb{F}_q^m$, and integer $\kappa \geq 1$, let $C' \in \mathbb{F}_{q^\kappa}^{m \times n}$ and $\boldsymbol{t}' \in \mathbb{F}_{q^\kappa}^m$ be obtained by applying the (unique) embedding from $\mathbb{F}_q$ into $\mathbb{F}_{q^\kappa}$ coordinate-wise to C and $\boldsymbol{t}$ respectively. Then $\lambda(C') = \lambda(C)$ and $\operatorname{dist}(\boldsymbol{t}', C') = \operatorname{dist}(\boldsymbol{t}, C)$.*

In particular, this embedding yields a trivial reduction from (possibly approximate) NCP over $\mathbb{F}_q$ to NCP over $\mathbb{F}_{q^\kappa}$ that preserves the rank n, ambient dimension m, and approximation factor γ; and likewise for MDP.

Proof. This follows from the fact that a system of linear equations over $\mathbb{F}_q$ has a solution (or, alternatively, a non-zero solution) if and only if the same system embedded in $\mathbb{F}_{q^\kappa}$ has a solution (or a non-zero solution).

More formally, let $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_m \in \mathbb{F}_q^n$ be the rows of C and we view them as embedded in $\mathbb{F}_{q^\kappa}^n$. Consider a system of linear equations over $\boldsymbol{z} = (z_1, \ldots, z_n) \in \mathbb{F}_q$ given by $\langle \boldsymbol{c}_i, \boldsymbol{z} \rangle = a_i$ for all $i \in S$ for some set $S \subset [m]$ and some $a_i \in \mathbb{F}_q$. Using the natural embedding of $\mathbb{F}_q$ into $\mathbb{F}_{q^\kappa}$ again, we can view this as a system of linear equations over $\mathbb{F}_{q^\kappa}$ as well. The key observation is that this system has a (non-zero) solution over $\mathbb{F}_q$ if and only if it has a (non-zero) solution over $\mathbb{F}_{q^\kappa}$. (One can see this, e.g., by noting that the system may be solved via Gaussian elimination, which works by repeatedly performing field operations on the coordinates of $\boldsymbol{c}_i$ and a_i and therefore never needs elements outside of the base field $\mathbb{F}_q$.)

Finally, we note that $m - \lambda(C)$ is equal to the maximum size of a subset S of the rows such that the corresponding system has a non-zero solution over $\mathbb{F}_q$ for $a_i = 0$. And, $m - \operatorname{dist}(\boldsymbol{t}, C)$ is the maximum size of such a set S for which any solution exists for $a_i = t_i$. Similarly, $m - \lambda(C')$ and $m - \operatorname{dist}(\boldsymbol{t}, C')$ are the maximal sizes for the corresponding sets, now allowing solutions over $\mathbb{F}_{q^\kappa}$. The result follows. □

Claim 2.4. *For any subset $V \subseteq \mathbb{F}_q^{n^\dagger}$ and $N \leq |V|/q^n$,*

$$\Pr_{\boldsymbol{z} \sim \mathbb{F}_q^n, T \sim \mathbb{F}_q^{n^\dagger \times n}} [|\{\boldsymbol{v} \in V \ : \ T\boldsymbol{v} = \boldsymbol{z}\}| \leq N] \leq \frac{q^n |V|}{(|V| - q^n N)^2} \, .$$

Proof. Notice that the expectation satisfies

$$\mathbb{E}[|\{\boldsymbol{v} \in V \ : \ T\boldsymbol{v} = \boldsymbol{z}\}|] = |V|/q^n \, ,$$

since $T\boldsymbol{v} - \boldsymbol{z}$ is a uniformly random element in $\mathbb{F}_q^n$. Furthermore, for distinct $\boldsymbol{v}_1, \boldsymbol{v}_2 \in V$, the events $T\boldsymbol{v}_i = \boldsymbol{z}$ are pairwise independent because $T(\boldsymbol{v}_1 - \boldsymbol{v}_2) - \boldsymbol{z}$ is also a uniformly random element in $\mathbb{F}_q^n$, independently of $T\boldsymbol{v}_1 - \boldsymbol{z}$. Therefore, the variance satisfies $\operatorname{Var}[|\{\boldsymbol{v} \in V \ : \ T\boldsymbol{v} = \boldsymbol{z}\}|] = |V| q^{-n}(1 - q^{-n}) \leq |V|/q^n$. The result follows from Chebyshev's inequality. □

2.2 Reed-Solomon codes

For $1 \le n \le m \le q$, a *Reed-Solomon code* is a code generated by a Vandermonde matrix of the form

$$C = \begin{pmatrix} 1 & z_1 & z_1^2 & \cdots & z_1^{n-1} \\ 1 & z_2 & z_2^2 & \cdots & z_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & z_m & z_m^2 & \cdots & z_m^{n-1} \end{pmatrix} \in \mathbb{F}_q^{m \times n},$$

where $z_1, \ldots, z_m \in \mathbb{F}_q$ are distinct field elements. Equivalently, we can think of the coordinates $\boldsymbol{a} = (a_0, \ldots, a_{n-1}) \in \mathbb{F}_q^n$ of a codeword $C\boldsymbol{a}$ as representing a polynomial $p_{\boldsymbol{a}}(x) := a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} \in \mathbb{F}_q[x]$ of degree less than n over $\mathbb{F}_q$. Then, the codeword $C\boldsymbol{a}$ is the vector of evaluations of the polynomial, $C\boldsymbol{a} = (p_{\boldsymbol{a}}(z_1), p_{\boldsymbol{a}}(z_2), \ldots, p_{\boldsymbol{a}}(z_m)) \in \mathbb{F}_q^m$.

In particular, we see that $\lambda(C) = m - (n-1)$, since (1) the polynomial $p(x) = (x - z_1)(x - z_2) \cdots (x - z_{n-1})$ corresponds to a codeword with Hamming weight exactly $m - (n-1)$; and (2) no non-zero codeword can have Hamming weight less than $m - (n-1)$, since this would correspond to a non-zero polynomial with degree at most $m - 1$ but more than $m - 1$ zeros. Indeed, the shortest non-zero codewords are given by the polynomials $p_{S,\alpha}(x) = \alpha \cdot \prod_{z \in S}(x - z)$ for any $\alpha \in \mathbb{F}_q^*$ and $S \subset \{z_1, \ldots, z_m\}$ with $|S| = n - 1$.

2.3 SAT, SETH, and Gap-ETH

Given Boolean variables $x_1, \ldots, x_n$, a *literal* y is a variable $y = x_i$ or its negation $y = \neg x_i$. An assignment to these variables maps each variable to true or false. For an integer $k \ge 2$, a k-clause is a list of k literals on *distinct* variables, which we write as a disjunction $y_1 \vee y_2 \vee \cdots \vee y_k$. E.g., $x_1 \vee \neg x_5 \vee x_8$ is a 3-clause. A clause is *satisfied* by an assignment if at least one of its literals is mapped to true by the assignment. A k-SAT formula on n variables is a list of distinct k-clauses over these variables.

Definition 2.5. *For an integer $k \ge 3$, the k-SAT problem is defined as follows. The input is a k-SAT formula Φ. It is a YES instance if there exists an assignment to the variables of Φ that satisfies all the clauses of Φ simultaneously. It is a NO instance if no such assignment exists.*

Definition 2.6. *For an integer $k \ge 2$, the Max-k-SAT problem is defined as follows. The input is a k-SAT formula Φ and an integer r. It is a YES instance if there exists an assignment to the variables of Φ that satisfies at least r of the clauses simultaneously. It is a NO instance if no such assignment exists.*

Definition 2.7. *For an integer $k \ge 2$ and $0 < s \le c \le 1$, the (s, c)-Gap-k-SAT problem is the promise problem defined as follows. The input is a k-SAT formula Φ with m clauses. It is a YES instance if there exists an assignment to the variables of Φ that satisfies at least cm clauses simultaneously. It is a NO instance if no assignment satisfies at least sm clauses simultaneously.*

A simple probabilistic argument shows that (s, c)-Gap-k-SAT is trivial for $s \le 1 - 2^{-k}$. (In particular, if we choose an assignment at random, then the expected number of satisfied clauses is $(1 - 2^{-k})m$.)

Definition 2.8 ([IP99, IPZ01])**.** *The Strong Exponential-Time Hypothesis (SETH) is the conjecture that for every $\varepsilon > 0$, there exists an integer $k \ge 3$ such that no $2^{(1-\varepsilon)n}$-time algorithm solves k-SAT.*

Randomized SETH is the analogous conjecture for randomized algorithms, and non-uniform SETH is the same conjecture for non-uniform algorithms (i.e., circuits).

Definition 2.9 ([Din16, MR17])**.** *The Gap-Exponential-Time Hypothesis (Gap-ETH) is the conjecture that there exist a constant $s \in (0,1)$ such that no $2^{o(n)}$-time algorithm solves $(s,1)$-Gap-3-SAT. (I.e., no such algorithm can distinguish satisfiable k-SAT instances from instances in which the maximum fraction of simultaneously satisfied clauses is less than s.) Randomized Gap-ETH is the analogous conjecture for randomized algorithms, and non-uniform Gap-ETH is the analogous conjecture for non-uniform algorithms (i.e., circuits).*

3 q-ary SAT, SETH, and Gap-ETH

In this section, we show that natural q-ary variants of SETH and Gap-ETH hold under the standard SETH and Gap-ETH assumptions, respectively. In particular, for q-ary variables $x_1, \dots, x_n \in \mathbb{Z}_q$, a (q,k)-clause is simply a *forbidden assignment* to k distinct variables, which we write as $(x_{i_1}, x_{i_2}, \dots, x_{i_k}) \neq (z_1, \dots, z_k)$ for $z_i \in \mathbb{Z}_q$. Such a clause is satisfied if there exists at least one j such that $x_{i_j} \neq z_j$. A (q,k)-SAT formula on n variables is a list of distinct (q,k)-clauses. Notice that this is exactly k-SAT when $q = 2$. We then write (q,k)-SAT, Max-(q,k)-SAT, and (s,c)-Gap-(q,k)-SAT for the natural q-ary analogues of k-SAT, Max-k-SAT, and (s,c)-Gap-k-SAT. In the sequel, we will always take q to be a prime power (since we work with codes over fields), in which case we may equivalently take our q-ary alphabet to be $\mathbb{F}_q$, rather than $\mathbb{Z}_q$.

The next proposition shows some straight-forward reductions between (q,k)-SAT for different values of q, from which we will show that SETH implies a strong q-ary analogue. We note that Items 1 and 3 already appeared in [Tra08].

Proposition 3.1. *For integers $k, q \geq 2$ and $\kappa \geq 1$, we have the following reductions.*

1. *There is a $\mathrm{poly}(m, \kappa, k, \log q)$-time (Karp) reduction that maps any (q^κ, k)-SAT instance with m clauses and $n \leq m$ variables to a $(q, \kappa k)$-SAT instance with m clauses and κn variables.*

2. *There is a $\mathrm{poly}(m, q^{\kappa k})$-time (Karp) reduction that maps any (q,k)-SAT instance with m clauses and $n \leq m$ variables to a (q^κ, k)-SAT instance with $\lceil n/\kappa \rceil$ variables and at most $q^{\kappa k} m$ clauses.*

3. *For $q' \geq q$, there is a $\mathrm{poly}(m, (q')^k)$-time (Karp) reduction that maps any (q,k)-SAT instance with m clauses and $n \leq m$ variables to (q',k)-SAT with n variables and $m + n(q')^k$ clauses.*

Proof. We first prove Item 1. Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ be the q^κ-ary input variables to a (q^κ, k)-SAT instance. Here, we think of the $\boldsymbol{x}_i$ as κ-dimensional vectors over $\mathbb{Z}_q$, $\boldsymbol{x}_i = (x_{i,1}, \dots, x_{i,\kappa})$ (which is why we use bold letters). The reduction creates an instance of $(q, \kappa k)$-SAT with variables $x_{i,j}$ over $\mathbb{Z}_q$ as follows. For each (q^κ, k)-clause of the form $(\boldsymbol{x}_{i_1}, \dots, \boldsymbol{x}_{i_k}) \neq (\boldsymbol{z}_1, \dots, \boldsymbol{z}_k)$ for $\boldsymbol{z}_\ell = (z_{\ell,1}, \dots, z_{\ell,\kappa}) \in \mathbb{Z}_{q^\kappa}$, the reduction creates the $(q, \kappa k)$-clause given by $(x_{i_j,\ell})_{1 \leq j \leq k, 1 \leq \ell \leq \kappa} \neq (z_{j,\ell})_{j,\ell}$. It is clear that the resulting $(q, \kappa k)$-SAT instance is equivalent to the input instance, has the desired properties, and can be constructed in time $\mathrm{poly}(m, \kappa, k, \log q)$.

Turning to Item 2, let $x_1, \dots, x_n$ be the q-ary input variables to a (q,k)-SAT instance. By adding at most $\kappa\lceil n/\kappa \rceil - n$ dummy variables, we may assume that n is divisible by κ. The reduction then groups the variables into groups of size κ. I.e., let $x'_{i,j} := x_{\kappa(i-1)+j}$ for $1 \leq i \leq n/\kappa$

and $1 \le j \le \kappa$. Let $\boldsymbol{x}'_i := (x_{i,1}, \ldots, x_{i,\kappa})$ for $1 \le i \le n/\kappa$. And, the reduction associates with each κ-tuple $\boldsymbol{z} = (z_1, \ldots, z_\kappa) \in \mathbb{Z}_q^\kappa$ an arbitrary unique element in $\mathbb{Z}_{q^\kappa}$, which by abuse of notation we also write as $\boldsymbol{z} = (z_1, \ldots, z_\kappa) \in \mathbb{Z}_{q^\kappa}$.

Then, for each (q,k)-clause in the input instance of the form $(x'_{i_1,j_1}, \ldots, x'_{i_k,j_k}) \neq (z_1, \ldots, z_k)$, the reduction creates a (q^κ, k)-clauses of the form $(\boldsymbol{x}'_{i_1}, \ldots, \boldsymbol{x}_{i_k}) \neq (\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_k)$ for *every* $(\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_k) \in \mathbb{F}_{q^\kappa}^k$ with $(z'_{1,j_1}, \ldots, z'_{k,j_k}) = (z_1, \ldots, z_k)$. (E.g., for $q = 2$ and $\kappa = 2$, the $(2,2)$-clause $(x'_{1,1}, x'_{2,1}) \neq (0,0)$ yields the collection of $(4,2)$-clauses $\{(\boldsymbol{x}_1, \boldsymbol{x}_2) \neq (0,0,0,0), (\boldsymbol{x}_1, \boldsymbol{x}_2) \neq (0,0,0,1), (\boldsymbol{x}_1, \boldsymbol{x}_2) \neq (0,1,0,0), (\boldsymbol{x}_1, \boldsymbol{x}_2) \neq (0,1,0,1)\}$.) There are at most $q^{\kappa k}$ such clauses. It is clear that the resulting (q^κ, k)-SAT instance is satisfiable if and only if the input instance is, that it has the appropriate parameters, and that the reduction be constructed in time $\mathrm{poly}(m, q^{\kappa k})$.

Finally, for Item 3, let $x_1, \ldots, x_n$ be the q-ary input variables to a (q,k)-SAT instance. Since $\mathbb{Z}_q \subseteq \mathbb{Z}_{q'}$, we can view the $x_1, \ldots, x_n$ as q'-ary variables, and we can treat each (q,k)-clause in the input instance $(x_{i_1}, \ldots, x_{i_k}) \neq (z_1, \ldots, z_k)$ as a (q',k)-clause. The output instance will consist of these clauses together with at most $n(q')^k$ additional clauses that are equivalent to the statement that $x_i \neq z$ for all i and all $z \notin \mathbb{Z}_q$. For example, when $q = 2$, $q' = 3$, $k = 2$, and $n = 2$, the reduction adds the clauses $(x_1, x_2) \neq (2,0)$, $(x_1, x_2) \neq (2,1)$, $(x_1, x_2) \neq (0,2)$, $(x_1, x_2) \neq (1,2)$, and $(x_1, x_2) \neq (2,2)$, which together are equivalent to $x_1 \neq 2$ and $x_2 \neq 2$. Again, it is clear that the resulting (q',k)-SAT instance has the desired properties and that the reduction runs in time $\mathrm{poly}(m, (q')^k)$ □

Corollary 3.2. *For all integers $q = q(n) \ge 2$, $q' = q'(n) \ge 2$, $k \ge 2$, and $\kappa = \kappa(n) \ge 2$, there is a $\mathrm{poly}(m, q^{\kappa k}, q')$-time (Karp) reduction that maps any (q,k)-SAT instance with m clauses and $n \le m$ variables to $(q', \lceil \kappa / \log_q q' \rceil k)$-SAT with at most $q^{\kappa k} m + \lceil n/\kappa \rceil \cdot q^{\kappa k} (q')^k$ clauses and at most*

$$n / \log_q q' + \kappa / \log_q q' + n/\kappa + 1$$

variables.

Proof. Let $\kappa' := \lceil \kappa / \log_q q' \rceil$. By Item 2 above, there is a $\mathrm{poly}(m, q^{\kappa k})$-time reduction from (q,k)-SAT with n variables and m clauses to a (q^κ, k)-SAT instance with $\lceil n/\kappa \rceil$ variables and at most $q^{\kappa k} m$ clauses. Notice that $(q')^{\kappa'} \ge q^\kappa$. Therefore, by Item 3, there is a $\mathrm{poly}(m, (q')^{\kappa' k})$-time reduction from this problem to $((q')^{\kappa'}, k)$-SAT with $\lceil n/\kappa \rceil$ variables and at most

$$q^{\kappa k} m + \lceil n/\kappa \rceil \cdot (q')^{\kappa' k} \le q^{\kappa k} m + \lceil n/\kappa \rceil \cdot q^{\kappa k} (q')^k$$

clauses. Finally, by Item 1, there is a $\mathrm{poly}(m, \kappa', k, \log q')$-time reduction from this problem to $(q', \kappa' k)$-SAT with $\kappa' \lceil n/\kappa \rceil$ variables and the same number of clauses.

The running time of the full reduction is $\mathrm{poly}(m, q^{\kappa k}, (q')^{\kappa'})$, and the result follows by plugging in $\kappa' := \lceil \kappa / \log_q q' \rceil$. □

From this, we derive the following theorem, which says that SETH implies a strong q-ary variant of SETH for all q. Traxler proved an analogous result for ETH [Tra08]. (His reduction yields $2^{(1-\varepsilon)\lfloor \log_2 q \rfloor n}$-hardness under SETH.)

Theorem 3.3. *For every constant $\varepsilon \in (0, 1/2)$ and every integer $q = q(n) \ge 2$, there is a constant integer $k \ge 3$ such that no $q^{(1-\varepsilon)n}$-time algorithm solves (q,k)-SAT with n variables, unless SETH is false. (The same result holds for randomized algorithms and randomized SETH; and for non-uniform algorithms and non-uniform SETH.)*

Proof. Assuming SETH, there exists a constant integer $k' \geq 3$ such that no $2^{(1-\varepsilon/2)n'}$-time algorithm solves k'-SAT (i.e., $(2,k')$-SAT) on n' variables.

Let $\kappa := \lceil 10 \log_2(q)/\varepsilon \rceil$. Let $k := \lceil \kappa / \log_2 q \rceil k'$. Notice that $k \leq 20k'/\varepsilon$ is bounded by a constant, independent of n and q. By Corollary 3.2, there is a reduction running in time $\mathrm{poly}(m, 2^{\kappa k'}) = \mathrm{poly}(m,q)$ from k'-SAT with n' variables to (q,k)-SAT with at most

$$n := n'/\log_2 q + \kappa/\log_2 q + n'/\kappa + 1 \leq (1+\varepsilon/5)n'/\log_2 q + 1 \leq (1+\varepsilon/4)n'/\log_2 q$$

variables, where the last inequality assumes that n' is sufficiently large.

Therefore, a $q^{(1-\varepsilon)n}$-time algorithm for a (q,k)-SAT on n variables would imply a $q^{(1-\varepsilon)n} + \mathrm{poly}(m,q) \leq 2^{(1-\varepsilon/2)n'}$-time algorithm for k'-SAT on n' variables, where we have assumed again that n' is sufficiently large. This is a contradiction, and the result follows. □

We also show that Gap-ETH implies a q-ary variant of Gap-ETH for constant $q \geq 2$. (We cannot hope to prove hardness of constant-factor approximation for superconstant q, since any (q,k)-SAT instance has an assignment that satisfies at least $(1-q^{-k}) \cdot m$ clauses.) We first need the following theorem due to [Din16, MR17].

Theorem 3.4. *There exist constants $C \geq 2$ and $s \in (0,1)$ such that no $2^{o(n)}$-time (randomized) algorithm solves $(s,1)$-Gap-3-SAT on n variables and at most Cm clauses in which each variable appears in at most C clauses, unless Gap-ETH is false.*

The following theorem shows how to make Item 3 of Proposition 3.1 work in the more delicate setting of Gap-SAT.[1]

Theorem 3.5. *For any $0 < s \leq c \leq 1$, $C \geq 2$, and any integer $q = q(n) \geq 2$ there is a $\mathrm{poly}(m,q)$-time (Karp) reduction that maps any (s,c)-Gap-3-SAT instance with n variables and $m \leq Cn$ clauses in which each variable appears in at most C clauses to a (s',c')-Gap-$(q,3)$-SAT instance with n variables and $m + q^2(q-2)n$ clauses, where*

$$s' := \frac{C - (1-s) + q^2(q-2)}{C + q^2(q-2)} ,$$

and

$$c' := \frac{c + q^2(q-2)/C}{1 + q^2(q-2)/C} \geq c .$$

In particular, for every constant $q \geq 2$, there exists $s_q \in (0,1)$ such that no $2^{o(n)}$-time randomized algorithm solves $(s_q,1)$-Gap-$(3,q)$-SAT on n variables, unless Gap-ETH is false. (The same result holds for non-uniform algorithms and non-uniform Gap-ETH.)

Proof. The reduction is essentially the same as the one used to prove Item 3 in Proposition 3.1, though the analysis is more difficult. In particular, the reduction treats the binary variables $x_1, \ldots, x_n$ in the input instance as q-ary variables and each $(2,3)$-clause in the input instance $(x_{i_1}, x_{i_2}, x_{i_3}) \neq (z_1, z_2, z_3)$ with $z_i \in \{0,1\}$ as a $(q,3)$-clause. The output (s',c)-$(q,2)$-SAT instance

[1] It is possible to reduce the number of variables in the output instance of Theorem 3.5 to roughly $n/\log_2 q$ as in Corollary 3.2 at the expense of a worse approximation factor. However, we do not attempt to do that here, since this reduction is primarily interesting for constant q, and we are not concerned with such constant factors in the context of Gap-ETH.

consists of these clauses, and for each $1 \le i \le n$ an additional $q^2(q-2)$ distinct (q,k)-clauses that together are equivalent to the statement that $x_i \in \{0,1\}$. E.g., for each $i < n-1$, we can add the clauses $(x_i, x_{i+1}, x_{i+2}) \neq (z_1, z_2, z_3)$ for all $z_1 \in \mathbb{Z}_q \setminus \{0,1\}$ and all $z_2, z_3 \in \mathbb{Z}_q$, and for $i = n-1$ and $i = n$, we can add $(x_i, x_1, x_2) \neq (z_1, z_2, z_3)$ for the same values of z_1, z_2, z_3.

Clearly the reduction runs in the claimed time. Furthermore, if there is a boolean assignment satisfying at least cm clauses of the input instance, then there is an assignment satisfying at least a

$$\frac{cm + q^2(q-2)n}{m + q^2(q-2)n} \ge c'$$

fraction of the clauses in the output instance. I.e., the output (s', c')-(q,k)-SAT is a YES, as needed.

Now, suppose that no boolean assignment satisfies sm clauses in the input instance. For an assignment $(z_1, \ldots, z_n) \in \mathbb{Z}_q^n$, let ℓ be the number of variables z_i with $z_i \notin \{0,1\}$. Let S_1 be the number of "original clauses" in the output instance satisfied by this assignment and S_2 be the number of "unoriginal clauses" satisfied by this assignment, where here by "original clauses," we mean the clauses that came from the original input instance. Since all variables assigned a boolean value collectively satisfy at most $sm - 1$ original clauses and each variable appears in at most C original clauses, we have $S_1 \le \min\{m, sm + C\ell - 1\}$. And, by construction, we have $S_2 \le q^2(q-2)(n-\ell) + (q^2(q-2)-1)\ell = q^2(q-2)n - \ell$. Therefore, the fraction of satisfied clauses is

$$\begin{aligned}
\frac{S_1 + S_2}{m + q(q-2)n} &\le \frac{\min\{m - \ell, sm + (C-1)\ell - 1\} + q^2(q-2)n}{m + q^2(q-2)n} \\
&\le \frac{(1 - (1-s)/C)m + q(q-2)n - 1/C}{m + q^2(q-2)n} \\
&\le \frac{C - (1-s) + q^2(q-2) - 1/C}{C + q^2(q-2)} \\
&< s' \,,
\end{aligned}$$

as needed, where the second inequality follows from the fact that this expression is maximized when $\ell = \frac{(1-s)m+1}{C}$. □

4 SETH-hardness for NCP

We will need the gadget guaranteed by the following claim. The code is the Hadamard code, whose key property from our perspective is that all non-zero codewords are permutations of each other. (Bonisoli proved that the Hadamard code is the only such code [Bon84], up to trivial equivalences, even if we only ask that all non-zero codewords have the same Hamming weight.) In particular, when $\boldsymbol{z} = \boldsymbol{0}$, we will take $\boldsymbol{t} = (1, 1, \ldots, 1)$ in Claim 4.1, so that Hamming distance to $\boldsymbol{t}$ corresponds to the number of 1s in a codeword. (Any non-zero multiple of $(1, 1, \ldots, 1)$ would work.)

Claim 4.1. *For every integer $k \ge 2$ and prime power $q \ge 2$, there exists a matrix $C \in \mathbb{F}_q^{(q^k-1)\times k}$ with the following property. For any $\boldsymbol{z} \in \mathbb{F}_q^k$, there is a target vector $\boldsymbol{t} \in \mathbb{F}_q^{q^k-1}$ such that*

$$\|C\boldsymbol{z} - \boldsymbol{t}\|_H = q^k - 1 \,,$$

but

$$\|C\boldsymbol{z}' - \boldsymbol{t}\|_H = q^k - q^{k-1} - 1$$

for all $\boldsymbol{z}' \in \mathbb{F}_q^k$ *with* $\boldsymbol{z}' \neq \boldsymbol{z}$.

Furthermore, given q, k, *and* $\boldsymbol{z}$, *such a* C *and* $\boldsymbol{t}$ *can be computed in time* $\mathrm{poly}(q^k)$.

Proof. Let $x_1, \ldots, x_{q-1}$ be all non-zero elements of $\mathbb{F}_q$. Let $C \in \mathbb{F}_q^{(q^k-1)\times k}$ be the matrix

$$C := \begin{pmatrix} 0 & 0 & \cdots & 0 & x_1 \\ 0 & 0 & \cdots & 0 & x_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{q-1} & x_{q-1} & \cdots & x_{q-1} & x_{q-2} \\ x_{q-1} & x_{q-1} & \cdots & x_{q-1} & x_{q-1} \end{pmatrix}$$

consisting of all possible non-zero rows in $\mathbb{F}_q^k$. (Row i can be viewed as the base q representation of i, and C generates the Hadamard code.) Take $\boldsymbol{t} := \boldsymbol{u} + C\boldsymbol{z}$, where $\boldsymbol{u} \in \mathbb{F}_q^{q^k-1}$ is the all ones vector, $\boldsymbol{u} := (1, 1, \ldots, 1)$. Clearly, this can be computed in time $\mathrm{poly}(q^k)$ as claimed.

Furthermore,

$$\|C\boldsymbol{z} - \boldsymbol{t}\|_H = \|\boldsymbol{u}\|_H = q^k - 1 \,.$$

On the other hand, for $\boldsymbol{z}' \neq \boldsymbol{z}$, we have

$$\|C\boldsymbol{z}' - \boldsymbol{t}\|_H = \|C(\boldsymbol{z} - \boldsymbol{z}') - \boldsymbol{u}\|_H = q^k - q^{k-1} - 1 \,,$$

where we have used the fact that every non-zero codeword in this code has exactly q^{k-1} coordinates equal to one. □

We can now present our main reduction for NCP. The SETH-hardness of NCP in Theorem 1.1 follows from this reduction together with Theorem 3.3 (which shows that SETH implies a similar statement for (q,k)-SAT). The Gap-ETH hardness of γ-NCP in Theorem 1.2 for constant $q \geq 2$ follows from this reduction together with Theorem 3.5 (which shows that Gap-ETH implies a similar statement for (s,c)-Gap-(q,k)-SAT for constant q). Finally, Claim 2.3 lets us extend this to any $q = p^\kappa$ for constant $p \geq 2$, and therefore all finite fields with constant characteristic.

Theorem 4.2. *There is a* $\mathrm{poly}(n, m, q^k)$ *(Karp) reduction that maps any Max-*(q,k)*-SAT instance with* m *clauses on* $n \leq m$ *variables and value* $r \leq m$ *to an NCP instance with rank at most* n*, ambient dimension* $(q^k - 1)m$*, and distance* $(q^k - 1)m - q^{k-1}r$.

In particular, for any $0 < s \leq c \leq 1$, *the reduction maps* (s,c)*-Gap-*(q,k)*-SAT to* γ*-GapNCP, where*

$$\gamma := \frac{1 - s/q - q^{-k}}{1 - c/q - q^{-k}} \,.$$

Proof. The reduction takes as input (q,k)-clauses $\phi_1, \ldots, \phi_m$ and an integer $0 \leq r \leq m$ and constructs the matrix and target

$$\Phi := \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_m \end{pmatrix} \text{ and } \boldsymbol{t} := \begin{pmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \\ \vdots \\ \boldsymbol{t}_m \end{pmatrix},$$

with $\Phi_i \in \mathbb{F}_q^{(q^k-1)\times n}$ and $\boldsymbol{t}_i \in \mathbb{F}_q^{q^k-1}$ as follows.

For each (q,k)-clause $\phi_i = ((x_{\ell_{i,1}}, \ldots, x_{\ell_{i,k}}) \neq (z_{i,1}, \ldots, z_{i,k}))$, let $\boldsymbol{z}_i := (z_{i,1}, \ldots, z_{i,k}) \in \mathbb{F}_q^k$ be the vector corresponding to the unique assignment that does not satisfy ϕ_i. Let $C \in \mathbb{F}_q^{(q^k-1)\times k}$ and $\boldsymbol{t}_i \in \mathbb{F}_q^{2^k-1}$ be as in Claim 4.1 with $\boldsymbol{z} = \boldsymbol{z}_i$. For each $j = 1, \ldots k$, the reduction sets the $\ell_{i,j}$th column of Φ_i to equal the jth column of C. All other columns of Φ_i are zero. (For example, if $k = 2$ and ϕ_i is the clause $(x_1, x_3) \neq (z_1, z_2)$, then $\Phi_i = (C_1, 0, C_2, 0, 0, \cdots, 0)$, where here C_j is the jth column of C.)

Finally, the reduction outputs the NCP instance consisting of generator matrix Φ, target $\boldsymbol{t}$, and distance $d := (q^k-1)(m-r) + (q^k - q^{k-1} - 1)r = (q^k-1)m - q^{k-1}r$.

It is clear that this reduction runs in time $\mathrm{poly}(n, m, q^k)$ as claimed. To prove correctness, we consider an assignment vector $\boldsymbol{a} \in \mathbb{F}_q^n$. Notice that $\Phi_i \boldsymbol{a} = C\boldsymbol{z}_i$ if and only if the assignment $\boldsymbol{a}$ fails to satisfy clause ϕ_i. Otherwise, $\Phi_i \boldsymbol{a} = C\boldsymbol{z}'$ for some other $\boldsymbol{z}' \in \mathbb{F}_q^k$. Therefore, by Claim 4.1,

$$\|\Phi_i \boldsymbol{a} - \boldsymbol{t}_i\|_H = \begin{cases} q^k - q^{k-1} - 1 & \boldsymbol{a} \text{ satisfies } \phi_i \\ q^k - 1 & \text{otherwise.} \end{cases}$$

It follows that

$$\|\Phi \boldsymbol{a} - \boldsymbol{t}\|_H = (q^k-1)(m - S_{\boldsymbol{a}}) + (q^k - q^{k-1} - 1)S_{\boldsymbol{a}} \,,$$

where $S_{\boldsymbol{a}}$ is the number of clauses satisfied by $\boldsymbol{a}$. So, $\mathrm{dist}(\boldsymbol{t}, C) \leq d$ if and only if the value of the number of maximum number satisfiable clauses is at least r, as needed. □

4.1 SETH-hardness of NCP with Preprocessing

We now sketch how to modify the above reduction to show essentially the same SETH-hardness result for NCP with preprocessing (though not hardness of approximation). The idea is to modify the gadget in Claim 4.1 so that we can choose a special target $\boldsymbol{t}_{\mathsf{off}}$ that "turns off" a clause—i.e., $\boldsymbol{t}_{\mathsf{off}}$ has the property that $\|C\boldsymbol{z} - \boldsymbol{t}_{\mathsf{off}}\|_H$ is the same for all assignments $\boldsymbol{z}$. In fact, any gadget that satisfies Claim 4.1 already implies such a gadget with at most q^k times as many rows,[2] but we build one with just q times as many rows (though this does not affect our results asymptotically).

Given such a gadget, we can then construct a code whose rows consist of such gadgets for all $m = m(q, n, k)$ possible (q,k)-clause on n variables. Then, given a (q,k)-SAT instance consisting of a subset of these clauses, we construct the corresponding target $\boldsymbol{t} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_m)$ by taking $\boldsymbol{t}_i = \boldsymbol{t}_{\mathsf{off}}$ if clause i is not in the input instance and mimicking the reduction from the previous section for the other clauses. I.e., we effectively "turn off the rows" corresponding to clauses that are not in our input instance.

Claim 4.3. *For every integer $k \geq 2$ and prime power $q \geq 2$, there exists a matrix $C \in \mathbb{F}_q^{q(q^k-1)\times k}$ and vector $\boldsymbol{t}_{\mathsf{off}} \in \mathbb{F}_q^{q(q^k-1)}$ with the following property. For all $\boldsymbol{z}' \in \mathbb{F}_q^k$,*

$$\|C\boldsymbol{z}' - \boldsymbol{t}_{\mathsf{off}}\|_H = (q-1)q^{k-1} \,,$$

[2] To see this let $C, \boldsymbol{t}(\boldsymbol{z})$ be a gadget satisfying Claim 4.1. Let C' be the matrix consisting of q^k copies of C stacked on top of each other. Let $t'(\boldsymbol{z}) := (\boldsymbol{t}(\boldsymbol{z}), \ldots, \boldsymbol{t}(\boldsymbol{z}))$. Let $\boldsymbol{t}_{\mathsf{off}} := (\boldsymbol{t}(\boldsymbol{z}_1), \boldsymbol{t}(\boldsymbol{z}_2), \ldots, \boldsymbol{t}(\boldsymbol{z}_{q^k}))$, where the $\boldsymbol{z}_i \in \mathbb{F}_q^k$ represent all possible assignments. Then, clearly $\boldsymbol{t}_{\mathsf{off}}$ is equidistant from all codewords and $\boldsymbol{t}'(\boldsymbol{z})$ is equidistant from all codewords except $C'\boldsymbol{z}$. [ABGS19] show a slightly different way to modify gadgets like those in Claim 4.1 to the type of gadget that we need in this section.

and for any $\boldsymbol{z} \in \mathbb{F}_q^k$*, there is a target vector* $\boldsymbol{t} \in \mathbb{F}_q^{q(q^k-1)}$ *such that*

$$\|C\boldsymbol{z} - \boldsymbol{t}\|_H = q(q^k - 1) \ ,$$

but

$$\|C\boldsymbol{z}' - \boldsymbol{t}\|_H = q(q^k - q^{k-1} - 1)$$

for all $\boldsymbol{z}' \in \mathbb{F}_q^k$ *with* $\boldsymbol{z}' \neq \boldsymbol{z}$*.*

Furthermore, given q*,* k*, and* $\boldsymbol{z}$*, such a* C*,* $\boldsymbol{t}$*, and* $\boldsymbol{t}_{\mathsf{off}}$ *can be computed in time* $\mathrm{poly}(q^k)$*.*

Proof. Let $x_0, x_1, \ldots, x_{q-1}$ be all elements of $\mathbb{F}_q$. Let $\widehat{C} \in \mathbb{F}_q^{(q^k-1)\times k}$ and $\widehat{\boldsymbol{t}} \in \mathbb{F}_q^{q^k-1}$ be the matrix and target from Claim 4.1. Take

$$C := \begin{pmatrix} \widehat{C} \\ \widehat{C} \\ \vdots \\ \widehat{C} \end{pmatrix} \in \mathbb{F}_q^{q(q^k-1)\times k}$$

to be the matrix consisting of q copies of $\widehat{C}$ and similarly take $\boldsymbol{t} := (\widehat{\boldsymbol{t}}, \ldots, \widehat{\boldsymbol{t}})$. It is immediate from that claim that C and $\boldsymbol{t}$ satisfy the desired properties.

Finally, take $\boldsymbol{t}_{\mathsf{off}} := (x_0, x_0, \ldots, x_0, x_1, x_1, \ldots, x_1, \ldots, x_{q-1}, x_{q-1}, \ldots, x_{q-1}) \in \mathbb{F}_q^{q(q^k-1)}$ to be the vector consisting of $q^k - 1$ consecutive copies of each element in $\mathbb{F}_q$. Notice that, for every non-zero $\boldsymbol{z}' \in \mathbb{F}_q^k$, the codeword $\widehat{C}\boldsymbol{z}'$ is a vector whose entries contain $q^{k-1} - 1$ zeros and q^{k-1} copies of each non-zero field element. (The reason that there are fewer zeros is because we excluded the zero row in $\widehat{C}$.) It follows immediately that $\|C\boldsymbol{z}' - \boldsymbol{t}_{\mathsf{off}}\|_H = (q^k - q^{k-1}) + (q-1)(q^k - q^{k-1} - 1) = (q-1)(q^k - 1)$ for non-zero $\boldsymbol{z}' \in \mathbb{F}_q^k$. Finally, for $\boldsymbol{z}' = \boldsymbol{0}$, we also have $\|C\boldsymbol{z}' - \boldsymbol{t}_{\mathsf{off}}\|_H = \|\boldsymbol{t}_{\mathsf{off}}\|_H = (q-1)(q^k - 1)$, as needed. □

Theorem 4.4. *There is a* $\mathrm{poly}(n^k, q^k)$*-time reduction that maps any Max-*(q,k)*-SAT instance on* n *variables to an NCP instance with rank* n *and ambient dimension* $M := q^{k+1}(q^k - 1) \cdot \binom{n}{k}$*. Furthermore, the code of the NCP instance depends only on* n*,* q*, and* k *(and not on the clauses of the input SAT instance).*

Proof. We write $m := q^k \binom{n}{k}$ for the total number of possible (q,k)-clauses on n variables. Notice that $M = q(q^k - 1) \cdot m$. The reduction constructs the generator matrix

$$\Phi_{q,k,n} := \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_m \end{pmatrix} \in \mathbb{F}_q^{M\times n}$$

of the code for fixed n, q, and k as follows. Let $\phi_1, \ldots, \phi_m$ be a list of all possible (q,k)-clauses over n variables. The reduction constructs Φ_i exactly as in Theorem 4.2, using the gadget C from Claim 4.3, rather than the gadget from Claim 4.1. I.e., it places the jth column of C in the column of Φ_i corresponding to the jth variable in ϕ_i.

Given some Max-(q,k)-SAT on n variables, consisting of some subset $T \subseteq \{1,\dots,m\}$ of the clauses and a value $0 \le r \le |T|$, the reduction constructs the target

$$\boldsymbol{t} := \begin{pmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \\ \vdots \\ \boldsymbol{t}_m \end{pmatrix} \in \mathbb{F}_q^{M \times n}$$

as follows. If $i \notin T$, then $\boldsymbol{t}_i = \boldsymbol{t}_{\mathsf{off}}$ as defined in Claim 4.3. In particular, for such i we have $\|\Phi_i \boldsymbol{a} - \boldsymbol{t}_i\|_H = (q-1)(q^k-1)$ for all $\boldsymbol{a} \in \mathbb{F}_q^n$. If $i \in T$, then the reduction computes $\boldsymbol{t}_i$ corresponding to ϕ_i as in Theorem 4.2, except using Claim 4.3 rather than Claim 4.1. In particular, we have $\|\Phi_i \boldsymbol{a} - \boldsymbol{t}_i\|_H = q(q^k-1)$ for any assignment $\boldsymbol{a} \in \mathbb{F}_q^n$ that does not satisfy ϕ_i and $\|\Phi_i \boldsymbol{a} - \boldsymbol{t}_i\|_H = q(q^k - q^{k-1} - 1)$ if $\boldsymbol{a}$ does satisfy ϕ_i. Finally, the reduction outputs the NCP instance given by generator matrix Φ, target $\boldsymbol{t}$, and distance $d := (q^k-1)(q-1)m - |T| + q^k(|T| - r)$.

The running time is clearly as claimed. Furthermore, by the analysis above, we see that for any assignment $\boldsymbol{a} \in \mathbb{F}_q^n$,

$$\begin{aligned} \|\Phi \boldsymbol{a} - \boldsymbol{t}\|_H &= q(q^k - q^{k-1} - 1)S_{\boldsymbol{a}} + q(q^k-1)U_{\boldsymbol{a}} + (q-1)(q^k-1)(m-|T|) \\ &= (q^k-1)(q-1)m - |T| + q^k(|T| - S_{\boldsymbol{a}}) \ , \end{aligned}$$

where $S_{\boldsymbol{a}}$ is the number of clauses in T satisfied by $\boldsymbol{a}$, $U_{\boldsymbol{a}}$ is the number of clauses in T not satisfied by $\boldsymbol{a}$, and we have used the fact that $U_{\boldsymbol{a}} = |T| - S_{\boldsymbol{a}}$. In particular, $\mathrm{dist}(\boldsymbol{t}, C) \le d$ if and only if there exists an assignment satisfying at least r of the clauses in T, as needed. □

5 Hardness of MDP

To prove fine-grained hardness of MDP, we first present (our interpretation of) Dumer, Micciancio, and Sudan's reduction from NCP to MDP [DMS03], which requires a certain type of gadget to work. Our contribution in this section is therefore entirely in the construction of the gadgets. We first define the gadget and then discuss our constructions. (Our gadget differs slightly from [DMS03] in that we require all vectors to be at the same distance. The same idea is used in [AS18] in the context of lattices and in [KM19] in a very different context.)

Definition 5.1 (Locally dense codes)**.** *For integers $1 \le n \le m$ and $1 \le M \le q^n$, we say that a generator matrix $C \in \mathbb{F}_q^{m \times n}$, a target $\boldsymbol{t} \in \mathbb{F}_q^m$, and distance $1 \le d \le m$ form an M-*locally dense triple *if $d := \mathrm{dist}(\boldsymbol{t}, C) < \lambda(C)$, and*

$$\left|\left\{\boldsymbol{z} \in \mathbb{F}_q^n \ : \ \|C\boldsymbol{z} - \boldsymbol{t}\|_H = d\right\}\right| \ge M \ .$$

We will construct two different families of locally dense codes in this section. (In Appendix A, we construct a third such code with an additional property.) Our first construction uses Reed-Solomon codes and is quite similar to the construction in [DMS03]. The main difference is just in the setting of parameters. We choose our parameters to minimize the rank n of the gadget code relative to M, achieving $M = q^{(1-\varepsilon)n}$ for any $\varepsilon > 0$. Together with Theorem 4.2, this simple change is enough to prove essentially optimal hardness of *exact* MDP under randomized SETH. In contrast, [DMS03] were content with any $M = q^{n^{\Omega(1)}}$ but focused on achieving a ratio $\mathrm{dist}(\boldsymbol{t}, C)/\lambda(C)$ that is bounded away from one, which allowed them to prove hardness of γ-MDP for constant $\gamma > 1$.

Our second construction is based on Ashikhmin, Barg, and Vlăduţ's codes over $\mathbb{F}_2$ with kissing number that is exponential in the ambient dimension [ABV01]. (A similar idea was used to show hardness of lattice problems in [AS18].) With this, we simultaneously achieve $M = 2^{\Omega(n)}$ and a ratio $\mathrm{dist}(\boldsymbol{t}, C)/\lambda(C) < 1 - \Omega(1)$ that is bounded away from one, over $\mathbb{F}_2$. I.e., up to the hidden constants in the asymptotic notation, we achieve the best of both worlds for the case $q = 2$. This allows us to prove $2^{\Omega(n)}$ hardness of approximation under (non-uniform) Gap-ETH for all fields of characteristic two.

We can now present (a version of) Dumer, Micciancio, and Sudan's reduction. (Notice that the reduction is only meaningful if $\gamma' \geq 1$.)

Theorem 5.2. *There is an efficient (randomized) reduction that takes as auxiliary input an M-locally dense triple $(C^\dagger \in \mathbb{F}_q^{m^\dagger \times n^\dagger}, \boldsymbol{t} \in \mathbb{F}_q^{n^\dagger}, d^\dagger \leq m^\dagger)$ with $M \geq 10q^n$ and reduces any γ-NCP instance over $\mathbb{F}_q$ with rank n, ambient dimension m, and distance d to a γ'-MDP instance over $\mathbb{F}_q$ with rank at most $n^\dagger + 1$, ambient dimension $m + m^\dagger$, and distance $d + d^\dagger$, where*

$$\gamma' := \frac{\min\left\{\lambda(C^\dagger),\ \gamma d + d^\dagger\right\}}{d + d^\dagger} \ .$$

The reduction succeeds with probability at least $1 - q^n/M$

Proof. Given as input the generator matrix for a code $C \in \mathbb{F}_q^{m \times n}$, target vector $\boldsymbol{t} \in \mathbb{F}_q^m$, distance $1 \leq d \leq m$ and an M-locally dense triple $(C^\dagger, \boldsymbol{t}^\dagger, d^\dagger)$, the reduction behaves as follows. It samples $T \in \mathbb{F}_q^{n \times n^\dagger}$ and $\boldsymbol{z}' \in \mathbb{F}_q^n$ uniformly at random and sets

$$C' := \begin{pmatrix} CT & -\boldsymbol{t} - C\boldsymbol{z}' \\ C^\dagger & -\boldsymbol{t}^\dagger \end{pmatrix} \in \mathbb{F}_q^{(m+m^\dagger)\times(n^\dagger+1)} \ .$$

The MDP instance is simply C' and $d' := d + d^\dagger$.

Clearly the reduction is efficient and achieves the parameters claimed in the theorem. To prove correctness, it will be convenient to define $W := \{(CT\boldsymbol{z}^\dagger, C^\dagger\boldsymbol{z}^\dagger) \ : \ \boldsymbol{z}^\dagger \in \mathbb{F}_q^{n^\dagger}\}$ to be the subspace generated by the first $n^\dagger$ columns of C'. Then, the code generated by C' is just W together with $W - (\boldsymbol{t} + C\boldsymbol{z}', \boldsymbol{t}^\dagger)\mathbb{F}_q^*$, where we write $\mathbb{F}_q^*$ for the non-zero elements of $\mathbb{F}_q$. Notice that the shortest non-zero codeword in W has length at least $\lambda(C^\dagger) + \lambda(C) > \lambda(C^\dagger)$, and the shortest codeword in $W - (\boldsymbol{t} + C\boldsymbol{z}', \boldsymbol{t}^\dagger)\mathbb{F}_q^*$ has length at least $\max_{\alpha \in \mathbb{F}_q^*} \mathrm{dist}(\alpha\boldsymbol{t}, C) + \mathrm{dist}(\alpha\boldsymbol{t}^\dagger, C^\dagger) = \mathrm{dist}(\boldsymbol{t}, C) + d^\dagger$.

So, suppose $\mathrm{dist}(\boldsymbol{t}, C) > \gamma d$. It follows that

$$\lambda(C') > \min\left\{\lambda(C^\dagger),\ \gamma d + d^\dagger\right\} = \gamma' d' \ .$$

I.e., the MDP instance is a NO.

On the other hand, suppose that $\mathrm{dist}(\boldsymbol{t}, C) \leq d$. Then, let $\boldsymbol{z} \in \mathbb{F}_q^n$ be such that $\|C\boldsymbol{z} - \boldsymbol{t}\|_H \leq d$. By Claim 2.4, with probability at least $1 - M/q^n$, there exists a $\boldsymbol{z}^\dagger \in \mathbb{F}_q^{n^\dagger}$ such that $\|C^\dagger\boldsymbol{z}^\dagger - \boldsymbol{t}^\dagger\|_H = d^\dagger$ *and* $T\boldsymbol{z}^\dagger = \boldsymbol{z} + \boldsymbol{z}'$. Assuming that such $\boldsymbol{z}^\dagger$ exists, then $(CT\boldsymbol{z}^\dagger, C^\dagger\boldsymbol{z}^\dagger) - (\boldsymbol{t} + C\boldsymbol{z}', \boldsymbol{t}^\dagger)$ is a non-zero codeword in the code generated by C' with weight at most $d + d^\dagger$. So, with probability at least $1 - M/q^n$, the MDP instance is a YES, as needed. □

The next simple corollary shows that we can "add k copies" of the gadget in order to increase γ' a bit. (In particular, this allows us to achieve $\gamma' \geq 1$, and thus a non-trivial reduction.)

Corollary 5.3. *There is an efficient (randomized) reduction that takes as auxiliary input an M-locally dense triple $(C^\dagger \in \mathbb{F}_q^{m^\dagger \times n^\dagger}, \boldsymbol{t} \in \mathbb{F}_q^{n^\dagger}, d^\dagger \le m^\dagger)$ with $M \ge 10q^n$ and an integer $k \ge 1$ (represented in unary) and reduces any γ-NCP instance with rank n, ambient dimension m, and distance d to a γ'-MDP instance with rank at most $n^\dagger+1$, ambient dimension $m+km^\dagger$, and distance $d+kd^\dagger$, where*

$$\gamma' := \frac{\min\left\{k\lambda(C^\dagger),\ \gamma d + kd^\dagger\right\}}{d + kd^\dagger}\ .$$

The reduction succeeds with probability at least $1 - q^n/M$

In particular, for $k \ge d$, we have $\gamma' \ge 1$ (since $\lambda(C^\dagger) \ge d^\dagger + 1$ by the definition of an M-locally dense triple).

Proof. The reduction simply constructs the new gadget

$$(C^\dagger)' := \begin{pmatrix} C^\dagger \\ \vdots \\ C^\dagger \end{pmatrix} \in \mathbb{F}_q^{km^\dagger \times n^\dagger}, \qquad (\boldsymbol{t}^\dagger)' := \begin{pmatrix} \boldsymbol{t}^\dagger \\ \vdots \\ \boldsymbol{t}^\dagger \end{pmatrix} \in \mathbb{F}_q^{km^\dagger}\ ,$$

and $(d^\dagger)' := kd^\dagger$ and then runs the reduction from Theorem 5.2 with this new gadget. □

5.1 SETH hardness via Reed-Solomon codes

We now use Reed-Solomon codes to construct locally dense triples with the key property that we can take $M = q^{(1-\varepsilon)n}$ for any constant $\varepsilon > 0$. We first construct our gadget for sufficiently large $q > n$ and then use concatenation to extend this to all q.

Proposition 5.4. *There is a $\mathrm{poly}(q)$-time deterministic algorithm that takes as input a prime power $q \ge 2$ and integer $n < q$ and outputs a generator matrix $C \in \mathbb{F}_q^{(q-1)\times n}$ and target $\boldsymbol{t} \in \mathbb{F}_q^{q-1}$ such that $(C, \boldsymbol{t}, d)$ form an M-locally dense triple, where $d := q - n - 1$ and $M := \binom{q-1}{n}$.*

Proof. Let

$$C := \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{q-1} & x_{q-1}^2 & \cdots & x_{q-1}^{n-1} \end{pmatrix} \in \mathbb{F}_q^{q-1\times n}$$

be a generator matrix of a Reed-Solomon code, where $x_1, \ldots, x_{q-1} \in \mathbb{F}_q^*$ are all distinct non-zero field elements. Recall that $\lambda(C) = q - n > d$. Let $\boldsymbol{t} := (-x_1^{-1}, -x_2^{-1}, \ldots, -x_{q-1}^{-1}) \in \mathbb{F}_q^{q-1}$. Then, for any codeword $\boldsymbol{p} := (p(x_1), \ldots, p(x_{q-1}))$ corresponding to a polynomial $p(x) \in \mathbb{F}_q^n$ with degree less than n, the quantity $(q-1) - \|\boldsymbol{t} - \boldsymbol{p}\|_H$ is exactly the number of non-zero roots of the polynomial $xp(x) + 1$. This polynomial has degree at most n and therefore has at most n roots. So, $\mathrm{dist}(\boldsymbol{t}, C) \ge q - n - 1 = d$.

Notice that any polynomial with degree at most n and with constant term one can be written (uniquely) as $xp(x) + 1$ for some $p(x)$ with $\deg(p) < n$. In particular, for every $S \subset \{1, \ldots, q-1\}$ with $|S| = n$, there exists a (unique) $p(x)$ with degree $n-1$ satisfying

$$xp(x) + 1 = \frac{\prod_{i\in S}(x - x_i)}{\prod_{i\in S}(-x_i)}\ .$$

For such $p(x)$, we have $\|\boldsymbol{t}-\boldsymbol{p}\|_H = d$. Therefore, $\mathrm{dist}(\boldsymbol{t}, C) = d$ and there are exactly $\binom{q-1}{n}$ vectors $\boldsymbol{p}$ with $\|\boldsymbol{t}-\boldsymbol{p}\|_H = d$ (one for each such subset S), as needed. □

Corollary 5.5. *There is a* $\mathrm{poly}(q^\kappa)$*-time deterministic algorithm that takes as input a prime power* $q \geq 2$ *and positive integers* n *and* κ *satisfying* $q^\kappa > n$ *and outputs a generator matrix* $C \in \mathbb{F}_q^{\kappa(q^\kappa-1)^2 \times \kappa n}$*, target* $\boldsymbol{t} \in \mathbb{F}_q^{\kappa(q^\kappa-1)^2}$*, and distance* $d := \kappa(q^\kappa - q^{\kappa-1})(q^\kappa - n - 1)$ *such that* $(C, \boldsymbol{t}, d)$ *form an* M*-locally dense triple, where* $M := \binom{q^\kappa-1}{n}$*.*

Proof. Let $\widehat{C} \in \mathbb{F}_{q^\kappa}^{q^\kappa-1 \times n}$ and $\widehat{\boldsymbol{t}} \in \mathbb{F}_{q^\kappa}^{q^\kappa-1}$ be as in Proposition 5.4. Let $\phi : \mathbb{F}_{q^\kappa} \to \mathbb{F}_q^{\kappa(q^\kappa-1)}$ be a linear map defined as follows. Recall that the finite field $\mathbb{F}_{q^\kappa}$ is isomorphic as an $\mathbb{F}_q$-vector space to $\mathbb{F}_q^\kappa$, and let $\psi : \mathbb{F}_{q^\kappa} \to \mathbb{F}_q^\kappa$ be an isomorphism between them (which can be found and computed efficiently). Let $x_1, \ldots, x_{q^\kappa-1} \in \mathbb{F}_{q^\kappa}^*$ be all distinct non-zero field elements. Then, let

$$\phi(z) := \begin{pmatrix} \psi(x_1 z) \\ \psi(x_2 z) \\ \vdots \\ \psi(x_{q^\kappa-1} z) \end{pmatrix} \in \mathbb{F}_q^{\kappa(q^\kappa-1)} .$$

We extend ϕ to a map from $\mathbb{F}_{q^\kappa}^{q^\kappa-1}$ to $\mathbb{F}_q^{\kappa(q^\kappa-1)^2}$ by letting it act coordinate-wise, and we let $C \in \mathbb{F}_q^{\kappa(q^\kappa-1)^2 \times \kappa n}$ be a generator matrix of the code $\{\phi(\widehat{C}\boldsymbol{z}) \ : \ \boldsymbol{z} \in \mathbb{F}_{q^\kappa}^n\}$ and $\boldsymbol{t} := \phi(\widehat{\boldsymbol{t}})$. (Since ψ is an isomorphism of vector spaces, this is in fact a code.)

It is clear that C and $\boldsymbol{t}$ can be computed in $\mathrm{poly}(q^\kappa)$ time. Notice that for every non-zero $z \in \mathbb{F}_{q^\kappa}$, we have $\|\phi(z)\|_H = \alpha := \kappa(q^\kappa - q^{\kappa-1})$. (In particular, $\phi(z)$ consists of all $q^\kappa - 1$ possible non-zero vectors in $\mathbb{F}_q^\kappa$. Its Hamming weight is therefore q^κ times the average Hamming weight of a vector in $\mathbb{F}_q^\kappa$, which is $(1-1/q)\kappa$.) Therefore, for $\boldsymbol{z} \in \mathbb{F}_{q^\kappa}^{q^\kappa-1}$, $\|\phi(\boldsymbol{z})\|_H = \alpha \cdot \|\boldsymbol{z}\|_H$. It follows that $\lambda(C) = \alpha \cdot \lambda(\widehat{C})$, $\mathrm{dist}(\boldsymbol{t}, C) = \alpha \cdot \mathrm{dist}(\widehat{\boldsymbol{t}}, \widehat{C})$, and

$$\left|\{\boldsymbol{z} \in \mathbb{F}_q^n \ : \ \|C\boldsymbol{z} - \boldsymbol{t}\|_H = \mathrm{dist}(\boldsymbol{t}, C)\}\right| = \left|\{\boldsymbol{z} \in \mathbb{F}_{q^\kappa}^n \ : \ \|\widehat{C}\boldsymbol{z} - \widehat{\boldsymbol{t}}\|_H = \mathrm{dist}(\widehat{\boldsymbol{t}}, \widehat{C})\}\right| .$$

The result follows from the properties of $\widehat{C}$ and $\widehat{\boldsymbol{t}}$ guaranteed by Proposition 5.4. □

Corollary 5.6. *For every constant* $\varepsilon \in (0, 1/2)$*, there is a* $\mathrm{poly}_\varepsilon(n, q)$*-time deterministic algorithm that takes as input a prime power* $q = q(n) \geq 2$ *and sufficiently large integer* n *and outputs a code* $C \in \mathbb{F}_q^{m \times n}$*, target* $\boldsymbol{t} \in \mathbb{F}_q^m$*, and distance* $d \geq (1-\varepsilon)(1-1/q)m$ *such that* $(C, \boldsymbol{t}, d)$ *is a* $q^{(1-\varepsilon)n}$*-locally dense triple, where* $m = \mathrm{poly}_\varepsilon(n, q)$*.*

Proof. Let $\kappa := 10\lceil(1 + \log_q n)/\varepsilon\rceil$ and $\widehat{n} := \lfloor n/\kappa \rfloor$ such that $\binom{q^\kappa-1}{\widehat{n}} \geq q^{(1-\varepsilon)n}$ and

$$\frac{d}{m} = \frac{\kappa(q^\kappa - q^{\kappa-1})(q^\kappa - n - 1)}{\kappa(q^\kappa - 1)^2} \geq (1-\varepsilon)(1-1/q) .$$

The result then follows by Corollary 5.5. □

Corollary 5.7. *For every constant* $\varepsilon \in (0, 1/2)$ *and every prime power* $q = q(n) \geq 2$*, there is an* $\mathrm{poly}_\varepsilon(m, q)$*-time (randomized) reduction that maps any NCP instance over* $\mathbb{F}_q$ *with rank* $n \geq 2$ *and ambient dimension* m *to an MDP instance over* $\mathbb{F}_q$ *with rank at most* $(1+\varepsilon)n$ *and ambient dimension* $\mathrm{poly}_\varepsilon(m, q)$*.*

Proof. On input $C \in \mathbb{F}_q^{m \times n}$, $\boldsymbol{t} \in \mathbb{F}_q^m$, and $1 \le d \le m$, the reduction sets $n^\dagger := \lfloor (1+\varepsilon)n \rfloor$ and runs the algorithm from Corollary 5.6 to obtain $(C^\dagger, \boldsymbol{t}^\dagger, d^\dagger)$, a $q^{(1-\varepsilon/10)n^\dagger} > 10q^n$-locally dense triple with rank $n^\dagger$. It then runs the reduction from Corollary 5.3 with $k := d$. □

The SETH-hardness of MDP (i.e., the MDP part of Theorem 1.1) now follows immediately from the SETH-hardness of NCP and Corollary 5.7.

5.2 Hardness of approximation via kissing codes

We now prove Gap-ETH hardness of MDP by constructing a different locally dense triple that has $\lambda(C) \le d/\gamma$ for constant $\gamma > 1$ while still achieving $M \ge 2^{\varepsilon n}$ for constant $\varepsilon > 0$. Recall that the kissing number of a code (or, by abuse of notation, the kissing number of its generator matrix) is the number of non-zero vectors whose length is exactly $\lambda(C)$.

Our construction will use the following result of Ashikhmin, Barg, and Vlăduţ, showing the existence of codes over $\mathbb{F}_2$ with kissing number exponential in the ambient dimension m [ABV01]. (In this section, we will only prove the existence of a gadget that suffices for our needs. We do not know if it can be constructed efficiently, which is why we require a non-uniform reduction.) We work over $\mathbb{F}_2$ throughout this section and then simply use Claim 2.3 to extend our hardness result to all fields of characteristic two. (All of the unspecified constants in this section can be made explicit.)

Theorem 5.8 ([ABV01])**.** *There exist constants $\varepsilon > 0$ and $\delta > 0$ such that for all integers $m \ge 2$, there is $C \in \mathbb{F}_2^{m \times n}$ with kissing number at least $2^{\varepsilon m}$ and $\lambda(C) \ge \delta m$.*

The basic idea behind the next corollary is straightforward. Given a code as in Theorem 5.8, we can sample a uniformly random target $\boldsymbol{t} \in \mathbb{F}_2^m$ with low Hamming norm, $\|\boldsymbol{t}\|_H = \delta' m < \delta m$ and argue that the expected number of codewords at distance $d := \lambda(C) - \|\boldsymbol{t}\|_H = \lambda(C)/\gamma$ from $\boldsymbol{t}$ is $2^{\varepsilon' m}$. We then "remove $\mathbf{0}$" from the code to make $\operatorname{dist}(\boldsymbol{t}, C) = d$. I.e., we translate $\boldsymbol{t}$ by a random codeword, take a random low-co-dimension subcode of C, and argue that with high probability the subcode will still contain many vectors at distance d from our new target but no closer codeword.

Corollary 5.9. *There exist constants $\varepsilon > 0$ and $\gamma \in (1,2)$ such that for all sufficiently large integers m, there exist $C \in \mathbb{F}_2^{m \times n}$, $\boldsymbol{t} \in \mathbb{F}_2^m$, and $0 \le d \le m$ that form an M-locally dense triple with $M \ge 2^{\varepsilon m}$ and $\lambda(C) \ge \gamma d$.*

Proof. Let $\widehat{C} \in \mathbb{F}_2^{m \times \widehat{n}}$ generate the code guaranteed by Theorem 5.8 with kissing number at least $2^{\widehat{\varepsilon} m}$. Let $\widehat{\boldsymbol{t}} \in \mathbb{F}_2^m$ be uniformly random with Hamming norm $w := \lambda(\widehat{C}) - \lfloor \lambda(\widehat{C})/\gamma \rfloor$. Let $\boldsymbol{a}, \widehat{\boldsymbol{z}} \in \mathbb{F}_2^{\widehat{n}}$ be uniformly random. Then, set $\boldsymbol{t} := \widehat{\boldsymbol{t}} + \widehat{C}\widehat{\boldsymbol{z}}$, and let $C \in \mathbb{F}_2^{m \times n}$ be the generator matrix of the code

$$\{\widehat{C}\boldsymbol{z} \ : \ \langle \boldsymbol{z}, \boldsymbol{a} \rangle = 0\} .$$

Let $d := \lambda(\widehat{C}) - w = \lfloor \lambda(\widehat{C})/\gamma \rfloor$.

We trivially have $\lambda(C) \ge \lambda(\widehat{C}) \ge \gamma d$. Below, we argue that $(C, \boldsymbol{t}, d)$ is an M-locally dense triple with non-zero probability.

Notice that, in the code generated by $\widehat{C}$, the closest codeword to $\widehat{\boldsymbol{t}}$ is $\mathbf{0}$, and all other codewords are at distance at least d away from $\widehat{\boldsymbol{t}}$. Let $\boldsymbol{y} \in \widehat{C}\mathbb{F}_2^{\widehat{n}}$ be a codeword with $\|\boldsymbol{y}\|_H = \lambda(\widehat{C})$. Then,

$$\Pr[\|\boldsymbol{y} - \widehat{\boldsymbol{t}}\|_H = d] = \frac{\binom{\lambda(\widehat{C})}{w}}{\binom{m}{w}} .$$

So, the expected number of codewords $\boldsymbol{y} \in \widehat{C}\mathbb{F}_2^{\widehat{n}}$ with $\|\boldsymbol{y} - \widehat{\boldsymbol{t}}\|_H = d$ is at least

$$2^{\widehat{\varepsilon} m} \cdot \frac{\binom{\lambda(\widehat{C})}{w}}{\binom{m}{w}} ,$$

which for sufficiently large m and sufficiently small constants ε and γ is at least $2^{\varepsilon m + 2}$.[3] Therefore, there exists a choice of $\widehat{\boldsymbol{t}}$ with at least $2^{\varepsilon m+2}$ codewords $\boldsymbol{y} \in \widehat{C}\mathbb{F}_2^{\widehat{n}}$ at distance d from $\widehat{\boldsymbol{t}}$.

Fix such $\widehat{\boldsymbol{t}}$, and notice that there are also at least $2^{\varepsilon m+2}$ codewords $\boldsymbol{y} \in \widehat{C}\mathbb{F}_2^{\widehat{n}}$ with $\|\boldsymbol{y} - \boldsymbol{t}\|_H = d$. (Recall that we chose $\boldsymbol{t} := \widehat{\boldsymbol{t}} + \widehat{C}\widehat{\boldsymbol{z}}$.) And, the only codeword $\boldsymbol{y} \in \widehat{C}\mathbb{F}_2^{\widehat{n}}$ satisfying $\|\boldsymbol{y} - \boldsymbol{t}\|_H < d$ is $\boldsymbol{y} = \widehat{C}\widehat{\boldsymbol{z}}$. It follows that $\operatorname{dist}(\boldsymbol{t}, C) = d$ if and only if $\langle \boldsymbol{a}, \widehat{\boldsymbol{z}} \rangle \neq 0$. Furthermore, for any $\boldsymbol{z} \neq \widehat{\boldsymbol{z}}$, we have

$$\Pr_{\boldsymbol{a},\widehat{\boldsymbol{z}}}[\langle \boldsymbol{a}, \boldsymbol{z} \rangle = 0 \mid \operatorname{dist}(\boldsymbol{t}, C) = d] \geq \Pr_{\boldsymbol{a},\widehat{\boldsymbol{z}}}[\langle \boldsymbol{a}, \boldsymbol{z} \rangle = 0 \mid \langle \boldsymbol{a}, \widehat{\boldsymbol{z}} \rangle = 1]/2 \geq 1/4 .$$

Therefore, for a choice of $\widehat{\boldsymbol{t}}$ as above, we have

$$\mathop{\mathbb{E}}_{\boldsymbol{a},\widehat{\boldsymbol{z}}}[\#\{\boldsymbol{z} \in \mathbb{F}_2^n \ : \ \|C\boldsymbol{z} - \boldsymbol{t}\|_H = d\} \mid \operatorname{dist}(\boldsymbol{t}, C) = d] \geq 2^{\varepsilon m} .$$

Since this holds in expectation, there must exist $C, \boldsymbol{t}$ such that $(C, \boldsymbol{t}, d)$ is an M-locally dense triple, as needed. □

Corollary 5.10. *For any constants $\gamma > 1$ and $\varepsilon > 0$, there is an efficient (non-uniform) reduction that maps any γ-NCP instance over $\mathbb{F}_2$ with rank n, ambient dimension d, and distance $d \geq \varepsilon n$ to a γ'-MDP instance over $\mathbb{F}_2$ with rank at most αn and ambient dimension at most $\operatorname{poly}_{\gamma,\varepsilon}(m)$ for some constants $\gamma' > 1$ and $\alpha > 1$ depending only on γ and ε.*

Proof. Simply combine Corollary 5.9 with Corollary 5.3. In particular, let $\varepsilon^\dagger, \gamma^\dagger$ be the constants guaranteed by Corollary 5.9. Let $\delta := \gamma^\dagger - 1 > 0$. Let $\alpha := 1 + 1/\varepsilon^\dagger$, and $m^\dagger := \lfloor \alpha n \rfloor$. Then, let $(C^\dagger, \boldsymbol{t}^\dagger, d^\dagger)$ be the M-locally dense triple guaranteed by Corollary 5.9, and notice that the rank of $C^\dagger$ is at most the ambient dimension $m^\dagger$. Also notice that for our choice of parameters $M \geq 2^{\varepsilon^\dagger m^\dagger} > 10 \cdot 2^n$ (for n sufficiently large).

Then, Corollary 5.3 with

$$k := \left\lceil \frac{2d}{\delta d^\dagger} \right\rceil \leq \frac{3\alpha d}{\varepsilon \delta d^\dagger}$$

guarantees a reduction with auxiliary input $(C^\dagger, \boldsymbol{t}^\dagger, d^\dagger)$ that reduces any γ-NCP instance with rank n, ambient dimension m, and distance $d \leq m$ to a γ'-MDP with rank at most $n^\dagger \leq m^\dagger = \lfloor \alpha n \rfloor$ and ambient dimension $m + km^\dagger \leq \operatorname{poly}(d)$, where we may take γ' as large as

$$\frac{\min\left\{k\lambda(C^\dagger),\ \gamma d + kd^\dagger\right\}}{d + kd^\dagger} \geq \min\left\{\frac{k\gamma^\dagger d^\dagger}{d + kd^\dagger},\ \frac{\gamma d + 3\alpha d/(\varepsilon\delta)}{d + 3\alpha d/(\varepsilon\delta)}\right\} \geq \min\left\{\frac{2\gamma}{\gamma + 1},\ \frac{\varepsilon\delta\gamma + 3\alpha}{\varepsilon\delta + 3\alpha}\right\} .$$

This is a constant strictly greater than one, as needed. □

[3] By a suitable effective form of Stirling's formula, we have

$$\frac{\binom{\lambda(\widehat{C})}{w}}{\binom{m}{w}} \geq \frac{\binom{\lfloor \delta m \rfloor}{\lfloor \delta m - \delta m/\gamma \rfloor}}{\binom{m}{\lceil \delta m - \delta m/\gamma \rceil}} \geq \frac{2^{(\delta H(1-1/\gamma) - H(\delta(1-1/\gamma)))m}}{\operatorname{poly}(m)} ,$$

where $H(p) := p \log_2(1/p) + (1-p)\log_2(1/(1-p))$ is the binary entropy function. It then suffices to note that $\lim_{\gamma \to 1} \delta H(1-1/\gamma) - H(\delta(1-1/\gamma)) = 0$. Therefore, there exists $\gamma > 1$ such that, e.g., $\delta H(1-1/\gamma) - H(\delta(1-1/\gamma)) > -\widehat{\varepsilon}/2$. We can therefore take γ to satisfy this and, e.g., $\varepsilon = \widehat{\varepsilon}/4$, so that for sufficiently large m, the ratio in question is at least $2^{\varepsilon m - \widehat{\varepsilon} m + 2}$, as needed.

Theorem 1.2 for $q = 2$ now follows immediately by combining Theorem 4.2 and Corollary 5.10. To extend this to all $q = 2^\ell$, we can simply apply Claim 2.3.

5.3 An exponential-time derandomized reduction

We now show how to derandomize the reduction from [DMS03] presented in Theorem 5.2 at the expense of making it run in exponential time. More specifically, we show a reduction from NCP on codes with rank n to MDP on codes with rank roughly $(1+\varepsilon)n/4$ (and polynomial ambient dimension) that runs in time roughly $q^{3n/4}$. This is enough to show a tight SETH-hardness result for MDP, as in Theorem 1.1.

We first show a minor variant of the reduction above, complete with its own variant of the necessary gadget. We omit the proof as it is essentially identical to that of Theorem 5.2, and quite similar to that of Theorem A.2.

Definition 5.11. *An* augmented code *is a generator matrix $C \in \mathbb{F}_q^{m \times n^\dagger}$, a target $\boldsymbol{t} \in \mathbb{F}_q^m$, distance $1 \le d \le m$ with $d := \operatorname{dist}(\boldsymbol{t}, C) < \lambda(C)$, matrices $T_1, \dots, T_\ell \in \mathbb{F}_q^{n \times n^\dagger}$, and vectors $\boldsymbol{z}_1, \dots, \boldsymbol{z}_\ell$ such that*

$$\left\{T_i \boldsymbol{z}^\dagger + \boldsymbol{z}_i \ : \ \|C\boldsymbol{z}^\dagger - \boldsymbol{t}\|_H = d\right\} = \mathbb{F}_q^n \, .$$

Theorem 5.12. *There is an efficient* deterministic *Cook reduction that takes as auxiliary input an augmented code $(C^\dagger \in \mathbb{F}_q^{m^\dagger \times n^\dagger}, \boldsymbol{t} \in \mathbb{F}_q^{n^\dagger}, d^\dagger \le m^\dagger, T_1, \dots, T_\ell \in \mathbb{F}_q^{n \times n^\dagger}, \boldsymbol{z}_1, \dots, \boldsymbol{z}_\ell \in \mathbb{F}_q^n)$ and reduces any NCP instance over $\mathbb{F}_q$ with rank n, ambient dimension m, and distance d to an MDP instance over $\mathbb{F}_q$ with rank at most $n^\dagger + 1$, ambient dimension $m + dm^\dagger$, and distance $(d^\dagger + 1)d$.*

5.3.1 Constructing the gadget deterministically

Below, we show how to find an augmented code deterministically. Unfortunately, our algorithm for finding these T_i will run in time greater than $q^{n+n^\dagger}$. In Section 5.3.2, we show how to use the algorithm anyway.

Proposition 5.13. *There exists a deterministic algorithm that takes as input $S \subseteq \mathbb{F}_q^n$ and $V \subseteq \mathbb{F}_q^{n^\dagger}$ with $|V| \ge 10n^2 q^n$ and outputs $T \in \mathbb{F}_q^{n \times n^\dagger}$ and $\boldsymbol{z} \in \mathbb{F}_q^n$ such that $|(TV + \boldsymbol{z}) \cap S| \ge \exp(-10n^2 q^n/|V|) \cdot |S|$ in time $|V||S|q^{n^\dagger+1} \cdot \operatorname{poly}(n^\dagger, \log q)$.*

Proof. The algorithm uses the method of conditional expectations and constructs T and $\boldsymbol{z}$ by finding the rows $\boldsymbol{t}_1, \dots, \boldsymbol{t}_n \in \mathbb{F}_q^{n^\dagger}$ of T and the coordinates $z_1, \dots, z_n \in \mathbb{F}_q$ of $\boldsymbol{z}$ one at a time. In particular, for $i = 1, \dots, n$, the algorithm behaves as follows. For each $\widetilde{\boldsymbol{t}} \in \mathbb{F}_q^{n^\dagger}$, $\widetilde{z} \in \mathbb{F}_q$, and $\boldsymbol{s} \in S$, let

$$U_{\boldsymbol{s}, \widetilde{\boldsymbol{t}}, \widetilde{z}} := \{\boldsymbol{v} \in V \ : \ \forall j < i, \ \langle \boldsymbol{v}, \boldsymbol{t}_j \rangle + z_j = s_j \text{ and } \langle \boldsymbol{v}, \widetilde{\boldsymbol{t}} \rangle + \widetilde{z} = s_i\}$$

be the set of all $\boldsymbol{v}$ that are "compatible with" $\boldsymbol{s}$, $\widetilde{\boldsymbol{t}}$, and $\widetilde{z}$. Let

$$E_{\widetilde{\boldsymbol{t}}, \widetilde{z}} := |\{\boldsymbol{s} \in S \ : \ |U_{\boldsymbol{s}, \widetilde{\boldsymbol{t}}, \widetilde{z}}| \ge N_i\}| \, ,$$

for some $1 \le N_i < N_{i-1}/q$ to be chosen later. ($E_{\widetilde{\boldsymbol{t}}, \widetilde{z}}$ is a rough approximation to the expected value of $|(TV + \boldsymbol{z}) \cap S|$ when the first i rows of T are fixed to $\boldsymbol{t}_1, \dots, \boldsymbol{t}_{i-1}, \widetilde{\boldsymbol{t}}$ and the first i coordinates of $\boldsymbol{z}$ are fixed to $z_1, \dots, z_{i-1}, \widetilde{z}$.) The algorithm sets $\boldsymbol{t}_i$ and z_i so that $E_{\boldsymbol{t}_i, z_i}$ is maximized. Finally, the algorithm outputs $T \in \mathbb{F}_q^{n \times n^\dagger}$ whose rows are the $\boldsymbol{t}_i$ and $\boldsymbol{z} := (z_1, \dots, z_n) \in \mathbb{F}_q^n$.

It is clear that the algorithm runs in time $|V||S|q^{n^\dagger+1} \cdot \mathrm{poly}(n^\dagger, \log q)$, as claimed. Let E_i be the maximizing value of $E_{\widetilde{\boldsymbol{t}},\widetilde{z}}$ at the ith step. For convenience, we define $E_0 := |S|$ and $N_0 := |V|$. Notice that $E_n \geq |(TV + \boldsymbol{z}) \cap S|$ for the final output T and $\boldsymbol{z}$, so that it suffices to show that the E_i do not decay too quickly.

We claim that

$$E_i \geq \left(1 - \frac{qN_{i-1}}{(N_{i-1} - qN_i)^2}\right) \cdot E_{i-1} \,. \tag{1}$$

Indeed, let $\boldsymbol{s} \in S$ such that in step $i-1$ we have $|U_{\boldsymbol{s},\boldsymbol{t}_{i-1},z_{i-1}}| \geq N_{i-1}$. Then in step i, by Claim 2.4, we have

$$\Pr_{\widetilde{\boldsymbol{t}},\widetilde{z}}[|U_{\boldsymbol{s},\widetilde{\boldsymbol{t}},\widetilde{z}}| < N_i] < \frac{qN_{i-1}}{(N_{i-1} - qN_i)^2} \,.$$

Therefore,

$$\mathop{\mathbb{E}}_{\widetilde{\boldsymbol{t}},\widetilde{z}}[E_{\widetilde{\boldsymbol{t}},\widetilde{z}}] \geq \left(1 - \frac{qN_{i-1}}{(N_{i-1} - qN_i)^2}\right) \cdot E_{i-1} \,,$$

which implies Eq. (1).

Applying Eq. (1) for all i and recalling that $E_0 := |S|$, we see that

$$E_n \geq |S| \cdot \prod_{i=1}^{n} \left(1 - \frac{qN_{i-1}}{(N_{i-1} - qN_i)^2}\right) .$$

Finally, we simply choose values of N_i that make the above expression relatively easy to analyze. It suffices to take $N_i := |V|/q^i \cdot (n - i + 1)/(n + 1)$. Then, we have

$$\frac{E_n}{|S|} \geq \prod_{i=1}^{n} \left(1 - (n+1)(n-i+2)q^i/|V|\right) \geq \prod_{i=1}^{n} \exp(-5n^2q^i/|V|) \geq \exp(-10n^2q^n/|V|) \,,$$

as needed. □

Corollary 5.14. *There exists a deterministic algorithm that takes as input an integer $n \geq 1$ and an M-locally dense triple $(C^\dagger \in \mathbb{F}_q^{m^\dagger \times n^\dagger}, \boldsymbol{t} \in \mathbb{F}_q^{n^\dagger}, d^\dagger \leq m^\dagger)$ with $M > 10n^2q^n$ and outputs $T_1, \ldots, T_\ell \in \mathbb{F}_q^{n \times n^\dagger}$ and $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_\ell \in \mathbb{F}_q^n$ such that*

$$\bigcup_{i=1}^{\ell} (T_i V + \boldsymbol{z}_i) = \mathbb{F}_q^n$$

in time $q^{n+2n^\dagger+1} \cdot \mathrm{poly}(n^\dagger, \log q, \ell)$, where

$$V := \{\boldsymbol{z}^\dagger \in \mathbb{F}_q^{n^\dagger} \;:\; \|C^\dagger \boldsymbol{z}^\dagger - \boldsymbol{t}^\dagger\|_H = d^\dagger\}$$

is the set of coordinates of closest codewords to $\boldsymbol{t}^\dagger$ and

$$\ell := \left\lceil \frac{n \log q + 1}{\log M - \log(10n^2q^n)} \right\rceil .$$

Proof. On input $n \geq 1$ and $(C^\dagger \in \mathbb{F}_q^{m^\dagger \times n^\dagger}, \boldsymbol{t} \in \mathbb{F}_q^{n^\dagger}, d^\dagger \leq m^\dagger)$, the algorithm behaves as follows. It first computes the set V by simply enumerating all codewords $C^\dagger \mathbb{F}_q^{n^\dagger}$. Let $S_1 := \mathbb{F}_q^n$.

The algorithm does the following for $i = 1, \ldots, \ell$. It runs the procedure from Proposition 5.13 on input S_i and V, receiving as output $T_i, \boldsymbol{z}_i$. It then sets $S_{i+1} := S_i \setminus (T_i V + \boldsymbol{z}_i)$. Finally, it outputs $T_1, \ldots, T_\ell$ and $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_\ell$.

The running time is clear. For correctness, we note that, by Proposition 5.13,

$$|(T_i V + \boldsymbol{z}_i) \cap S_i| \geq \exp(-10n^2 q^n/|V|) \cdot |S_i| \,.$$

Therefore,

$$|S_i| \leq (1 - \exp(-10n^2 q^n/|V|)) \cdot |S_{i-1}| \leq (10n^2 q^n/|V|) \cdot |S_{i-1}| \,.$$

Applying this inequality for all i and recalling that $S_1 = \mathbb{F}_q^n$, we have

$$|S_{\ell+1}| \leq q^n (10n^2 q^n/|V|)^\ell < 1 \,.$$

I.e., $S_{\ell+1}$ is empty. The result follows by noting that $S_{\ell+1} = \emptyset$ is the complement of the union $\bigcup_{i=1}^{\ell} (T_i V + \boldsymbol{z}_i)$. □

Finally, we note that we can combine the locally dense triple construction in Corollary 5.6 with Corollary 5.14 in order to construct an augmented code.

Corollary 5.15. *For every constant $\varepsilon \in (0, 1/2)$, there is a deterministic algorithm that takes as input a prime power $q = q(n) \geq 2$ and sufficiently large integer n and outputs an augmented code consisting of a generator matrix $C \in \mathbb{F}_q^{m \times n^\dagger}$, target $\boldsymbol{t} \in \mathbb{F}_q^m$, distance $d := \lceil (1 - \varepsilon/10)(1 - 1/q)m \rceil$, matrices $T_1, \ldots, T_\ell \in \mathbb{F}_q^{n \times n^\dagger}$, and vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_\ell \in \mathbb{F}_q^n$ in time $q^{n+2n^\dagger+1} \cdot \mathrm{poly}_\varepsilon(n^\dagger, \log q)$, where*

$$\ell := \left\lceil \frac{10n \log q}{\varepsilon n \log(q) - \log(10n^2)} \right\rceil ,$$

and $n^\dagger := \lceil (1 + \varepsilon)n \rceil$.

5.3.2 Completing the reduction

To complete the reduction, we need to somehow obtain a reduction that runs in time less than $q^{(1-\varepsilon)n}$ using the fact that Corollary 5.15 allows us to build an augmented code in time roughly q^{3n}. We do this by guessing most of the coordinates of the potential solution to our NCP instance. E.g., for each of the $q^{3n/4}$ choices for the first $3n/4$ coordinates, we create an NCP instance with rank $n/4$. We then use Corollary 5.15 to construct an augmented code for $n' := n/4$, which we can do in time roughly $q^{3n'} = q^{3n/4}$. Notice that we only need to construct this gadget once, and we can use it to reduce each of the $q^{3n/4}$ NCP instances with rank n' to an MDP instance with rank just slightly larger than n'. In particular, this implies that a $q^{(1-\varepsilon)n'}$-time algorithm for MDP on instances with rank n' would imply a roughly $q^{(1-\varepsilon/4)n}$-time algorithm for NCP on instances with rank n, contradicting SETH.

Theorem 5.16. *For any constant $\varepsilon \in (0, 1/2)$, there is a $q^{3(1+\varepsilon)n/4} \cdot \mathrm{poly}_\varepsilon(n, m, q)$-time reduction that maps any NCP instance with rank n and ambient dimension m to $q^{3n/4}$ instances of MDP with rank $\lceil (1 + \varepsilon)n/4 \rceil + 1$ and ambient dimension $\mathrm{poly}_\varepsilon(n, m)$.*

Proof. On input an NCP instance $C \in \mathbb{F}_q^{m \times n}$, $\boldsymbol{t} \in \mathbb{F}_q^n$, and $1 \leq d \leq m$, the reduction behaves as follows. We assume for simplicity that n is divisible by four and sufficiently large. The reduction first uses the procedure from Corollary 5.15 with input $n/4$ to find an augmented code $C \in \mathbb{F}_q^{m^\dagger \times n^\dagger}$, target $\boldsymbol{t} \in \mathbb{F}_q^{m^\dagger}$, distance $d := \lceil (1-\varepsilon/10)(1-1/q)m^\dagger \rceil$, matrices $T_1, \ldots, T_\ell \in \mathbb{F}_q^{n/4 \times n^\dagger}$, and vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_\ell \in \mathbb{F}_q^{n/4}$ in time $q^{n/4+2n^\dagger+1} \cdot \mathrm{poly}_\varepsilon(n^\dagger, \log q)$, where $\ell \leq \mathrm{poly}_\varepsilon(n, \log q)$ and $n^\dagger := \lceil (1+\varepsilon)n/4 \rceil$.

Let $\widehat{C} \in \mathbb{F}_q^{m \times n/4}$ be the matrix consisting of the last $n/4$ columns of C. Let $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{q^{3n/4}} \in \mathbb{F}_q^m$ be the targets given by $\boldsymbol{t}_i := \boldsymbol{t} - C\boldsymbol{z}_i$, where $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{q^{3n/4}} \in \mathbb{F}_q^n$ are all the distinct vectors whose last $n/4$ coordinates are zero. Then, for each $i = 1, \ldots, q^{3n/4}$, the reduction uses the procedure from Theorem 5.12 together with the augmented code to reduce the NCP instance given by $(\widehat{C}, \boldsymbol{t}_i, d)$ to an MDP instance with the desired parameters. The reduction then calls its MDP oracle on each such MDP instance and outputs YES if the oracle ever responds YES. Otherwise, it outputs NO.

The running time of the reduction is clear. To prove correctness, fix $\boldsymbol{z} \in \mathbb{F}_q^n$. Let $\boldsymbol{z}_i \in \mathbb{F}_q^n$ be the vector whose first $3n/4$ coordinates match $\boldsymbol{z}$ and whose last $n/4$ coordinates are zero, and let $\boldsymbol{z}' \in \mathbb{F}_q^{n/4}$ be the projection of $\boldsymbol{z}$ onto its last $n/4$ coordinates. Then, $C\boldsymbol{z} - \boldsymbol{t} = C\boldsymbol{z}_i + \widehat{C}\boldsymbol{z}' - \boldsymbol{t} = \widehat{C} - \boldsymbol{t}_i$. It follows that $\mathrm{dist}(\boldsymbol{t}_i, \widehat{C}) \geq \mathrm{dist}(\boldsymbol{t}, C)$, which implies that the reduction always correctly outputs NO on a NO instance. Furthermore, by taking $\boldsymbol{z}$ so that $\|C\boldsymbol{z} - \boldsymbol{t}\|_H \leq d$, we see that the reduction always correctly outputs YES when such a $\boldsymbol{z}$ exists. I.e., the reduction is correct as needed. □

Acknowledgments. We thank Ryan Williams for pointing out the work of [Tra08]. We thank Huck Bennett and Sasha Golovnev for helpful discussions. We also thank the anonymous reviewers for their helpful comments. We are particularly thankful to the reviewer who suggested that we use the method of conditional expectations to obtain a deterministic reduction that runs in exponential time, as in Section 5.3.

References

[ABGS19] Divesh Aggarwal, Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. Fine-grained hardness of CVP(P)—Everything that we can prove (and nothing else). http://arxiv.org/abs/1911.02440, 2019. 1, 2, 3, 13

[ABSS97] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. 2

[ABV01] Alexei Ashikhmin, Alexander Barg, and Serge Vlăduţ. Linear codes with exponentially many light vectors. *Journal of Combinatorial Theory, Series A*, 96(2), 2001. 4, 16, 19

[ADS15] Divesh Aggarwal, Daniel Dadush, and Noah Stephens-Davidowitz. Solving the Closest Vector Problem in 2^n time–The discrete Gaussian strikes again! In *FOCS*, 2015. 1

[AK14] Per Austrin and Subhash Khot. A simple deterministic reduction for the gap minimum distance of code problem. *IEEE Trans. Information Theory*, 60(10):6636–6645, 2014. 2, 4

[App17] B. Applebaum. Exponentially-hard Gap-CSP and local PRG via local hardcore functions. In *FOCS*, 2017. 2

[APY09] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. Deterministic approximation algorithms for the nearest codeword problem. In *APPROX*, 2009. 1

[AS18] Divesh Aggarwal and Noah Stephens-Davidowitz. (Gap/S)ETH hardness of SVP. In *STOC*, 2018. 1, 4, 15, 16

[BGS17] Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. On the quantitative hardness of CVP. In *FOCS*, 2017. 1, 2, 3

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In *Eurocrypt*, 2012. 1

[BK02] Piotr Berman and Marek Karpinski. Approximating minimum unsatisfiability of linear equations. In *SODA*, 2002. 1, 2

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: Ball-collision decoding. In *CRYPTO*, 2011. 1

[BMvT78] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Information Theory*, 24(3):384–386, 1978. 1

[Bon84] Arrigo Bonisoli. Every equidistant linear code is a sequence of dual Hamming codes. *ARS Combinatoria*, 18, 1984. 11

[CW10] Qi Cheng and Daqing Wan. Complexity of decoding positive-rate primitive Reed-Solomon codes. *IEEE Trans. Information Theory*, 56(10):5217–5222, 2010. 2

[CW12] Qi Cheng and Daqing Wan. A deterministic reduction for the gap Minimum Distance Problem. *IEEE Trans. Information Theory*, 58(11):6935–6941, 2012. 2, 4, 26, 28

[Din16] Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. https://eccc.weizmann.ac.il/report/2016/128/, 2016. 2, 8, 10

[DKRS03] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003. 2

[DMS03] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Information Theory*, 49(1):22–37, 2003. 1, 2, 4, 15, 21, 26

[IP99] Russell Impagliazzo and Ramamohan Paturi. Complexity of k-SAT. In *CCC*, 1999. 1, 7

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. 1, 5, 7

[KM19] Karthik C. S. and Pasin Manurangsi. On Closest Pair in Euclidean metric: Monochromatic is as hard as bichromatic. In *ITCS*, 2019. 15

[Mic12] Daniele Micciancio. Inapproximability of the Shortest Vector Problem: Toward a deterministic reduction. *Theory of Computing*, 8, 2012. 4

[Mic14] Daniele Micciancio. Locally dense codes. In *CCC*, 2014. 4

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\widetilde{O}(2^{0.054N})$. In *Asiacrypt*, 2011. 1

[MR17] Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense CSPs. In *ICALP*, 2017. 2, 8, 10

[MV10] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In *STOC*, 2010. 1

[Tra08] Patrick Traxler. The time complexity of constraint satisfaction. In *International Conference on Parameterized and Exact Computation*, 2008. 4, 8, 9, 24

[Var97] Alexander Vardy. Algorithmic complexity in coding theory and the minimum distance problem. In *STOC*, 1997. 1

[Wan97] Daqing Wan. Generators and irreducible polynomials over finite fields. *Mathematics of Computation*, 66(219), 1997. 27

A A deterministic reduction in the spirit of Cheng and Wan

We now modify the reduction in Theorem 5.2 to make it deterministic. To do so, we largely follow Cheng and Wan [CW12] (who themselves follow the high-level framework of [DMS03] used in Section 5). We first introduce a strengthening of the gadgets used in the previous section. We will construct this gadget in essentially the same way as Cheng and Wan. The main difference is that we focus on minimizing the rank $n^\dagger$ relative to n, while Cheng and Wan focus on minimizing the ratio $\mathrm{dist}(\boldsymbol{t}, C)/\lambda(C)$.

Definition A.1 (Projecting codes)**.** *For integers $1 \le n < n^\dagger \le m$, we say that a generator matrix $C \in \mathbb{F}_q^{m \times n^\dagger}$, a target $\boldsymbol{t} \in \mathbb{F}_q^m$, and distance $1 \le d \le m$ form a* n-projecting code *if $d = \mathrm{dist}(\boldsymbol{t}, C) < \lambda(C)$, and if for every $\boldsymbol{z} \in \mathbb{F}_q^n$, there is a $\boldsymbol{z}^\dagger \in \mathbb{F}_q^{n^\dagger - n}$ such that $\|C(\boldsymbol{z}, \boldsymbol{z}^\dagger) - \boldsymbol{t}\|_H = d$.*

I.e., let S be the set of $\boldsymbol{z} \in \mathbb{F}_q^{n^\dagger}$ with $\|C\boldsymbol{z} - \boldsymbol{t}\|_H = d$ from $\boldsymbol{t}$. Then, $(C, \boldsymbol{t}, d)$ is a projecting code if and only if $\pi(S) = \mathbb{F}_q^n$, where π is the map that projects a vector onto its first n coordinates. The next theorem shows how such a gadget can be used to deterministically reduce NCP to MDP.

Theorem A.2. *There is an efficient* deterministic *reduction that takes as auxiliary input an n-projecting code $(C^\dagger \in \mathbb{F}_q^{m^\dagger \times n^\dagger}, \boldsymbol{t}^\dagger \in \mathbb{F}_q^{m^\dagger}, 0 \le d^\dagger \le m^\dagger)$ and reduces any NCP instance over $\mathbb{F}_q$ with rank n and ambient dimension m to an MDP instance over $\mathbb{F}_q$ with rank $n^\dagger + 1$ and ambient dimension $m(m^\dagger + 1)$.*

Proof. Given input $\boldsymbol{t} \in \mathbb{F}_q^m$, $C \in \mathbb{F}_q^{m\times n}$, $1 \le d \le m$, and an n-projecting code $(C^\dagger \in \mathbb{F}_q^{m^\dagger \times n^\dagger}, \boldsymbol{t}^\dagger \in \mathbb{F}_q^{m^\dagger}, 0 \le d^\dagger \le m^\dagger)$, the reduction first constructs the matrix

$$C_m^\dagger := \begin{pmatrix} C^\dagger \\ C^\dagger \\ \vdots \\ C^\dagger \end{pmatrix} \in \mathbb{F}_q^{mm^\dagger \times n^\dagger}$$

and target $\boldsymbol{t}_m^\dagger := (\boldsymbol{t}^\dagger, \boldsymbol{t}^\dagger, \ldots, \boldsymbol{t}^\dagger) \in \mathbb{F}_q^{mm^\dagger}$—i.e., it stacks m copies of $C^\dagger$ and $\boldsymbol{t}^\dagger$ vertically. Notice that $(C_m^\dagger, \boldsymbol{t}_m^\dagger, md^\dagger)$ is also an n-projecting code. Let $C_0 := (C, 0) \in \mathbb{F}_q^{m\times n^\dagger}$ be the matrix C padded with with $n^\dagger - n$ columns of zeros. (Notice that we must have $n^\dagger \ge n$, so that this definition makes sense.) The reduction simply constructs the generator matrix

$$C' := \begin{pmatrix} C_0 & -\boldsymbol{t} \\ C_m^\dagger & -\boldsymbol{t}_m^\dagger \end{pmatrix} \in \mathbb{F}_q^{m(m^\dagger+1)\times(n^\dagger+1)} .$$

Finally, it simply calls its MDP oracle on input C' and $d' := d + md^\dagger$ and returns the result.

Clearly the reduction is efficient and achieves the parameters claimed in the theorem. To prove correctness, it will be convenient to define $W := \{(C_0(\boldsymbol{z}, \boldsymbol{z}^\dagger), C_m^\dagger(\boldsymbol{z}, \boldsymbol{z}^\dagger)) \ : \ \boldsymbol{z} \in \mathbb{F}_q^n,\ \boldsymbol{z}^\dagger \in \mathbb{F}_q^{n^\dagger - n}\}$ to be the subspace generated by its first $n^\dagger$ columns of C'. Then, the code generated by C' is just $W \cup (W - (\boldsymbol{t}, \boldsymbol{t}_m^\dagger)\mathbb{F}_q^*)$. Notice that the shortest non-zero codeword in W has length at least $\lambda(C_m^\dagger) = m\lambda(C^\dagger)$, and the shortest codeword in $W - (\boldsymbol{t}, \boldsymbol{t}_m^\dagger)\mathbb{F}_q^*$ has length at least $\max_{\alpha\in\mathbb{F}_q^*} \mathrm{dist}(\alpha\boldsymbol{t}, C) + \mathrm{dist}(\alpha\boldsymbol{t}_m^\dagger, C_m^\dagger) = \mathrm{dist}(\boldsymbol{t}, C) + md^\dagger$.

So, suppose $\mathrm{dist}(\boldsymbol{t}, C) > d$. It follows that

$$\lambda(C') > \min\left\{m\lambda(C^\dagger),\ d + md^\dagger\right\} \ge d' ,$$

where we have used the fact that $m\lambda(C^\dagger) \ge m(1 + d^\dagger) \ge d + md^\dagger$. I.e., the MDP oracle must return NO.

On the other hand, suppose that $\mathrm{dist}(\boldsymbol{t}, C) \le d$. Then, let $\boldsymbol{z} \in \mathbb{F}_q^n$ be such that $\|C\boldsymbol{z} - \boldsymbol{t}\|_H \le d$. By the definition of a projecting code, there exists $\boldsymbol{z}^\dagger \in \mathbb{F}_q^{n^\dagger - n}$ such that $\|C^\dagger(\boldsymbol{z}, \boldsymbol{z}^\dagger) - \boldsymbol{t}^\dagger\|_H = d^\dagger$. Then, clearly $(C(\boldsymbol{z}, \boldsymbol{z}^\dagger), C_m^\dagger(\boldsymbol{z}, \boldsymbol{z}^\dagger)) - (\boldsymbol{t}, \boldsymbol{t}_m^\dagger)$ is a non-zero codeword in the code generated by C' with weight $\|C\boldsymbol{z} - \boldsymbol{t}\|_H + md^\dagger \le d'$. So, the MDP oracle must return YES. □

A.1 Constructing the gadget using Reed-Solomon codes and character sums

We will need a version of Weil's character sum bound for the affine line. Recall that a character χ of some group A is a homomorphism $\chi : A \to \mathbb{T}$ from A to the multiplicative group of complex numbers with norm one. For a polynomial $h(x)$ over $\mathbb{F}_q$, we write $(\mathbb{F}_q[x]/h(x))^*$ for the multiplicative group of units in $\mathbb{F}_q[x]/h(x)$. I.e., the elements of $(\mathbb{F}_q[x]/h(x))^*$ are residue classes of polynomials that are coprime to $h(x)$.

Theorem A.3 (Weil's character sum bound for the affine line, [Wan97, Theorem 2.1])**.** *For any polynomial $h(x) \in \mathbb{F}_q[x]$ and any non-constant character χ of $(\mathbb{F}_q[x]/h(x))^*$,*

$$\left| \sum_{v\in\mathbb{F}_q,\ h(v)\neq 0} \chi(x - v) \right| \le (\deg h - 1)\sqrt{q} .$$

From this, we derive the following result, which is a slight variant of [CW12, Theorem 2.2]. The proof is essentially identical, but we reproduce it for completeness.

Theorem A.4. *For any $h(x) \in \mathbb{F}_q[x]$ with $d := \deg(h) > 1$, any integer g satisfying $q > (g+d)^2$ and $g > (2+\varepsilon)d$, and any element $\beta \in (\mathbb{F}_q[x]/h(x))^*$, there exist distinct $v_1, \ldots, v_g \in \mathbb{F}_q$ such that*

$$\beta = (x - v_1)(x - v_2) \cdots (x - v_g) \bmod h(x) \ ,$$

where

$$\varepsilon := 20 \cdot \frac{\log d + \log(q)/d}{\log q - 2\log d} \ .$$

Proof. Let $\phi(h) := |(\mathbb{F}_q[x]/h(x))^*| < g^d$ denote the cardinality of $(\mathbb{F}_q[x]/h(x))^*$. Let G be the group of characters of $(\mathbb{F}_q[x]/h(x))^*$, and let $\chi_0 \in G$ be the unique constant character satisfying $\chi_0(\alpha) = 1$ for all $\alpha \in (\mathbb{F}_q[x]/h(x))^*$. We recall that $|G| = \phi(h)$; for any $\alpha \in (\mathbb{F}_q[x]/h(x))^*$,

$$\sum_{\chi \in G} \chi(\alpha) = \begin{cases} \phi(h) & \alpha = 1 \bmod h(x) \\ 0 & \text{otherwise;} \end{cases}$$

and similarly for any $\chi \in G$,

$$\sum_{\alpha \in (\mathbb{F}_q[x]/h(x))^*} \chi(\alpha) = \begin{cases} \phi(h) & \chi = \chi_0 \\ 0 & \text{otherwise.} \end{cases}$$

Notice that for $v \in \mathbb{F}_q$, $x - v \in (\mathbb{F}_q[x]/h(x))^*$ if and only if $h(v) \neq 0$. Let $S \subseteq \mathbb{F}_q$ be the set of all such elements, and let $S_m \subset S^m$ be the set of all m-tuples $(v_1, \ldots, v_m) \in S^m$ with $v_i \neq v_j$ for $i \neq j$. Let $N_m(\beta)$ be the number of such tuples $(v_1, \ldots, v_m) \in S_m$ with $\beta = (x - v_1)(x - v_2) \cdots (x - v_m) \bmod h(x)$. We wish to show that $N_g(\beta) > 0$.

Let $\chi(\boldsymbol{v}) := \chi(x - v_1)\chi(x - v_2) \cdots \chi(x - v_m)$ for $\boldsymbol{v} = (v_1, \ldots, v_m) \in S^m$. By the above, we can write

$$\begin{aligned}
N_g(\beta) &= \frac{1}{\phi(h)} \cdot \sum_{\boldsymbol{v} \in S_g} \sum_{\chi \in G} \chi(\beta^{-1} \cdot (x - v_1)(x - v_2) \cdots (x - v_g)) \\
&= \frac{1}{\phi(h)} \cdot \sum_{\boldsymbol{v} \in S^g} \sum_{\chi \in G} \chi(\beta^{-1})\chi(\boldsymbol{v}) - \frac{1}{\phi(h)} \cdot \sum_{\boldsymbol{v} \in S^g \setminus S_g} \sum_{\chi \in G} \chi(\beta^{-1})\chi(\boldsymbol{v}) \\
&\geq \frac{1}{\phi(h)} \cdot \sum_{\boldsymbol{v} \in S^g} \sum_{\chi \in G} \chi(\beta^{-1})\chi(\boldsymbol{v}) - \frac{1}{\phi(h)} \cdot \binom{g}{2} \cdot \sum_{\substack{\boldsymbol{v} \in S^g \\ v_1 = v_2}} \sum_{\chi \in G} \chi(\beta^{-1})\chi(\boldsymbol{v}) \\
&= \frac{|S|^g - \binom{g}{2}|S|^{g-1}}{\phi(h)} + \frac{1}{\phi(h)} \cdot \sum_{\chi \in G \setminus \{\chi_0\}} \chi(\beta^{-1}) \Big(\sum_{\boldsymbol{v} \in S^g} \chi(\boldsymbol{v}) - \binom{g}{2} \cdot \sum_{\substack{\boldsymbol{v} \in S^g \\ v_1 = v_2}} \chi(\boldsymbol{v}) \Big) \\
&\geq (q-d)^{g-1} \cdot \frac{q - d - g^2}{q^d} - \max_{\chi \neq \chi_0} \Big| \sum_{\boldsymbol{v} \in S^g} \chi(\boldsymbol{v}) \Big| - g^2 \cdot \max_{\chi \neq \chi_0} \Big| \sum_{\boldsymbol{v} \in S^{g-1}} \chi(\boldsymbol{v}) \Big| \\
&= (q-d)^{g-1} \cdot \frac{q - d - g^2}{q^d} - \max_{\chi \neq \chi_0} \Big| \sum_{v \in S} \chi(\boldsymbol{v}) \Big|^g - g^2 \cdot \max_{\chi \neq \chi_0} \Big| \sum_{v \in S} \chi(\boldsymbol{v}) \Big|^{g-1} \\
&\geq (q-d)^{g-1} \cdot \frac{q - d - \binom{g}{2}}{q^d} - \big(1 + g^2\big) \cdot (d-1)^g q^{g/2} \ ,
\end{aligned}$$

where the first inequality follows from the fact that the inner summand is non-negative and the last inequality is Theorem A.3. The result now follows from the constraints on g, q, and d, which imply that the above quantity is positive. □

Proposition A.5. *Let $h(x) \in \mathbb{F}_q[x]$ be an irreducible polynomial with $d := \deg(h) > 1$, let $x_1, \ldots, x_n \in \mathbb{F}_q$ be distinct field elements, and let $1 \le n \le g$ be integers satisfying $q > (g+n+d)^2$ and $g > (2+\varepsilon)(n+d)$, where*

$$\varepsilon := 20\frac{\log(n+d) + \log(q)/(n+d)}{\log q - 2\log(n+d)}$$

Then, for every $z_1, \ldots, z_n \in \mathbb{F}_q$, there exists a polynomial $p(x)$ such that $p(x_i) = z_i$ for $1 \le i \le n$ and $p(x)h(x) - 1 = \prod_j (x - v_j)$ for some distinct $v_1, \ldots, v_g \in \mathbb{F}_q$.

Proof. We will have to treat the z_i slightly differently depending on whether $z_i h(x_i) = 1$. By reordering the x_i and z_i, we may assume for convenience that $z_i h(x_i) \neq 1$ for all $i \le \ell$ and $z_i h(x_i) = 1$ for all $i > \ell$, for some $0 \le \ell \le n$.

Let $\alpha(x) \in \mathbb{F}_q[x]$ be the unique polynomial with degree at most n such that $\alpha(x_i) = z_i$ for all i. Let $\pi_1(x) := \prod_{i=1}^{\ell}(x - x_i)$, $\pi_2 := \prod_{i=\ell+1}^{n}(x - x_i)$, and $\widetilde{h}(x) := h(x)\pi_1(x)$. Notice that $\pi_2(x)$ is relatively prime to $\widetilde{h}(x)$, so that $\pi_2(x)$ has an inverse modulo $\widetilde{h}(x)$, which we simply write as $\pi_2^{-1}(x)$. Furthermore, $\alpha(x)h(x) - 1$ is relatively prime to $\widetilde{h}(x)$. (Here, we have used the fact that $\alpha(x_i)h(x_i) - 1 \neq 0$ for all $i \le \ell$.) Therefore, Theorem A.4 guarantees the existence of $v_1, \ldots, v_{g+\ell-n}$ with

$$\prod_{j=1}^{g+\ell-n} (x - v_j) = \pi_2^{-1}(x) \cdot (\alpha(x)h(x) - 1) \mod \widetilde{h}(x) .$$

I.e., there exists a $r(x)$ such that

$$\pi_2(x)\prod_{j=1}^{\ell+d}(x - v_j) = \alpha(x)h(x) - 1 + r(x)h(x)\pi_1(x) = (\alpha(x) + r(x)\pi_1(x))h(x) - 1 .$$

The result follows by taking $p(x) := \alpha(x) + r(x)\pi_1(x)$, which must have degree exactly $\ell + \deg(\pi_2) = g$. □

We are now ready to show the existence of our gadget for large enough q. The existence for all q will follow.

Corollary A.6. *There exists a deterministic $\mathrm{poly}(q)$-time algorithm that takes as input a sufficiently large positive integer and prime power $q \ge 10n^2$ and outputs $C \in \mathbb{F}_q^{q \times n^\dagger}$, $\boldsymbol{t} \in \mathbb{F}_q^q$, and $d := q - n^\dagger - 1$ that form an n-projecting code, where $n^\dagger := \lceil (2+\varepsilon)n \rceil$ with*

$$\varepsilon := 40\frac{\log n + \log(q)/n}{\log q - 2\log n} .$$

Proof. Our code is simply a Reed-Solomon code of rank $n^\dagger$ over $\mathbb{F}_q$ with an appropriate generator matrix. Recall that codewords in such a code have the form $(p(x_1), p(x_2), \ldots, p(x_q)) \in \mathbb{F}_q^q$, where $p(x) \in \mathbb{F}_q[x]$ is a polynomial with degree less than $n^\dagger$ and $x_1, \ldots, x_q \in \mathbb{F}_q$ are all distinct field

elements. We choose our generator matrix $C \in \mathbb{F}_q^{q \times n^\dagger}$ so that for $\boldsymbol{z} = (z_1, \ldots, z_{n^\dagger}) \in \mathbb{F}_q^{n^\dagger}$, $C\boldsymbol{z} = (z_1, z_2, \ldots, z_n, y_1, \ldots, y_{q-n})$. I.e., the first n rows of C are the identity matrix. (Such a matrix C can be computed efficiently by Gaussian elimination.) We of course have $\lambda(C) = q - n^\dagger + 1$.

Let $h(x) \in \mathbb{F}_q[x]$ be an irreducible polynomial of degree two (which can be found efficiently). Let $\boldsymbol{t} := (1/h(x_1), 1/h(x_2), \ldots, 1/h(x_q)) \in \mathbb{F}_q^q$. (Notice that $h(x_i) \neq 0$ so that these inverses exist.) For a codeword $\boldsymbol{a} \in \mathbb{F}_q^q$ corresponding to the polynomial $p_{\boldsymbol{a}}(x) \in \mathbb{F}_q[x]$, the distance $\|\boldsymbol{a} - \boldsymbol{t}\|_H$ is exactly $q - g$, where g is the number of distinct roots of the polynomial $p_{\boldsymbol{a}}(x)h(x) - 1$. Therefore, it suffices to show that for every $\boldsymbol{z} \in \mathbb{F}_q^n$, there exists a polynomial $p(x)$ with degree less than $n^\dagger$ such that $p(x_i) = z_i$ for $1 \leq i \leq n$ and $p(x)h(x) - 1$ splits completely (i.e., $p(x)h(x) - 1$ has $n^\dagger + 1$ distinct roots). This is what Proposition A.5 guarantees. □

The following corollary extends the above result to all prime powers $q \geq 2$ by working over a field extension $\mathbb{F}_{q^\kappa}$. We omit the proof as it is essentially identical to the proof of Corollary 5.5.

Corollary A.7. *There exists a deterministic* $\mathrm{poly}(\widehat{n}, q^\kappa)$*-time algorithm that takes as input a sufficiently large integer* $\widehat{n}$*, prime power* $q \geq 2$*, and integer* $\kappa \geq 1$ *with* $q^\kappa \geq 10\widehat{n}^2$ *and outputs* $C \in \mathbb{F}_q^{m \times n^\dagger}$*,* $\boldsymbol{t} \in \mathbb{F}_q^m$*, and* $d := \kappa(q^\kappa - q^{\kappa-1})(q^\kappa - n^\dagger - 1)$ *that form a* n*-projecting code, where* $n := \kappa\widehat{n}$*,* $m := \kappa(q^\kappa - 1)q^\kappa$*,* $n^\dagger := \kappa\lceil(2+\varepsilon)\widehat{n}\rceil$*, and*

$$\varepsilon := 40\frac{\log \widehat{n} + \kappa \log(q)/\widehat{n}}{\kappa \log q - 2\log \widehat{n}} \,.$$

Corollary A.8. *For any constant* $\varepsilon \in (0,1)$ *and any prime power* $q = q(n) \geq 2$*, there is a* $\mathrm{poly}_\varepsilon(m,q)$*-time deterministic reduction that maps any NCP instance over* $\mathbb{F}_q$ *with rank* n *and ambient dimension* m *to a MDP instance over* $\mathbb{F}_q$ *with rank at most* $(2+\varepsilon)n$ *and dimension at most* $\mathrm{poly}_\varepsilon(m,q)$*.*

Proof. By Theorem A.2, it suffices to show a deterministic $\mathrm{poly}_\varepsilon(m,q)$-time algorithm that constructs an n-projecting code with rank $n^\dagger \leq (2+\varepsilon)n$.

Let $\kappa := 100\lceil(1 + \log_q(n))/\varepsilon\rceil$, $\widehat{n} := \lceil n/\kappa \rceil$, and

$$\varepsilon^\dagger := 40\frac{\log \widehat{n} + \kappa \log(q)/\widehat{n}}{\kappa \log q - 2\log \widehat{n}} < \varepsilon/2 \,,$$

where the inequality holds for sufficiently large n. Then, by Corollary A.7, there is a $\mathrm{poly}(n, q^\kappa) \leq \mathrm{poly}_\varepsilon(n,q)$-time deterministic algorithm that constructs an n-projecting code with rank $n^\dagger := \kappa\lceil(2+\varepsilon^\dagger)\widehat{n}\rceil \leq (2+\varepsilon)n$ for sufficiently large n, as needed. (Formally, Corollary A.7 gives a $(\kappa\widehat{n})$-projecting code, but $\kappa\widehat{n} \geq n$ and an n'-projecting code is also an n-projecting code for any $n' \geq n$.) □

The deterministic hardness of MDP stated in Theorem 1.1 follows immediately from Corollary A.8.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il