Learning and Testing Convex Functions

Renato Ferreira Pinto Jr.
Columbia University, New York, NY, USA

Cassandra Marcussen
Harvard University, Cambridge, MA, USA

Elchanan Mossel
MIT, Cambridge, MA, USA

Shivam Nadimpalli
MIT, Cambridge, MA, USA

Abstract

We consider the problems of *learning* and *testing* real-valued convex functions over Gaussian space. Despite the extensive study of function convexity across mathematics, statistics, and computer science, its learnability and testability have largely been examined only in discrete or restricted settings – typically with respect to the Hamming distance, which is ill-suited for real-valued functions.

In contrast, we study these problems in high dimensions under the standard Gaussian measure, assuming sample access to the function and a mild smoothness condition, namely Lipschitzness. A smoothness assumption is natural and, in fact, necessary even in one dimension: without it, convexity cannot be inferred from finitely many samples. As our main results, we give:

- Learning Convex Functions: An agnostic proper learning algorithm for Lipschitz convex functions that achieves error ε using $n^{O(1/\varepsilon^2)}$ samples, together with a complementary lower bound of $n^{\mathrm{poly}(1/\varepsilon)}$ samples in the *correlational statistical query (CSQ)* model.
- Testing Convex Functions: A tolerant (two-sided) tester for convexity of Lipschitz functions with the same sample complexity (as a corollary of our learning result), and a one-sided tester (which never rejects convex functions) using $O(\sqrt{n}/\varepsilon)^n$ samples.

2012 ACM Subject Classification Theory of computation → Sample complexity and generalization bounds; Theory of computation → Models of learning; Computing methodologies → Supervised learning by regression; Theory of computation → Convex optimization

Keywords and phrases Property testing, learning theory, Gaussian distribution, convex functions

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.62

Category RANDOM

Related Version *Full Version*: `https://arxiv.org/abs/2511.11498`

Funding *Renato Ferreira Pinto Jr.*: Supported by an NSERC Postdoctoral Fellowship and by NSF grant CCF-2106429.
Cassandra Marcussen: This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-23-F-0014 (NDSEG Fellowship). Supported in part by NSF Award 2152413 and a Simons Investigator Award to Madhu Sudan.
Elchanan Mossel: Supported by ARO MURI N000142412742, by NSF grant DMS-2031883, by Vannevar Bush Faculty Fellowship ONR-N00014-20-1-2826 and by a Simons Investigator Award.

Acknowledgements The authors would like to thank Ramon van Handel for several enjoyable discussions concerning Question 5. E.M. would like to thank Dan Mikulincer for interesting discussions.

1 Introduction

Few mathematical ideas are as pervasive or as powerful as convexity: recall that a function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if for every $x, y \in \mathbb{R}^n$ and $\lambda \in [0, 1]$, we have $f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$. The mathematics of convex functions has been intensively studied for over a century, and continues to be central to a number of theoretical and applied disciplines ranging from mathematical analysis to machine learning [68, 15]. Within mathematical programming, for example, convexity of the objective function often governs computational tractability; to quote Rockafellar [69], "the great watershed in optimization isn't between linearity and nonlinearity, but convexity and non-convexity."

This centrality of convexity motivates the following basic questions, which form the focus of this work:

How many samples are needed to *learn* a convex function?

How many samples are needed to *test* if a function $f : \mathbb{R}^n \to \mathbb{R}$ is essentially convex?

We study each of these questions through the lens of theoretical computer science, viewing the former as a problem in *learning theory* [48] and the latter in *property testing* [36, 7]. For these problems to be well-defined, we must first specify a distance metric between functions on $\mathbb{R}^n$. Throughout this work, we employ the $L^2(\gamma)$ distance where $\gamma = \gamma_n$ denotes the measure corresponding to the n-dimensional standard Gaussian distribution $N(0, I_n)$.

Unsurprisingly, given their fundamental nature, both problems have been studied previously:

- There is work in statistics [72, 40, 41, 61, 32] that deals with optimal rates for convex regression where the perspective is that the dimension is fixed, and the number of samples is very large (could be more than exponential) and the question is to study the error rate as the number of samples goes to infinity. This is different than the standard perspective in learning theory where we are interested to more exactly quantify the number of samples needed as a function of the dimension.
- Turning to the problem of *testing* convexity, prior work has largely concerned discrete or highly constrained settings. Much of this literature [65, 6, 63, 5, 4, 54] focuses on testing functions over finite domains or ranges under the Hamming distance, reflecting the field's early connections to complexity theory [13, 71]. A few works such as [6] extend this line of work to real-valued functions under L^p metrics, but still assume bounded range and full query access.

We refer the reader to Section 1.2 for a detailed discussion of related work on both testing and learning convex functions.

In this work, we study both *agnostic proper learning* and *(tolerant) testing* of convexity in high dimensions, in the statistically natural setting where one only has access to *samples* labeled by an unknown function f. The query-access model used extensively in property testing, while natural and powerful for applications to complexity theory [13, 71], is ill-suited to statistical problems. By contrast, sample-based access is the standard framework in settings such as empirical risk minimization, convex regression, and stochastic optimization, where functions are observed only through (possibly noisy) random samples rather than arbitrary queries [67, 51, 44, 14, 73, 39].

Before proceeding, we note that some smoothness assumption is essential for both problems: without any regularity, convexity cannot be meaningfully inferred from finitely many samples, even in one dimension. Accordingly, we restrict attention to *Lipschitz* functions – a natural and mild assumption also adopted in prior work on convex regression [61].

1.1 Our Results

We begin by introducing notation that will be used throughout the paper. Given a function $f \in L^2(\gamma)$, we will write $\mathrm{d}^L_{\mathrm{conv}}(f)$ for the $L^2(\gamma)$ distance of f to its nearest convex L-Lipschitz function. More formally,

$$\mathrm{d}^L_{\mathrm{conv}}(f) := \inf_{\substack{g:\mathbb{R}^n\to\mathbb{R} \\ g \text{ convex, } L\text{-Lipschitz}}} \|f-g\|_{L^2(\gamma)}\,,$$

where

$$\|f-g\|_{L^2(\gamma)} := \mathop{\mathbf{E}}_{\boldsymbol{x}\sim N(0,I_n)}\left[(f(\boldsymbol{x})-g(\boldsymbol{x}))^2\right]^{1/2}.$$

The $L^2(\gamma)$ metric is a standard choice in learning theory, property testing, and derandomization (see, for example, [52, 60, 78, 43, 53, 20, 22, 28, 23, 49, 26, 27], among many others).

Although we focus on learning and testing convex functions over Gaussian space, all of our techniques and results extend naturally to functions over $[0,1]^n$ under the uniform measure. We adopt the Gaussian setting as it is a more canonical model for studying real valued learning and testing problems.

1.1.1 Agnostic Proper Learning and Sample-Based Testing of Convex Functions

We construct an agnostic proper learning algorithm for Lipschitz convex functions with the following guarantees:

▶ **Theorem 1** (Agnostic proper learning of Lipschitz convex functions). *Let $\varepsilon, L > 0$ and suppose $f:\mathbb{R}^n\to\mathbb{R}$ is a L-Lipschitz function. There exists an algorithm which, given i.i.d. access to labeled samples $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x}\sim N(0,I_n)$, draws $n^{O(L^2/\varepsilon^2)}$ samples, runs in time $\exp(\widetilde{O}(nL^2/\varepsilon^2))$, and outputs a function $g\in L^2(\gamma)$ such that*

$$\|f-g\|_{L^2(\gamma)} \le \mathrm{d}^L_{\mathrm{conv}}(f)+\varepsilon\,.$$

Additionally, the function g is convex and L-Lipschitz.

Note that our learning algorithm is agnostic in the sense that it produces a nearly optimal (convex) hypothesis even if the input function f is arbitrarily far from convex, as long as it satisfies the Lipschitz condition.

Using the proof of the Theorem above and the "testing-via-learning" paradigm [37], we obtain a *tolerant* testing algorithm for Lipschitz convex functions:

▶ **Theorem 2** (Tolerant two-sided testing of Lipschitz convex functions). *Let $\varepsilon, \varepsilon_0, L \ge 0$ and suppose $f:\mathbb{R}^n\to\mathbb{R}$ is L-Lipschitz. There is an algorithm which, given i.i.d. labeled samples $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x}\sim N(0,I_n)$, draws $n^{O(L^2/\varepsilon^2)}$ samples, runs in time $\exp(\widetilde{O}(nL^2/\varepsilon^2))$, and has the following performance guarantee:*

- *If $\mathrm{d}^L_{\mathrm{conv}}(f)\le\varepsilon_0$, then it outputs "accept" with probability $9/10$;*
- *If $\mathrm{d}^L_{\mathrm{conv}}(f)\ge\varepsilon+\varepsilon_0$, then it outputs "reject" with probability $9/10$.*

In particular, setting $\varepsilon_0 = 0$ shows that for every fixed $\varepsilon > 0$, there exists an (essentially) $\exp(n)$-time algorithm that distinguishes Lipschitz convex functions from those that are ε-far (in $L^2(\gamma)$) from all Lipschitz convex functions. The "testing-via-learning" approach underlies

the best known sample-based testing algorithms for several other function classes including convex subsets of $\mathbb{R}^n$ [52, 20], monotone Boolean functions [17, 56, 58], and k-monotone Boolean functions [8].

Our proof of Theorem 1 proceeds in two steps:

1. **Low-Degree Approximation.** We begin with a standard observation (see, for example, Theorem 4.1 of [2] or alternatively [46, 23]) that every L-Lipschitz function over Gaussian space is well-approximated in L_2 by a $O(L^2)$-degree polynomial. This allows us to learn such an approximation $\widetilde{f}$ from random samples via polynomial regression.
2. **Convex Regression.** We then *convexify* the learned polynomial $\widetilde{f}$ by projecting it onto the space of convex functions via convex regression [61]. This step is essential to obtain a *proper* learner – one whose output lies within the target class of convex L-Lipschitz functions. For our tolerant testing result, this is essential: the "testing via learning" reduction requires the learner to be proper.

Although the first step is conceptually straightforward, it has an independent implication:

> Any $O(1)$-Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$ can be learned – and hence tested – from $n^{O(1/\varepsilon^2)}$ random samples in an information-theoretic sense.

Note, however, that this guarantee is purely information-theoretic and does not yield a computationally efficient learning algorithm for the function class. In general, however, fitting a function from the target class to the observed samples may be computationally intractable. The main conceptual contribution of Theorem 1 is that we may combine standard low-degree learning with tools specific to convexity to obtain a *proper* learning algorithm.

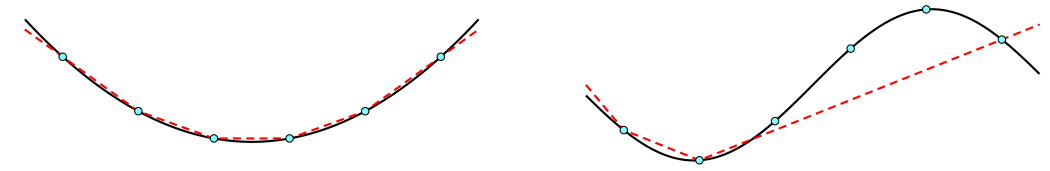
Accordingly, the second step above introduces the main technical challenge: ensuring that the projection onto convex functions can be performed in finite (albeit exponential) time while preserving the $L^2(\gamma)$ error bound. Using queries to the learned low-degree approximation $\widetilde{f}$ on a sufficiently fine grid, we learn via convex regression a function $\widetilde{g}$ (namely a supremum of affine functions) which approximates $\widetilde{f}$ well on a discrete grid. Following [61], quadratic constraints in the convex regression ensure that $\widetilde{g}$ is L-Lipschitz. However, convex regression says nothing about the quality of the approximation over (continuous) Gaussian space, and the low-degree $\widetilde{f}$ need not inherit the Lipschitz behavior of f. To bridge this gap, we draw on Hermite analysis to show that the low-degree approximation $\widetilde{f}$ is L'-Lipschitz for some $L' \gg L$ over a large box in $\mathbb{R}^n$, where both L' and the size of the box can be controlled in terms of n, L and ε. This lets us recover an $L^2(\gamma)$ approximation guarantee for the convex $\widetilde{g}$ against the original target function f.

1.1.2 Lower Bounds for Learning Convex Functions

We complement this upper bound with corresponding *lower bounds* for learning convex functions. In the information-theoretic setting, we show that learning an arbitrary linear function to constant accuracy requires $\Omega(n)$ queries, not just samples; see the full version. We further strengthen this by proving a quantitatively sharper bound for algorithms in the *correlational statistical query* (CSQ) model:

▶ **Theorem 3** (CSQ lower bound; informal, see full version)**.** *Suppose algorithm A, given access to L-Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$ via correlational statistical queries, outputs a function $g \in L^2(\gamma)$ satisfying $\|f - g\|_{L^2(\gamma)} \le \mathrm{d}^L_{\mathrm{conv}}(f) + \varepsilon$. Then A requires $n^{\mathrm{poly}(L/\varepsilon)}$ correlational statistical queries with tolerance $n^{-\mathrm{poly}(L/\varepsilon)}$.*

CSQ algorithms only have access to the input f via approximate inner product queries of the form $\mathbf{E}_{\boldsymbol{z}\sim N(0,I_n)}[f(\boldsymbol{z})h(\boldsymbol{z})]$, for query h chosen by the algorithm; we formally define the CSQ model in the full version. This model captures, for example, gradient descent



(a) (b)

Figure 1 One-dimensional illustration of the empirical convex envelope (red, dashed) of (a) a convex function and (b) a far-from-convex function, with respect to the sampled points shown in cyan. (See Definition 19 for a precise definition).

with respect to squared loss, and the algorithm attaining our $n^{O(L^2/\varepsilon^2)}$ upper bound from Theorem 1 via low-degree approximation is a CSQ algorithm. Thus, Theorem 3 poses a barrier to improving upon the low-degree learning approach: any agnostic learning algorithm that fits the CSQ model, even if it uses properties of convex functions to do something more sophisticated than low-degree approximation, must use $n^{\mathrm{poly}(L/\varepsilon)}$ statistical queries.

Our proof relies on the construction of a large family $\mathcal{F}$ of functions with low pairwise-correlation, which is the standard approach for showing *weak learning* lower bounds in the CSQ model [12, 75, 33]. Additionally, we require each function in $\mathcal{F}$ to be non-trivially correlated with a convex Lipschitz function – namely a "projected ReLU", which is a function of the form $x \mapsto \mathrm{ReLU}(\langle x, u\rangle)$ for some $u \in \mathbb{S}^{n-1}$. This implies that an agnostic learning algorithm for convex functions gives a weak learner for $\mathcal{F}$, and yields our CSQ lower bound.

Our construction and proof strategy are inspired by and leverage the results of [30], who showed an $n^{\mathrm{poly}(1/\varepsilon)}$ lower bound for agnostically learning ReLUs under the Gaussian distribution using statistical queries. While one might hope to immediately conclude the same lower bound for agnostically learning convex functions in our model (since ReLUs are convex), the key difference is that in our setting the input must be Lipschitz, whereas [30] build their hard family of inputs from Boolean functions. We adapt their construction to our purposes via careful application of the Ornstein-Uhlenbeck noise operator P_t, which satisfies a few crucial properties for our application:

- It takes bounded functions into Lipschitz ones, so that from a hard input $f : \mathbb{R}^n \to \{\pm 1\}$, we may obtain a hard Lipschitz input $P_t f$.
- It satisfies the self-adjointness property $\langle P_t f, g\rangle = \langle f, P_t g\rangle$, which allows us to map the approximation quality of a learned hypothesis g for the Lipschitz input $P_t f$ into the approximation quality of hypothesis $P_t g$ for the hard input f.
- Lipschitz functions are stable under noise in the sense that $\|P_t f - f\|_2^2 = O(t)$, which informally implies that most of the "signal" in the reduction survives our application of noise.

1.1.3 One-Sided Testing of Lipschitz Convex Functions

Our final result is a *one-sided* testing algorithm for Lipschitz convex functions – that is, an algorithm which never rejects a Lipschitz convex function. In general, there can be dramatic separations between the complexity of one-sided and two-sided testing (cf. [37, §1.5]), and in particular, testing by learning is not a suitable approach to one-sided testing. Therefore, this goal requires us to exploit the properties of convex and Lipschitz functions in a different way. Our result is:

▶ **Theorem 4** (One-sided testing of Lipschitz convex functions). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be L-Lipschitz, and let $\varepsilon \geq 0$. There is an algorithm, One-Sided-Tester (Algorithm 1), which, given access to i.i.d. labeled samples $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x} \sim N(0, I_n)$, makes $O(\sqrt{n}L/\varepsilon)^n$ draws, runs in time $(nL/\varepsilon)^{O(n)}$, and has the following performance guarantee:*

- *If f is convex, then One-Sided-Tester outputs "accept" with probability 1;*
- *If $\mathrm{d}^L_{\mathrm{conv}}(f) \geq \varepsilon$, then One-Sided-Tester outputs "reject" with probability $9/10$.*

Our tester operates by constructing an *empirical convex envelope* (cf. Definition 19) of the sampled function values, which can be viewed as the function analogue of the convex hull of a collection of points. Given random samples $(\boldsymbol{x}_i, f(\boldsymbol{x}_i))$, the tester checks whether f coincides with its empirical convex envelope at these points. If f is convex, this always holds; conversely, we show that if f is L-Lipschitz and ε-far from convex, then with sufficiently many samples the test will detect a violation with high probability. See Figure 1 for an illustration of the empirical convex envelope.

In particular, our analysis implies an associated *agnostic learning guarantee in $L^\infty(\gamma)$*: we show that when the input function f is an L-Lipschitz convex function, the empirical convex envelope learned by One-Sided-Tester uniformly approximates the convex envelope of f over all but an exponentially-small fraction of Gaussian space. Thus, the algorithm may also be viewed as a form of *uniform convex regression*.

1.2 Related Work

We briefly situate our results within the broader work on learning and testing convexity.

Learning Convex Functions

As discussed earlier, the statistical perspective for *convex regression* is interested in the error rate as the number of samples goes to infinity for fixed dimensions. This falls under the broader umbrella of *shape-constrained regression*. We refer the interested reader to [72, 40, 41, 61, 32] and the references therein for further background. By contrast, our results address the more general problem of *agnostic proper learning* from i.i.d. samples, with the goal of establishing PAC-style sample-complexity bounds rather than asymptotic error rates.

We note that low-degree approximation of Lipschitz functions over Gaussian space has been used a few times in the past; see, for example, [2, 46, 23]. Such approximations also are the source of $n^{\mathrm{poly}(1/\varepsilon)}$ sample-complexity bounds appearing in several agnostic learning problems beyond convexity, e.g., in the context of learning halfspaces [31, 57]. However, with the exception of [2], most of these works focus on Boolean-valued function classes rather than real-valued ones.

Approximation Theory

As indicated earlier in Section 1.1, our upper bounds for convexity testing follow from semi-agnostic learning algorithms for convex functions under the Gaussian L_2 distance. Related problems have been studied in approximation theory, both in the low-dimensional setting [74, 70] as well as in high-dimensions [45], with exponential-in-n sample lower bounds known for deterministic approximation under L^p distance [45].

More broadly, classical results in convex approximation typically focus on deterministic settings, approximating a convex function by piecewise-linear or polynomial surrogates and bounding the error in terms of smoothness or curvature (see, e.g., [29, 45]). Our setting differs in that we consider randomized, sample-based approximation under a fixed measure, leading to dimension-dependent but probabilistic guarantees.

Testing Convexity over Discrete Domains

The computational problem of recognizing convexity of quartic polynomials is known to be NP-hard [1]. (Deciding convexity of polynomials of odd-degree or quadratics is trivial, making degree 4 the first interesting case.) See [1] and the references therein for a history of the computational problem of recognizing *exact* convexity of functions. The relaxed problem of recognizing *approximate* convexity of functions has primarily been considered in the discrete setting. The study of testing convexity of functions $f : [N] \to \mathbb{R}$ under the Hamming distance was initiated by Parnas, Ron, and Rubinfeld [64], where convexity is defined as non-negativity of a discrete second-derivative. Their was later extended by Ben-Eliezer [5], who obtained improved quantitative bounds; by Pallavoor, Raskhodnikova, and Verma [63], who gave an improved bound parametrized by the size of the range of the function being tested; by Blais, Raskhodnikova, and Yaroslavtsev [11], who gave lower bounds on testing convexity of real-valued functions over the hypergrid $[N]^d$; and by Belovs, Blais, and Bommireddi [4], who gave upper and lower bounds on the number of queries required to test convexity of real-valued functions over various discrete domains including the discrete line, the "stripe" $[3] \times N$, and the hypergrid $[N]^d$. These results, however, do not have any bearing on this work due to differences in domain and choice of distance metric.

L^p Testing of Convexity

The work most closely related to ours is that of Berman, Raskhodnikova, and Yaroslavtsev [6], who study L^p testing of convexity of an unknown $f : [0,1]^n \to [0,1]$ where $[0,1]^n$ is endowed with the uniform measure. They provide tight bounds in the one-dimensional setting and exponential-in-n bounds for the high-dimensional case (see Corollary B.3 of [6]). Our results, in contrast, apply specifically to Lipschitz functions $f : \mathbb{R}^n \to \mathbb{R}$, giving polynomial-in-$n$ complexity algorithms for every fixed choice of the distance parameter ε. (We remark that our exponential-in-n algorithm from Theorem 4 has the additional property of being a tester with one-sided error, which does not follow from a black-box application of testing-by-learning.)

Although one might hope to relate the two settings, this connection does not appear to be straightforward. The two frameworks are best viewed as complementary: ours imposes regularity assumptions that enable polynomial dependence on n, whereas [6] addresses unrestricted (i.e., possibly highly irregular) albeit bounded functions at exponential cost. Indeed, assuming Lipschitzness or bounded gradient is the standard regularity condition in convex analysis, optimization, and high-dimensional learning, and thus provides a more natural setting for analytic property testing.

Testing Set Convexity

A different line of research considers testing convexity of high-dimensional sets (i.e., Boolean-valued indicator functions of subsets of $\mathbb{R}^n$). Rademacher and Vempala [66] established exponential-in-n upper bounds for testing set convexity under the Lebesgue measure; they also gave a strong lower bound against a simple "line" tester for convexity. The [66] lower bound was subsequently strengthened by Blais and Bommireddi [10]. For testing set convexity under the Gaussian measure, the best-known upper bound follows from the agnostic learning algorithm for convex sets due to Klivans, O'Donnell, and Servedio [52].

Strong lower bounds for sample-based testing under the Gaussian measure were established by Chen, Freilich, Servedio, and Sun [20], and the first lower bounds in the query model were obtained recently by Chen, De, Nadimpalli, Servedio, and Waingarten [21]. While our work shares conceptual and technical similarities with [52, 20, 21], there does not appear to be a

reduction or formal connection between the two settings. In particular, natural attempts to reduce testing convexity of functions to testing convexity of sets (such as mapping a function to its epigraph) fail to preserve the relevant distance measures, so neither problem appears to subsume the other.

Finally, beyond the Gaussian setting, the work of Black, Blais, and Harms [9] also considers consider testing and learning convex subsets of the ternary hypercube $\{0, \pm 1\}^n$.

1.3 Discussion

As evidenced above, convexity is a powerful and pervasive principle whose associated computational problems are notoriously challenging and enjoy long lines of study from a plethora of perspectives. We view the present work as a first step in posing and investigating the problems of learning and testing convex functions in the continuous setting from the perspective of learning theory and property testing. Our results establish that these problems are computationally feasible, albeit not at present with efficient algorithms, and suggest barriers to be tackled for overcoming this limitation: any agnostic learning algorithm with truly $\mathrm{poly}(n)$ sample complexity, beating the low-degree learning bound from Theorem 1, must not be a CSQ algorithm (Theorem 3); and any algorithm (for learning or testing) with subexponential running time must bypass the grid-based approaches of Theorems 1 and 4.

We conclude with a number of open directions suggested by our work.

Analogy with Boolean Monotonicity

Convexity is formally connected to the monotonicity of the gradient: in one dimension, a function is convex if and only if its derivative is monotone, and more generally, convexity corresponds to the monotonicity of the gradient map. This connection suggests a natural analogy between convexity and monotonicity, and one might hope that insights from the rich body of work on learning and testing monotone Boolean functions can inform the study of convex function learning and testing.

For *learning*, our results may at first appear considerably stronger than existing bounds for learning monotone Boolean functions. (Recall that the sample complexity of agnostically learning monotone Boolean functions over $\{0,1\}^n$ is $2^{\widetilde{O}(\sqrt{n})}$ [17].) However, this stems from the $O(1)$-Lipschitz assumption in our setting, which has no meaningful analog in the Boolean function setting. While all Boolean functions are technically $O(1)$-Lipschitz, the underlying metric geometry differs between the Boolean setting and Gaussian space: in the Boolean space (see, for example, [55]), the typical L_1 (Hamming) distance between points in dimension n is of order n while in the Gaussian space the typical distance is of order $\sqrt{n}$. So, at least in some sense, our polynomial agnostic learning algorithm does correspond to the results for learning monotone functions with $2^{\Theta(\sqrt{n})}$ samples.

As for *testing*, in the Boolean setting the analogous quantity to $\mathrm{d}_{\mathrm{conv}}$ is the *distance to monotonicity* $\mathrm{d}_{\mathrm{mon}}$, which has been extensively investigated in a rich line of work on monotonicity testing [38, 18, 50]. Classical Boolean monotonicity testers rely on upper bounds on $\mathrm{d}_{\mathrm{mon}}$ in terms of the *directed edge boundary*, and these structural relationships can be viewed as strengthenings of the classical edge-isoperimetric inequalities on the hypercube [59, 76].

This analogy raises a natural question: can $\mathrm{d}_{\mathrm{conv}}$ likewise give rise to structural inequalities that quantify the "non-convexity" of real-valued functions? Motivated by the role of the directed edge boundary in monotonicity testing, the following question suggests itself:

▶ **Question 5.** *Given an L-Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$, is*

$$\mathrm{d}_{\mathrm{conv}}(f) \leq \mathrm{poly}(n) \cdot \mathop{\mathbf{E}}_{\boldsymbol{x} \sim N(0,I_n)} \left[\left\| \nabla^2_- f(\boldsymbol{x}) \right\|_{\mathrm{op}} \right]$$

where $\nabla^2_- f(\boldsymbol{x})$ is the negative-definite part of $\nabla^2 f(\boldsymbol{x})$?

If true, such an inequality would likely imply a convexity tester based on random samples of (or queries simulating) the Hessian of f, mirroring how monotonicity testers – both in discrete and continuous settings – leverage local gradient information [50, 34]. Beyond its algorithmic implications, this inequality would also be of intrinsic geometric interest, as it would relate a *global* measure of non-convexity (namely, $\mathrm{d}_{\mathrm{conv}}(f)$) to *local* curvature data through the Hessian.

As a step toward understanding $\mathrm{d}_{\mathrm{conv}}$, we establish in the full version a quantitative relationship between $\mathrm{d}_{\mathrm{conv}}(f)$ and the negative degree-2 Hermite coefficients of f, which yields a lower bound on $\mathrm{d}_{\mathrm{conv}}(f)$ in terms of their magnitude.

We note that an independent line of work has identified an unexpected but powerful analogy between monotone Boolean functions and *convex subsets of Gaussian space* [24, 28, 25, 26, 27]. It is natural to ask which aspects of this correspondence extend further to the setting of convex functions over Gaussian space. As a first step in this direction, we note that [24] establishes a quantitative sharpening of Hu's correlation inequality for symmetric convex sets [47], in the spirit of Talagrand's quantitative correlation inequality [77].

Testing Lower Bounds

Since we are concerned with functions $f : \mathbb{R}^n \to \mathbb{R}$ whose outputs are real-valued, standard information-theoretic approaches to proving lower bounds become infeasible. A basic but significant obstacle – shared with many real-valued testing problems – is that each sample reveals a real number $f(x)$, which, at least naively, may contain an arbitrarily large amount of information. Presently, there is no general framework for establishing lower bounds for testing convexity of real-valued functions, even in the purely sample-based setting.

Therefore, we pose the following two questions on the true sample complexity of testing convexity of Lipschitz functions in the two-sided and one-sided error regimes. For two-sided testing, it is conceivable – again in analogy with monotonicity testing – that some polynomial dependence on n is unavoidable. We ask:

▶ **Question 6.** *Does testing convexity of 1-Lipschitz functions under the Gaussian distribution (with two-sided error) require $n^{\Omega(1)}$ samples? What if arbitrary queries are allowed?*

Even for one-sided testing algorithms, we currently lack any nontrivial lower bounds on the sample complexity of testing convexity. It is instructive to recall how an exponential lower bound is obtained for the analogous problem in the setting of *set* convexity [20]. In particular, Lemma 29 of [20] establishes that for sufficiently small constant c, if 2^{cn} points are drawn independently from $N(0, I_n)$, then with high probability no point lies in the convex hull of the others. Consequently, such a draw of samples is consistent with the indicator function of a convex set, and a one-sided sample-based tester cannot detect any violation of convexity.

The difficulty in our setting is that a sample with no witnessed violation of convexity need not be *consistent* with any L-Lipschitz function that is convex. In particular, a naive interpolation through the sampled points may fail to be $O(L)$-Lipschitz. Thus, it is not immediate that a one-sided tester must accept unless the sample explicitly contains a convexity violation. This leads to the following question:

▶ **Question 7.** *Is there a tester for convexity of* 1*-Lipschitz functions under the Gaussian distribution with one-sided error and sample complexity that is sub-exponential in* n*?*

Sample vs. Runtime Gaps

We note that while Theorem 1 achieves a sample complexity of $n^{O(1/\varepsilon^2)}$, its runtime remains exponential in n. An analogous gap long persisted for properly learning monotone Boolean functions from samples: the best-known sample complexity was $2^{\widetilde{O}(\sqrt{n})}$, whereas the best runtime was $2^{O(n)}$. Recent work by [56, 58] closed this gap, giving a $2^{\widetilde{O}(\sqrt{n})}$-time agnostic proper learner. It is natural to ask whether a similarly efficient proper learning algorithm exists for convex functions, potentially closing the gap between information-theoretic and computational efficiency in our setting as well.

Lower Bounds in the Full SQ Model

Our $n^{\mathrm{poly}(L/\varepsilon)}$ lower bound from Theorem 3 applies to correlational statistical query (CSQ) algorithms, a setting that already captures the algorithm attaining our $n^{O(L^2/\varepsilon^2)}$ upper bound. Recall that CSQ algorithms only have access to the input f via approximate inner product queries of the form $\mathbf{E}_{\boldsymbol{z}\sim N(0,I_n)}[f(\boldsymbol{z})h(\boldsymbol{z})]$. In contrast, the full SQ model allows approximate oracle access to $\mathbf{E}_{\boldsymbol{z}\sim N(0,I_n)}[q(\boldsymbol{z}, f(\boldsymbol{z}))]$ for general query functions $q(x, y)$.

In the (distribution-specific) setting of *Boolean* functions, the CSQ model is equivalent to the full SQ model [16], because SQ queries to Boolean functions may be simulated using correlational queries. However, the same is not true in the real-valued setting, where there exist separations between the CSQ and SQ models, for example for the problem of ReLU regression [42]. Indeed, proving lower bounds against SQ algorithms in the real-valued setting is a notoriously challenging problem; see [35, 19] and the discussion therein. We leave as an open question whether our $n^{\mathrm{poly}(L/\varepsilon)}$ CSQ lower bound can be extended to the full SQ model, or whether our problem witnesses a separation between these models.

Organization

In the rest of this paper, we outline the proofs of our main upper bound results – Theorems 1, 2, and 4. We defer proofs of technical lemmas and our lower bounds to the full version.

2 Preliminaries

We use boldfaced letters – such as $\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{A}$ – to denote random variables (which may be real-valued, vector-valued, function-valued, or set-valued; the intended type will be clear from the context). We write $\boldsymbol{x} \sim \mathcal{D}$ to indicate that the random variable $\boldsymbol{x}$ is distributed according to the probability distribution $\mathcal{D}$. Throughout, $\mathbb{S}^{n-1} := \{x \in \mathbb{R}^n : \|x\|_2 = 1\}$ will denote the unit sphere in $\mathbb{R}^n$. We will write $B_p(r) \subseteq \mathbb{R}^n$ for the ℓ_p-ball of radius r, i.e.,

$$B_p(r) := \{x \in \mathbb{R}^n : \|x\|_p \leq r\}.$$

Hermite Analysis over Gaussian Space

Our notation and terminology follow Chapter 11 of [62]. We say that an n-dimensional *multi-index* is a tuple $\alpha \in \mathbb{N}^n$, and we define $|\alpha| := \sum_{i=1}^n \alpha_i$.

For $n \in \mathbb{N}_{>0}$, we write $L^2(\gamma_n)$ to denote the space of functions $f : \mathbb{R}^n \to \mathbb{R}$ that have finite second moment under the Gaussian distribution, i.e., $f \in L^2(\gamma_n)$ if

$$\|f\|^2_{L^2(\gamma_n)} := \mathop{\mathbf{E}}_{\boldsymbol{z} \sim N(0,I_n)} \left[f(\boldsymbol{z})^2\right] < \infty \,.$$

When the dimension n is clear from context, we will write $\gamma = \gamma_n$ instead. We view $L^2(\gamma)$ as an inner product space with

$$\langle f, g \rangle_{L^2(\gamma)} := \mathop{\mathbf{E}}_{\boldsymbol{z} \sim N(0,I_n)} [f(\boldsymbol{z}) g(\boldsymbol{z})] \,.$$

We will sometimes write $\langle f, g \rangle = \langle f, g \rangle_{L^2(\gamma)}$ for notational convenience. We recall the Hermite basis for $L^2(\gamma_1)$:

▶ **Definition 8** (Hermite basis). *The* Hermite polynomials $(h_j)_{j \in \mathbb{N}}$ *are the univariate polynomials defined as*

$$h_j(x) = \frac{(-1)^j}{\sqrt{j!}} \exp\left(\frac{x^2}{2}\right) \cdot \frac{d^j}{dx^j} \exp\left(-\frac{x^2}{2}\right).$$

In particular, we have

$$h_2(x) := \frac{x^2 - 1}{\sqrt{2}} \,. \tag{1}$$

The following fact is standard:

▶ **Fact 9** (Proposition 11.33 of [62]). *The Hermite polynomials $(h_j)_{j \in \mathbb{N}}$ form a complete, orthonormal basis for $L^2(\gamma_1)$. For $n > 1$ the collection of n-variate polynomials given by $(h_\alpha)_{\alpha \in \mathbb{N}^n}$ where*

$$h_\alpha(x) := \prod_{i=1}^{n} h_{\alpha_i}(x)$$

forms a complete, orthonormal basis for $L^2(\gamma_n)$.

Given a function $f \in L^2(\gamma)$ and $\alpha \in \mathbb{N}^n$, we define its *Hermite coefficient on* α as $\widehat{f}(\alpha) = \langle f, h_\alpha \rangle$. It follows that $f : \mathbb{R}^n \to \mathbb{R}$ can be uniquely expressed as

$$f = \sum_{\alpha \in \mathbb{N}^n} \widehat{f}(\alpha) h_\alpha$$

with the equality holding in $L^2(\gamma)$; we will refer to this expansion as the *Hermite expansion* of f. One can check that Parseval's and Plancharel's identities hold in this setting:

$$\langle f, f \rangle = \sum_{\alpha \in \mathbb{N}^n} \widehat{f}(\alpha)^2 \qquad \text{and} \qquad \langle f, g \rangle = \sum_{\alpha \in \mathbb{N}^n} \widehat{f}(\alpha) \widehat{g}(\alpha).$$

We recall the Gaussian Poincaré inequality (see, e.g., [3]):

▶ **Proposition 10.** *Suppose $f \in L^2(\gamma)$ is differentiable. Then*

$$\mathop{\mathbf{Var}}_{\boldsymbol{x} \sim N(0,I_n)} [f(\boldsymbol{x})] \leq \mathop{\mathbf{E}}_{\boldsymbol{x} \sim N(0,I_n)} \left[\|f(\boldsymbol{x})\|_2^2\right] .$$

3 Agnostic Proper Learning and Two-Sided Sample-Based Testing

Write $\mathcal{C}(L)$ for the class of convex, L-Lipschitz functions:

$$\mathcal{C}(L) := \{f : \mathbb{R}^n \to \mathbb{R} : f \text{ is } L\text{-Lipschitz and convex}\}.$$

Recall our notation for the L_2 distance to the set of convex L-Lipschitz functions, which we may now write as

$$\mathrm{d}^L_{\mathrm{conv}}(f) = \inf_{g \in \mathcal{C}(L)} \|f - g\|_{L^2(\gamma)}.$$

By convention, given a function $f : \mathbb{R}^n \to \mathbb{R}$, by "sample from f" we mean a sample of the form $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x} \sim N(0, I_n)$. The main result in this section is the following.

▶ **Theorem 1** (Agnostic proper learning of Lipschitz convex functions). *Let $\varepsilon, L > 0$ and suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a L-Lipschitz function. There exists an algorithm which, given i.i.d. access to labeled samples $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x} \sim N(0, I_n)$, draws $n^{O(L^2/\varepsilon^2)}$ samples, runs in time $\exp(\widetilde{O}(nL^2/\varepsilon^2))$, and outputs a function $g \in L^2(\gamma)$ such that*

$$\|f - g\|_{L^2(\gamma)} \le \mathrm{d}^L_{\mathrm{conv}}(f) + \varepsilon.$$

Additionally, the function g is convex and L-Lipschitz.

Combining the agnostic proper learning algorithm with a distance estimation step, we obtain a sample-based tolerant tester for Lipschitz convex functions:

▶ **Theorem 2** (Tolerant two-sided testing of Lipschitz convex functions). *Let $\varepsilon, \varepsilon_0, L \ge 0$ and suppose $f : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz. There is an algorithm which, given i.i.d. labeled samples $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x} \sim N(0, I_n)$, draws $n^{O(L^2/\varepsilon^2)}$ samples, runs in time $\exp(\widetilde{O}(nL^2/\varepsilon^2))$, and has the following performance guarantee:*

- *If $\mathrm{d}^L_{\mathrm{conv}}(f) \le \varepsilon_0$, then it outputs "accept" with probability $9/10$;*
- *If $\mathrm{d}^L_{\mathrm{conv}}(f) \ge \varepsilon + \varepsilon_0$, then it outputs "reject" with probability $9/10$.*

Proof. Note that

$$\mathrm{d}^L_{\mathrm{conv}}(f)^2 \le \mathbf{Var}[f] \le \mathbf{E}\left[\|\nabla f\|^2\right] \le L^2$$

by the Poincaré inequality (Proposition 10), the fact that f is L-Lipschitz, and the fact that constant functions are convex. Hence we may assume that $\varepsilon \le L$, since otherwise the tester can immediately accept.

On input $f : \mathbb{R}^n \to \mathbb{R}$, the tester uses the learning algorithm from Theorem 1 to learn a function $g \in \mathcal{C}(L)$ satisfying $\|f - g\|_2 \le \mathrm{d}^L_{\mathrm{conv}}(f) + \varepsilon/4$ with probability at least $2/3$; assume henceforth that this event occurs. It then takes $m = O(L^4/\varepsilon^4)$ samples $(\boldsymbol{x}_i, f(\boldsymbol{x}_i))$, and accepts if and only if

$$\boldsymbol{Y} := \frac{1}{m}\sum_{i=1}^{m}(f(\boldsymbol{x}_i) - g(\boldsymbol{x}_i))^2 \le \left(\varepsilon_0 + \frac{\varepsilon}{2}\right)^2.$$

The sample and time complexity claims are immediate, and we now prove correctness. Let $h := f - g$, so that $\mathbf{E}[\boldsymbol{Y}] = \mathbf{E}[h^2] = \|f - g\|^2_{L^2(\gamma)}$ and $\mathbf{Var}[\boldsymbol{Y}] = \frac{1}{m}\mathbf{Var}[h^2]$. Define the centered function $\bar{h} := h - \mathbf{E}[h]$. Then $\bar{h}$ is centered and $2L$-Lipschitz, and hence sub-Gaussian with sub-Gaussian norm $O(L)$. Then

$$\mathbf{Var}[h^2] = \mathbf{Var}[(\bar{h} + \mathbf{E}[h])^2] = \mathbf{Var}[\bar{h}^2 + 2\,\mathbf{E}[h]\bar{h}] \le 2\,\mathbf{Var}[\bar{h}^2] + 2 \cdot \mathbf{E}[h]^2\,\mathbf{Var}[\bar{h}].$$

By the sub-Gaussianity of $\bar{h}$, we have $\mathbf{Var}[\bar{h}^2] = O(L^4)$ and $\mathbf{Var}[\bar{h}] = O(L^2)$. Moreover, we have

$$\mathbf{E}[h]^2 \le \mathbf{E}[h^2] = \|f-g\|_{L^2(\gamma)}^2 \le \left(\mathrm{d}_{\mathrm{conv}}^L(f) + \frac{\varepsilon}{4}\right)^2 \le 2\mathrm{d}_{\mathrm{conv}}^L(f)^2 + \varepsilon^2 \le O(L^2)\,.$$

We conclude that $\mathbf{Var}[h^2] \le O(L^4)$. By Chebyshev's inequality,

$$\mathbf{Pr}\left[\left|\boldsymbol{Y} - \|f-g\|_{L^2(\gamma)}^2\right| \ge \frac{\varepsilon^2}{16}\right] \le \frac{\frac{1}{m}\cdot O(L^4)}{\varepsilon^4/256} \le \frac{1}{10}\,,$$

for suitable choice of $m = O(L^4/\varepsilon^4)$. Suppose $\left|\boldsymbol{Y} - \|f-g\|_{L^2(\gamma)}^2\right| \le \varepsilon^2/16$. Then if $\mathrm{d}_{\mathrm{conv}}^L(f) \le \varepsilon_0$, we obtain

$$\boldsymbol{Y} \le \|f-g\|_{L^2(\gamma)}^2 + \frac{\varepsilon^2}{16} \le \left(\|f-g\|_{L^2(\gamma)} + \frac{\varepsilon}{4}\right)^2 \le \left(\mathrm{d}_{\mathrm{conv}}^L(f) + \frac{\varepsilon}{2}\right)^2 \le \left(\varepsilon_0 + \frac{\varepsilon}{2}\right)^2,$$

and the tester accepts. On the other hand, suppose $\mathrm{d}_{\mathrm{conv}}^L(f) > \varepsilon_0 + \varepsilon$. Then, recalling that g is a convex, L-Lipschitz function, we obtain

$$\begin{aligned}\boldsymbol{Y} &\ge \|f-g\|_{L^2(\gamma)}^2 - \frac{\varepsilon^2}{16} > (\varepsilon_0+\varepsilon)^2 - \frac{\varepsilon^2}{16} = \varepsilon_0^2 + 2\varepsilon_0\varepsilon + \frac{15\varepsilon^2}{16} \\ &> \varepsilon_0^2 + 2\varepsilon_0\cdot\frac{3\varepsilon}{4} + \left(\frac{3\varepsilon}{4}\right)^2 = \left(\varepsilon_0 + \frac{3\varepsilon}{4}\right)^2,\end{aligned}$$

and the tester rejects. By a union bound, the tester correctly accepts/rejects except with probability at most $\frac{2}{3} + \frac{1}{10} < \frac{1}{2}$, which may be made as small as desired by a standard majority vote. ◀

Proof of the Main Result

We require the following lemmas, whose proofs we defer to the full version. The first lemma says that a Lipschitz function is well-approximated by its low-degree Hermite expansion. This fact has been observed before, e.g. in [2, 46] for domain $[0,1]^n$; here we give a Gaussian version. Recall the notation $f^{\le d} := \sum_{|\alpha|\le d}\widehat{f}(\alpha)h_\alpha$ for the degree-d Hermite expansion of f.

▶ **Lemma 11** (Lipschitz functions are approximately low-degree). *Let $L, \varepsilon > 0$, let $f : \mathbb{R}^n \to \mathbb{R}$ be L-Lipschitz, and let $d := \lceil L^2/\varepsilon^2 \rceil$. Then $\|f - f^{\le d}\|_{L^2(\gamma)} \le \varepsilon$.*

Additionally, we now give upper bounds on the number of samples required to approximately learn the low-degree Hermite coefficients of a Lipschitz function. We assume for simplicity that f has expectation bounded by some constant (this assumption is not essential, and the general case is recovered in the proof of Theorem 1).

▶ **Lemma 12** (Learning a single Hermite coefficient). *Let $L, \xi > 0$, $\delta \in (0,1)$, $\alpha \in \mathbb{N}^n$, and let $f : \mathbb{R}^n \to \mathbb{R}$ be L-Lipschitz and satisfy $|\mathbf{E}[f]| \le O(1)$. Then*

$$O\left(\frac{L^2 3^{|\alpha|}}{\xi^2}\log\left(\frac{1}{\delta}\right)\right) \text{ samples from } f$$

suffice to compute an estimate $\widetilde{\boldsymbol{f}}(\alpha)$ satisfying $\left|\widetilde{\boldsymbol{f}}(\alpha) - \widehat{f}(\alpha)\right| \le \xi$ with probability at least $1-\delta$. The running time is polynomial in the number of samples.

We have the following immediate consequence:

▶ **Corollary 13** (Learning a low-degree expansion). *Let $L, \varepsilon > 0$, $\delta \in (0,1)$, $d \in \mathbb{N}$, and let $f : \mathbb{R}^n \to \mathbb{R}$ be L-Lipschitz and satisfy $|\mathbf{E}[f]| \le O(1)$. Then*

$$O\left(\frac{L^2 n^{2.01d}}{\varepsilon^2} \log\left(\frac{1}{\delta}\right)\right) \text{ samples from } f$$

suffice to obtain coefficient estimates $\widetilde{\boldsymbol{f}}(\alpha)$, for all $\alpha \in \mathbb{N}^n$ with $|\alpha| \le d$, such that the function $\boldsymbol{f} := \sum_{|\alpha| \le d} \widetilde{\boldsymbol{f}}(\alpha) h_\alpha$ satisfies $\|f^{\le d} - \boldsymbol{f}\|_{L^2(\gamma)} \le \varepsilon$ with probability at least $1 - \delta$. The running time is polynomial in the number of samples.

We will require a simple a priori upper bound on the Lipschitz constant of the low-degree approximation above in terms of distance to the origin:

▶ **Lemma 14** (Lipschitz constant of low-degree polynomials). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a degree-d polynomial. Then for all $x \in \mathbb{R}^n$, we have*

$$\|\nabla f(x)\|_2 \le \|f\|_{L^2(\gamma)} d^{d+\frac{1}{2}} n^{d/2} (1 + |x|)^d .$$

The next lemma says that, given *query* access to a Lipschitz function g which is close to some convex Lipschitz function over a ball, we may learn (via convex regression) a good convex approximation to g using a finite number of queries. We will apply this result to an approximation $\boldsymbol{f}$ of the low-degree restriction $f^{\le d}$ over the restriction of the Gaussian measure to a sufficiently large ball. It is convenient to work with the centered radius-r ℓ^∞-ball $B_\infty(r)$, i.e. an axis-aligned hypercube.

▶ Remark 15. This number of queries used for the convex regression is exponential in n; however, we make these queries to the learned approximation $\boldsymbol{f}$ rather than the input function f, so we pay this cost in the running time but not in the sample complexity.

▶ **Lemma 16** (Learning a Lipschitz convex approximation). *Let $r > 0$, and let μ be the probability measure γ conditioned to $B_\infty(r)$. There exists an algorithm which, given parameters $L, L', \varepsilon > 0$, with $L' \ge L$, and query access to some L'-Lipschitz function $g : B_\infty(r) \to \mathbb{R}$, makes $(L' r \sqrt{n}/\varepsilon)^{O(n)}$ queries to g and outputs a function $\widetilde{g} \in \mathcal{C}(L)$ that is a nearly optimal convex approximation of g over $B_\infty(r)$ with respect to L-Lipschitz functions, in the following sense: for all convex, L-Lipschitz $h : B_\infty(r) \to \mathbb{R}$, we have*

$$\|\widetilde{g} - g\|_{L^2(B_\infty(r),\mu)} \le \|h - g\|_{L^2(B_\infty(r),\mu)} + \varepsilon.$$

The running time is polynomial in the number of queries.

To translate back and forth between full Gaussian space and its restriction to an ℓ^∞-ball, while approximately preserving the distance between Lipschitz functions, we use the following lemma.

▶ **Lemma 17** (Restricting Gaussian space to a box). *There exists a universal constant $C > 0$ such that the following holds. Let $L, \tau \ge 1$, $\varepsilon \in (0, L)$, and let $r := C\sqrt{\log(nL\tau/\varepsilon)}$. Let μ be probability measure γ conditioned to $B_\infty(r)$. Then for all L-Lipschitz functions $f, g : \mathbb{R}^n \to \mathbb{R}$ such that $|f(x^*) - g(x^*)| \le \tau$ for some $x^* \in B_\infty(r)$, we have*

$$\left| \|f - g\|_{L^2(\gamma)} - \|f - g\|_{L^2(B_\infty(r),\mu)} \right| \le \varepsilon .$$

▶ Remark 18. The condition that $|f(x) - g(x)| \le \tau$ for some $x \in B_\infty(r)$ in Lemma 17 holds in particular if $\|f - g\|_2 \le \tau/2$, since otherwise the choice of radius r is large enough to yield (say) $\|f - g\|_2 \ge 3\tau/4$, a contradiction.

Equipped with these results, we may prove Theorem 1.

Proof of Theorem 1. We first claim that we may assume without loss of generality that $|\mathbf{E}[f]| \le \varepsilon/100$. Indeed, the algorithm may first compute an approximation $\bar{\boldsymbol{f}}$ of $\mathbf{E}[f]$ up to $\varepsilon/100$ additive error using $O(L^2/\varepsilon^2)$ samples (by Chebyshev's inequality, since $\mathbf{Var}[f] \le \mathbf{E}[\|\nabla f\|_2^2] \le L^2$ by Proposition 10, and the Lipschitz assumption), then proceed with the modified input function $f - \bar{\boldsymbol{f}}$ and shift the final output again by $\bar{\boldsymbol{f}}$.

We may also assume that $L \ge 1$ (by rescaling the input and ε if necessary) and that $\varepsilon < \frac{100}{99}L$ (otherwise, we get $\|f\|_{L^2(\gamma)} \le \|f - \mathbf{E}[f]\|_{L^2(\gamma)} + |\mathbf{E}[f]| \le L + \varepsilon/100 \le \varepsilon$ by the triangle and Poincaré inequalities, and we may immediately output the constant zero function).

Let $\varepsilon' := \varepsilon/8$ and $d := \lceil (L/\varepsilon')^2 \rceil$. The algorithm first uses

$$O\left(\frac{L^2 n^{2.01d}}{\varepsilon^2}\right) = n^{O(L^2/\varepsilon^2)} \text{ samples from } f$$

to learn, via Corollary 13, a degree-d approximation $\boldsymbol{f}$ satisfying $\|f^{\le d} - \boldsymbol{f}\|_{L^2(\gamma)} \le \varepsilon'$ with probability at least $2/3$; assume henceforth that this event occurs. By Lemma 11 and the triangle inequality, we have $\|f - \boldsymbol{f}\|_{L^2(\gamma)} \le 2\varepsilon'$. Note that $\boldsymbol{f}$ satisfies $\|\boldsymbol{f}\|_{L^2(\gamma)} \le \|f\|_{L^2(\gamma)} + 2\varepsilon' \le L + \varepsilon/100 + 2\varepsilon' \le 2L$.

Let $r = \Theta\left(\frac{L}{\varepsilon}\sqrt{\log(nL/\varepsilon)}\right)$ be large enough to ensure that, when we set

$$L' := 2L(2ndr)^{2d} \ge \|\boldsymbol{f}\|_{L^2(\gamma)}(2ndr)^{2d} \ge \|\boldsymbol{f}\|_{L^2(\gamma)} d^{d+\frac{1}{2}} n^{d/2}(1 + r\sqrt{n})^d$$

as the upper bound on the Lipschitz constant of $\boldsymbol{f}$ over $B_\infty(r)$ via Lemma 14, we obtain

$$r \ge C'\sqrt{\log(nL'/\varepsilon)}$$

for sufficiently large choice of absolute constant $C' > 0$ so that, by Lemma 17, ball $B_\infty(r)$ gives an ε'-approximation for the distance between $\boldsymbol{f}$ and L-Lipschitz functions g for parameter $\tau = 6L$. (One may check that it is possible to choose r satisfying this.)

Let $\widetilde{g} \in \mathcal{C}(L)$ be the function produced by Lemma 16 with input $g = \boldsymbol{f}$, the choice of r above, and parameters L, L' and ε'. Note that the number of queries to $\boldsymbol{f}$, and thus the running time of this step, is

$$(L'r\sqrt{n}/\varepsilon)^{O(n)} = (Lndr/\varepsilon)^{O(dn)} = (nL/\varepsilon)^{O(nL^2/\varepsilon^2)} = \exp(\widetilde{O}(nL^2/\varepsilon^2))\,.$$

We claim that $\widetilde{g}$ satisfies the guarantee of Theorem 1. We already have that $\widetilde{g} \in \mathcal{C}(L)$, so it remains to show that $\|\widetilde{g} - f\|_{L^2(\gamma)} \le \mathrm{d}^L_{\mathrm{conv}}(f) + \varepsilon$. Let $h \in \mathcal{C}(L)$ be any function satisfying $\|f - h\|_{L^2(\gamma)} \le L$ (which exists because f is L-close to the constant $|\mathbf{E}[f]|$ function by the Poincaré inequality), so that $\|\boldsymbol{f} - h\|_{L^2(\gamma)} \le L + 2\varepsilon' \le 2L$. We have

$$\begin{aligned}
\|\widetilde{g} - f\|_{L^2(\gamma)} &\le \|\widetilde{g} - \boldsymbol{f}\|_{L^2(\gamma)} + \|\boldsymbol{f} - f\|_{L^2(\gamma)} \\
&\le \|\widetilde{g} - \boldsymbol{f}\|_{L^2(B_\infty(r),\mu)} + 3\varepsilon' && \text{(Lemma 17, see below)} \\
&\le \|h - \boldsymbol{f}\|_{L^2(B_\infty(r),\mu)} + 4\varepsilon' && \text{(Lemma 16)} \\
&\le \|h - \boldsymbol{f}\|_{L^2(\gamma)} + 5\varepsilon' && \text{(Lemma 17 via Remark 18)} \\
&\le \|h - f\|_{L^2(\gamma)} + \|f - \boldsymbol{f}\|_{L^2(\gamma)} + 5\varepsilon' \\
&\le \|h - f\|_{L^2(\gamma)} + 7\varepsilon'\,.
\end{aligned}$$

To justify the first use of Lemma 17, we claim that $\widetilde{g}$ satisfies $|\widetilde{g}(x) - \boldsymbol{f}(x)| \leq 6L = \tau$ at some point $x \in B_\infty(r)$. Indeed, note that the function h satisfies (by Lemma 17 via Remark 18) $\|h - \boldsymbol{f}\|_{L^2(B_\infty(r),\mu)} \leq \|h - \boldsymbol{f}\|_{L^2(\gamma)} + \varepsilon' \leq 3L$; hence, if we had $|\widetilde{g}(x) - \boldsymbol{f}(x)| > 6L$ for all $x \in B_\infty(r)$, we would obtain $\|\widetilde{g} - \boldsymbol{f}\|_{L^2(B_\infty(r),\mu)} > 6L > \|h - \boldsymbol{f}\|_{L^2(B_\infty(r),\mu)} + \varepsilon'$, contradicting Lemma 16.

Since the above holds for all $h \in \mathcal{C}(L)$ sufficiently close to f, we obtain $\|\widetilde{g} - f\|_{L^2(\gamma)} \leq \mathrm{d}^L_{\mathrm{conv}}(f) + \varepsilon$ as desired. ◀

4 One-Sided Sample-Based Testing

We now turn to convexity testing algorithms with *one-sided* error, i.e., algorithms that never incorrectly reject a convex function. Our main result here is the following:

▶ **Theorem 4** (One-sided testing of Lipschitz convex functions). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be L-Lipschitz, and let $\varepsilon \geq 0$. There is an algorithm, One-Sided-Tester (Algorithm 1), which, given access to i.i.d. labeled samples $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x} \sim N(0, I_n)$, makes $O(\sqrt{n}L/\varepsilon)^n$ draws, runs in time $(nL/\varepsilon)^{O(n)}$, and has the following performance guarantee:*

- *If f is convex, then One-Sided-Tester outputs "accept" with probability 1;*
- *If $\mathrm{d}^L_{\mathrm{conv}}(f) \geq \varepsilon$, then One-Sided-Tester outputs "reject" with probability 9/10.*

Note that in contrast to our upper bound for standard testing (Theorem 2), our sample complexity is exponentially worse.

Our one-sided testing algorithm proceeds by constructing an empirical *convex envelope* of the function using labeled samples. We will ensure by design that when f is convex, its empirical convex envelope is close (in an $L^\infty(\gamma)$ sense) to f. On the other hand, when f is far from convex, it will be far from its empirical convex envelope, which will be revealed by the samples. The strong $L^\infty(\gamma)$ guarantee of the convex envelope may be contrasted with our two-sided tester from Theorem 2, which also proceeds via learning but rather in $L^2(\gamma)$ sense (which is a weaker guarantee that more precisely matches our target metric for testing).

4.1 The Empirical Convex Envelope

We will require the following definition:

▶ **Definition 19** (Empirical convex envelope). *For a function $f : \mathbb{R}^n \to \mathbb{R}$ and m points $Y = \{y_1, y_2, \ldots, y_m\} \subset \mathbb{R}^n$, we define the* empirical convex envelope *of f with respect to Y as the function $\overline{f_Y} : \mathbb{R}^n \to \mathbb{R}$ defined by*

$$\overline{f_Y}(x) := \sup\left\{\langle p, x\rangle + a : p \in \mathbb{R}^n, a \in \mathbb{R}, \|p\|_2 \leq L, \langle p, y\rangle + a \leq f(y) \text{ for all } y \in Y\right\}.$$

Informally, $\overline{f_Y}$ is the "largest" convex function that is below f on all the sampled points $y \in Y$. We record two immediate properties of the empirical convex envelope for future use:

▶ **Lemma 20.** *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz, and let $Y = \{y_1, \ldots, y_m\} \subseteq \mathbb{R}^n$. Let $\overline{f_Y} : \mathbb{R}^n \to \mathbb{R}$ be the empirical convex envelope of f with respect to Y.*

1. *The empirical convex envelope $\overline{f_Y}$ is convex and L-Lipschitz.*
2. *Furthermore, if f is itself convex, then for all $y \in Y$, $\overline{f_Y}(y) = f(y)$.*

Proof. Note that convexity of $\overline{f_Y}$ is immediate from Definition 19 since the supremum of affine functions is convex. Additionally, the constraint that $\|p\|_2 \leq L$ ensures that $\overline{f_Y}$ is L-Lipschitz. Together, these imply the first item.

We now turn to the second item. Since $\overline{f_Y}(y_j) \leq f(y_j)$ for all $y_j \in Y$ by construction, it remains to show that $f(y_j) \leq \overline{f_Y}(y_j)$ for $y_j \in Y$. This is readily seen to hold by taking $p = \nabla f(y_j)$ and $a = f(y_j)$. Since f is L-Lipschitz, and by Rademacher's theorem Lipschitz functions are differentiable almost everywhere, it follows that $\|\nabla f(y_j)\|_2 \leq L$. Since $\overline{f_Y}(y_j)$ is a supremum over all feasible p and a, it follows that $\overline{f_Y}(y_j) \geq f(y_j)$. ◀

Next, we note that the empirical convex envelope $\overline{f_Y}$ can be computed on an input $x \in \mathbb{R}^n$ by the quadratically constrained convex program CECE (short for "compute empirical convex envelope") given in Figure 2.

▶ **Lemma 21.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be L-Lipschitz, and let $Y = \{y_1, \ldots, y_m\} \subseteq \mathbb{R}^n$. Let $\overline{f_Y} : \mathbb{R}^n \to \mathbb{R}$ be the empirical convex envelope of f with respect to Y. Then for each $x \in \mathbb{R}^n$, the program CECE(x) is feasible and satisfies CECE$(x) = \overline{f_Y}(x)$. Additionally, CECE(x) runs in time* $\mathrm{poly}(m, n)$.

Here, we are assuming the real-valued model of computation, where infinite-length real-valued computations can be achieved with one unit operation. However, our results work equally well in the bounded-bit model, where we use and assume finite bits of precision, with some minor (and routine) modifications to the analysis.

Proof. Note that the quadratic program in Figure 2 coincides with the definition of the empirical convex envelope from Definition 19, and so correctness holds. (Feasibility can be verified by taking $p = 0$ and a small enough, since the set Y is finite.) The runtime is also immediate via standard guarantees for solving convex programs [15]. ◀

$$
\begin{array}{ll}
\text{maximize} & \langle p, x \rangle + a \\
\text{subject to} & p \in \mathbb{R}^n, a \in \mathbb{R} \\
& \|p\|_2 \leq L \\
& \langle p, y \rangle + a \leq f(y) \quad \forall y \in Y
\end{array}
$$

Figure 2 The quadratic program CECE(x) which takes as input a collection of samples $Y = \{y_1, \ldots, y_m\}$ labeled by an L-Lipschitz function f.

4.2 Algorithm 1 and Technical Results

We now formally describe our one-sided convexity testing algorithm, One-Sided-Tester(cf. Algorithm 1). We also record some useful lemmas before proving Theorem 4. For the remainder of this section, we set $d := \frac{\varepsilon}{8L}$, where ε, L are as in Theorem 4.

The first lemma shows that the distance from f to its convex envelope $\overline{f_Y}$ can be well-approximated by considering only the points in the set Y.

▶ **Lemma 22.** *Let $Y = \{y_1, y_2, \ldots, y_m\}$ be a set of points in $\mathbb{R}^n$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a L-Lipschitz function and let $\overline{f_Y}$ be its empirical convex envelope with respect to Y. Then for all $x \in \mathbb{R}^n$, we have*

$$|\overline{f_Y}(x) - f(x)| \leq \min_{y \in Y} \left\{ \left|\overline{f_Y}(y) - f(y)\right| + 2L\|x - y\|_2 \right\}.$$

Algorithm 1 A one-sided testing algorithm for Lipschitz convex functions.

Input: Access to i.i.d. labeled samples $(\boldsymbol{x}, f(\boldsymbol{x}))$ where f is L-Lipschitz, $\varepsilon \geq 0$

Output: "Accept" or "reject"

One-Sided-Tester:

1. Set
$$t_1 := \left(\frac{cL\sqrt{n}}{\varepsilon}\right)^n$$
for a universal constant c, and draw t_1 samples $(\boldsymbol{x}^{(1)}, f(\boldsymbol{x}^{(1)})), \ldots, (\boldsymbol{x}^{(t_1)}, f(\boldsymbol{x}^{(t_1)}))$. Let $\boldsymbol{Y} := \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t_1)}\}$.
2. Let $\mathcal{P}$ be a partition of $B_2(1.1\sqrt{n})$ into at most $(3nL/\varepsilon)^n$ (solid) hypercubes of side length $\ell = \lceil \varepsilon/(4L\sqrt{n}) \rceil$. For each hypercube $C \in \mathcal{P}$, check if there exists $\boldsymbol{x}^{(i)} \in C$; if not, halt and return "accept."
3. If $f(\boldsymbol{x}^{(j)}) \neq \mathrm{CECE}(\boldsymbol{x}^{(j)})$ for any $j \in [t_1]$, then halt and return "reject." Otherwise, return "accept."

Proof. Fix $x \in \mathbb{R}^n$. For each $y \in Y$, we have

$$\begin{aligned} |\overline{f_Y}(x) - f(x)| &\leq |\overline{f_Y}(x) - \overline{f_Y}(y)| + |\overline{f_Y}(y) - f(y)| + |f(x) - f(y)| && \text{(triangle inequality)} \\ &\leq |\overline{f_Y}(y) - f(y)| + 2L \cdot \|x - y\|_2 && (f, \overline{f_Y} \text{ are } L\text{-Lipschitz}) \end{aligned}$$

as desired. ◀

The following lemma shows that with high probability, a draw of roughly $\exp(n \log n)$ samples from $N(0, I_n)$ will be sufficiently "dense" in $B_2(1.1\sqrt{n})$, which will help us bound the error term in Lemma 22. The proof computes a lower bound of the Gaussian density inside $B_2(1.1\sqrt{n})$, and then uses a union bound to show that, with high constant probability, each small cube in a partition of $B_2(1.1\sqrt{n})$ contains at least one sample.

▶ **Lemma 23.** *Let t_1 be as in Algorithm 1, and suppose $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t_1)} \sim N(0, I_n)$ are i.i.d. draws. Then with probability 0.99, for all $x \in B_2(1.1\sqrt{n})$ we have*

$$\min_{i \in [t_1]} \|x - \boldsymbol{x}^{(i)}\| \leq d\,.$$

Proof. Note that $B_2(1.1\sqrt{n}) \subseteq B_\infty(1.1\sqrt{n})$. Partition $B_\infty(1.1\sqrt{n})$ into a hypergrid of at most $(1.1\sqrt{n}\ell^{-1})^n \leq \left(3nd^{-1}\right)^n$ (hyper)cubes, each of side length $\ell := \lceil d/\sqrt{n} \rceil$. Suppose for every cube in the grid there exists a point $y \in Y$ in the cube. Then note that

$$\min_{y \in Y} \|x - y\|_2 \leq \ell \cdot \sqrt{n} \leq d$$

thanks to our choice of ℓ.

Thus, in order to establish Lemma 23, it suffices to show that after drawing t_1 samples, with probability at least 9/10, every cube in the hypergrid contains at least one sample. We will show this by taking a union bound over the $(3nd^{-1})^n$ cubes in the hypergrid.

In this argument, it will be necessary to consider the cube in the hypergrid with minimal Gaussian measure. Note that the maximum ℓ_2 norm of any point in the hypergrid is at most $(1.1 + \ell)\sqrt{n}$. Therefore for all $x \in B_\infty(1.1\sqrt{n})$, we have

$$\frac{\exp\left(-||x||_2^2/2\right)}{(2\pi)^{n/2}} \geq \frac{\exp\left(-(1.1+\ell)^2/(2n)\right)}{(2\pi)^{n/2}} = \frac{1}{(2\pi)^{n/2}} \cdot \exp\left(-\left(1.1 + \frac{\varepsilon}{(4L\sqrt{n})}\right)^2 \cdot \frac{1}{2n}\right).$$

Since the Lebesgue measure of any cube is ℓ^n, the minimum Gaussian measure of any of the cubes in our hypergrid is at least

$$\alpha := \frac{1}{(2\pi)^{n/2}} \cdot \exp\left(-\left(1.1 + \frac{\varepsilon}{(4L\sqrt{n})}\right)^2 \cdot \frac{1}{2n}\right) \cdot \ell^n \,.$$

The probability that t_1 independent draws from $N(0, I_n)$ all fail to land in this cube is

$$(1-\alpha)^{t_1} \leq \exp(-\alpha t_1) \,.$$

Applying a union bound, it follows that the probability that there exists a cube in the hypergrid not containing any sample is at most

$$\exp(-\alpha t_1) \cdot 2^{O(n \log(1.1n/d)} \,,$$

and it is readily verified that this is at most 0.01 for our choice of t_1 and ℓ, for some universal constant c. ◀

Finally, the next lemma shows that, under the density condition above, if f and $\overline{f_Y}$ agree on all points in Y then they are close in $L^2(\gamma)$.

▶ **Lemma 24.** *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is an L-Lipschiz function and $Y = \{y_1, y_2, \ldots, y_m\}$ is a set of points in $\mathbb{R}^n$ such that for all $x \in B(1.1\sqrt{n})$, there exists $i \in [m]$ such that $\|x - y_i\|_2 \leq d$, for $d = \varepsilon/(8L)$. Let $\overline{f_Y}$ be the empirical convex envelope of f with respect to Y. If $f(y_i) = \overline{f_Y}(y_i)$ for all $i \in [m]$, then $\left\|f - \overline{f_Y}\right\|_{L^2(\gamma)} \leq \varepsilon/2$.*

Proof. We decompose $\left\|f - \overline{f_Y}\right\|^2_{L^2(\gamma)}$ into the regions inside and outside $B_2(1.1\sqrt{n})$ and apply Lemma 22:

$$\begin{aligned}
&\left\|f - \overline{f_Y}\right\|^2_{L^2(\gamma)} \\
&= \mathop{\mathbf{E}}_{\boldsymbol{x}\sim N(0,I_n)}\left[\left|f(\boldsymbol{x}) - \overline{f_Y}(\boldsymbol{x})\right|^2\right] \\
&\leq \mathop{\mathbf{E}}_{\boldsymbol{x}\sim N(0,I_n)}\left[\left|f(\boldsymbol{x}) - \overline{f_Y}(\boldsymbol{x})\right|^2 \,\middle|\, \|\boldsymbol{x}\|_2 \leq 1.1\sqrt{n}\right] \\
&\quad + \mathop{\mathbf{E}}_{\boldsymbol{x}\sim N(0,I_n)}\left[\left|f(\boldsymbol{x}) - \overline{f_Y}(\boldsymbol{x})\right|^2 \,\middle|\, \|\boldsymbol{x}\|_2 > 1.1\sqrt{n}\right] \cdot \mathbf{Pr}\left[\|\boldsymbol{x}\|_2 > 1.1\sqrt{n}\right] \\
&\leq (2Ld)^2 + (2L)^2 \mathop{\mathbf{E}}_{\boldsymbol{x}\sim N(0,I_n)}\left[\min_{y\in Y} \|\boldsymbol{x} - y\|_2^2 \mid \|\boldsymbol{x}\|_2 > 1.1\sqrt{n}\right] \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim N(0,I_n)}\left[\|\boldsymbol{x}\|_2 > 1.1\sqrt{n}\right] \\
&= (2Ld)^2 + (2L)^2 \cdot \mathrm{poly}(n)\exp(-\Omega(n)) \\
&= \varepsilon^2/16 + o(1) \,,
\end{aligned}$$

which is less than $\varepsilon^2/4$ for large enough n. ◀

4.3 Proof of Theorem 4

Proof. First, note that the sample complexity of One-Sided-Tester is $O\left(\sqrt{n}L/\varepsilon\right)^n$ as desired. We establish correctness as well as the runtime bound below, starting with the former.

Correctness. We first show that if f is L-Lipschitz and convex, then One-Sided-Tester accepts with probability 1. Indeed, Algorithm 1 only outputs "reject" when $f(y) \neq \mathrm{CECE}(y)$ for some $y \in Y$, which cannot happen by Lemmas 20 and 21.

For soundness, suppose that $\mathrm{d}^L_{\mathrm{conv}}(f) \geq \varepsilon$, so that in particular $\left\|f - \overline{f_Y}\right\|_{L^2(\gamma)} \geq \varepsilon$. Suppose the algorithm samples a set Y of points in $\mathbb{R}^n$ such that for all $x \in B_2(1.1\sqrt{n})$, there exists $y \in Y$ such that $\|x - y\|_2 \leq d$, for $d = \varepsilon/(8L)$; by Lemma 23 this occurs with probability at least 0.99. In this case the algorithm rejects, because it would only accept if we had $f(y) = \mathrm{CECE}(y)$ for every $y \in Y$; since $\mathrm{CECE}(y) = \overline{f_Y}(y)$ for each $y \in Y$ by Lemma 21, this would imply via Lemma 24 that $\left\|f - \overline{f_Y}\right\|_{L^2(\gamma)} \leq \varepsilon/2$, a contradiction. Therefore the algorithm rejects with probability at least 0.99.

Runtime. Note that Step 1 takes time t_1 and Step 2 takes time

$$|\mathcal{P}| \cdot t_1 = (3nL/\varepsilon)^n \cdot \left(\frac{50L\sqrt{n}}{\varepsilon}\right)^n .$$

Finally, Step 3 requires time $\mathrm{poly}(n, t_1)$ by Lemma 21. It follows that Algorithm 1 uses time $(nL/\varepsilon)^{O(n)}$. ◀

References

1. Amir Ali Ahmadi, Alex Olshevsky, Pablo A Parrilo, and John N Tsitsiklis. NP-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical programming*, 137:453–476, 2013. doi:10.1007/s10107-011-0499-2.
2. Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning sparse polynomial functions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 500–510. SIAM, 2014. doi:10.1137/1.9781611973402.37.
3. Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and Geometry of Markov Diffusion Operators*. Grundlehren der mathematischen Wissenschaften. Springer Cham, 2014. doi:10.1007/978-3-319-00227-9.
4. Aleksandrs Belovs, Eric Blais, and Abhinav Bommireddi. Testing convexity of functions over finite domains. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2030–2045. SIAM, 2020. doi:10.1137/1.9781611975994.125.
5. Omri Ben-Eliezer. Testing Local Properties of Arrays. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 11:1–11:20, 2019. doi:10.4230/LIPIcs.ITCS.2019.11.
6. Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 164–173, 2014. doi:10.1145/2591796.2591887.
7. Arnab Bhattacharyya and Yuichi Yoshida. *Property Testing: Problems and Techniques*. Springer, 2022. doi:10.1007/978-981-16-8622-1.
8. Hadley Black. Nearly optimal bounds for sample-based testing and learning of k-monotone functions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2024)*, pages 37–1. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.APPROX/RANDOM.2024.37.
9. Hadley Black, Eric Blais, and Nathaniel Harms. Testing and Learning Convex Sets in the Ternary Hypercube. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, pages 15–1. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ITCS.2024.15.
10. Eric Blais and Abhinav Bommireddi. On testing and robust characterizations of convexity. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 18:1–18:15, 2020. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.18.

11 Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions over hypergrid domains. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 309–320. IEEE, 2014. doi:10.1109/CCC.2014.38.

12 Avrim Blum, Merrick L. Furst, Jeffrey C. Jackson, Michael J. Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using fourier analysis. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 253–262. ACM, 1994. doi:10.1145/195058.195147.

13 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 73–83, 1990. doi:10.1145/100216.100225.

14 Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT 2010*, pages 177–186. Physica-Verlag HD, 2010. doi:10.1007/978-3-7908-2604-3_16.

15 Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi:10.1017/CBO9780511804441.

16 Nader H. Bshouty and Vitaly Feldman. On using extended statistical queries to avoid membership queries. *J. Mach. Learn. Res.*, 2:359–395, 2002. URL: https://jmlr.org/papers/v2/bshouty02a.html.

17 Nader H Bshouty and Christino Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM (JACM)*, 43(4):747–770, 1996. doi:10.1145/234533.234564.

18 Deeparnab Chakrabarty and Comandur Seshadhri. A $o(n)$ monotonicity tester for boolean functions over the hypercube. In *Proceedings of the forty-fifth annual ACM symposium on Theory of Computing*, pages 411–418, 2013. doi:10.1145/2488608.2488660.

19 Sitan Chen, Aravind Gollakota, Adam R. Klivans, and Raghu Meka. Hardness of noise-free learning for two-hidden-layer neural networks. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/45a7ca247462d9e465ee88c8a302ca70-Abstract-Conference.html.

20 X. Chen, A. Freilich, R. Servedio, and T. Sun. Sample-based high-dimensional convexity testing. In *Proceedings of the 17th Int. Workshop on Randomization and Computation (RANDOM)*, pages 37:1–37:20, 2017. doi:10.4230/LIPIcs.APPROX-RANDOM.2017.37.

21 Xi Chen, Anindya De, Shivam Nadimpalli, Rocco A Servedio, and Erik Waingarten. Lower bounds for convexity testing. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 446–488. SIAM, 2025. doi:10.1137/1.9781611978322.13.

22 Anindya De, Elchanan Mossel, and Joe Neeman. Is your function low dimensional? In Alina Beygelzimer and Daniel Hsu, editors, *Conference on Learning Theory, COLT 2019*, volume 99 of *Proceedings of Machine Learning Research*, pages 979–993. PMLR, 2019. URL: http://proceedings.mlr.press/v99/de19a.html.

23 Anindya De, Elchanan Mossel, and Joe Neeman. Robust testing of low dimensional functions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 584–597, 2021. doi:10.1145/3406325.3451115.

24 Anindya De, Shivam Nadimpalli, and Rocco A. Servedio. Quantitative correlation inequalities via semigroup interpolation. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021*, volume 185 of *LIPIcs*, pages 69:1–69:20, 2021. doi:10.4230/LIPIcs.ITCS.2021.69.

25 Anindya De, Shivam Nadimpalli, and Rocco A. Servedio. Convex influences. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS*, volume 215 of *LIPIcs*, pages 53:1–53:21, 2022. doi:10.4230/LIPIcs.ITCS.2022.53.

26 Anindya De, Shivam Nadimpalli, and Rocco A. Servedio. Testing Convex Truncation. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4050–4082, 2023. doi:10.1137/1.9781611977554.ch155.

27 Anindya De, Shivam Nadimpalli, and Rocco A Servedio. Gaussian approximation of convex sets by intersections of halfspaces. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1911–1930. IEEE, 2024. doi:10.1109/FOCS61266.2024.00115.

28 Anindya De and Rocco Servedio. Weak learning convex sets under normal distributions. In *Conference on Learning Theory*, pages 1399–1428. PMLR, 2021. URL: http://proceedings.mlr.press/v134/de21a.html.

29 Ronald A. DeVore and George G. Lorentz. *Constructive Approximation*, volume 303 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, Berlin, 1993.

30 Ilias Diakonikolas, Daniel Kane, and Nikos Zarifis. Near-optimal SQ lower bounds for agnostically learning halfspaces and relus under gaussian marginals. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/9d7311ba459f9e45ed746755a32dcd11-Abstract.html.

31 Ilias Diakonikolas, Daniel M Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Agnostic proper learning of halfspaces under gaussian marginals. In *Conference on Learning Theory*, pages 1522–1551. PMLR, 2021. URL: http://proceedings.mlr.press/v134/diakonikolas21b.html.

32 Lutz Dümbgen. Shape-constrained statistical inference. *Annual Review of Statistics and Its Application*, 11, 2024. doi:10.1146/annurev-statistics-033021-014937.

33 Vitaly Feldman. A complete characterization of statistical query learning with applications to evolvability. *J. Comput. Syst. Sci.*, 78(5):1444–1459, 2012. doi:10.1016/j.jcss.2011.12.024.

34 Renato Ferreira Pinto Jr. Directed Isoperimetry and Monotonicity Testing: A Dynamical Approach. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2295–2305. IEEE, 2024. doi:10.1109/FOCS61266.2024.00134.

35 Surbhi Goel, Aravind Gollakota, and Adam R. Klivans. Statistical-query lower bounds via functional gradients. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/17257e81a344982579af1ae6415a7b8c-Abstract.html.

36 Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. doi:10.1017/9781108135252.

37 Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

38 Oded Goldreich, Shafi Goldwassert, Eric Lehman, and Dana Ron. Testing monotonicity. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 426–435. IEEE, 1998. doi:10.1109/SFCS.1998.743493.

39 Bram L Gorissen, İhsan Yanıkoğlu, and Dick Den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015. doi:10.1016/j.omega.2014.12.006.

40 Piet Groeneboom and Geurt Jongbloed. *Nonparametric estimation under shape constraints*. Cambridge University Press, 2014. doi:10.1017/CBO9781139020893.

41 Adityanand Guntuboyina and Bodhisattva Sen. Nonparametric shape-restricted regression. *Statistical Science*, 33(4):568–594, 2018. doi:10.1214/18-STS665.

42 Anxin Guo and Aravindan Vijayaraghavan. Agnostic Learning of Arbitrary ReLU Activation under Gaussian Marginals. In *The Thirty Eighth Annual Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 2592–2631. PMLR, 2025. URL: https://proceedings.mlr.press/v291/guo25a.html.

43 Prahladh Harsha, Adam Klivans, and Raghu Meka. An invariance principle for polytopes. *Journal of the ACM (JACM)*, 59(6):1–25, 2013. doi:10.1145/2395116.2395118.

44 Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009. doi:10.1007/978-0-387-84858-7.

45 Aicke Hinrichs, Erich Novak, and Henryk Woźniakowski. The curse of dimensionality for the class of monotone functions and for the class of convex functions. *Journal of Approximation Theory*, 163(8):955–965, 2011. doi:10.1016/j.jat.2011.02.009.

46 Daniel Hsu, Clayton H. Sanford, Rocco A. Servedio, and Emmanouil Vasileios Vlatakis-Gkaragkounis. On the Approximation Power of Two-Layer Networks of Random ReLUs. In *Conference on Learning Theory, COLT 2021*, volume 134 of *Proceedings of Machine Learning Research*, pages 2423–2461. PMLR, 2021. URL: http://proceedings.mlr.press/v134/hsu21a.html.

47 Yaozhong Hu. Itô-wiener chaos expansion with exact residual and correlation, variance inequalities. *Journal of Theoretical Probability*, 10(4):835–848, 1997. doi:10.1023/A:1022654314791.

48 Michael J Kearns and Umesh Vazirani. *An introduction to computational learning theory*. MIT Press, 1994. URL: https://mitpress.mit.edu/books/introduction-computational-learning-theory.

49 Zander Kelley and Raghu Meka. Random Restrictions and PRGs for PTFs in Gaussian Space. In *37th Computational Complexity Conference (CCC 2022)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 21:1–21:24, 2022. doi:10.4230/LIPIcs.CCC.2022.21.

50 Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and Boolean isoperimetric-type theorems. *SIAM Journal on Computing*, 47(6):2238–2276, 2018. doi:10.1137/16M1065872.

51 Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002. doi:10.1137/S1052623499363220.

52 A. Klivans, R. O'Donnell, and R. Servedio. Learning geometric concepts via gaussian surface area. In *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 541–550, 2008. doi:10.1109/focs.2008.64.

53 Pravesh Kothari, Amir Nayyeri, Ryan O'Donnell, and Chenggang Wu. Testing surface area. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*. SIAM, 2014. doi:10.1137/1.9781611973402.89.

54 Abhiruk Lahiri, Ilan Newman, and Nithin Varma. Parameterized convexity testing. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 174–181. SIAM, 2022. doi:10.1137/1.9781611977066.12.

55 Jane Lange, Ephraim Linder, Sofya Raskhodnikova, and Arsen Vasilyan. Local lipschitz filters for bounded-range functions with applications to arbitrary real-valued functions. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2881–2907. SIAM, 2025. doi:10.1137/1.9781611978322.93.

56 Jane Lange, Ronitt Rubinfeld, and Arsen Vasilyan. Properly learning monotone functions via local correction. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 75–86. IEEE, 2022. doi:10.1109/FOCS54457.2022.00015.

57 Jane Lange and Arsen Vasilyan. Robust learning of halfspaces under log-concave marginals. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL: https://proceedings.neurips.cc/paper_files/paper/2025/hash/91a5bb5e5939cb075f5f2464d7b8bbf0-Abstract-Conference.html.

58 Jane Lange and Arsen Vasilyan. Agnostic proper learning of monotone functions: beyond the black-box correction barrier. *SIAM Journal on Computing*, 55(3):FOCS23–1–FOCS23–32, 2026. doi:10.1137/24M1630463.

59 Grigorii Aleksandrovich Margulis. Probabilistic characteristics of graphs with large connectivity. *Problemy peredachi informatsii*, 10(2):101–108, 1974. URL: http://mi.mathnet.ru/ppi1034.

60 K. Matulef, R. O'Donnell, R. Rubinfeld, and R. Servedio. Testing halfspaces. *SIAM J. on Comput.*, 39(5):2004–2047, 2010. doi:10.1137/070707890.

61 Rahul Mazumder, Arkopal Choudhury, Garud Iyengar, and Bodhisattva Sen. A computational framework for multivariate convex regression and its variants. *Journal of the American Statistical Association*, 114(525):318–331, 2019. doi:10.1080/01621459.2017.1407771.

62 R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. URL: http://www.cambridge.org/de/academic/subjects/computer-science/algorithmics-complexity-computer-algebra-and-computational-g/analysis-boolean-functions.

63 Ramesh Krishnan S Pallavoor, Sofya Raskhodnikova, and Nithin Varma. Parameterized property testing of functions. *ACM Transactions on Computation Theory (TOCT)*, 9(4):1–19, 2017. doi:10.1145/3155296.

64 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. On testing convexity and submodularity. *SIAM Journal on Computing*, 32(5):1158–1184, 2003. doi:10.1137/S0097539702414026.

65 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006. doi:10.1016/j.jcss.2006.03.002.

66 Luis Rademacher and Santosh Vempala. Testing geometric convexity. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 469–480, 2005. doi:10.1007/978-3-540-30538-5_39.

67 Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. URL: https://www.jstor.org/stable/2236626.

68 R. Tyrrell Rockafellar. *Convex Analysis*, volume 28 of *Princeton Landmarks in Mathematics and Physics*. Princeton University Press, 1970.

69 R Tyrrell Rockafellar. Lagrange multipliers and optimality. *SIAM review*, 35(2):183–238, 1993. doi:10.1137/1035044.

70 Günter Rote. The convergence rate of the sandwich algorithm for approximating convex functions. *Computing*, 48(3):337–361, 1992. doi:10.1007/BF02238642.

71 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151.

72 Emilio Seijo and Bodhisattva Sen. Nonparametric least squares estimation of a multivariate convex regression function. *The Annals of Statistics*, 39(3):1633, 2011. doi:10.1214/10-AOS852.

73 Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. URL: http://www.cambridge.org/de/academic/subjects/computer-science/pattern-recognition-and-machine-learning/understanding-machine-learning-theory-algorithms.

74 Gyorgy Sonnevend. An optimal sequential algorithm for the uniform approximation of convex functions on $[0,1]^2$. *Applied Mathematics and Optimization*, 10(1):127–142, 1983. doi:10.1007/BF01448382.

75 Balázs Szörényi. Characterizing statistical query learning: Simplified notions and proofs. In *Algorithmic Learning Theory, 20th International Conference, ALT*, pages 186–200. Springer, 2009. doi:10.1007/978-3-642-04414-4_18.

76 Michel Talagrand. Isoperimetry, logarithmic sobolev inequalities on the discrete cube, and margulis' graph connectivity theorem. *Geometric & Functional Analysis GAFA*, 3(3):295–314, 1993. doi:10.1007/BF01895691.

77 Michel Talagrand. How much are increasing sets positively correlated? *Combinatorica*, 16(2):243–258, 1996. doi:10.1007/BF01844850.

78 Santosh S. Vempala. Learning Convex Concepts from Gaussian Distributions with PCA. In *51th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 124–130, 2010. doi:10.1109/FOCS.2010.19.