On the Sample Complexity of Lipschitz Learning Algorithms

Anonymous authors
Paper under double-blind review

Abstract

Estimating the Lipschitz constant of a function, also known as Lipschitz learning, is a fundamental problem with broad applications in fields such as control and global optimization. In this paper, we study the Lipschitz learning problem with minimal parametric assumptions on the target function. As a first theoretical contribution, we derive novel lower bounds on the sample complexity of this problem for both noise-free and noisy settings under mild assumptions. Moreover, we propose a simple Lipschitz learning algorithm called *Lipschitz Constant Estimation by Least Squares Regression* (referred to as LCLS). We show that LCLS is asymptotically consistent and offers finite sample guarantees that can be translated to new upper bounds on the sample complexity of the Lipschitz learning problem. Our analysis shows that the sample complexity of LCLS is optimal in the general noise-free setting. Furthermore, we show that by design, the LCLS algorithm is computationally faster than existing theoretically consistent methods, and can be readily adapted to various noise assumptions with little to no prior knowledge of the target function properties or noise distribution.

1 Introduction

Many applications are based on the Lipschitz continuity of an objective or target function and depend explicitly on the value of a Lipschitz constant. Examples include: robustness analysis which utilises Lipschitz constants to characterise worst-case behaviour in control settings (Limon et al. (2017), Canale et al. (2014)), system identification which leverages the Lipschitz continuity property to obtain worst-case prediction error bounds for interpolation methods (Milanese and Novara (2004),Beliakov (2006), Calliess et al. (2020)), global optimization algorithms which rely on precise Lipschitz constant estimates to ensure speedy convergence (Jones et al. (1993), González et al. (2016), Malherbe and Vayatis (2017)), multi-armed bandit problems which utilise the Lipschitz constants to obtain asymptotic lower bounds and design algorithms (Magureanu et al. (2014)) or reinforcement learning which utilises the Lipschitz constant to construct safe initial policies (Chakrabarty et al. (2020)). For these applications it is critical that the Lipschitz constant estimate used is sufficiently precise in order to ensure satisfactory performance in their specified goal. A main practical drawback is therefore the dependency on the assumption of prior knowledge of the Lipschitz constant or, in the case that this assumption is not made, the necessity of having to learn a precise Lipschitz constant estimate from data.

Consequently, a number of Lipschitz constant estimation methods (also called Lipschitz learning algorithms) have been developed. For target functions belonging to families of parametric models, Lipschitz learning approaches generally utilise the structure of the model class to obtain precise estimates (e.g. see extensive literature on the estimation of Lipschitz constants of Neural networks (Scaman and Virmaux (2018)) (Fazlyab et al. (2019))). For frameworks that don't consider a particular parametric family, a majority of the existing approaches are black-box methods that utilise and extend Strongin's classical estimator (Strongin (1973)): $\hat{L} := r \max_{i \neq j} \frac{|f_i - f_j|}{\|x_i - x_j\|}$ where $r \in \mathbb{R}$ is a multiplicative factor and (x_i, f_i) is a data sample with $f_i = f(x_i)$. In particular, we highlight: (Wood and Zhang (1996)) builds on Strongin's estimator by fitting an approximate reverse Weibull distribution to the Lipschitz estimate in the one-dimensional case and using the location parameter as a Lipschitz estimate, (Sergeyev (1995)) utilises Strongin's approach to compute local Lipschitz constant estimates and extends the approach to the multidimensional case by using space filling curves in

order to solve a global optimization problem and a more recent approach (Strongin et al. (2019)) proposes dual Strongin Lipschitz estimates: with two differing "local" and "global" multiplicative factors. We remark that the class of Lipschitz learning algorithms described so far does not consider the possibility of observational noise and can explode in value if it exists. In this case, we can consider the Lispchitz constant estimator proposed by both (Novara et al. (2013)) and (Calliess et al. (2020)) which specifically extends Strongin's estimate to deal with bounded observational noise.

Alternative Lipschitz learning approaches that do not directly utilise Strongin's estimate have also been developed. These generally can consider the case of observational noise and include: (Beliakov (2005)) which utilises a short optimisation problem and cross validation/sample splitting to obtain Lipschitz constant estimates, (Bubeck et al. (2011)) which employs a similar idea to Strongin's estimate in order to propose a Lipschitz constant estimator in the context of the Lipschitz multi-armed bandit problem, (González et al. (2016)) which generates Lipschitz constant estimates using the mean function of the gradient function estimate of a fitted Gaussian Process (GP) and is directly computable using the GP-associated covariance function, and (Calliess (2017)) which obtains Lipschitz constant estimates by optimising a Lipschitz interpolation method. Unfortunately, while this class of Lipschitz learning algorithms tends to work well in practice, they generally do not guarantee asymptotic convergence and are of limited theoretical interest.

Despite the wide range of proposed Lipschitz learning algorithms, there has been little theoretical investigation into the Lipschitz constant estimation problem other than various consistency proofs of Strongin-based estimators. It is generally understood that this learning problem, without making further restrictive assumptions on the underlying space of target functions, is inevitably subject to the curse of dimensionality. However, to the best of our knowledge, this intuition has not been explored formally. A first goal of this paper is therefore to provide a theoretical investigation into the Lipschitz learning problem by deriving lower bounds on the sample complexity in the case of both a noiseless and noisy sampling settings. We confirm the general intuition on the difficulty of the Lipschitz learning problem by demonstrating that the problem has a sample complexity lower bound that scales at least exponentially on the function input dimension in the noiseless case and at least exponentially on two times the function input dimension in the noisy sampling case.

Our theoretical results imply that a precise estimation of the Lipschitz constant requires a significant number of samples. This is computationally problematic for classical Strongin-based Lipschitz learning algorithms due to the fact that the computational complexity of these methods can be shown to be quadratic in the number of samples: $O(n_{samples}^2)$. While the non-Strongin based estimators discussed above could potentially be less computationally expensive, they only provide heuristic or experimental convergence guarantees and are generally difficult to study from a theoretical perspective. Therefore, in light of our lower bounds on sample complexity, we propose a novel algorithm for Lipschitz learning called LCLS (short for *Lipschitz Constant estimation by Least Squares regression*) that has a linear computational complexity in the number of sample points and for which we can derive theoretical guarantees on asymptotic convergence and finite sample behaviour. The optimality of the lower bound on the sample complexity of the Lipschitz learning problem in the noiseless sampling setting derived in the first part of the paper and a relatively tight upper bound on the sample complexity of the Lipschitz learning problem in the noisy sampling setting follow from these theoretical results.

In practice, the proposed LCLS algorithm provides a theoretically motivated and computationally quick way of estimating the Lipschitz constant. With minimal fine-tuning, LCLS can be plugged into any computational method that utilises a Lipschitz constant estimate – see above discussion – under any sampling noise assumptions. We provide an example of such a procedure in the context of nonparametric regression for system identification in control by combining the LCLS algorithm with a classical nonlinear set-membership/Lipschitz interpolation framework and illustrating the empirical performance of the combined regression method through a series of short experiments.

A more comprehensive list of the contributions of this paper is given below.

1.1 Contributions and Outline of Paper

In this paper, we provide a rigorous treatment of the Lipschitz constant estimation problem discussed above. In particular, we make the following contributions:

1. In Section 2, we provide novel theoretical lower bounds on the sample complexity of the general Lipschitz learning problem (for $p \in \{1, 2\}$) in noiseless sampling (see Theorem 2.2) and noisy sampling (see Theorem 2.5) set-ups when the target function $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ satisfies a regularity condition $C^2(\mathcal{X}, K)$ defined in Assumption 2. We note that these bounds can be equivalently stated as lower bounds on the convergence rate of the general Lipschitz learning problem (see Corollaries 2.3 and 2.7). As far as the authors are aware, the sample complexity and convergence bounds derived in this paper are the first theoretical results pertaining to the convergence of the general Lipschitz learning problem. We show in Section 3 that our proposed lower bound on the sample complexity rate is optimal in the noiseless sampling setting and differs only slightly in the noisy sampling setting.

2. In Section 3.1, we propose a least squares-based Lipschitz learning (LCLS) approach that utilises a partition of the input space $\mathcal{X}$ and local least squares estimates in order to generate a Lipschitz constant estimate. As discussed in the introduction, our motivation for developing the LCLS algorithm rests on the following two points:
   - both theoretically and computationally tractable.
   - directly applicable across all noise settings considered in the literature and without any prior knowledge of target function properties or of the noise structure.

3. In Sections 3.2 and 3.3, we investigate theoretical properties of the proposed algorithm:
   - Asymptotic convergence for general partition choice in noiseless and noisy sampling set-ups (Section 3.2, see Theorem 3.6).
   - Finite sample guarantees in the noiseless and noisy sampling set-ups when the partition is constructed using regular hypercubes (Section 3.3, see Theorem 3.9). These guarantees provide an upper bound on the sample complexity of the general Lipschitz learning problem and show that the complexity rates derived in the first part of the paper match in the noiseless case and are off by a factor of $\sim (\frac{K^2}{\epsilon^4})$ in the noisy one. (Section 3.3, see Remark 3.10, 3.11).

4. In Section 3.4, we illustrate and compare the empirical performance of the LCLS algorithm against Strongin-based Lipschitz learning algorithms on a set of test functions. We consider both the noiseless and noisy sampling settings. We find that while the benchmark Strongin-based algorithms converges slightly faster in terms of number of samples, our proposed algorithm converges faster in terms of computation time for all functions in the test set (see Figure 4). This is despite the fact that we consider settings for which the benchmark algorithms are specifically designed.

5. In Section 4, we explore the application of the various theoretical results and the LCLS algorithm derived in this paper to the fields of *Global Optimisation* and *Nonparametric Regression for System Identification*. More specifically, we propose a lower bound on the sample complexity of adaptive Lipschitz optimisation algorithms that follows from one of the theoretical results stated in Section 2 and a new nonparametric regression method constructed by combining the LCLS algorithm of Section 3 with a classical nonlinear set membership framework (see Milanese and Novara (2004)).

2 Assumptions and Sample Complexity Lower Bound

In this section, we provide the standing assumptions of the paper and state the main results pertaining to theoretical lower bounds on the sample complexity of Lipschitz learning algorithms.

2.1 Basic Assumptions

Let $p \in \mathbb{N}$, $d \in \mathbb{N}$, a function $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ is said to be Lipschitz continuous with respect to a norm $\|.\|_p$ if there exists a positive real value $L_p(f) \in \mathbb{R}$ such that $\|x - y\|_p \leq L_p(f)|f(x) - f(y)|$, for all $x, y \in \mathcal{X}$. The

smallest constant satisfying this condition denoted $L_p^*(f)$ is called the (best) Lipschitz constant and can be interpreted as the tightest bound on the rate of change of f. Furthermore, for $L \geq 0$, we define the class of Lipschitz continuous functions as

$$\mathcal{F}_p(L) := \{h : \mathcal{X} \to \mathbb{R} | h \text{ is Lipschitz } \wedge L_p^*(h) = L\}.$$

The Lipschitz learning problem therefore considers the estimation of $L_p^*(f)$ when f is an unknown target function. As described in the introduction, we consider a general version of this problem where f is considered black-box and can can only be accessed through queries to a, possibly noisy, oracle. As f is not assumed to belong to any parametric family, other assumptions are needed[1] in order to derive theoretical bounds on the sample complexity. For our results, we make the following two assumptions on the input space $\mathcal{X}$ and the regularity of f.

Assumption 1 *(Domain) The domain $\mathcal{X}$ of the target function f is a convex and compact sub-set of $\mathbb{R}^d$.*

Assumption 2 *(Functional) The target function $f \in C^2(\mathcal{X})$ and there exists an upper bound $K \in \mathbb{R}_+$ on the second-order partial derivatives of f, i.e. $|\frac{\partial f}{\partial x_i \partial x_j}| \leq K$ for all $x \in \mathcal{X}$ and $i, j \in \{1, ..., d\}$.*

For a given $K \in \mathbb{R}_+$, we denote by $C^2(\mathcal{X}, K)$ the class of functions that satisfies Assumption 2 with an upper bound K on the second degree partial derivatives. It is important to point out that this bound does need to be tight and that if Assumption 1 holds then any $f \in C^2(\mathcal{X})$ automatically belongs to $C^2(\mathcal{X}, \bar{K})$ for some $\bar{K} \in \mathbb{R}_+$. Finally, we assume that we have access to the target function f through an oracle $\Omega : \mathcal{X} \to \mathbb{R}$ – defined formally below for each sampling setting – which can be queried in order to generate observations of f. In particular, this oracle can be freely used by any Lipschitz learning algorithm as described in the following definition.

Definition 2.1 *(Lipschitz Learning Algorithms) We define $\mathcal{L}_{n,p}(\mathcal{X})$ as the set of all $\|.\|_p$-Lipschitz learning algorithms that utilise at most $n \in \mathbb{N}$ queries to the Oracle Ω with inputs in $\mathcal{X}$. The sampling procedure is considered to be a part of the Lipschitz learning algorithm and $\forall \hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})$ we denote the set of generated samples by $G_{\mathcal{X}}^{\hat{L}} = \{(x_i^{\hat{L}}, \Omega(x_i^{\hat{L}}))_{i=1,...,n}\}$.*

We note that Definition 2.1 defines a general class of Lipschitz learning algorithms without any structural specifications and that the inclusion of the sampling procedure in the algorithm is common for applications in both control and global optimisation.

2.2 Noiseless Sampling Setting

Assumptions (1)-(2) are sufficient to formulate a lower bound on the sample complexity rate of the Lipschitz learning problem in the case where one has access to an oracle[2] Ω that can be queried to obtain noiseless observations of the underlying target function. Formally, the noiseless Oracle is described by

$$\begin{aligned} \Omega : \ & \mathcal{X} \to \mathbb{R} \\ & x \overset{\Omega}{\mapsto} f(x). \end{aligned}$$

The lower bound on the sample complexity of any Lipschitz learning algorithm is given in the following theorem.

Theorem 2.2 *(Sample Complexity Bound – Noiseless) Let $M \in \mathbb{R}_+$, $d \in \mathbb{N}$, $p \in \{1, 2\}$ and suppose $\mathcal{X} := [0, M]^d$. Assume that Assumption (2) holds and that a noiseless Oracle Ω, (described above) is available. $\forall L^* \geq 0, \forall \epsilon \in (0, MK)$, if*

$$\inf_{\hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})} \sup_{f \in C^2(\mathcal{X}, K) \cap \mathcal{F}_p(L^*)} |\hat{L}(f) - L^*| < \epsilon$$

[1] Otherwise, a theoretical characterisation of the Lipschitz learning problem is not feasible.
[2] Note: in the noiseless case, the oracle and the target function are equivalent.

then

$$n > \left(C(d,p)\frac{MK}{\epsilon} \right)^d$$

In this paper, we find $C(d,p) = \frac{1}{20d^{\max\{\frac{1}{p}-\frac{1}{2},0\}}}$*, however this value has not been optimized.*

Theorem 2.2 provides a lower bound on the minimum number of oracle queries that are needed in order for a Lipschitz learning algorithm to ensure a precise estimate of the Lipschitz constant for all underlying target functions in $C^2(\mathcal{X}, K)$. As speculated in the introduction, it shows that the Lipschitz Learning problem is a computationally expensive one that depends heavily on the input dimension. The lower bounding expression is given as a function of the size of the input space (M), the assumed bound on the second order partial derivatives (K) and the precision parameter (ϵ) but is independent of the true Lipschitz constant (L^*) of the target function. The product MK can be understood as a bound on the maximum change in the gradient values of functions in $C^2(\mathcal{X}, K)$. In Section 3, the proposed LCLS algorithm will be shown capable of estimating the Lipschitz constant of all functions f in $C^2(\mathcal{X}, K)$ using $O((\frac{MK}{\epsilon})^d)$ queries to the noiseless sampling oracle Ω implying that the lower bound on the sample complexity rate stated in Theorem 2.2 is optimal.

An equivalent reformulation of Theorem 2.2 in the form of a lower bound on the convergence rate of Lipschitz learning algorithms is provided in the following corollary.

Corollary 2.3 *(Convergence Rate Bound – Noiseless) Assume the same setting as Theorem 2.2. Then,* $\forall L^* \geq 0$*,*

$$\inf_{\hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})} \sup_{f \in C^2(\mathcal{X},K) \cap \mathcal{F}_p(L^*)} |\hat{L}(f) - L^*| \geq C(d,p)\frac{MK}{\sqrt[d]{n}}$$

where $C(d,p)$ *is defined in Theorem 2.2.*

Corollary 2.3 is generally more practical to use then Theorem 2.2 when considering convergence properties of applications of Lipschitz constant estimators. In Section 4, we show how Corollary 2.3 can be applied in conjunction with recent theoretical results (Bachoc et al. (2021)) to derive lower bounds on the sample complexity of adaptive Lipschitz optimisation algorithms.

2.3 Noisy Setting

In many practical cases, the sampling oracle cannot be assumed reliable and only approximative observations of the target function are obtainable. In this case, we model $\Omega : \mathcal{X} \to \mathbb{R}$ as being corrupted by additive noise and a new lower bound on the sample complexity of Lipschitz learning algorithms can be derived. In order to do so, additional assumptions must be made on the additive observational noise process. Let $\sigma^2 > 0$ and denote by $\mathcal{D}(0, \sigma^2)$ the set of all probability distributions on $\mathbb{R}$ with zero mean and finite variance $\sigma^2 > 0$.

Assumption 3 *(Noisy Oracle) We define a noisy sampling oracle as*

$$\tilde{\Omega} : \ \mathcal{X} \to \mathbb{R}$$
$$x \overset{\tilde{\Omega}}{\mapsto} \tilde{f}_x := f(x) + \gamma_x$$

where $\gamma_x \sim D_{\gamma_x} \in \mathcal{D}(0, \sigma^2)$ *are uncorrelated random variables* ($x \in \mathcal{X}$)*. Note:* γ_x *is an abuse of notation as the noise is not dependent on the input* x*. In other words: if* $x \in \mathcal{X}$ *is sampled twice, then* $\gamma_x^1 \neq \gamma_x^2$*.*

As the class of Lipschitz learning algorithms has been loosely defined so far – with no parametric or functional assumptions – an additional technical assumption is needed. Let $\mathcal{L}_{n,p}(\mathcal{X})$ and $G_{\mathcal{X}}^{\hat{L}}$ be as defined in Definition 2.1. We assume that the following assumption holds for $\mathcal{D}(0, \sigma^2)$ and $\mathcal{L}_{n,p}(\mathcal{X})$.

Notation 2.4 *To alleviate notation when discussing the distribution of a vector* $\{\gamma_{x_1}, ..., \gamma_{x_n}\}$ *of noise variables, we denote by* D_γ^n *their n-dimensional distribution and write* $D_\gamma^n \in \mathcal{D}(0, \sigma^2)$ *as a shorthand for* $D_{\gamma_{x_i}} \in \mathcal{D}(0, \sigma^2), \forall i \in \{1, ..., n\}$ *with* $\gamma_{x_i}, \gamma_{x_j}$ *uncorrelated for* $i, j \in \{1, ..., n\}$*.*

Assumption 4 *Let $\mathcal{X} := [0, M]^d$, $K \in \mathbb{R}_+$ and fix $\bar{\epsilon} \leq \frac{MK}{20}$. We assume that the $\mathcal{L}_{n,p}(\mathcal{X})$ class satisfies the following property: $\forall L^* \geq 0$, $\forall \delta \in (0, \frac{1}{2})$, $\exists Q(\delta) \in \mathbb{R}_+$ such that $\forall \epsilon \in (0, \bar{\epsilon})$, $\forall \hat{L}_n \in \mathcal{L}_{n,p}(\mathcal{X})$, $\forall D_\gamma^n \in \mathcal{D}(0, \sigma^2)$ one of the two following statements hold :*

1. $\forall f \in C^2(\mathcal{X}, K) \cap \mathcal{F}_p(L^*)$, $\mathbb{P}_{D_\gamma^n}\left(\frac{1}{n_\mathcal{Y}} \sum_{x_i^{\hat{L}} \in \mathcal{Y}} \left|\gamma_{x_i^{\hat{L}}}\right| > Q(\delta)\right) = 0$ *or,*
2. $\exists f \in C^2(\mathcal{X}, K) \cap \mathcal{F}_p(L^*)$ *s.t.* $\mathbb{P}_{D_\gamma^n}\left(|\hat{L}_n(f) - L_p^*(f)| \leq \epsilon \left| \frac{1}{n_\mathcal{Y}} \sum_{x_i^{\hat{L}} \in \mathcal{Y}} \left|\gamma_{x_i^{\hat{L}}}\right| > Q(\delta)\right.\right) \leq 1 - \delta$.

where $\mathcal{Y} := B_{\frac{20\epsilon}{K}}(c) \subset \mathcal{X}$[3] (with arbitrary $c \in \mathcal{X}$) and $n_\mathcal{Y} := |G_\mathcal{X}^{\hat{L}} \cap \mathcal{Y}| < n$.

Assumption 4 states that there exists a level of local sampling noise for which no Lipschitz learning algorithm can guarantee precise estimation of the Lipschitz constant with a high degree of certainty. More precisely: if a subset of observations generated by the oracle (Ω) with inputs belonging to a hyperball[4] ($B_{\frac{20\epsilon}{K}}(c)$) in $\mathcal{X}$ has been significantly corrupted by noise ($Q(\delta)$), then it becomes impossible to precisely (ϵ) estimate the Lipschitz constant of all target functions ($f \in C^2(\mathcal{X}, K) \cap \mathcal{F}_p(L^*)$) with high probability ($1 - \delta$). Here, the notion of significantly corrupted is dependent on the probabilistic estimation guarantee δ and can be set without constraints. We note that Assumption 4 is not the most general assumption which can be made on $\mathcal{L}_{n,p}(\mathcal{X})$ in order for Theorem 2.5 to hold[5] and that most existing Lipschitz constant estimators can be shown to satisfy Assumption 4. This assumption is relatively intuitive and additional comments are given in appendix

Using Assumptions (3) and (4), the following lower bound on the sample complexity rate can be derived in the noisy sampling setting.

Theorem 2.5 *(Sample Complexity Bound – Noisy) Let $M \in \mathbb{R}_+$, $d \in \mathbb{N}$, $p \in \{1, 2\}$ and suppose $\mathcal{X} := [0, M]^d$. Assume that Assumptions (2)-(4) hold and that one has access to a noisy oracle $\tilde{\Omega} : \mathcal{X} \to \mathbb{R}$ as specified in Assumption (3). Define $\mathcal{L}_{n,p}(\mathcal{X})$ as in Theorem 2.2. $\forall L^* \geq 0, \forall \epsilon > 0, \forall \delta \in (0, \frac{1}{4})$, if*

$$\sup_{\hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})} \inf_{f \in C^2(\mathcal{X}, K) \cap \mathcal{F}_p(L^*)} \mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| < \epsilon) > 1 - \delta.$$

for all $D_\gamma^n \in \mathcal{D}(0, \sigma^2)$ then

$$n > \left(\tilde{C}(\sigma^2, \delta, d) \frac{MK}{\epsilon}\right)^{2d}$$

where $\tilde{C}(\sigma^2, \delta, d) = \frac{\sigma^2}{20 \log(2) Q^{*2} d^{\max\{\frac{1}{p} - \frac{1}{2}, 0\}}}$.

The lower bound on the sample complexity proposed in Theorem 2.5 is approximately the square of the one given in Theorem 2.2 due to the additional noise consideration. A comparison with an upper bound on the sample complexity derived from applying the LCLS estimator proposed in this paper is given in Remark 3.10. The two bounds are shown to differ by a factor of $\sim \left(\frac{K^2}{\epsilon^4}\right)$.

We note that the lower bound stated in Theorem 2.5 is conservative as it considers all distributions in $\mathcal{D}(0, \sigma^2)$. This represents the fact that in practice one would want a Lipschitz learning algorithm to work without any prior knowledge of the additive noise distribution. In future work, we will aim to improve the sample complexity bound given in Theorem 2.5 under additional noise assumptions (see Remark 2.6 for details).

Remark 2.6 *If a more precise characterisation of the noise distribution is available (e.g. γ is bounded or sub-Gaussian), then the bound given in Theorem 2.5 can be improved significantly by replacing the Chebyshev-type inequality used in the proof of Theorem 2.5 by another classical concentration inequality.*

[3] where $B_r(c)$ denotes the ball of radius r and center c w.r.t. $\|.\|_2$.

[4] The choice of a hyperball is arbitrary.

[5] A more general version of Assumption 4 can be used, however this would require spending a significant amount of the paper defining additional notation.

Finally, as in the noiseless sampling setting, an equivalent reformulation of Theorem 2.5 is provided in the form of a probabilistic lower bound on the convergence rate of Lipschitz learning algorithms.

Corollary 2.7 *(Convergence Rate Bound – Noisy)) Assume the same setting as Theorem 2.5. Then, there exists at least one distribution $D_\gamma^n \in \mathcal{D}(0, \sigma^2)$ such that*

$$\sup_{\hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})} \inf_{f \in C^2(\mathcal{X},K) \cap \mathcal{F}_p(L^*)} \mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| < C(\sigma^2, \delta, d) \frac{MK}{\sqrt[2d]{n}}) \leq 1 - \delta.$$

where $C(\sigma^2, \delta, d)$ is defined in Theorem 2.5.

Theorem 2.5 and Corollary 2.7 are particularly interesting in the context of system identification for control applications (e.g. Milanese and Novara (2004), Calliess et al. (2020)) where robustness properties depend explicitly on estimating a feasible Lipschitz constant from noisy data. These frameworks often ignore the modeling error arising from the Lipschitz constant estimation which is problematic when the goal is to provide worst-case guarantees. One source of usefulness for the two bounds stated in this subsection is therefore to provide a theoretical understanding of the worst-case estimation of Lipschitz constants in this context and therefore to make possible a more realistic robustness analysis of Lipschitz constant-based system identification methods in practice. A short illustrative comparison of the convergence rate given in Corollary 2.7 with the convergence of existing Lipschitz learning algorithms used for system identification purposes is given in Figure 3.

3 Lipschitz Constant estimation by Least Squares regression (LCLS)

The theoretical results of Section 2 imply that a significant number of samples must be used in order to obtain a precise estimation of the best Lipschitz constant. As noted in the introduction, this is problematic computationally for classical Strongin-based Lipschitz learning algorithms due to the fact that the computational complexity of these methods can be shown to be quadratic in the number of samples. Using existing non-Strongin based methods could resolve this computational problem, however obtaining convergence guarantees would then be difficult as this class of methods is generally complicated to study from a theoretical perspective. Therefore, in the goal of obtaining a Lipschitz learning approach that can provide asymptotically consistency guarantees, has low computational complexity and for which we can derive upper bounds on the sample complexity (in the goal of comparing with the sample complexity lower bounds derived in Section 2), we define a new estimator: *Lipschitz Constant estimation by Least Squares regression* (LCLS).

3.1 Overview

The general intuition behind the Lipschitz learning algorithm proposed in this paper follows from the simple observation that the coefficients from a least squares regression can be interpreted as a local approximation of the gradient and that the maximum q-norm of the gradient of f on $\mathcal{X}$ coincides[6] (for certain values of $p \in \mathbb{N}$) with the best Lipschitz constant associated to the p-norm, where q is the Holder-conjugate of p, i.e. $\frac{1}{p} + \frac{1}{q} = 1$. Therefore, by using a partition $\mathcal{H}$ of the input space $\mathcal{X}$ that is sufficiently refined to properly capture the gradient variation of f and computing the maximum q-norm of the least squares coefficients associated to each subset of $\mathcal{H}$, a precise estimate of the Lipschitz constant is obtainable. Practically, in order to ensure that the refinement of the partition suffices[7], the proposed estimation framework is designed as an iterative method that utilises a sequence of increasingly fine convex partitions $(\mathcal{H}_I)_{I \in \mathbb{N}}$ that are given as input. A brief technical description of an iteration of the algorithm can be described as follows: *For a given iteration, indexed by $I \in \mathbb{N}$, a set of observations $D_I^H := \{(x_i^H, \tilde{f}_i^H)\}_{i \in \{1,\dots,N_I^H\}}$ is generated by an oracle $\Omega : \mathcal{X} \to \mathbb{R}$ (defined in Section 2) for each subset H of the partition $\mathcal{H}_I$ and used individually to compute the coefficients $\{\beta_I^H\}_{H \in \mathcal{H}}$ of an ordinary least squares regression for each subset $H \in \mathcal{H}_I$. The Lipschitz constant estimate can then be directly computed: $\hat{L}_I := \max_{H \in \mathcal{H}_I} \{\|\beta_I^H\|_q\}$ where q is the Holder-conjugate of p.*

[6] see Lemma D.2 in Appendix for a formal statement

[7] In the case where the upper bound K given in Assumption 2 is known beforehand it is possible to directly partition at the required refinement level (See Theorem 3.9 for example).

Algorithm 1 General LCLS[8]

Input: $\tilde{\Omega}$ (Oracle), $(\mathcal{H}_I)_{I\in\mathbb{N}}$ (Partition Sequence)
Output: $\{\hat{L}_I\}$ (Lipschitz Estimates)
procedure: LCLS($\tilde{\Omega}$, $(\mathcal{H}_I)_{I\in\mathbb{N}}$)
  initialise: $I \leftarrow 1$
  repeat
    $\hat{L}_I \leftarrow 0$
    for $H \in \mathcal{H}_\mathcal{I}$ **do**
      $(X_I^H, \tilde{f}_I^H) \leftarrow D_I^H$ generated by $\tilde{\Omega}$
      $\hat{\beta}_I^H \leftarrow ({X_I^H}^\top X_I^H)^{-1} {X_I^H}^\top \tilde{f}_I^H$
      $\hat{L}_I \leftarrow \max(\|\hat{\beta}_I^H\|_q, \hat{L}_I)$
    end
    $I \leftarrow I + 1$
return $\{\hat{L}_I\}$

Algorithm 2 Hypercube LCLS[9] on $[0, M]^d$

Input: $\tilde{\Omega}$ (Oracle), K (Bound from (2)), η (covering constant), σ^2 (noise variance), (ϵ, δ) (precision)
Output: $\hat{L}$ (Lipschitz Constant Estimation)
procedure: LCLS($\tilde{\Omega}$, K, η, σ^2, (ϵ, δ))
  initialise: $\hat{L} \leftarrow 0$ $I \leftarrow (C_1(d)\frac{MK}{\sqrt{\eta}\epsilon})$,
  $N_I \leftarrow (C_2(d,q)\frac{\sigma^2}{\eta\delta}\frac{I^{d+2}}{M^2\epsilon^2})$
  $\mathcal{H} \leftarrow$ hypercube partition of $[0, M]^d$ with side-length $\frac{M}{I}$
  for $H \in \mathcal{H}$ **do**
    $(X^H, \tilde{f}^H) \leftarrow D^H$ generated by $\tilde{\Omega}$
    $\hat{\beta}^H \leftarrow ({X^H}^\top X^H)^{-1} {X^H}^\top \tilde{f}^H$
    $\hat{L} \leftarrow \max(\|\hat{\beta}^H\|_q, \hat{L})$
  end
return $\hat{L}_I$

Figure 1: *Algorithm 1 details the implementation of the LCLS algorithm for a general input space and partition choice. Algorithm 2 details the implementation of the LCLS algorithm when the input space is a hypercube $[0, M]^d$ and the partitions are regular. We note that the generated data points X_I^H used by the two algorithms are selected arbitrarily in each $H \in \mathcal{H}_I$. In order to ensure convergence of the LCLS algorithm, X_I^H will need to verify Assumption 5 for all $I \in \mathbb{N}$ and $H \in \mathcal{H}_I$.*

The LCLS algorithm is described here in its most general form in order to allow flexibility in the choice of the input space partitions and sampling scheme. Algorithm 1 provides an algorithmic description of this approach. A more specific implementation of the LCLS algorithm which utilises a regular hypercube partition of the input space is given in Algorithm 2 and discussed later on in this section in Theorem 3.9 and ensuing discussions. As one might expect, the structure of $(\mathcal{H}_I)_{I\in\mathbb{N}}$ is a key part of the LCLS estimator. In practice, these partitions can be defined using domain or functional knowledge in order to better estimate the gradient variation and therefore speed up the convergence of the algorithm. The distribution of the sample points given by $(N_I^H)_{n\in\mathbb{N}}$ should also be considered carefully and can be selected in a partition dependent way to take advantage of any prior knowledge of f or of the underlying noise distribution. We note that the relation between the structure of $(\mathcal{H}_I)_{I\in\mathbb{N}}$ and $(N_I^H)_{n\in\mathbb{N}}$ is essential in the proofs of Theorem 3.6 and Theorem 3.9.

The following variables are used to formally describe a partition belonging to $(\mathcal{H}_I)_{I\in\mathbb{N}}$.

Notation 3.1 *Let $\delta(A) = \sup_{x,y\in A} \|x - y\|_2$ denote the diameter function and $B_r(x)$ the d-dimensional ball centered in $x \in \mathcal{X}$ and with radius r with respect to $\|.\|_2$.*

Definition 3.2 *(Partition Variables) Let $\mathcal{H}_J \in (\mathcal{H}_I)_{I\in\mathbb{N}}$, we define the following two $\mathcal{H}_J$ related quantities: the maximum diameters of $\mathcal{H}_J$: $\{\Delta_J^H\}_{H\in H_J}$, $\Delta_J^H := \delta(H)$ and the minimum diameters of the biggest subset-inscribed balls of $\mathcal{H}_J$: $\{\delta_J^H\}_{H\in H_J}$, $\delta_J^H := 2\max\{r \in \mathbb{R}_+ | \exists x \in H$ such that $B_r(x) \subset H\}$.*

The quantities $\{\Delta_J^H\}_{H\in H_J}$ and $\{\delta_J^H\}_{H\in H_J}$ are used in Definition 3.4 and in Theorem 3.6 to define sufficient conditions on the structure of the $(\mathcal{H}_I)_{I\in\mathbb{N}}$ partitions in order for the general version of the LCLS algorithm to converge.

We conclude this subsection by giving a result on the computational complexity of the proposed algorithm.

Proposition 3.3 *(Computational Complexity of LCLS) The computational complexity of the Lipschitz Constant Least Squares Estimator is $O(n_{samples})$ where $n_{samples}$ denotes the number of observations sampled by the algorithm.*

The computational complexity derived in Proposition 3.3 is significantly smaller than the complexity of Strongin-based approaches which is $O(n_{samples}^2)$. The difference in computation speed is illustrated empirically on a set of test functions in Section 3.4.

3.2 General Theoretical Analysis

An investigation of the theoretical behaviour and performance of the proposed LCLS algorithm is carried out in this section. This analysis provides an understanding of the design constraints required for the construction of the input space partitions and for the choice of sampling schemes in order to ensure satisfactory performance – see Remark 3.7. We begin by stating an asymptotic convergence result for the general form of the algorithm in the noiseless and noisy sampling settings before stating and discussing finite sample results for a more concrete application of LCLS when the partition of the input space is constructed to be a set of regular hypercubes.

The following definition defines two quantities $(a_I)_{I\in\mathbb{N}}$, $(b_I)_{I\in\mathbb{N}}$ as a function of $\{\Delta_J^H\}_{H\in H_J}$, $\{\delta_J^H\}_{H\in H_J}$, $(\{N_I^H\}_{H\in\mathcal{H}_\mathcal{I}})_{I\in\mathbb{N}}$ and $(|\mathcal{H}_I|)_{I\in\mathbb{N}}$ in order to alleviate notation[10]. They will be used to describe the conditions on the structure of the input partition sequence needed in order to ensure asymptotic consistency.

Definition 3.4 *For any sequence of convex and compact partitions, $(\mathcal{H}_I)_{I\in\mathbb{N}}$, we construct the following sequences:*

- $(a_I)_{I\in\mathbb{N}}$, $a_I = \max_{H\in\mathcal{H}_I}\left(\frac{(\Delta_I^H)^2}{\delta_I^H}\right)$
- $(b_I)_{I\in\mathbb{N}}$, $b_I = \max_{H\in\mathcal{H}_I}\left(\frac{|\mathcal{H}_I|}{N_I^H(\delta_I^H)^2}\right)$.

Before stating the first main result of this section, a condition on the sampling procedure used by the LCLS algorithm must be given. This condition uses the concept of ϵ-coverings in order to ensure that the inputs queried by the oracle are well-distributed in each subset H belonging to a partition $\mathcal{H}_I$, $I\in\mathbb{N}$ and is described formally in the following assumption.

Definition 3.5 *Let $H\subseteq\mathcal{X}$ be compact and convex and denote by $\{(x_i^H,\hat{f}_i^H)\}_{i\in\{1,\dots,N_I^H\}}$ the subset of generated or archived data samples in H. We say that H is (ϵ,η)-covered if there exists $\epsilon>0,\eta\in]0,1]$ such that D_I^H contains ηN_I^H disjointed ϵ-covers (with respect to $\|.\|_2$) of H.*

Assumption 5 *(Sampling) For a given $\eta\in(0,1]$, the sampling scheme selected for LCLS is such that $\forall I\in\mathbb{N}$ and $\forall H\in\mathcal{H}_I$, H is $(\frac{\delta_I^H}{4},\eta)$-covered.*

The sampling condition stated in Assumption 5 is necessary in order to ensure stability of the least squares coefficient as the sequence of partitions becomes increasingly refined and can be satisfied by using quasi-Monte Carlo schemes in practice.

Theorem 3.6 formalizes the consistency of the proposed Lipschitz learning framework for the noisy sampling setting.

Theorem 3.6 *(General Convergence Rate) If Assumptions (1)-(5) (for a given $\eta\in(0,1]$) hold and the following conditions are verified:*

1. *$\forall I\in\mathbb{N}$, $\mathcal{H}_I$ is a convex partition of $\mathcal{X}$,*
2. *$\lim_{I\to\infty} a_I = 0$, $\lim_{I\to\infty} b_I = 0$, $\lim_{I\to\infty}\max_{H\in\mathcal{H}_I}(\Delta_I^H) = 0$,*

then $\forall\ D_\gamma^n\in\mathcal{D}(0,\sigma^2)$, $f\in C^2(\mathcal{X},K)$,

$$\hat{L}_I(f)\xrightarrow[I\to\infty]{\mathbb{P}} L_p(f)$$

where $L_p(f)=L_p^*(f)$ for $p=1,2$, $L_p\geq L_p^*(f)$ for $p>2$. $\mathbb{P}$ denotes convergence in probability and $(\hat{L}_I(f))_{I\in\mathbb{N}}$ is the sequence of Lipschitz constant estimates generated by the LCLS estimator.

[6] Notation is provided in Appendix A.

[9] The method described in this algorithm corresponds to the specific case where the (K,σ^2) variables are known.

[10] Here. $|.|$ denotes the cardinality operator.

Remark 3.7 *(Design Constraints) Condition 2 of Theorem 3.6 specifies the design constraints needed in the construction of the partition sequence $(\mathcal{H}_I)_{I\in\mathbb{N}}$ and the number of sample points $(\{N_I^H\}_{H\in\mathcal{H}_\mathcal{I}})_{I\in\mathbb{N}}$ required per hypercube in order to ensure convergence. In particular:*

1. *$\lim_{I\to\infty} a_I = 0$ provides the limitations on the shape of the sets in each partition $\mathcal{H}_I$ as I goes to infinity. In particular, as $I \to \infty$, $(\Delta_I^H)^2 << \delta_I^H < \Delta_I^H$.*
2. *$\lim_{I\to\infty} b_I = 0$ specifies a condition on the number of samples needed per hypercube. As $I \to \infty$, $N_I^H >> \frac{|\mathcal{H}_I|}{(\delta_I^H)^2}$. This is made precise and used for a comparison to the sample complexity rates found in the previous section in Remark 3.10.*
3. *$\lim_{I\to\infty} \max_{H\in\mathcal{H}_I}(\Delta_I^H) = 0$ ensures that the partitions are increasingly refined.*

In practice, applying the theoretical conditions used in Theorem 3.6 produces an overly conservative estimator in terms of required number of queries made to the oracle – see Section 3.4 for an illustration of the empirical convergence of the LCLS estimator. This is due to the fact that the LCLS estimator makes minimal functional assumptions and therefore has to explore all of $\mathcal{X}$ to generate a precise Lipschitz estimate. In order to avoid this issue, the number of samples per hypercube as measured by $(b_I)_{I\in\mathbb{N}}$ can be set heuristically in order to improve the empirical performance.

In the noiseless sampling setting, the stopping and sampling rules given in Theorem 3.6 and Remark 3.7 can be modified in order to obtain a quicker convergence. This is detailed in the following corollary.

Corollary 3.8 *(Noiseless Sampling) If Assumptions (1),(2),(5) (for a given $\eta \in (0, 1]$) hold, a noiseless oracle $\Omega : \mathcal{X} \to \mathbb{R}$ is available and the following conditions are verified:*

1. *$\forall I \in \mathbb{N}$, $\mathcal{H}_I$ is a convex partition of $\mathcal{X}$,*
2. *$\lim_{I\to\infty} a_I = 0$, $\lim_{I\to\infty} \max_{H\in\mathcal{H}_I}(\Delta_I^H) = 0$,*
3. *$\forall I \in \mathbb{N}, H \in \mathcal{H}_I$, $N_I^H \geq d + 1$,*

then $\forall\ D_\gamma^n \in \mathcal{D}(0, \sigma^2)$, $f \in C^2(\mathcal{X}, K)$,

$$\hat{L}_I(f) \xrightarrow[I\to\infty]{} L_p(f)$$

where $L_p(f) = L_p^*(f)$ for $p = 1, 2$, $L_p(f) \geq L_p^*(f)$ for $p > 2$ and the right arrow denotes deterministic convergence.

While, the conditions on the design constraints of the partition sequence needed to ensure asymptotic convergence of the LCLS algorithm remain the same as in Theorem 3.6, the sampling conditions specified in Corollary 3.8 imply that a much smaller number of samples are required per hypercube. More precisely, the only sampling condition stated in Corollary 3.8 is related to the minimum number of samples needed to ensure that the local linear regressions computed by the LCLS algorithm are well-defined.

Using the general results developed in this section, we now explore a more specific application to the $[0, M]^d$ input space. Theorem 3.8 provides convergence rates that can be compared to the ones proposed in Theorem 2.2 and Theorem 2.5. This is discussed in Remark 3.10 and Remark 3.11.

3.3 LCLS with Regular Partitions and Sample Complexity Upper Bound

In the previous section, we considered a general form of the LCLS algorithm and stated the conditions on the design constraints of the input partition sequence and the sampling scheme required to ensure convergence. Here, we assume that the input space is the d-dimensional hypercube $[0, M]^d$ and consider the case where every input partition $\mathcal{H}_I$ is a regular hypercube partition of side-length $\frac{M}{I}$. The associated sampling scheme is then defined based on the sampling condition given in Assumption 5 and the desired precision of the Lipschitz constant estimate.

Under these additional constraints, the following finite sample guarantee can be obtained for the LCLS algorithm.

Theorem 3.9 *(Finite Sample Guarantee) Let $\mathcal{X} := [0, M]^d$ and $(\mathcal{H}_I)_{I\in\mathbb{N}_{>1}}$ denote the regular partition of sub-hypercubes of $\mathcal{X}$ with side-length $\frac{M}{I}$. If Assumptions (2)-(5) (for a given $\eta \in (0, 1]$) hold and if*

$\forall \epsilon > 0$, $\delta \in (0, \frac{1}{2}]$, $I \geq \left(C_1(d)\frac{MK}{\sqrt{\eta}\epsilon}\right)$ *and* $\forall H \in \mathcal{H}_I$ $N_I^H \geq \left(C_2(d,q)\frac{\sigma^2}{\eta\delta}\frac{I^{d+2}}{M^2\epsilon^2}\right)$ *for* $C_1(d), C_2(d,q) \in \mathbb{R}_+$, *then* $\forall$ $D_\gamma^n \in \mathcal{D}(0,\sigma^2)$*:*

$$\inf_{f \in C^2(\mathcal{X},K)} \mathbb{P}(|L_p(f) - \hat{L}_I(f)| \leq \epsilon) \geq 1 - \delta. \tag{1}$$

where $L_p(f) = L_p^*(f)$ *for* $p = 1, 2$ *and* $L_p \geq L_p^*$ *for* $p > 2$*. Here* $C_1(d) = 8d^2\sqrt{d}d^{\max\{\frac{1}{q}-\frac{1}{2},0\}}$ *and* $C_2(d,q) = 2^5 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}$ *however these constants have not been optimized.*

The theoretical guarantees of Theorem 3.9 can be extended to include any $\mathcal{X} \subset \mathbb{R}^d$ that satisfies Assumption 1. Indeed, trivially there exists a hypercube $[a,b]^d \subset \mathbb{R}^d$ with $a, b \in \mathbb{R}$ such that $\mathcal{X} \subset [a,b]^d$ which can be partitioned according to the iterative regular hypercube partitioning approach. The partition sequence inputted into the LCLS algorithm then consists of the regular hypercube subsets partitions of $[a,b]^d$ that intersect with $\mathcal{X}$. In this case, under Assumptions (1)-(5), a modified version of Theorem 3.9 holds: the condition on I remains the same, but the lower bound condition on N_I^H can be weakened to become $N_I^H \geq \left(C_2(d,q)\frac{\sigma^2}{\eta\delta}\frac{I^{d+2}-\Gamma}{(b-a)^2\epsilon^2}\right)$, $\forall H \in \mathcal{H}_I, I \in \mathbb{N}$, where $\Gamma = |\{H \in \mathcal{H}_I \big| H \cap \mathcal{X} = \emptyset\}|$.

Since Theorem 3.9 holds under Assumption 3, i.e. for any $D_\gamma^n \in \mathcal{D}(0,\sigma^2)$, $n \in \mathbb{N}$ and any $f \in C^2([0,M]^d, K)$ it also holds for $\sup_{D_\gamma^n \in \mathcal{D}(0,\sigma^2)} \sup_{f \in C^2([0,M]^d,K)}$. The sample complexity of the LCLS algorithm implied by Theorem 3.9 is therefore comparable to the general lower bound on the sample complexity of the noisy Lipschitz learning problem stated in Theorem 2.5. This is done in the following remark.

Remark 3.10 *(Sample Complexity Comparison – Noisy) For* $p = 1, 2$*, assuming that the lower bounds:* $I \geq \left(C_1(d)\frac{MK}{\sqrt{\eta}\epsilon}\right)$ *and* $\forall H \in \mathcal{H}_I$ $N_I^H \geq \left(C_2(d,q)\frac{\sigma^2}{\eta\delta}\frac{I^{d+2}}{M^2\epsilon^2}\right)$ *are satisfied with an equality, the total number* n_1 *of points required to ensure* $\mathbb{P}(|L_p - \hat{L}_I| \leq \epsilon) \geq 1 - \delta$ *is given by*

$$n_1 = |\mathcal{H}_I|N_I = C_2(d,q)\frac{\sigma^2}{\eta\delta}\left(\frac{C_1(d)}{\sqrt{\eta}}\frac{MK}{\epsilon}\right)^{2d+2}\frac{1}{M^2\epsilon^2} = O\left(\left(\frac{MK}{\epsilon}\right)^{2d+2}\frac{1}{M^2\epsilon^2}\right).$$

This convergence rate differs from the lower bound on the noisy sample complexity rate stated in the Theorem 2.5 by a rate of $\frac{K^2}{\epsilon^4}$*. While this difference is small, it means that we cannot conclude on the optimality of the obtained sample complexity rate. The improvement of these bounds is an open question and will be continued to be worked on in future work.*

By slightly modifying the necessary conditions used in Theorem 3.9, we can also compare the sample complexity of the LCLS algorithm implied by Theorem 3.9 in the noiseless sampling setting to the lower bound on the sample complexity of the noiseless Lipschitz learning problem stated in Theorem 2.2. In order to do so, we define

$$N(I) := \max_{H \in \mathcal{H}_\mathcal{I}} \min\{n \in \mathbb{N} | D_I^H \text{ contains a disjointed } \delta_I^H\text{-cover of } H\}$$

which is constant $\forall I \in \mathbb{N}$ when $(\mathcal{H}_I)_{I \in \mathbb{N}}$ is defined as a sequence of regular hypercube partitions on $[0,M]^d$. In this case, we remove the dependence on I and write $N := N(I)$. We note that the following two inequalities hold: (1) $\eta \leq \frac{1}{N}$ (tight) and (2) $N < \sqrt{d}^d$ (loose).

Remark 3.11 *(Sample Complexity Comparison – Noiseless) In the case of noiseless sampling, the lower bound on* N_I^H *stated in Theorem 3.9 can be replaced by condition 3. of Corollary 3.8 and the definition of* N *given above, i.e.* $\forall I \in \mathbb{N}, H \in \mathcal{H}_I$, $N_I^H \geq \max(d+1, N)$*. Proceeding as in Remark 3.10, we have in this case:*

$$n_2 = |\mathcal{H}_I|N_I = \max(d+1, N)\left(C_1(d)\frac{MK}{\sqrt{\eta}\epsilon}\right)^d = O\left(\left(\frac{MK}{\epsilon}\right)^d\right).$$

This convergence rate corresponds exactly to the lower bound on the noiseless sample complexity rate stated in the Theorem 2.2 and therefore implies that the sample complexity rate $\left(\frac{MK}{\epsilon}\right)^d$ *is optimal (up to constant factors dependent on* d *and* p*) in the sense that it characterises the minimum number of samples that are needed to obtain an* ϵ*-precise Lipschitz constant estimate for any* $f \in C^2(\mathcal{X}, K)$*.*

As in Section 2, we can reformulate the sample complexity rates of the LCLS algorithm given in Remarks 3.10 and 3.11 as convergence rates and therefore as upper bounds on the convergence rate of the general Lipschitz learning problem. This is done in the following corollary.

Corollary 3.12 *(Convergence Rate Comparison)*

1. *(Noiseless) Assume the same setting as Remark 3.11. Then,*

$$\inf_{\hat{L}\in\mathcal{L}_{n,p}(\mathcal{X})} \sup_{f\in C^2(\mathcal{X},K)} |\hat{L}(f) - L_p^*(f)| \leq C(d,p)\frac{MK}{\sqrt[d]{n}}$$

   where $C(d,p)$ can be determined from Remark 3.11.
2. *(Noisy) Assume the same setting as Remark 3.10. Then, $\forall$ distribution $D_\gamma^n \in \mathcal{D}(0,\sigma^2)$:*

$$\sup_{\hat{L}\in\mathcal{L}_{n,p}(\mathcal{X})} \inf_{f\in C^2(\mathcal{X},K)} \mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| < C(\sigma^2,\delta,d)\frac{M^{\frac{d}{d+2}}K^{\frac{d+1}{d+2}}}{\sqrt[2d+4]{n}}) \geq 1-\delta$$

   where $C(\sigma^2,\delta,d)$ can be determined from Remark 3.10.

An interesting consequence of Corollary 3.12 is that it provides a way of generating a sequence of feasible[11] Lipschitz constant estimates that converge to the best Lipschitz constant if a potentially loose upper bound on the second degree partial derivatives is known. More precisely, one can consider the Lipschitz constant estimates:

- $\hat{L}_{up} := \hat{L}(f) + C(d,p)\frac{MK}{\sqrt[d]{n}}$ in the noiseless sampling setting
- $\hat{L}_{up} := \hat{L}(f) + C(\sigma^2,\delta,d)\frac{M^{\frac{d}{d+2}}K^{\frac{d+1}{d+2}}}{\sqrt[2d+4]{n}}$ in the noisy sampling setting

where $\hat{L}(f)$ denotes the Lipschitz constant estimate generated by the LCLS algorithm. Such an approach is useful in practice as Lipschitz constant-based computational frameworks often rely on the assumption that the estimated Lipschitz constant used is feasible. This is briefly discussed further in Section 4 where a direct application of the LCLS algorithm in the context of nonparametric regression for system identification is developed.

3.4 Empirical Performance:

The focus so far in this section has been on developing the theoretical properties of the LCLS algorithm. While that discussion is useful in itself as it provides performance guarantees for LCLS as well as upper bounds on the sample complexity of the general Lipschitz learning problem, we are also interested in how the proposed algorithm performs empirically. In particular, we would like to compare the convergence speed of the LCLS algorithm to other theoretically well-behaved methods and to verify whether the theoretical computational advantage of LCLS (see Proposition 3.3) is observed in practice. In this subsection, we investigate these questions by illustrating the convergence rate and computation time of the proposed Lipschitz constant estimation method and comparing it against existing Strongin-based algorithms on a set of test functions with interesting properties in noiseless, bounded noise and unbounded noise sampling settings.

3.4.1 Experimental Setup

Table 1 provides an overview of the four test functions that are used in the experiments discussed in this section. The choice of these functions represents different testing points that are of interest: Function (a) reaches the maximum of the normed gradient in a single unique point of the input space, Function (b) is a classical optimisation testing function which we have also defined to have large second degree partial derivatives, Function (c) is a trigonometric function which provides an illustration of convergence for simple target functions and finally, Function (d) is a higher dimensional version of Function (a) with 3 dimensional inputs. We do not explore higher dimensional versions (>3) of Function (a) as the convergence speed with

[11] i.e. which upper bound the best Lipschitz constant and satisfy the Lipschitz continuity condition.

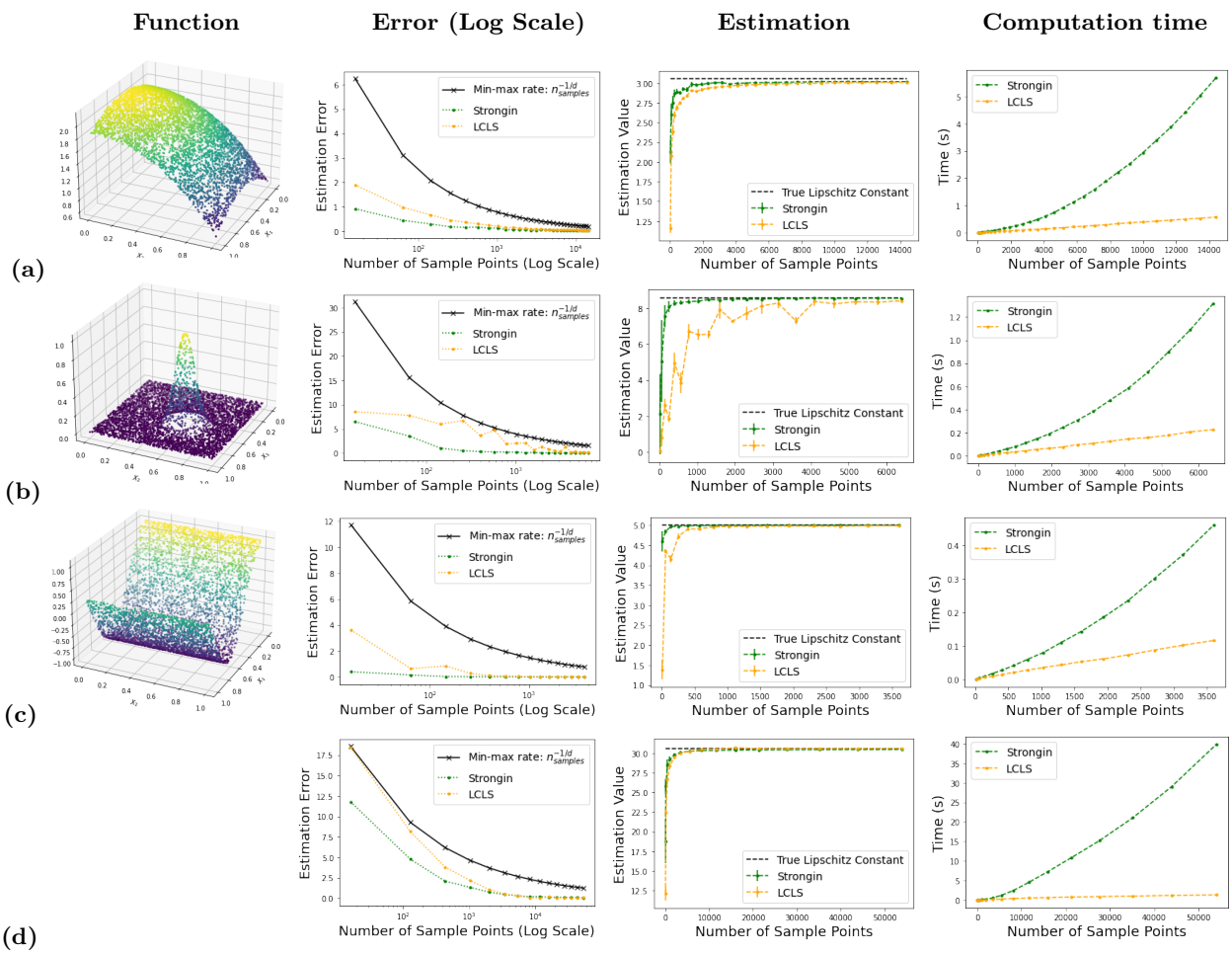

Figure 2: Comparison between the performance of the LCLS algorithm (in orange) and the classical Strongin algorithm (in green) in the noiseless setting. Each row corresponds to a different test function ((a) - (d)) and each column represents a different point of comparison between the two algorithms. From left to right: Column 1: Illustration of the target function where applicable. Column 2: Error of Lipschitz constant estimate - the bound on the sample complexity rate derived in Corollary 2.3 is plotted (in black). Column 3: Behaviour of the sequence of Lipschitz constant estimates. Column 4: Computation time required for each algorithm.

respect to computation time of the Strongin-based benchmark algorithms is already very slow for Function (d) - see Figure 4 and ensuing discussion.

Function	Expression	Lipschitz Const.	Key Property
(a)	See Lemma C.1	3.054	Lipschitz constant reached in a unique point.
(b)	$e^{-(x_1^2+x_2^2)}cos(x_1)cos(x_2)$	8.5776	Large second degree partial derivatives (K).
(c)	$cos(5x_1)$	5	Simple test function.
(d)	See Lemma C.1	30.5399	Higher dimensional input ($\mathbb{R}^3$).

Table 1: Test Functions

As benchmarks we utilise the classical Strongin Lipschitz learning algorithm (Strongin (1973)) in the noiseless setting and the popular modified Strongin-based Lipschitz constant estimator in the bounded noise setting (see in particular Novara et al. (2013), Calliess et al. (2020) and Khajenejad et al. (2021) for applications in control problems). We note that this modified Strongin estimator is strongly dependent on a precise estimate of the smallest upper bound of the noise $\bar{b} > 0$ in order to properly specify $\bar{e} \in \mathbb{R}_+$ hyper-parameter. Indeed,

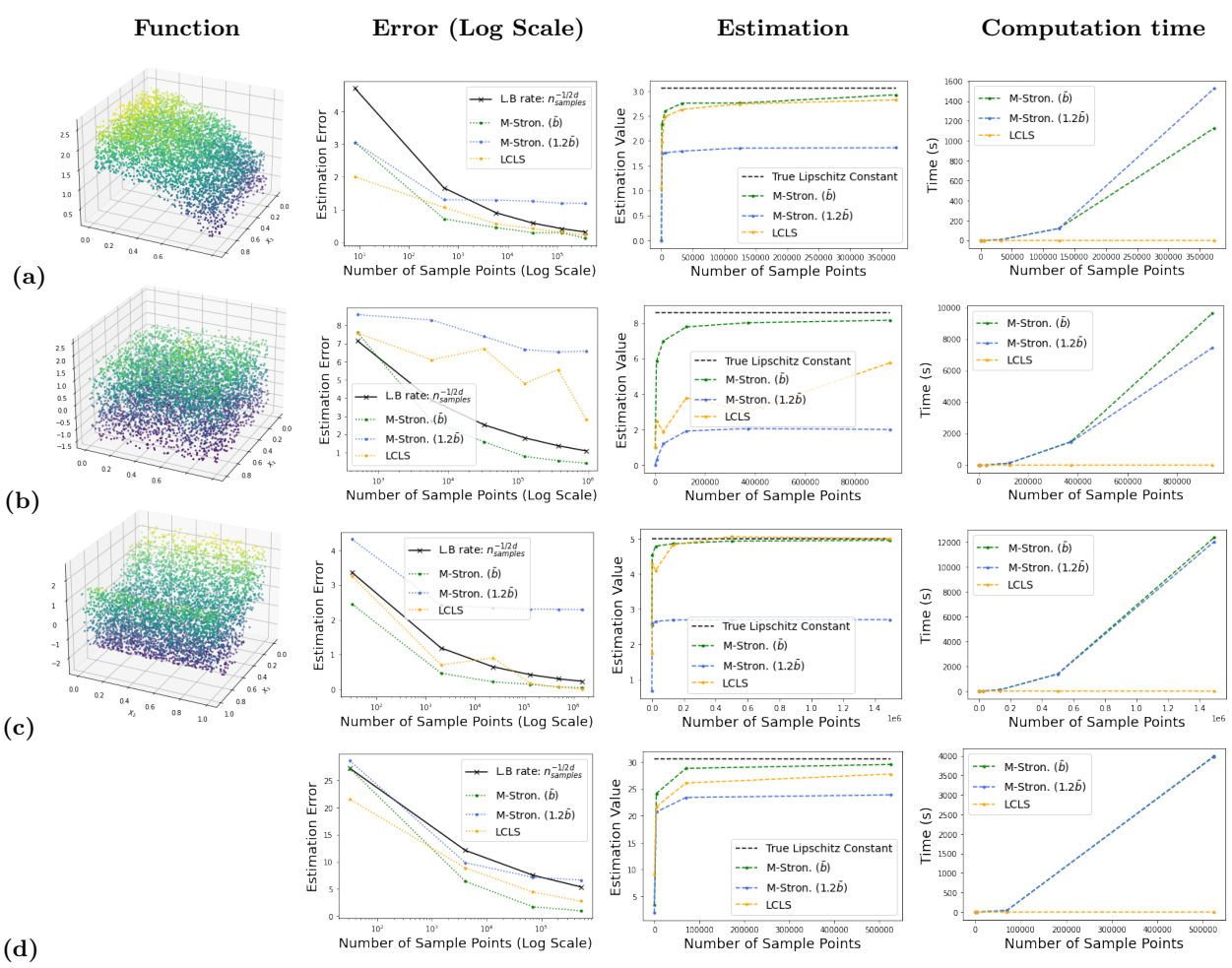

Figure 3: Comparison between the performance of the LCLS algorithm (in orange) and the modified-Strongin algorithm with a correctly (in green) and incorrectly (in blue) specified hyper-parameter in the noisy setting. ach row corresponds to a different test function ((a) - (d)) and each column represents a different point of comparison between the two algorithms. From left to right: Column 1: Illustration of the target function where applicable. Column 2: Error of Lipschitz constant estimate - the bound on the sample complexity rate derived in Corollary 2.3 is plotted (in black). Column 3: Behaviour of the sequence of Lipschitz constant estimates. Column 4: Computation time required for each algorithm.

if $\bar{e}$ is smaller than $\bar{b}$, then the Lipschitz constant estimates generated by the modified Strongin estimator converge to $+\infty$ as the number of observations increases. In contrast, if $\bar{e}$ is bigger then $\bar{b}$ then the generated Lipschitz constant estimates will converge to an underestimate of $L_p^*(f)$ and never be feasible.

Benchmarking algorithms:

- *(Noiseless Setting) Strongin Estimator:*

$$\hat{L} := \max_{i \neq j} \frac{|\tilde{f}_i - \tilde{f}_j|}{\|x_i - x_j\|}$$

.

- *(Noisy Setting) Modified Strongin Estimator:*

$$\hat{L} := \max_{i \neq j} \frac{|\tilde{f}_i - \tilde{f}_j| - 2\bar{e}}{\|x_i - x_j\|}$$

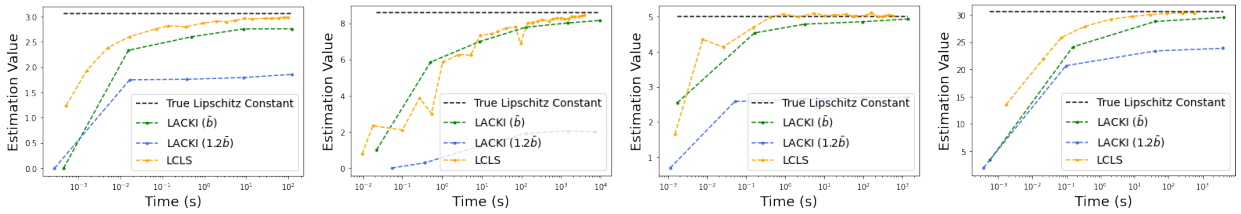

Figure 4: Illustration of convergence speed relative to computation time in the bounded noise setting using the set of test functions given in Table 1. We compare the LCLS algorithm (in orange) and the modified-Strongin algorithm with a correctly (in green) and incorrectly (in blue) specified hyper-parameter. We observe that the LCLS algorithm performs better on all test functions.

where $\bar{e}$ is a hyper-parameter that estimates the tightest upper bound $\bar{b}$ on the noise. We consider modified Strongin Lipschitz estimators with a correctly specified hyper-parameter ($\bar{e} = \bar{b}$) and a hyper-parameter that is slightly larger than the true upper bound ($\bar{e} = 1.2\bar{b}$) as benchmarks.

3.4.2 Discussion

In Figure 2, we illustrate the performance of the LCLS algorithm against the classical Strongin algorithm on the proposed set of test functions. We plot the theoretical lower bounds on the sample complexity rate found in Section 2 in order to provide an intuition for the theoretical bounds. As one would expect, due to the fact that the Strongin algorithm was specifically designed for the noiseless setting, our proposed approach converges more slowly in terms of number of samples on all four test functions. However the difference in convergence speed is not significant and is mitigated by the substantial divergence in computation time. We also note that the plotted sample complexity rate implied by the lower bound of Section 2 does not appear to be tight which is unsurprising as it represents a min-max type bound.

Remark 3.13 *(Link between the proof of Theorem 3.6 and convergence of LCLS) From the proof of Theorem 3.6, we have that the convergence of the LCLS algorithm depends on two factors:*

1. *the diameter of the subsets of the regular partition (upper bounded theoretically using a Taylor expansion).*
2. *the number of samples in each subset (upper bounded theoretically using a multivariate Chebyshev inequality).*

The approximate Lipschitz constant estimation of LCLS in the case of low sample size is mainly due to (1.) especially in the case where the second derivatives of the target function are large (e.g. Function (b) - Easom function).

In Figure 3, we observe the performance of the LCLS algorithm in the bounded noise setting. Here, the convergence speed relative to sample size of the LCLS method differs more significantly from the convergence speed of the correctly specified modified Strongin benchmark algorithm. This is again unsurprising as the correctly specified modified Strongin algorithm makes use of additional information on the noise distribution and the choice of a uniform noise distribution in the experiment is beneficial towards its convergence speed[12]. We note that the modified Strongin algorithm with a slightly incorrectly specified tightest upper bound fails to show any sign of convergence and that the difference in computation time is more significant than in the noiseless setting. The relation between computational complexity and convergence rate of the LCLS and modified Strongin Lipschitz constant estimators is illustrated more precisely in Figure 4 by plotting the convergence rate relative to computation time. We observe that the LCLS estimator performs better on all functions in the test set despite the fact that the modified Strongin estimator utilises additional information on the noise distribution. In particular, for Function (d) which takes inputs in $\mathbb{R}^3$, the LCLS algorithm

[12]If a truncated Gaussian distribution had been used instead, the convergence speed of the modified Strongin estimator could have been arbitrarily slowed by decreasing the variance of the distribution.

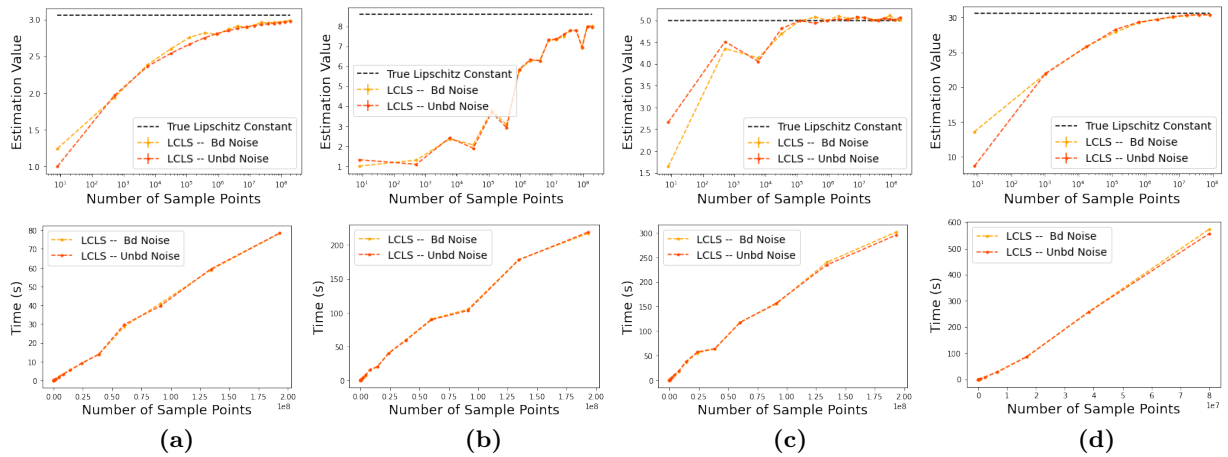

Figure 5: Illustration of the LCLS algorithm in bounded (in light orange) and unbounded noise settings (in dark orange) using the set of test functions given in Table 1. We observe no significant impact of the unboundedness of the noise distribution on the Lipschitz constant estimates produced by LCLS.

needs 8.5 seconds to generate estimates with an estimation error < 0.5, while the Strongin approach needs approximately 4000 seconds. This suggests that for application settings with high sampling capacity and time constraints, the LCLS method should be used even when the modified Strongin algorithm can be properly specified.

In Figures 2, 3 and 4, the performance of the LCLS method seems to be more dependent on the value of the maximum of the second degree partial derivatives than the Strongin-based methods. This can be observed by noting the difference in convergence performance for Function (b) relative to the other test functions[13] and is due to the fact that the LCLS algorithm depends on the maximum to define a sufficiently refined partition of the input space in order to "localise" the computations and generate local Lipschitz constant estimates, see Remark 3.7 and Theorem 3.9 for a precise characterisation of the relationship. In some sense, the stronger dependency on the maximum of the second degree partial derivatives of the target function can be interpreted as the necessary trade-off for the improvement in computation time obtained by the LCLS algorithm.

The last illustration provided in Figure 5 shows the convergence and computation time of the LCLS algorithm in the unbounded noise setting. We do not provide a benchmark as no alternative theoretically backed approaches exist in this setting: the approaches of Beliakov (2005) and Calliess (2017) could be used but do not have any asymptotic convergence guarantees. Instead, we compare the convergence rate to the one obtained by LCLS in the bounded noise setting and observe the fact that no significant performance loss has occurred when the noise is unbounded on any of the test functions.

We conclude this section by remarking that throughout the experiments, our proposed method has been relatively unaffected by the changes in sample setting assumptions and can be used with minimal fine-tuning. Indeed, only the relation between the number of samples in each subset and the diameter of each of these subsets needs to be modified (see Remark 3.13). This relation can be set in a theoretically principled manner by considering the results given in Remarks 3.10 and 3.11 or treated as a hyper-parameter and set more heuristically. The flexibility of the LCLS algorithm is in contrast to existing asymptotically consistent Lipschitz learning algorithms such as the benchmark approaches used in this section which either only consider noiseless sampling settings or require prior knowledge of the noise distribution in order to be applied.

[13] See also Figure 7 and ensuing discussion.

4 Connections to Machine Learning and Related Fields

The theoretical results derived in Section 2 are fundamental in nature. They can be used as a benchmark when developing novel Lipschitz constant estimators or more generally to provide a better theoretical understanding of algorithms that depend explicitly on Lipschitz constant estimates of an underlying target function. Utilising Corollaries 2.3 and 2.7 the worst-case estimation errors of Lipschitz constant estimation can be better understood and their negative impact on overall performance mitigated. This is particularly important as Lipschitz constant dependent algorithms often rely on heuristic or experimental arguments which might not always hold in practice to justify the Lipschitz constant estimation step.

In some settings, the LCLS estimator developed in Section 3 can be directly applied to improve existing computational frameworks in which case the finite sample guarantees derived in Theorem 3.9 and Corollary 3.12 can be used. In particular, when a (loose) bound on the second order partial derivatives of the target function is known, a sequence of feasible Lipschitz constants converging to the best Lispchitz constant at a known convergence speed is obtainable. Unfortunately, while this approach is possible in all the sampling set-ups considered in this paper, the convergence rate obtained for the noisy sampling set-up (see Corollaries 2.3 and 3.12) is generally too slow to be useful in practice. Instead, the LCLS estimator can be applied directly to estimate the Lipschitz constant without feasibility guarantees.

In the section below, we briefly discuss how the results and algorithms derived in this paper can be used in the fields of system identification and global optimisation.

4.1 Global Optimisation:

A major subfield of global optimisation research focuses on sequential search methods that explicitly utilise the Lipschitz constant of the target function to remove large sets in the search space and enhance the efficiency of exploration (Shubert (1972), Mladineo (1986)). As a good estimate of the Lipschitz constant is not always available in practice, work arounds must be be found (Jones et al. (1993)). In particular, a number of these optimisation frameworks make use of a Lipschitz constant estimator (Kvasov and Sergeyev (2012) and references therein, D'Agostino (2022)). The computation of these estimates is generally done heuristically without convergence analysis or error-certificate of the Lipschitz constant estimates. Therefore, the minimax bounds derived in Theorem 2.2 of Section 2 provide a context for the expected performance of these methods. More precisely, given recent work by Malherbe and Vayatis (2017) and Bachoc et al. (2021) which derives optimal sample complexity rates for Lipschitz Optimisation when a Lipschitz constant is known, it becomes possible to derive a lower bound on the sample complexity of adaptive Lipschitz Optimisation algorithms that separate the optimisation procedure and the Lipschitz constant estimation. We derive such a lower bound below as an example of how this can be done.

Following the set-up of certified online learning algorithms described in Bachoc et al. (2021), we assume that we have access to a black-box target function f that can be queried to obtain noiseless observations. The goal of certified global optimisation is to design an algorithm that systematically queries f in order to generate an output sequence $((x_n, f(x_n^*), \zeta_n))_{n\in\mathbb{N}}$ where x_n is the n-th query point, $f(x_n^*)$ is the generated estimate of $\max_{x\in\mathcal{X}} f(x)$ after n queries and $\zeta_n \geq 0$ is an error certificate that guarantees: $\max_{x\in\mathcal{X}} f(x) - f(x_n^*) \leq \zeta_n$.

Given an accuracy $\epsilon \in \mathbb{R}_+$, we can then define the sample complexity [14] $N(A, f, \epsilon)$ of a certified global optimisation algorithm A as the smallest number of queries needed in order to obtain an error certificate smaller than ϵ for all f belonging to a function class $\mathcal{C}$, or in other words:

$$N(A, f, \epsilon) := \min\{n \in \mathbb{N} \cup \{+\infty\} | \zeta_n < \epsilon\}.$$

Utilising this theoretical set-up, we can then combine the theoretical results of Bachoc et al. (2021) with Corollary 2.3 in order to obtain the following statement on the worst case lower sample complexity bound of the adaptive Lipschitz optimisation problem.

[14] Note: this differs slightly from the definition used in Bachoc et al. (2021)

Proposition 4.1 *(Sample Complexity - Adaptive Lipschitz Optimisation) Assume that $\mathcal{X}$ is the hypercube and consider the set $\mathcal{A}$ of adaptive Lipschitz optimisation algorithms which combine classical Lipschitz optimisation methods with a separable[15] feasible Lipschitz constant estimator. There exists constants $C_1, C_2 > 0$ such that $\forall L^* \geq 0$, $A \in \mathcal{A}$ and $\epsilon \in (0, \epsilon_0)$ where $\epsilon_0 \in (0, 2^{d-1}ML^*)$:*

$$C_1\alpha_d(M, L^*, K)(1 + C_2 \max(\min(\frac{3}{C_2}, \frac{1}{\lceil(1 + \log_2(\frac{\epsilon_0}{\epsilon})\rceil)^{\frac{1}{d}} + \beta(L^*, K, \epsilon)}), \frac{\gamma_d(M, L^*, \epsilon)}{\beta(L^*, K, \epsilon)}\Big)^{\frac{1}{2}} - 1))^d \tag{2}$$
$$\leq \sup_{f \in C^2(\mathcal{X}, K) \cap \mathcal{F}_p(L^*)} N(A, f, \epsilon)$$

where $m := \max_{y \in \mathcal{X}} f(y)$, $V_{\mathcal{X}} = M^d$, K is as defined in Assumption 2 and

- $\alpha_d(M, L^*, K) := (\frac{ML^*}{K})^d$ *represents the dependency on the desired precision of the optimisation algorithm and the input space size.*
- $\beta(L^*, K, \epsilon) := (1 + \lceil \log_2(\frac{\epsilon_0}{\epsilon})\rceil)^{1/d} \frac{K\epsilon}{L^{*2}}$ *represents the dependency second degree partial derivatives and true best Lipschitz constant of the target function.*
- $\gamma_d(M, L^*, \epsilon) := \sqrt[d]{\frac{L^*\epsilon(d-1)}{M}}$ *represents the dependency second degree partial derivatives and true best Lipschitz constant of the target function.*

To our knowledge, (2) is the first lower bound on the sample complexity of adaptive Lipschitz optimisation frameworks (see Malherbe and Vayatis (2017) for a possible sample complexity upper bound provided by the adaLIPO algorithm). It depends on the input space, desired precision and upper bounds on the first two orders of differentiation of f. The structure of the proof of Proposition 4.1 as well as the two terms contained in the max expression of the lower bound can be interpreted as a comparison between the sample complexity arising from the optimisation procedure and the one arising from the Lipschitz constant estimation. In particular, $\frac{\gamma_d(L^*, M, \epsilon)}{\beta(L^*, K, \epsilon)}$ is computed by considering the subset of linear functions of $C^2(\mathcal{X}, K) \cap \mathcal{F}_p(L^*)$ which is trivial to optimise in the case where the Lipschitz constant is known but becomes complicated to certify if the Lipschitz constant estimation is difficult.

Unfortunately, the proposed bound is loose as the sampling scheme for the Lipschitz optimisation algorithm can differ significantly from the sampling scheme of Lipschitz constant estimator and is of moderate interest as it only considers a subset of adaptive Lipschitz optimisation algorithms. It does however provide an example of how the lower bounds derived in Section 2 of this paper can be utilised to theoretically analyse existing computational frameworks that rely on Lipschitz learning and future work will consider refining the lower bound given in (2).

Finally, we note that the lower bounds derived in Section 2 can also be considered in the application of recently proposed batch Bayesian optimisation frameworks (González et al. (2016), Alvi et al. (2019)). Indeed, while these methods provide interesting experimental results, the convergence bound stated in Corollary 2.3 shows that in the worst case the Lipschitz constant estimate generated from the fitted Gaussian Process can differ significantly from the true Lipschitz constant - severely impacting the performance of the algorithm in high dimensional settings. At best, the Lipschitz constant estimate used in these papers: $\max_{x \in \mathcal{X}} \|\mu_\nabla(x)\|$ must be replaced by $\max_{x \in \mathcal{X}} \|\mu_\nabla(x)\| + C(d, p)\frac{MK}{\sqrt[d]{n}}$ in order to ensure that the estimated value is a feasible Lipschitz constant. Here μ_∇ denotes the mean function of the gradient function estimate associated to the fitted GP which can be computed efficiently using the covariance function of the GP.

4.2 Nonparametric Regression for System Identification:

A popular system identification method in control settings known as *Nonlinear Set Membership* (Milanese and Novara (2004)) and also referred to as *Lipschitz interpolation* (Beliakov (2006)) or *Kinky inference* (Calliess et al. (2020)) by subsequent authors, explicitly utilises the Lipschitz constant of an underlying Lipschitz continuous target function to define the smallest set of all possible systems that is consistent with the observed data and to provide optimal[16] point estimates. In the relevant literature, a number of

[15] In other words, only knowledge of the Lipschitz constant estimate is used in the optimisation part of the algorithm

[16] see Milanese and Novara (2004).

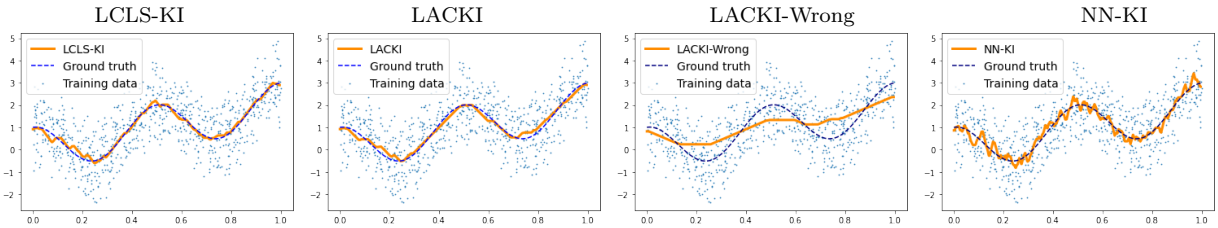

Figure 6: Illustration of several nonparametric methods applied to noisy data. The target function is $f : x \mapsto cos(4\pi x) + 2x$ and the noise is distributed according to a truncated Gaussian distribution (std: 1, upper/lower bound: $-2/2$). The predictions of the trained methods are plotted in red and the training data in light blue (800 observations). From left to write: LCLS-KI: Kinky inference using the Lipschitz constant estimate generated by the LCLS algorithm. LACKI: Adaptive Kinky Inference proposed by Calliess et al. (2020) with correctly set error bounds. LACKI-wrong: LACKI method with error bounds set at the wrong observational error bound (i.e. at 1.3× the true error bound). NN-KI: Kinky inference method using the Lipschitz constant of a fitted Neural Network model with sigmoid activation as proposed by Milanese and Novara (2004).

approaches have been used to estimate the Lispchitz constant however these either rely on heuristic estimation (Milanese and Novara (2004), Calliess (2017)) or on knowledge of often unavailable hyper-parameters such as tight bounds on the noise (Novara et al. (2013),Calliess et al. (2020)) which underestimate the true Lipschitz constant. Utilising the LCLS algorithm developed in Section 3 would therefore be an interesting alternative approach to constructing an adaptive Nonlinear Set Membership framework. As noted at the beginning of the section, we directly utilise the Lipschitz estimate produced by the LCLS estimator as the worst-case error guarantee stated in Corollary 3.12 is of the order $n^{-\frac{1}{2d+4}}$ and is too conservative to be useful in the considered use case.

In Figure 6, we illustrate the performance of a hybrid LCLS - Kinky Inference method in comparison to other nonparametric methods that depend explicitly on the Lipschitz constant of the target function. The variation of the plotted nonparametric predictors is a direct function of the Lipschitz constant estimated from the data – when the Lipschitz constant estimate underestimates the true Lipschitz constant flatter prediction curves that do not fully capture the nonlinearity of the target function are produced while Lipschitz constant estimates that overestimate the true Lipschitz constant produce overly input sensitive predictions. In fact, the kinky inference framework converges to a nearest neighbour estimator as the Lispchitz constant goes to infinity (Maddalena and Jones (2020)).

In Figure 7, we observe that under the truncated Gaussian noise assumptions, the proposed LCLS-KI approach seems to performs best in comparison to the other nonparametric methods as long as the bound on the second derivative K (see Assumption 2) is not too large relative to the number of observations in the training data. As noted in Section 3.4, this is due to the fact that the LCLS algorithm is more dependent on K than other classes of Lipschitz learning algorithms and can significantly underestimate the true Lipschitz

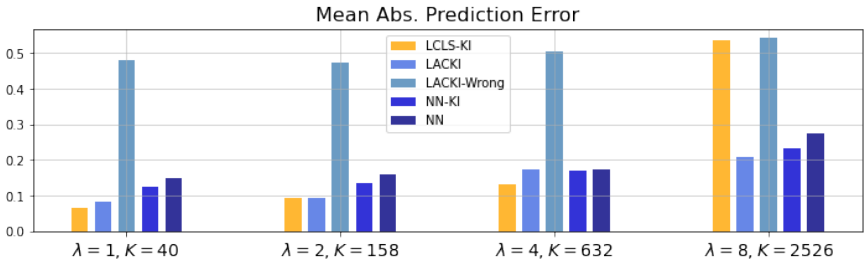

Figure 7: Mean absolute error of the nonparametric methods discussed in Figure 6 and the neural network utilised to estimate the Lipschitz constant in the NN-KI method. The target functions are given by $f_\lambda : x \mapsto cos(2\lambda\pi x) + \lambda x$, for $\lambda = 1, 2, 4, 10$ and associated maximum second derivative: $K := \max_{x \in \mathcal{X}} f''_\lambda(x) = 40, 158, 632, 2526$. The values shown in the plot are computed on a test set contatining 500 independently sampled observations.

constant when K is too large. Therefore, when the second order derivatives are moderate and upper bounds on the noise on the noise are not precisely known, the LCLS-KI algorithm provides an interesting alternative to existing nonlinear set membership/Lipschitz interpolation methods. Applications of LCLS-KI to the common-use case of such methods, e.g. in learning-based model predictive control (Canale et al. (2014), Limon et al. (2017)), could be pursued in future work.

5 Conclusions

In this work, we have established precise lower and upper bounds on the sample complexity of the estimation of Lipschitz constants under minimal parametric constraints on the target function and relaxed noise assumptions. Instead, our bounds rely on the assumption of C^2 regularity of the target function which, given a compact input space, implies the existence of an upper bound on the second degree partial derivatives (this type of assumption is unavoidable as if the second degree partial derivatives are not assumed bounded, then the sample complexity can not be guaranteed to be finite and any theoretical characterisation of the general Lipschitz learning problem is trivial). The obtained bounds on the sample complexity are shown to be optimal in the noiseless sampling setting and to differ slightly by a rate of $\frac{K^2}{\epsilon^4}$ in the noisy sampling setting. These results can be used to provide a theoretical baseline for the Lipschitz learning problem and to help drive the design of future black-box Lipschitz constant estimators.

In order to derive the upper bound on the sample complexity, we have proposed a new algorithm for Lipschitz learning based on local least squares regression that is sample-optimal in the noiseless setting and performs well in the presence of noise. We have thoroughly investigated the theoretical properties of this algorithm showing asymptotic consistency, guarantees on finite sample behaviour and computational complexity in both noiseless and noisy sampling settings. A series of brief empirical experiments illustrate how these theoretical results translate into practice and how the LCLS algorithm compares to existing classical Lipschitz constant estimators. The proposed method provides a suitable solution for Lipschitz constant estimation when a theoretically principled and computationally flexible approach is needed.

Forthcoming work on LCLS will focus on extending the algorithm to recursively compute local Lipschitz constants on observed data and to provide theoretical guarantees on this extension. In addition, future work on theoretical Lipschitz learning will look to improve the sample complexity bound in the noisy setting derived in Section 2 under stronger assumptions on the noise distribution and by restricting the class of Lipschitz learning algorithms.

Appendix: Contents

A Notation

For a partition $\mathcal{H}$ of $\mathcal{X}$ defined following the discussion of subsection 3.1 and a set of samples $D := \{(x_i, \tilde{f}_i)\}_{i\in\{1,...,N\}}$ we recall the following notations. Note; if $\mathcal{H}$ is part of a sequence of partitions $(\mathcal{H})_{I\in\mathbb{N}}$ an additional I index may be used.

1. The subset of samples that belongs to $H \in \mathcal{H}_{\mathcal{I}}$ is denoted $D_I^H := \{(x_i^H, \tilde{f}_i^H)\}_{i\in\{1,...,N_I^H\}}$. Note: samples can only belong to one subset H. If a sample point is on the border between two sets, then it can be included in either design matrix.

2. We denote the design matrices of the least squares regression $X_I^H \in \mathbb{R}^{N_I^H\times(d+1)}$ and the observation vectors $\tilde{f}_I^H \in \mathbb{R}^{N_I^H}$;
$$X_I^H = \begin{bmatrix} 1 & x_{H_1}^\top \\ 1 & x_{H_2}^\top \\ \vdots & \vdots \\ 1 & x_{H_{N_I^H}}^\top \end{bmatrix}, \quad \tilde{f}_I^H = \begin{bmatrix} \tilde{f}_{H_1} \\ \tilde{f}_{H_2} \\ \vdots \\ \tilde{f}_{H_{N_I^H}} \end{bmatrix} = \underbrace{\begin{bmatrix} f_{H_1} \\ f_{H_2} \\ \vdots \\ f_{H_{N_I^H}} \end{bmatrix}}_{f_I^H :=} + \underbrace{\begin{bmatrix} \gamma_{H_1} \\ \gamma_{H_2} \\ \vdots \\ \gamma_{H_{N_I^H}} \end{bmatrix}}_{\gamma_I^H :=}, \text{ where } \forall k \in \{1,...,N_I^H\} \text{ where}$$
$(x_{H_k}, \tilde{f}_{H_k})$ is a sample point contained in D_I^H with $\tilde{f}_{H_k} := \tilde{\Omega}(x_{H_k})$ and by abuse of notation; $\gamma_{H_k} = \gamma_{x_{H_k}}$.

3. We denote by $[\hat{b}_I^H, \hat{\beta}_I^H] \in \mathbb{R}^{d+1}$ (where $b_I^H \in \mathbb{R}$ is the intercept) the least squares coefficients associated to $H \in \mathcal{H}_I$ and computed using X_I^H and $\tilde{f}_{H_{N_I^H}}$.

B Remark on Assumption 4

Remark B.1 *(Some additional comments on Assumption 4:)*

1. *We define "sufficiently large" by considering the balls $B_{\frac{20\epsilon}{K}}(c)$ of radius $\frac{20\epsilon}{K}$. This choice of set size is linked to the maximum fluctuation of the gradient given bounds on the second derivatives. This can be explained more rigorously by following a similar argument to the one presented for the construction of the set $\mathcal{B}$ of hyperspheres in the proof of Theorem 2.2 .*

2. *We consider $\delta \in (0, \frac{1}{2})$. (2) therefore only implies light upper bounds on the probability statement (i.e. upper bounds in $(\frac{1}{2}, 1)$). It is also important to note that $Q(\delta)$ does not depend on ϵ.*

3. *$\epsilon \leq \bar{\epsilon} \leq \frac{MK}{20}$ seems restrictive, however for all classes of Lipschitz constants that we will analyse below, $Q(\delta) > \sigma^2$ which implies that the $C(\sigma^2\delta, d)$ constant given in Theorem 2.5 is smaller than 1. Therefore in the case where $\epsilon \geq \frac{MK}{20}$ we have that $n = 0$ and Theorem 2.5 holds trivially (i.e. no need for Assumption 4).*

4. *Note that in (2) of Assumption 4, $f \in C^2(\mathcal{X}, K)$ can depend on both $\hat{L}_n \in \mathcal{L}_{n,p}(\mathcal{X})$ and $D_\gamma^n \in \mathcal{D}(0, \sigma^2)$ however the choice of $Q(\delta)$ must hold $\forall \hat{L}_n \in \mathcal{L}_{n,p}(\mathcal{X})$ and $\forall D_\gamma^n \in \mathcal{D}(0, \sigma^2)$.*

C Proofs: Lower bounds on Sample Complexity

Lemma C.1 *(Properties of $\mathcal{F}$) For $C_1, C_2 \in \mathbb{R}$, define the function $g_0 : \mathbb{R}^d \to \mathbb{R}$,*
$$g_0(x) = \begin{cases} C_1 e^{-\frac{1}{1-C_2\sum_{j=1}^d (x_j - z_j)^2}} & \text{if } C_2 \sum_{j=1}^d (x_j - z_j)^2 < 1 \\ 0 \text{ otherwise.} \end{cases} .$$

The following properties of g_0 can be shown;

1. $\max_{x\in\mathbb{R}^d} \|\nabla g_0(x)\|_2 \approx 0.8C_1\sqrt{C_2}$
2. $\max_{x\in\mathbb{R}^d, i,j\in\{1,...,d\}} \left|\frac{\partial^2 g_0}{\partial x_i x_j}(x)\right| \approx 7.75C_1C_2$.

Proof Let g_0 be as described above. It follows from construction that g_0 is a radial function and that there exists $u : [0,+\infty) \to \mathbb{R}$, such that $\forall x \in \mathbb{R}^d$, $u(\sum_{i=1}^d x_i^2) = g_0(x)$ (and in other terms, $\forall r \in [0,+\infty)$, $u(r) := g_0(\sqrt{r}, 0, ..., 0)$). We can compute the maximum magnitude (in $\|.\|_2$) of the gradient of g_0 as follows;

$$\max_{x\in\mathbb{R}^d} \|\nabla g_0(x)\|_2 = \max_{x\in\mathbb{R}^d} \|(2x_1u'(\sum_{i=1}^d x_i^2), 2x_2u'(\sum_{i=1}^d x_i^2), ...)\|_2 = \max_{x\in\mathbb{R}^d}\{|2u'(\sum_{i=1}^d x_i^2)|\|x\|_2\}$$

$$= \max_{r\in\mathbb{R}_+} |2u'(r^2)r| = \max_{0\le r\le\frac{1}{\sqrt{C_2}}} \{2C_1C_2r\frac{e^{-\frac{1}{1-C_2r^2}}}{(1-C_2r^2)^2}\} = C_1\sqrt{C_2}\max_{0\le r\le 1}\{2r\frac{e^{-\frac{1}{1-r^2}}}{(1-r^2)^2}\}.$$

Computing $\max_{0\le r\le 1}\{2r\frac{e^{-\frac{1}{1-r^2}}}{(1-r^2)^2}\}$ gives $\max_{0\le r\le 1}\{2r\frac{e^{-\frac{1}{1-r^2}}}{(1-r^2)^2}\} = \frac{6\sqrt[4]{3^3}e^{-\frac{1}{1-\frac{1}{\sqrt{3}}}}}{(\sqrt{3}-3)^2}$. Since g_0 is continuously differentiable and the support of ∇g_0 is compact, we have that there exists $x^* \in \mathbb{R}^d$ such that $\|\nabla g_0(x^*)\|_2 = C_1\sqrt{C_2}\frac{6\sqrt[4]{3^3}}{(\sqrt{3}-3)^2}e^{-\frac{\sqrt{3}}{\sqrt{3}-1}} \approx 0.8C_1\sqrt{C_2}$.

Similarly, we have that for $i \in \{1, ..., d\}$, $x \in \mathbb{R}^d$;

$$\frac{\partial^2 g_0}{\partial x_i^2}(x) = 2u'(\sum_{i=1}^d x_i^2)) + 4x_i^2u''(\sum_{i=1}^d x_i^2)).$$

Here it is clear that for $x^* \in \text{argmax}_{x\in\mathbb{R}^d}|\frac{\partial^2 g_0}{\partial x_i^2}(x)|$ either (1) $x_i^* = 0$ or (2) $x_j^* = 0$, $\forall i \neq j$. In the first case; we can compute $\max_{r\in\mathbb{R}_+}|2u'(r)| = \frac{8C_1C_2}{e^2} \approx 1.08C_1C_2$. In the second case, setting $x = re_i$, we consider the computation of $\max_{r\in\mathbb{R}_+}|2u'(r^2) + 4r^2u''(r^2)|$. We have

$$2u'(r^2) + 4r^2u''(r^2) = 2C_1C_2e^{-\frac{1}{1-r^2}}\frac{3C_2^2r^4-1}{(1-C_2r^2)^4}$$

and can compute

$$\max_{r\in\mathbb{R}_+}|2u'(r^2) + 4r^2u''(r^2)| = C_1C_2\max_{r\in\mathbb{R}_+}|2e^{-\frac{1}{1-r^2}}\frac{3C_2^2r^4-1}{(1-C_2r^2)^4}| \approx 7.75C_1C_2.$$

Therefore, we have $\max_{x\in\mathbb{R}^d}|\frac{\partial^2 g_0}{\partial x_i^2}(x)| \approx 7.75C_1C_2$. Finally, we check $\forall i \neq j \in \{1, ..., d\}$, $\max_{x\in\mathbb{R}^d}|\frac{\partial^2 g_0}{\partial x_i x_j}(x)| = \max_{x\in\mathbb{R}^d}|4x_ix_ju''(\sum_{i=1}^d x_i^2)|$. Clearly, we can set $x = re_i + se_j$ for $r, s \in \mathbb{R}_+$. Computing this quantity gives;

$$\max_{x\in\mathbb{R}^d}|4x_ix_ju''(\sum_{i=1}^d x_i^2)| = C_1C_2^2\max_{(r,s)\in\mathbb{R}_+\times\mathbb{R}_+}|\frac{4rse^{-\frac{1}{1-C_2(r^2+s^2)}}}{(1-C_2(r^2+s^2))^3}| = C_1C_2\frac{8\sqrt{2}e^{-2-\sqrt{2}}}{(\sqrt{2}-2)^3}.$$

We obtain $\forall i \neq j \in \{1, ..., d\}$, $\max_{x\in\mathbb{R}^d}|\frac{\partial^2 g_0}{\partial x_i x_j}(x)| \approx 1.85C_1C_2 \le \max_{x\in\mathbb{R}^d}|\frac{\partial^2 g_0}{\partial x_i^2}(x)|$.

■

Proof of Theorem 2.2 (Sample Complexity Bound 1).
Let $p = 2$. If we can show that there exists a set $\mathcal{F} \subseteq \sup_{f\in C^2(\mathcal{X},K)\cap\mathcal{F}_p(L^*)}$ of functions that can be constructed such that

$$\forall \hat{L} \in \mathcal{L}_{n,p}(\mathcal{X}), \sup_{f\in\mathcal{F}}|\hat{L}(f) - L_p^*(f)| > \epsilon$$

when $n \leq \big(C(d,p)\frac{MK}{\epsilon}\big)^d$, then Theorem 2.2 follows directly. This expression can be simplified to the equivalent statement;

$$\forall \hat{L} \in \mathcal{L}_{n,p}(\mathcal{X}), \exists f \in \mathcal{F} \text{ such that } |\hat{L}(f) - L_p^*(f)| > \epsilon.$$

In order to construct such a set, consider the following functional family. For $C_1, C_2 \in \mathbb{R}$,

$$\mathcal{F}_0(C_1,C_2) := \big\{ g_z : \mathbb{R}^d \to \mathbb{R} | z \in \mathcal{X}, g_z(x) = \begin{cases} C_1 e^{-\frac{1}{1-C_2\sum_{j=1}^d (x_j - z_j)^2}} \text{ if } C_2 \sum_{j=1}^d (x_j - z_j)^2 < 1 \\ 0 \text{ otherwise.} \end{cases} \big\}.$$

For any L^*, we can consider the family $\mathcal{F}_{L^*}(C_1,C_2)$ by adding a linear component, e.g. $L^* x_1$ to $g_z \in \mathcal{F}_0(C_1,C_2)$. In this case, we have by the construction of $\mathcal{F}_0(C_1,C_2)$ that for all $g_z^0 \in \mathcal{F}_0(C_1,C_2)$ with support in $\mathcal{X}$ and $g_z^{L^*} \in \mathcal{F}_0(C_1,C_2)$, $z \in \mathcal{X}$, $\max_{x\in\mathbb{R}^d} \|\nabla g_z^{L^*}\|_2 = \max_{x\in\mathbb{R}^d} \|\nabla g_z^0\|_2 + L^*$ and $\max_{x\in\mathbb{R}^d, i,j\in\{1,\dots,d\}} \big|\frac{\partial^2 g_z^{L^*}}{\partial x_i x_j}\big| = \max_{x\in\mathbb{R}^d, i,j\in\{1,\dots,d\}} \big|\frac{\partial^2 g_z^0}{\partial x_i x_j}\big|$. We can restrict our proof to considering the case where $L^* = 0$ (see comments in parenthesis).

In this proof, we will show that for carefully selected values $C_1^*, C_2^* \in \mathbb{R}$, $\mathcal{X}$ contains $\sim (\frac{AK}{\epsilon})^d$ disjointed $\|.\|_2$-hyperspheres $\mathcal{B} := \{B_i\}_{i\in\{1,\dots,(\frac{AK}{\epsilon})^d\}}$ of radius $\frac{1}{\sqrt{C_2^*}}$ such that; $\forall B_i, B_j \in \mathcal{B}$, $B_i \subset \mathcal{X}$, $B_i \cap B_j = \emptyset$ if $i \neq j$ and a set $\mathcal{F} \subset \mathcal{F}_0$ of associated functions with the following properties; $\forall g_{\bar{z}_i} \in \mathcal{F}$ associated to $B_i \in \mathcal{B}$,

1. $supp(g_{\bar{z}_i}) = B_i$,
2. $\max_{x\in\mathcal{X}} \|\nabla g_{\bar{z}_i}(x)\|_2 \geq 2\epsilon \quad (+L^* \text{ if } L^* \neq 0)$,
3. $\forall i,j \in \{1,\dots,d\}$, $\max_{x\in\mathcal{X}} |\frac{\partial^2 g_{\bar{z}_i}}{\partial x_i x_j}(x)| \leq K$.

Since every $g_{\bar{z}_i} \in \mathcal{F}$ is analytical, $\mathcal{F} \subset C^2(\mathcal{X}, K)$. This implies that a Lipschitz learning algorithm $\hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})$ needs to sample at least one data point in each hypersphere of $B_i \in \mathcal{B}$ in order to ensure that the difference between the Lipschitz constant estimate and the true Lipschitz constant is smaller than ϵ for every $g \in \mathcal{F} \cup \{h \equiv 0\}$ (in the case where L^* is not assumed to be equal to 0, we can consider the linear component of $\mathcal{F}_L^*(C_1,C_2)$ instead of the zero function). The number of sample points n utilised by a Lipschitz learning algorithm must therefore be greater than $\sim (\frac{AK}{\epsilon})^d$ in order to ensure this ϵ-precision for all $g_{\bar{z}_i} \in \mathcal{F}$.

We begin by showing that such values of C_1^*, C_2^* exist. To do so we consider the gradient and second order partial derivatives of the functions in $\mathcal{F}$. Let $g \in \mathcal{F}$, applying Lemma C.1, we have :

1. $\max_{x\in\mathbb{R}^d} \|\nabla g\|_2 \approx 0.8 C_1 \sqrt{C_2} \quad (+L^* \text{ if } L^* \neq 0)$
2. $\max_{x\in\mathbb{R}^d, i,j\in\{1,\dots,d\}} \big|\frac{\partial^2 g}{\partial x_i x_j}\big| \approx 7.75 C_1 C_2$.

Using these values, we can define the values of C_1^* and C_2^* discussed earlier in the proof. Firstly, in order to have $g \in C^2(\mathbb{R}^d, K)$, we need $\max_{x\in\mathbb{R}^d, i,j\in\{1,\dots,d\}} |\frac{\partial^2 g}{\partial x_i x_j}(x)| \leq K$. This implies the relation $C_1 = \frac{K}{7.75 C_2}$. Secondly, we set C_2 such that $\max_{x\in\mathcal{X}} \|\nabla g(x)\|_2 = 0.8 C_1 \sqrt{C_2} = 2\epsilon$. Plugging in the relation for C_1 given above;

$$\frac{0.1K}{\sqrt{C_2}} = 2\epsilon \Leftrightarrow (\frac{K}{20\epsilon})^2 = C_2^* \text{ and } C_1^* = \frac{51\epsilon^2}{K}.$$

Setting $l = \frac{1}{\sqrt{C_2^*}} = \frac{20\epsilon}{K}$ we have

$$supp(g) = \{x \in \mathbb{R}^d | C_2^* \sum_{i=1}^d x_i^2 < 1\} = B_l(c)$$

where $B_l(z)$ denotes the d-dimensional ball of radius l defined with respect to $\|.\|_2$ and centered in $c \in \mathcal{X}$. The last step before defining $\mathcal{F}$ is to count how many sphere of radius l can fit in $\mathcal{X} = [0, M]^d$. Here, we use a lower bound that is obtained by considering the regular hypercube partition of $\mathcal{X}$ of side-length $\tilde{l}$, defined by; $N := \lfloor \frac{M}{l} \rfloor$ and $\tilde{l} = \frac{M}{N}$. Let $\mathcal{B}$ denote the set of balls of radius l that can be inscribed in a subset belonging to the hypercube partition of $\mathcal{X}$. Then, for all $B_i, B_j \in \mathcal{B}$, $B_i \subset \mathcal{X}$ and $B_i \cap B_j = \emptyset$. Furthermore, we have $|\mathcal{B}| = (\frac{M}{\tilde{l}})^d \approx \left(\frac{MK}{20\epsilon}\right)^d$ (+ constant).

The associated set $\mathcal{F}$ of functions can be constructed by utilising the set $\mathcal{Z}$ of ball centers z_i for $B_i \in \mathcal{B}$ and the values C_1^*, C_2^* computed above to define

$$\mathcal{F} := \{g_z \in \mathcal{F}_0(C_1^*, C_2^*) | z \in \mathcal{Z}\} \cup \{f \equiv 0\}.$$

As noted in the beginning in the proof this construction implies that the number of sample points n utilised by a Lipschitz learning learning algorithm must be greater than $\frac{1}{20^d}\left(\frac{MK}{\epsilon}\right)^d$ in order to ensure that the difference between the Lipschitz estimate and the Lipschitz constant is smaller than ϵ for all $g \in \mathcal{F}$

Utilising norm equivalences and Lemma D.2, we have that in the case $p = 1$, the sample complexity can of Lipschitz learning problem can be lower bounded by $\left(C(d)\frac{MK}{\epsilon}\right)^d$, where $C(d) = \frac{1}{20d^{\frac{1}{2}}}$.

■

Proof of Theorem 2.5 (Sample Complexity Bound 2).
Theorem 2.5 is implied by the following statement: $\forall \epsilon > 0, \delta \in (0, \frac{1}{4})$,

$$n \leq \left(\tilde{C}(\sigma^2, \delta, d)\frac{MK}{\epsilon}\right)^{2d} \implies \inf_{\hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})} \sup_{f \in C^2(\mathcal{X}, K) \cap \mathcal{F}_p(L^*)} \sup_{D_\gamma^n \in \mathcal{D}(0, \sigma^2)} \mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| > \epsilon) > \delta$$

where $\tilde{C}(\sigma^2, \delta, d) = \frac{\sigma^2}{20log(2)Q^{*2}d^{\max\{\frac{1}{p}-\frac{1}{2}, 0\}}}$. In this proof we will show that the stronger statement given above holds. As with Theorem 2.2, we consider the case $L^* = 0$ and treat the case $L^* > 0$ by adding a linear component to functions in the set $\mathcal{F}$ defined in the proof of Theorem 2.2 and used in this proof. To alleviate notation, we remove $\mathcal{F}_p(L^*)$ from the sup expression.

Let $\hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})$ be arbitrary and denote by $G^{\hat{L}}$ the set of samples generated and used by $\hat{L}$. By the proof given for Theorem 2.2, $G^{\hat{L}}$ needs to contain at least one sample point in each of the $\left(C(d,p)\frac{MK}{\epsilon})\right)^d$ disjointed balls of radius $\frac{20\epsilon}{K}$ in $\mathcal{X}$ in order to obtain a Lipschitz estimate whose difference with the Lipschitz constant is smaller than ϵ in the noiseless sampling set-up. It is clear that this must hold true for the noisy sampling set-up as well. Indeed, let $\mathcal{B}$ denote the set of these disjointed balls and $\mathcal{F}$ denote the set of associated functions defined in the proof of Theorem 2.2. Suppose that there exists a $B \in \mathcal{B}$ that does not have a sample point and let $g_B \in \mathcal{F}$ be the associated function to B. For arbitrary $D_\gamma^n \in \mathcal{D}(0, \sigma^2)$, suppose that $\mathbb{P}_{D_\gamma^n}(|\hat{L}(g_B) - L_p^*(g_B)| > \epsilon) < \delta$. By construction of $\mathcal{F}$, $L_p^*(g_B) = 2\epsilon$ and this statement is equivalent to $\mathbb{P}_{D_\gamma^n}(\hat{L}(g_B) \in (\epsilon, 3\epsilon)) > 1 - \delta$. Now consider the zero function $f \in C^2(\mathcal{X}, K)$, $f \equiv 0$. As $g_B(x) = 0, \forall x \notin B$, we have that data samples $(x, \tilde{f}(x))$ generated for f and samples $(x, \tilde{g}_B(x))$ generated for g_B will be equal. Therefore, $\hat{L}(f) = \hat{L}(g_z)$ and $\mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| < \epsilon) = \mathbb{P}_{D_\gamma^n}(|\hat{L}(f)| < \epsilon) \leq 1 - \mathbb{P}_{D_\gamma^n}(\hat{L}(g_B) \in (\epsilon, 3\epsilon)) < \delta$ which means that $\mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| > \epsilon) > 1 - \delta > \delta$ since $\delta \in (0, \frac{1}{4})$.

For $B \in \mathcal{B}$, let $G_B^{\hat{L}} := (x_i^{\hat{L}}, \tilde{\Omega}(x_i^{\hat{L}}))_{x_i \in B}$ denote the set of generated samples that belong to B.
Consider an arbitrary Lipschitz learning algorithm $\hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})$. By the argument given in previous paragraph, we have $|G_B^{\hat{L}}| \geq 1$. Furthermore, since $\bar{B}$ (closure of B) is compact and convex and by construction $\sup_{f \in C^2(\mathcal{X}, K)} L_p^*(f|_B) - L_p^*(f|_{\mathcal{X} \setminus B}) \geq L_p^*(g_B|_B) - L_p^*(g_B|_{\mathcal{X} \setminus B}) = 2\epsilon - 0$, we can apply Assumption 4 on B in order to obtain $Q_B(2\delta) \in \mathbb{R}_+$ such that $\forall D_\gamma^n \in \mathcal{D}(0, \sigma^2)$, we have that one of the two following statements hold:

1. $\forall f_{B,\hat{L}} \in C^2(\mathcal{X}, K)$: $\mathbb{P}_{D_\gamma^n}\Big(\frac{1}{n_B^{\hat{L}}}\big|\sum_{i=1}^{n_B^{\hat{L}}} \tilde{\Omega}(x_i^{\hat{L}}) - f_{B,\hat{L}}(x_i^{\hat{L}})\ \big| > Q_B\Big) = 0.$

2. $\exists f_{B,\hat{L}} \in C^2(\mathcal{X}, K)$: $\mathbb{P}_{D_\gamma^n}\Big(|\hat{L}(f_{B,\hat{L}}) - L_p^*(f_{B,\hat{L}})| \leq \epsilon \Big| \frac{1}{n_B^{\hat{L}}}\big|\sum_{i=1}^{n_B^{\hat{L}}} \tilde{\Omega}(x_i^{\hat{L}}) - f_{B,\hat{L}}(x_i^{\hat{L}})\ \big| > Q_B(2\delta)\Big) \leq 1 - 2\delta.$

where $n_B^{\hat{L}} := |G_B^{\hat{L}}|$. Note: the dependency of $f_{B,\hat{L}}$ on D_γ^n is made implicit to alleviate notation. The second statement is equivalent to

$$\mathbb{P}_{D_\gamma^n}\Big(|\hat{L}(f_{B,\hat{L}}) - L_p^*(f_{B,\hat{L}})| > \epsilon \Big| \frac{1}{n_B^{\hat{L}}}\big|\sum_{i=1}^{n_B^{\hat{L}}} \tilde{\gamma}_{x_i^{\hat{L}}}\ \big| > Q_B(2\delta)\Big) > 2\delta$$

where the $\gamma_{x_i^{\hat{L}}}$ do not depend on the choice of sampling input $x_i^{\hat{L}}$. Applying this approach to all $B \in \mathcal{B}$, we obtain an associated set of $\{Q_B(2\delta)\}_{B\in\mathcal{B}}$ and $\{f_{B,\hat{L}}\}_{B\in\mathcal{B}}$. Defining, $Q^* := \max_{B\in\mathcal{B}} |Q_B(2\delta)|$, we have that $\forall D_\gamma^n \in \mathcal{D}(0,\sigma^2)$ for which the second statement given above holds:

$$\sup_{f\in C^2(\mathcal{X},K)} \mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| > \epsilon \big| \exists B \in \mathcal{B} \text{ s.t. } \frac{1}{n_B^{\hat{L}}}|\sum_{i=1}^{n_B^{\hat{L}}} \tilde{\gamma}_{x_i^{\hat{L}}}| > Q^*) > 2\delta.$$

This implies that $\forall D_\gamma^n \in \mathcal{D}(0,\sigma^2)$:

$$\sup_{f\in C^2(\mathcal{X},K)} \mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| > \epsilon) \geq \mathbb{P}_{D_\gamma^n}(\exists B \in \mathcal{B} \text{ such that } |\frac{1}{n_B^{\hat{L}}}\sum_{i=1}^{n_B^{\hat{L}}} \tilde{\gamma}_{x_i^{\hat{L}}}| > Q^*)2\delta$$

$$= 2\delta(1 - \prod_{B\in\mathcal{B}} \mathbb{P}_{D_\gamma^n}(|\frac{1}{n_B^{\hat{L}}}\sum_{i=1}^{n_B^{\hat{L}}} \tilde{\gamma}_{x_i^{\hat{L}}}| \leq Q^*)).$$

Here the product is maximised if $|G_B^{\hat{L}}|$ is constant and minimum for all $B \in \mathcal{B}$. Let us therefore assume that this is the case and set $N := \min_{B\in\mathcal{B}} |G_B^{\hat{L}}|$, we have,

$$\sup_{f\in C^2(\mathcal{X},K)} \mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| > \epsilon) \geq 2\delta(1 - \mathbb{P}_{D_\gamma^N}(|\frac{1}{N}\sum_{i=1}^{N} \tilde{\gamma}_{x_i^{\hat{L}}}| \leq Q^*)^{|\mathcal{B}|})$$

$$= 2\delta(1 - (1 - \mathbb{P}_{D_\gamma^N}(|\frac{1}{N}\sum_{i=1}^{N} \tilde{\gamma}_{x_i^{\hat{L}}}| > Q^*)^{|\mathcal{B}|}))$$

Let Γ_N be the random variable defined by $\Gamma_N := \frac{1}{N}\sum_{i=1}^{N} \tilde{\gamma}_{x_i^{\hat{L}}}$. Since the $\tilde{\gamma}_{x_i^{\hat{L}}}$ are uncorrelated $\forall (x_i^{\hat{L}}, \tilde{\Omega}(x_i^{\hat{L}})) \in G_B^{\hat{L}}$ and $D_{\gamma_{x_i^{\hat{L}}}} \in \mathcal{D}(0,\sigma^2)$ we have $\mathbb{E}[\Gamma_N] = 0$ and $\text{var}[\Gamma_N] = \frac{\sigma^2}{N}$. It is well known that the classical Chebyshev inequality is tight for the discrete distribution $\tilde{\Gamma} \in \mathcal{D}(0, \frac{\sigma^2}{N})$:

$$\mathbb{P}_{D_\gamma}(|\tilde{\Gamma} - 0| \geq Q^*) = \frac{\sigma^2}{NQ^{*2}}, \text{ for } \tilde{\Gamma} := \begin{cases} -Q^* \text{ with probability } \frac{\sigma^2}{2NQ^{*2}} \\ Q^* \text{ with probability } \frac{\sigma^2}{2NQ^{*2}} \\ 0 \text{ with probability } 1 - \frac{\sigma^2}{NQ^{*2}} \end{cases}.$$

Following (Rujeerapaiboon et al. (2018)), we have that since $\{\tilde{\gamma}_{x_i^{\hat{L}}}\}_{(x_i^{\hat{L}},\tilde{\Omega}(x_i^{\hat{L}}))\in G_B^{\hat{L}}}$ is assumed to be uncorrelated but not independent or identically distributed, then the projection properties of $D(0,\sigma^2)$ imply that Γ_N can follow any distribution of mean 0 and variance $\frac{\sigma^2}{N}$ (Yu et al. (2009)). Therefore, we have

$$\sup_{D_\gamma^N \in \mathcal{D}(0,\sigma^2)} 2\delta(1 - \mathbb{P}_{D_\gamma^N}(|\Gamma_N| > Q^*)^{|\mathcal{B}|}) = 2\delta(1 - (1 - \frac{\sigma^2}{NQ^{*2}})^{|\mathcal{B}|}).$$

In order to obtain the largest lower bound on the sample complexity, we compute the greatest N such that $2\delta(1 - (1 - \frac{\sigma^2}{NQ^{*2}})^{|\mathcal{B}|}) \geq \delta$. Setting $2\delta(1 - (1 - \frac{\sigma^2}{NQ^{*2}})^{|B|}) = \delta$, yields $N = \frac{\sigma^2}{Q^{*2}} \frac{1}{1 - \sqrt[|\mathcal{B}|]{\frac{1}{2}}}$. This last term behaves approximately linearly in $|\mathcal{B}|$; its Laurent series is given by $\frac{\sigma^2|\mathcal{B}|}{log(2)Q^{*2}} + \frac{\sigma^2}{2Q^{*2}} + \frac{\sigma^2 log(2)}{12Q^{*2}|\mathcal{B}|} + O(\frac{1}{|\mathcal{B}|^2})$ as $|\mathcal{B}|$ goes to infinity and it can be lower bounded by; $\frac{\sigma^2|\mathcal{B}|}{log(2)Q^{*2}}$. Therefore, since the above reasoning holds for arbitrary $\hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})$ we have $\forall \hat{L} \in \mathcal{L}_{n,p}$,

$$\text{If } \min_{B \in \mathcal{B}} |G_B^{\hat{L}}| < \frac{\sigma^2|\mathcal{B}|}{log(2)Q^{*2}}, \text{ then } \sup_{f \in C^2(\mathcal{X},K)} \sup_{D_\gamma^n \in \mathcal{D}(0,\sigma^2)} \mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| > \epsilon) > \delta.$$

This implies that $\forall \hat{L} \in \mathcal{L}_{n,p}$,

$$\text{If } n < \frac{\sigma^2|\mathcal{B}|^2}{log(2)Q^{*2}}, \text{ then } \sup_{f \in C^2(\mathcal{X},K)} \sup_{D_\gamma^n \in \mathcal{D}(0,\sigma^2)} \mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| > \epsilon) > \delta.$$

As the the bound $\frac{\sigma^2|\mathcal{B}|^2}{log(2)Q^{*2}}$ does not depend on $\hat{L}$ (as by Assumption 4, $\{Q_B\}_{B \in \mathcal{B}}$ does not depend on $\hat{L}$), we have that if

$$n < \frac{\sigma^2|\mathcal{B}|^2}{log(2)Q^{*2}} = \tilde{C}(\sigma^2, \delta, d)\big(\frac{AK}{\epsilon}\big)^{2d}$$

where $\tilde{C}(\sigma^2, \delta, d) = \frac{\sigma^2}{20log(2)Q^{*2}d^{\max\{\frac{1}{p} - \frac{1}{2}, 0\}}}$, then

$$\inf_{\hat{L} \in \mathcal{L}_{n,p}(\mathcal{X})} \sup_{f \in C^2(\mathcal{X},K)} \sup_{D_\gamma^n \in \mathcal{D}(0,\sigma^2)} \mathbb{P}_{D_\gamma^n}(|\hat{L}(f) - L_p^*(f)| > \epsilon) > \delta.$$

■

D Proofs: Theoretical Properties of LCLS

Proof of Proposition 3.3 (Computational Complexity of LCLS).
Follows directly from the computational complexity of the linear least squares regression algorithm which is $O(n_{samples})$.

■

D.1 Technical Lemmas

The proof of Theorem 3.6 relies on the following technical lemmas.

Lemma D.1 *(Fundamental logarithm inequalities) For all $x > 0$,*

$$1 - \frac{1}{x} \leq \log(x) \leq x - 1.$$

Lemma D.2 *Let $\mathcal{X}$ be as described in Assumption 1. If f verifies Assumption 2, then f is L_p^*-Lipschitz with respect to $\|.\|_p$. Furthermore, for $p = 1, 2$, $L_p^* = \max_{x\in\mathcal{X}}\{\|\nabla f(x)\|_q\}$ and for $p > 2$, $L_p^* \leq \max_{x\in\mathcal{X}}\{\|\nabla f(x)\|_q\}$ where q is the Holder conjugate of p.*

Proof f is L_p^*-Lipschitz follows directly from the fact that $\mathcal{X}$ is compact and $f \in C^1(\mathcal{X})$.

$\forall p \in \mathbb{N}$, $L_p^* \leq \max_{x\in\mathcal{X}}\{\|\nabla f(x)\|_q\}$ follows from the multidimensional mean-value theorem and an application of the Holder inequality.

For $p = 1, 2$, we show $L_p^* \geq \max_{x\in\mathcal{X}}\{\|\nabla f(x)\|_q\}$. Consider : $\forall x \in \mathcal{X}$ consider the Frechet derivative of f at x; $\lim_{t\to 0} \frac{|f(x+th)-f(x)-\nabla f(x)^\top(th)|}{t} = 0$ $\forall h \in \mathbb{R}^d$. Then, choosing $h = \nabla f(x)$ for $p = 2$ and $h = e_{i^*}$ such that $|\nabla f(x)^\top e_{i^*}| = \|\nabla f(x)\|_\infty$ for $p = 1$ gives $L_p^* \geq \max_{x\in\mathcal{X}}\{\|\nabla f(x)\|_q\}$. (Note that this reasoning is well defined because $f \in C^1(\tilde{\mathcal{X}})$ and $\tilde{\mathcal{X}}$ is an open set that contains $\mathcal{X}$ which implies that $f(x+th)$ is well-defined for any $h \in \mathbb{R}^d$ and small enough t).

■

Lemma D.3 *Consider a sequence of partitions $(\mathcal{H}_I)_{I\in\mathbb{N}}$ used by LCLS and assume that for a given $\eta \in (0, 1]$ the sampling distribution satisfies Assumption 5. Then,*

$$\forall I \in \mathbb{N},\ \forall H \in \mathcal{H}_I,\ \|\left({X_I^H}^\top X_I^H\right)^{-1}\|_2 \leq \frac{16}{\eta {\delta_I^H}^2 N_I^H}.$$

Proof Let $\lambda_{max}(M)$ denote the maximum eigenvalue of a matrix M if it exists. $\forall I \in \mathbb{N}$, $\forall H \in \mathcal{H}_I$, we have $\|\left({X_I^H}^\top X_I^H\right)^{-1}\|_2 = \frac{1}{\sigma_{min}({X_I^H}^\top X_I^H)}$ where $\sigma_{min}({X_I^H}^\top X_I^H)$ denotes the smallest singular value of ${X_I^H}^\top X_I^H$. Therefore, we can focus on showing the following relation that implies the Lemma statement:

$$\sigma_{min}({X_I^H}^\top X_I^H) \geq \frac{\eta N_I^H}{16} {\delta_I^H}^2.$$

Let $\bar{X}_I^H$ be the design matrix without the first column of ones, ie. $\bar{X}_I^H = \begin{bmatrix} x_{H_1}^\top \\ x_{H_2}^\top \\ \vdots \\ x_{H_{N_I^H}}^\top \end{bmatrix}$, we have

$$\sigma_{min}({X_I^H}^\top X_I^H) = \sigma_{min}({X_I^H}^\top)^2 = \min_{\|u\|_2=1} \|{X_I^H}^\top u\|_2^2 = \min_{\|u\|_2=1} \|\begin{bmatrix} \mathbf{1}_{N_I^H}^\top \\ \bar{X}_I^{H\top} \end{bmatrix} u\|_2^2$$

$$\geq \min_{\|u\|_2=1} \| \begin{bmatrix} \mathbf{0}_{N_I^H}^\top \\ \bar{X_I^H}^\top \end{bmatrix} u\|_2^2 = \min_{\|u\|_2=1} \|\bar{X_I^H}^\top u\|_2^2 = \sigma_{min}(\bar{X_I^H}^\top \bar{X_I^H}).$$

Therefore we can consider the smallest singular value of $\bar{X_I^H}$ instead of X_I^H which allows for a direct use of Assumption 5. We have

$$\sigma_{min}(\bar{X_I^H}^\top \bar{X_I^H}) = \lambda_{min}(\bar{X_I^H}^\top \bar{X_I^H}) = \min_{u\in\mathbb{R}^d, \|u\|_2=1} u^\top \bar{X_I^H}^\top \bar{X_I^H} u = \min_{u\in\mathbb{R}^d, \|u\|_2=1} \sum_{i=1}^{N_I^H} \langle x_{H_i}, u\rangle^2.$$

Let $u_{min}^* \in \mathbb{R}^d$ denote the eigenvector associated to $\lambda_{min}(\bar{X_I^H}^\top \bar{X_I^H})$ with $\|u_{min}^*\|_2 = 1$. Then, u_{min}^* satisfies $\sum_{i=1}^{N_I^H} \langle x_{H_i}, u_{min}^*\rangle^2 = \min_{\|u\|_2=1} \sum_{i=1}^{N_I^H} \langle x_{H_i}, u\rangle^2$.

Using u_{min}^*, we can construct an orthonormal basis of $\{u_{min}^*, u_2, ..., u_d\}$ of $\mathbb{R}^d$ (such a basis can be constructed using the Gramm-Schmitt algorithm). Then, since D_I^H contains ηN_I^H distinct $\frac{\delta_I^H}{4}$-covers, we have that there exists at least ηN_I^H pairs of datapoints $(x_{H_i}, \tilde{f}_{H_j})$, $(x_{H_j}, \tilde{f}_{H_j}) \in D_I^H$ such that $\exists \{\alpha_i\}_{i\in\{1,...,d\}}, \alpha_i \in \mathbb{R}$ with $|\alpha_1| > \frac{\delta_I^H}{2}$ and $(x_{H_i} - x_{H_j}) = \alpha_1 u_{min}^* + \sum_{k=2}^{d} \alpha_k u_k$. This implies that $\max(|\langle x_{H_i}, u_{min}^*\rangle|, |\langle x_{H_j}, u_{min}^*\rangle|) \geq \frac{\delta_I^H}{4}$. Indeed, if $|\langle x_{H_i}, u_{min}^*\rangle| < \frac{\delta_I^H}{4}$, then

$$|\langle x_{H_j}, u_{min}^*\rangle| = |\langle x_{H_j} - x_{H_i} + x_{H_i}, u_{min}^*\rangle| = |\langle x_{H_j} - x_{H_i}, u_{min}^*\rangle + \langle x_{H_i}, u_{min}^*\rangle| \geq \frac{\delta_I^H}{2} - \frac{\delta_I^H}{4} = \frac{\delta_I^H}{4}$$

Using this inequality ηN_I^H times we can conclude, $\sigma_{min}(\bar{X_I^H}^\top \bar{X_I^H}) = \sum_{i=1}^{N_I^H} \langle x_{H_i}, u_{min}^*\rangle^2 \geq \eta N_I^H (\frac{\delta_I^H}{4})^2$.

■

Lemma D.4 *Consider the constructions of Definition 3.2. The following relationship holds for all $I \in \mathbb{N}$,*

$$\frac{V(\mathcal{X})\Gamma(\frac{d}{2}+1)2^d}{\pi^{\frac{d}{2}} \max_{H\in\mathcal{H}_I} (\Delta_I^H)^d} \leq |\mathcal{H}_I| \leq \frac{V(\mathcal{X})\Gamma(\frac{d}{2}+1)2^d}{\pi^{\frac{d}{2}} \min_{H\in\mathcal{H}_I} (\delta_n)^d}$$

where $V(\mathcal{X})$ denotes the volume of $\mathcal{X}$.

Proof Follows directly from the definition of $\{\delta_I^H\}_{H\in\mathcal{H}_I}$, $\{\Delta_I^H\}_{H\in\mathcal{H}_I}$ and volume formula for the d-dimensional ball.

■

Lemma D.5 *Let the notation and assumptions be as described in Theorem 3.6 and define $x^* \in \mathcal{X}$ as $x^* \in argmax_{x\in\mathcal{X}} \|\nabla f(x)\|_q$. Then, $\forall I \in \mathbb{N}$,*

$$\left| \|\nabla f(x^*)\|_q - \max_{H\in\mathcal{H}_I} \{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\} \right| \leq \frac{4\sqrt{d} d n_q K}{\sqrt{\eta}} a_I.$$

where $n_q = d^{\max\{\frac{1}{q}-\frac{1}{2}, 0\}}$.

Proof Note: such an x^* exists by compactness of $\mathcal{X}$ and the fact that $f \in C^2(\mathcal{X})$.

By definition, $[\hat{b}_I^H, \hat{\beta}_I^H]^\top = (X_I^{H\top} X_I^H)^{-1} X_I^{H\top} \tilde{f}_I^H$. Computing the expectation of this expression yields

$$\mathbb{E}[[\hat{b}_I^H, \hat{\beta}_I^H]^\top] = \mathbb{E}[(X_I^{H\top} X_I^H)^{-1} X_I^{H\top} \tilde{f}_I^H]$$

$$= \mathbb{E}[(X_I^{H^\top} X_I^H)^{-1} X_I^{H^\top} f_I^H] + \mathbb{E}[(X_I^{H^\top} X_I^H)^{-1} X_I^{H^\top} \gamma_I^H] = (X_I^{H^\top} X_I^H)^{-1} X_I^{H^\top} f_I^H.$$

$\forall H \in \mathcal{H}_I$, let $c_H \in \overline{H}$ (closure of H) be such that $\|\nabla f(c_H)\|_q = \max_{x \in \bar{H}} \{\|\nabla f(x)\|_q\}$ which exists by compactness of $\bar{H}$ and the fact that $f \in C^2(\mathcal{X})$. Then, using the second order Taylor expansion of f around c_h, every coordinate f_{H_k} of f_I^H can be re-expressed as

$$f_{H_k} = f(c_H) + (x_{H_k} - c_H)^\top \nabla f(c_H) + (x_{H_k} - c_H)^\top Hess(c_H + r_{H_k}(x_{H_k} - c_H))(x_{H_k} - c_H), \tag{3}$$

where $r_{H_k} \in [0,1]$ and $Hess$ denotes the Hessian matrix of f. To alleviate notation, let $\|.\|_{\tilde{q}}$ denote a pseudo-norm on $\mathbb{R}^{d+1}$ defined by; $x \in \mathbb{R}^{d+1}$, $\|x\|_{\tilde{q}} := \sqrt[q]{\sum_{i=2}^{d+1} x_i^q}$ if $q < \infty$ and $\|x\|_\infty := \max_{i \in \{2,\dots,d+1\}} |x_i|$ otherwise. Then, using the definition of X_I^H,

$$\|\mathbb{E}[\hat{\beta}_I^H]\|_q = \| \begin{bmatrix} \mathbb{E}[\hat{b}_I^H] \\ \mathbb{E}[\hat{\beta}_I^H] \end{bmatrix} \|_{\tilde{q}} = \|(X_I^{H^\top} X_I^H)^{-1} X_I^{H^\top} f_I^H\|_{\tilde{q}}$$

$$= \|(X_I^{H^\top} X_I^H)^{-1} X_I^{H^\top} (X_I^H \begin{bmatrix} f(c_H) - c_H^\top \nabla f(c_H) \\ 0 \\ \vdots \\ 0 \end{bmatrix} + X_I^H \begin{bmatrix} 0 \\ \nabla f(c_H) \end{bmatrix}$$

$$+ \begin{bmatrix} (x_{H_1} - c_H)^\top Hess(r_{H_1})(x_{H_1} - c_H) \\ (x_{H_2} - c_H)^\top Hess(r_{H_2})(x_{H_2} - c_H) \\ \vdots \end{bmatrix}) \|_{\tilde{q}}$$

$$= \| \begin{bmatrix} f(c_H) - c_H^\top \nabla f(c_H) \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \nabla f(c_H) \end{bmatrix} + \underbrace{(X_I^{H^\top} X_I^H)^{-1} X_I^{H^\top} (\begin{bmatrix} (x_{H_1} - c_H)^\top Hess(r_{H_1})(x_{H_1} - c_H) \\ (x_{H_2} - c_H)^\top Hess(r_{H_2})(x_{H_2} - c_H) \\ \vdots \end{bmatrix})}_{=: J(H)} \|_{\tilde{q}}$$

Plugging this expression into the theorem statement yields:

$$\left| \|\nabla f(x^*)\|_q - \max_{H \in \mathcal{H}_I} \{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\} \right| = \|\nabla f(x^*)\|_q - \max_{H \in \mathcal{H}_I} \{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\}$$

$$\leq \|\nabla f(x^*)\|_q - (\max_{H \in \mathcal{H}_I} \{\| \begin{bmatrix} 0 \\ \nabla f(c_H) \end{bmatrix} \|_{\tilde{q}}\} - \max_{H \in \mathcal{H}_I} \{\|J(H)\|_{\tilde{q}}\}) \leq \|\nabla f(x^*)\|_q - \max_{H \in \mathcal{H}_I} \{\|\nabla f(c_H)\|_q\}$$

$$+ \max_{H \in \mathcal{H}_I} \{\|J(H)\|_q\} = \max_{H \in \mathcal{H}_I} \{\|J(H)\|_q\}$$

where the last equality follows from the fact that there exists $H \in \mathcal{H}_I$ such that $x^* \in H$. As $f \in C^2(\mathcal{X}, K)$, we have $\forall i,j \in \{1,...,d\}$, $\forall x \in \mathcal{X}$ that $|\frac{\partial^2 f}{\partial x_i \partial x_j}(x)| < K$. This implies that $\|Hess(x)\|_1 \leq dK$ $\forall x \in \mathcal{X}$ and by matrix norm equivalence; $\|Hess(x)\|_2 \leq \sqrt{d}\|Hess(x)\|_1 \leq d\sqrt{d}K$, $\forall x \in \mathbb{R}^d$. Therefore, since matrix p-norms are sub-multiplicative;

$$\|J(H)\|_q \leq n_q \|J(H)\|_2 \leq n_q \|(X_I^{H^\top} X_I^H)^{-1} X_I^{H^\top}\|_2 \| \begin{bmatrix} (x_{H_1} - c_1)^\top Hess(r_{H_1})(x_{H_1} - c_H) \\ (x_{H_2} - c_2)^\top Hess(r_{H_2})(x_{H_2} - c_H) \\ \vdots \end{bmatrix} \|_2$$

where $n_q = d^{\max\{\frac{1}{q} - \frac{1}{2}, 0\}}$.

$$\|(X_I^{H^\top} X_I^H)^{-1} X_I^{H^\top}\|_2 = \|X_I^H ((X_I^{H^\top} X_I^H)^\top)^{-1}\|_2 = \sqrt{\lambda_{max}((X_I^{H^\top} X_I^H)^{-1})}$$

$$= \sqrt{\|(X_I^{H^\top} X_I^H)^{-1}\|_2} \leq \frac{4}{\delta_I^H \sqrt{\eta N_I^H}}$$

where λ_{max} denotes the maximum eigenvalue of $(X_I^{H^\top} X_I^H)^{-1}$. Combining these two upper bounds we can conclude:

$$\left| \|\nabla f(x^*)\|_q - \max_{H \in \mathcal{H}_I} \{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\} \right| \leq \max_{H \in \mathcal{H}_I} \frac{4\sqrt{d} d K n_q}{\sqrt{\eta}} \frac{{\Delta_I^H}^2}{\delta_I^H} = \frac{4\sqrt{d} d n_q K}{\sqrt{\eta}} a_I.$$

■

Lemma D.6 *If the Assumptions of Theorem 3.6 hold, then $\forall I \in \mathbb{N}$, the difference between the Lipschitz estimate generated by the LCLS method with noisy sampling $\hat{L}_I$ and the Lipschitz estimate generated by the LCLS method with noiseless sampling $\bar{L}_I$ can be upper bounded by;*

$$\mathbb{P}(|\bar{L}_I - \hat{L}_I| > \frac{\epsilon}{2}) \leq 1 - \prod_{H \in \mathcal{H}_I} (1 - \frac{2^6 \sigma^2 d^{\max\{\frac{2}{q}-1,0\}} \sqrt{d}}{\eta \epsilon^2} \frac{1}{N_I^H {\delta_I^H}^2}). \tag{4}$$

Proof Let $I \in \mathbb{N}$. $\forall H \in \mathcal{H}_I$ denote by $[b_I^H, \beta_I^H]$ the least squares coefficients computed using (X_I^H, f_I^H) (instead of $(X_I^H, \tilde{f}_I^H)$), i.e. the noiseless least squares coefficients. Then,

$$\mathbb{E}[[\hat{b}_I^H, \hat{\beta}_I^H]^\top] = \mathbb{E}[({X_I^H}^\top X_I^H)^{-1} {X_I^H}^\top f_I^H] = [b_I^H, \beta_I^H]^\top$$

Therefore, we can write (with $n_q = d^{\max\{\frac{1}{q}-\frac{1}{2},0\}}$)

$$\mathbb{P}(|\bar{L}_I - \hat{L}_I| > \frac{\epsilon}{2}) = \mathbb{P}(|\max_{H \in \mathcal{H}_I} \{\|[\beta_I^H]\|_q\} - \max_{H \in \mathcal{H}_I} \{\|\hat{\beta}_I^H\|_q\}| > \frac{\epsilon}{2})$$

$$= \mathbb{P}(|\max_{H \in \mathcal{H}_I} \{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\} - \max_{H \in \mathcal{H}_I} \{\|\hat{\beta}_I^H\|_q\}| > \frac{\epsilon}{2}) \leq \mathbb{P}(|\max_{H \in \mathcal{H}_I} \{\|\mathbb{E}[\hat{\beta}_I^H]\|_q - \|\hat{\beta}_I^H\|_q\}| > \frac{\epsilon}{2})$$

$$\leq \mathbb{P}(\max_{H \in \mathcal{H}_I} \{\|\mathbb{E}[\hat{\beta}_I^H] - \hat{\beta}_I^H\|_q\} > \frac{\epsilon}{2}) \leq \mathbb{P}(\max_{H \in \mathcal{H}_I} \{\|\mathbb{E}[\hat{\beta}_I^H] - \hat{\beta}_I^H\|_2\} > \frac{\epsilon}{2n_q})$$

$$= 1 - \mathbb{P}(\max_{H \in \mathcal{H}_I} \{\|\mathbb{E}[\hat{\beta}_I^H] - \hat{\beta}_I^H\|_2\} < \frac{\epsilon}{2n_q}) \leq 1 - \prod_{H \in \mathcal{H}_I} \mathbb{P}(\|\mathbb{E}[\hat{\beta}_I^H] - \hat{\beta}_I^H\|_2 < \frac{\epsilon}{2n_q})$$

$$= 1 - \prod_{H \in \mathcal{H}_I} (1 - \mathbb{P}(\|\mathbb{E}[\hat{\beta}_I^H] - \hat{\beta}_I^H\|_2 \geq \frac{\epsilon}{2n_q}))$$

In order to upper bound the term given in product: $\mathbb{P}(\|\mathbb{E}[\hat{\beta}_I^H] - \hat{\beta}_I^H\|_2 \geq \frac{\epsilon}{2n_q})$, we use the covariance matrix: $var([\hat{b}_I^H, \hat{\beta}_I^H]) = \sigma^2({X_I^H}^\top X_I^H)^{-1}$ which follows from the fact that the components of γ_I^H are assumed to be uncorrelated with mean 0 and variance σ^2. We also denote by $Tr(M)$ the trace of a matrix $M \in \mathbb{R}^{d \times d}$. Then, by applying an extension of Chebyshev's inequality to finite dimensional vectors (Ferentios (1982)) and Lemma D.3, we have

$$\mathbb{P}(\|\mathbb{E}[\hat{\beta}_I^H] - \hat{\beta}_I^H\|_2 \geq \frac{\epsilon}{2n_q}) \overset{\substack{Chebychev's \\ Inequality}}{\leq} \frac{4n_q^2 \sigma^2 Tr(({X_I^H}^\top X_I^H)^{-1})}{\epsilon^2} \leq \frac{4n_q^2 \sigma^2 \sqrt{d} \| \left({X_I^H}^\top X_I^H \right)^{-1} \|_2}{\epsilon^2}$$

$$\overset{Lemma\ D.3}{\leq} \frac{4n_q^2 \sigma^2 \sqrt{d}}{\epsilon^2} \frac{16}{\eta {\delta_I^H}^2 N_I^H} = \frac{2^6 n_q^2 \sigma^2 \sqrt{d}}{\eta \epsilon^2} \frac{1}{N_I^H {\delta_I^H}^2}.$$

Plugging this expression into the product given above concludes the proof.

$$\mathbb{P}(|\bar{L}_I - \hat{L}_I| > \frac{\epsilon}{2}) \leq 1 - \prod_{H \in \mathcal{H}_I} (1 - \frac{2^6 \sigma^2 d^{\max\{\frac{2}{q}-1,0\}} \sqrt{d}}{\eta \epsilon^2} \frac{1}{N_I^H {\delta_I^H}^2})$$

■

D.2 Proof of Main Theoretical Properties of LCLS

Proof of Theorem 3.6 (General Convergence Rate).
We recall that $\forall I \in \mathbb{N}$, the Lipschitz estimate $\hat{L}_I$ is obtained by considering the partition $\mathcal{H}_I$ and computing $\max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\}$. Let $\epsilon > 0$ be arbitrary. We need to show for $p = 1, 2$:

$$\lim_{I\to\infty} \mathbb{P}(|L_p^* - \hat{L}_I| > \epsilon) = 0$$

and for $p > 2$

$$\lim_{I\to\infty} \mathbb{P}(|L_p - \hat{L}_I| > \epsilon) = 0 \text{ with } L_p \in \mathbb{R}_{\geq L_p^*}.$$

Since f verifies assumption 2 and $\mathcal{X}$ is convex and compact, Lemma D.2 guarantees the existence of $x^* \in \mathcal{X}$ such that $\|\nabla f(x^*)\|_q = L_p^*$ for $p = 1, 2$ and $L_p := \|\nabla f(x^*)\|_q = \max_{x\in\mathcal{X}} \|\nabla f(x)\|_p \geq L_p^*$ for $p > 2$.

Therefore, for all $p \geq 1$, we can consider the statement;

$$\lim_{I\to\infty} \mathbb{P}(|\|\nabla f(x^*)\|_q - \hat{L}_I| > \epsilon) = 0$$

Let $I \in \mathbb{N}$ and consider $\mathbb{P}(|\|\nabla f(x^*)\|_q - \hat{L}_I| > \epsilon)$. This expression can be split into two terms:

$$\begin{aligned}
&\mathbb{P}(|\|\nabla f(x^*)\|_q - \hat{L}_I| > \epsilon) \\
&\leq \underbrace{\mathbb{P}(|\|\nabla f(x^*)\|_q - \max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\}| > \frac{\epsilon}{2})}_{(I)} + \underbrace{\mathbb{P}(|\max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\} - \hat{L}_I| > \frac{\epsilon}{2})}_{(II)}.
\end{aligned}$$

In the following, we show that both (I) and (II) converge to 0 when I goes to infinity.

(I): From Lemma D.5, $\left|\|\nabla f(x^*)\|_q - \max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\}\right| \leq \frac{4\sqrt{d}dn_qK}{\sqrt{\eta}} a_I$. Plugging this upper bound into the above expression, we have

$$\mathbb{P}(|\|\nabla f(x^*)\|_q - \max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\}| > \frac{\epsilon}{2}) \leq \mathbb{P}(\frac{4\sqrt{d}dn_qK}{\sqrt{\eta}} a_I > \frac{\epsilon}{2}).$$

By hypothesis 2. $\lim_{I\to\infty} a_I = 0$ and therefore there exists $I_1 \in \mathbb{N}$ sufficiently large such that $\frac{4\sqrt{d}dn_qK}{\sqrt{\eta}} a_{I_1} \leq \frac{\epsilon}{2}$ and therefore $\mathbb{P}(\frac{4\sqrt{d}dn_qK}{\sqrt{\eta}} a_{I_1} > \frac{\epsilon}{2}) = 0$.

(II): We show that $\mathbb{P}(|\max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\} - \hat{L}_I| > \frac{\epsilon}{2})$ converges to 0 as I goes to infinity. Let $\bar{L}$ denote the Lipschitz estimate generated by LCLS with noiseless samples. Then, applying Lemma D.6, we have the following upper bound on $\mathbb{P}(|\max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\} - \hat{L}_I| > \frac{\epsilon}{2})$:

$$\begin{aligned}
\mathbb{P}(|\max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\} - \hat{L}_I| > \frac{\epsilon}{2}) = \mathbb{P}(|\bar{L}_I - \hat{L}_I| > \frac{\epsilon}{2}) &\leq 1 - \prod_{H\in\mathcal{H}_I}(1 - \frac{16\sigma^2 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}}{\eta\epsilon^2}\frac{1}{N_I^H{\delta_I^H}^2}) \\
&\leq 1 - (1 - \frac{2^6\sigma^2 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}}{\eta\epsilon^2 \min_{H\in\mathcal{H}_I}(N_I^H{\delta_I^H}^2)})^{|\mathcal{H}_I|}.
\end{aligned}$$

As by Theorem hypothesis 2 $\lim_{I\to\infty}\max_{H\in\mathcal{H}_I}(\Delta_I^H) = 0$, applying Lemma D.4 implies that $\lim_{I\to\infty}|\mathcal{H}_I| = \infty$. Therefore using the fact that $\lim_{I\to\infty} b_I = 0$, we have $\lim_{I\to\infty}\max_{H\in\mathcal{H}_I}\left(\frac{1}{N_I^H{\delta_I^H}^2}\right) = \lim_{I\to\infty}\frac{1}{\min_{H\in\mathcal{H}_I} N_I^H{\delta_I^H}^2} = 0$.

To alleviate notation, let $(\alpha_I)_{I\in\mathbb{N}}$ be the sequence defined by $\alpha_I := \frac{2^6\sigma^2 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}}{\eta\epsilon^2 \min_{H\in\mathcal{H}_I} N_I^H{\delta_I^H}^2}$, then $\lim_{I\to\infty}\frac{1}{\min_{H\in\mathcal{H}_I} N_I^H{\delta_I^H}^2} = 0$ implies that $\exists\bar{I} \in \mathbb{N}$ such that $\forall I \geq \bar{I}$, $\alpha_I < 0.5$. Utilising fundamental logarithm inequalities, we obtain:

$$1 - (1-\alpha_I)^{|\mathcal{H}_I|} \leq |\mathcal{H}_I|\log(\frac{1}{1-\alpha_I}) \leq |\mathcal{H}_I|\frac{\alpha_I}{1-\alpha_I} \leq |\mathcal{H}_I|\frac{\alpha_I}{2}$$

$$= \frac{2^5\sigma^2 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}}{\eta\epsilon^2}\frac{|\mathcal{H}_I|}{\min_{H\in\mathcal{H}_I} N_I^H {\delta_I^H}^2} = \Big(\frac{2^5\sigma^2 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}}{\eta\epsilon^2}\Big) b_I \overset{I\to\infty}{\longrightarrow} 0.$$

■

Proof of Corollary 3.8 (Noiseless Oracle).
As in the proof of Theorem 3.6, we can consider the statement; $p \in \mathbb{N}$,

$$\lim_{I\to\infty} \mathbb{P}(|\|\nabla f(x^*)\|_q - \hat{L}_I| > \epsilon) = 0$$

where $x^* \in \text{argmax}_{x\in\mathcal{X}}\|\nabla f(x)\|_p$. Since the data samples contain no noise, $\hat{L}_I = \max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\}$ and

$$\mathbb{P}(|\|\nabla f(x^*)\|_q - \hat{L}_I| > \epsilon) = \mathbb{P}(|\|\nabla f(x^*)\|_q - \max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\}| > \epsilon).$$

Then, applying Lemma D.5 and using $\lim_{I\to\infty} a_I = 0$ as in the proof of Theorem 3.6 gives the desired convergence result.
(Note: that the least squares estimation is well defined as $N_I^H \geq d+1$ and Assumption 5 holds.)

■

Proof of Theorem 3.9 (Finite Sample Guarantee).
We show the equivalent statement; $\mathbb{P}(|L_p - \hat{L}_I| > \epsilon) \leq \delta$. where as in proof of Theorem 3.6, $L_p := \|\nabla f(x^*)\|_q$ with $x^* := \text{argmax}_{x\in\mathcal{X}}\|\nabla f(x)\|_p$ and $L_p = L_p^*$ for $p = 1, 2$, $L_p \geq L_p^*$ for $p > 2$.
In the hypercube set-up, we have $\forall I \in \mathbb{N}_{>1}$, $\forall H \in \mathcal{H}_I$, $\Delta_I^H = \frac{\sqrt{d}M}{I}$, $\delta_I^H = \frac{M}{I}$ and $|\mathcal{H}_I| = I^d$. Let $\epsilon > 0$, $\delta \in (0, \frac{1}{2}]$: From the proof of Theorem 3.6 we have that three following inequalities need to be satisfied in order for (1) to hold. From **(I)** we need $\frac{4\sqrt{d}dn_qK}{\sqrt{\eta}}\frac{{\Delta_I^H}^2}{\delta_I^H} \leq \frac{\epsilon}{2}$ in order for $\mathbb{P}(|\|\nabla f(x^*)\|_q - \max_{H\in\mathcal{H}_I}\{\|\mathbb{E}[\hat{\beta}_I^H]\|_q\}| > \frac{\epsilon}{2}) = 0$. This implies that;

$$I \geq \frac{8d^2\sqrt{d}n_q}{\sqrt{\eta}}\frac{MK}{\epsilon}.$$

From **(II)**, we have the following two inequalities that need to be satisfied;

$$(1) \qquad \alpha_I = \frac{2^6\sigma^2 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}}{\eta\epsilon^2 \min_{H\in\mathcal{H}_I} N_I^H {\delta_I^H}^2} < 0.5$$

$$(2) \qquad \frac{2^5\sigma^2 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}}{\eta\epsilon^2}\frac{|\mathcal{H}_I|}{\min_{H\in\mathcal{H}_I} N_I^H {\delta_I^H}^2} < \delta$$

The first implies that

$$\frac{2^7 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}\sigma^2}{\eta}\frac{I^2}{M^2\epsilon^2} < \min_{H\in\mathcal{H}_I} N_I^H$$

and the second expression gives

$$\frac{2^5 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}\sigma^2}{\eta}\frac{I^2}{M^2\epsilon^2}\frac{|\mathcal{H}_I|}{\delta} < \min_{H\in\mathcal{H}_I} N_I^H.$$

Since $|\mathcal{H}_I| = I^d$, $I \in \mathbb{N}_{>1}$ and $\delta \in (0, \frac{1}{2}]$, we have that if the $\min_{H\in\mathcal{H}_I} N_I^H$ satisfies (2) then (1) is true as well. Therefore, we have $\forall H \in \mathcal{H}_I$;

$$\frac{2^5 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}\sigma^2}{\eta}\frac{I^{d+2}}{\delta M^2\epsilon^2} < \min_{H\in\mathcal{H}_I} N_I^H.$$

Setting $C_1(d) = 8d^2\sqrt{d}d^{\max\{\frac{1}{q}-\frac{1}{2},0\}}$ and $C_2(d,q) = 2^5 d^{\max\{\frac{2}{q}-1,0\}}\sqrt{d}$ concludes the proof.

■

E Proofs: Sample Complexity of Adaptive Lipschitz Optimisation

In this section we prove the lower bound on the sample complexity of certified adaptive Lipschitz optimisation algorithms given in Section 4.

Proof of Proposition 4.1 (Sample Complexity of Adaptive Lipschitz Optimisation).

Fix $\epsilon > 0$, $L^* \geq 0$ and let A be a non-adaptive certified optimisation algorithm which takes a given Lipschitz constant $\bar{L} > L^*$ as a hyperparameter. Using the notation given in Section 4: with n-queries to the oracle, A outputs a triplet $((x_n, f(x_n^*), \zeta_n))_{n\in\mathbb{N}}$ where x_n is the n-th query point, $f(x_n^*)$ is the generated estimate of $\max_{x\in\mathcal{X}} f(x)$ after n queries and $\zeta_n \geq 0$ is an error certificate that guarantees: $\max_{x\in\mathcal{X}} f(x) - f(x_n^*) \leq \zeta_n$. From Theorem 3 of Bachoc et al. (2021) with $\epsilon_0 < 2^{d-1}ML^*$ (this follows from the fact that $\mathcal{X}$ is a hypercube), we have that for all $f \in \{h : \mathcal{X} \to | h \text{ is Lipschitz cont. and } L_p^*(h) < \bar{L}\}$:

$$N(A, f, \epsilon) \geq \frac{c_d L^{*d}(1 - \frac{L^*}{\bar{L}})^d}{1 + \lceil \log_2(\frac{\epsilon_0}{\epsilon}) \rceil} \int_{\mathcal{X}} \frac{dx}{(f(x^*) - f(x) + \epsilon)^d}. \tag{5}$$

where $c_d > 0$ (It is important to note that the term $c_d L^{*d}$ is not optimised in (Bachoc et al. (2021)) and could be improved in future work). Now, consider an adaptive Lipschitz optimisation algorithm $\tilde{A}$ with a separable Lipschitz constant estimator $L_{\tilde{A}}(f)$. If $\tilde{L}_{\tilde{A}}(f)$ can be guaranteed to be feasible (e.g. see discussion after Corollary 3.12) then equation (5) holds for $\tilde{A}$ and $\forall f \in C^2(\mathcal{X}, K) \cap \mathcal{F}_p(L^*)$ with $\bar{L}$ replaced by $\tilde{L}_{\tilde{A}}(f)$ [17]. The precision at which $\tilde{L}_{\tilde{A}}(f)$ estimates $L^*(f)$ therefore directly impacts the lower bound on $N(\tilde{A}, f, \epsilon)$. From the Corollary 2.3 given in Section 2, we have that $\forall n \in \mathbb{N}$, any Lipschitz learning algorithm $\tilde{L} \in \mathcal{L}_{n,p}$ that guarantees feasible Lipschitz constants must satisfy

$$\sup_{f\in C^2(\mathcal{X},K)\cap\mathcal{F}_p(L^*)} \tilde{L}(f) - L^* \geq C\frac{MK}{\sqrt[d]{n}}.$$

for some $C > 0$. This implies that for all $A \in \mathcal{A}$, there exists a non-empty set $\mathcal{G}_A \subset C^2(\mathcal{X}, K) \cap \mathcal{F}_p(L^*)$ such that $\forall f^* \in \mathcal{G}_A$, $\tilde{L}_A(f^*) - L^* \geq \frac{C}{2}\frac{MK}{\sqrt[d]{n}}$. Then, denoting $I(f) := \frac{c_d L^{*d}}{1+\lceil \log_2(\frac{\epsilon_0}{\epsilon}) \rceil} \int_{\mathcal{X}} \frac{dx}{(f(x^*)-f(x)+\epsilon)^d}$ in order to alleviate notation, we have $\forall A \in \mathcal{A}$,

$$\begin{aligned} N(A, \epsilon) &:= \sup_{f\in C^2(\mathcal{X},K)\cap\mathcal{F}_p(L^*)} N(A, f, \epsilon) \geq \sup_{f\in C^2(\mathcal{X},K)\cap\mathcal{F}_p(L^*)} \big\{(1 - \frac{L^*}{\tilde{L}_A(f)})^d I(f)\big\} \\ &\geq (1 - \frac{L^*}{L^* + \frac{C}{2}\frac{MK}{\sqrt[d]{N(A,\epsilon)}}})^d \sup_{f\in\mathcal{G}_A} \big\{I(f)\big\}. \end{aligned}$$

Re-arranging the terms in the above expression, we can obtain:

$$\frac{C}{2}MK \sup_{f\in\mathcal{G}_A} \big\{\sqrt[d]{I(f)}\big\} \leq L^*(\sqrt[d]{N(A,\epsilon)})^2 + \frac{C}{2}MK\sqrt[d]{N(A,\epsilon)}$$

which can be solved to give the lower bound

$$\sqrt[d]{N(A,\epsilon)} \geq C_1\frac{MK}{L^*}\Big(\sqrt{1 + C_1\frac{L^* \sup_{f\in\mathcal{G}_A}\big\{\sqrt[d]{I(f)}\big\}}{MK}} - 1\Big)$$

where $C_1 > 0$ is a constant. In order to finish the proof, a lower bound on $\sup_{f\in\mathcal{G}_A}\big\{\sqrt[d]{I(f)}\big\}$ is needed. To do so, we note that $I(f)$ is minimised when f is constant. We therefore consider the set of functions $\mathcal{F}_0$ defined in the proof of Theorem 2.2. From the proof of Theorem 2.2, we have that if $N(A,\epsilon) \leq (\frac{MK}{L^*})^d(\frac{C_2}{2})^d$, then $L^* \leq \frac{C_2}{2}\frac{MK}{\sqrt[d]{N(A,\epsilon)}}$ and $\mathcal{F}_0(\frac{(L^*)^2}{0.8K}, \frac{0.8}{7.75}(\frac{K}{L^*})^2) \subset \mathcal{G}_A$. Using $f(x^*) = \frac{(L^*)^2}{0.8K}$, $\forall f \in \mathcal{F}_0(\frac{(L^*)^2}{0.8K}, \frac{0.8}{7.75}(\frac{K}{L^*})^2)$, we obtain the lower bound:

$$\sup_{f\in\mathcal{G}_A}\big\{I(f)\big\} \geq \frac{c_d L^{*d}}{1 + \lceil \log_2(\frac{\epsilon_0}{\epsilon}) \rceil}\frac{\mathcal{V}_{\mathcal{X}}}{(\epsilon + \frac{(L^*)^2}{0.8K})^d}.$$

[17] Note: this is only possible as we are considering adaptive Lipschitz optimization algorithms which are separable.

Therefore, if $N(A,\epsilon) \leq (\frac{MK}{L^*})^d(\frac{C_2}{2})^d$, the above expression can be plugged into the lower bound on $\sqrt[d]{N(A,\epsilon)}$. We obtain

$$\sqrt[d]{N(A,\epsilon)} \geq C_1 \frac{MK}{L^*}\left(\sqrt{1 + C_3 \frac{1}{\sqrt[d]{(1+\lceil \log_2(\frac{\epsilon_0}{\epsilon})\rceil)(\frac{\epsilon K}{L^{*2}}+1)}}} - 1\right)$$

(for some constant $C_3 > 0$) which corresponds to the first half of the lower bound stated in the Proposition 4.1. In order to derive the second part of the expression, we consider the case where $N(A,\epsilon) > (\frac{MK}{L^*})^d(\frac{C_2}{2})^d$. In this case, an alternative lower bound on $\sup_{f\in\mathcal{G}_A}\{I(f)\}$ needs to be derived. In order to do so, we consider the following class of functions,

$$\{g : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R} | \forall x \in \mathcal{X}, g(x) = f(x) + (L^* - \frac{C_2}{2}\frac{MK}{\sqrt[d]{N(A,\epsilon)}})x_1 \text{ where } f \in \mathcal{F}_0(\frac{(L^*)^2}{0.8K}, \frac{0.8}{7.75}(\frac{K}{L^*})^2)\}$$

which belongs to $\mathcal{G}_A$ by construction. However, as obtaining a tight lower bound on $\sup_{f\in\mathcal{G}_A}\{\sqrt[d]{I(f)}\}$ is technically infeasible for this class, we simplify the problem by removing the functional input from $\mathcal{F}_0(\frac{(L^*)^2}{0.8K}, \frac{0.8}{7.75}(\frac{K}{L^*})^2)$ and considering the simple linear function $f^* : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$, $f^*(x) = L^* x_1$ which belongs trivially to $\mathcal{G}_A$. In this case, we can compute the lower bound

$$\sup_{f\in\mathcal{G}_A}\{I(f)\} \geq c_d \frac{L^{*d+1}M^{d-1}}{(1+\lceil \log_2(\frac{\epsilon_0}{\epsilon})\rceil)\epsilon^{d-1}}(d-1)\frac{(\frac{LM}{\epsilon}+1)^{d-1}-1}{(\frac{LM}{\epsilon}+1)^{d-1}} \geq \frac{c_d(d-1)}{2}\frac{L^{*d+1}M^{d-1}}{(1+\lceil log_2(\frac{\epsilon_0}{\epsilon})\rceil)\epsilon^{d-1}}$$

where the last inequality follows from the fact that $LM \geq \epsilon$ since $\epsilon \in (0, \epsilon_0)$. Plugging this expression into the lower bound on $\sqrt[d]{N(A,\epsilon)}$, we obtain

$$\sqrt[d]{N(A,\epsilon)} \geq C_1 \frac{MK}{L^*}\left(\sqrt{1 + C_4 \frac{L^{*2}}{\epsilon K}\sqrt[d]{\frac{L^*(d-1)\epsilon}{M(1+\lceil log_2(\frac{\epsilon_0}{\epsilon})\rceil)}}} - 1\right)$$

(for some constant $C_4 > 0$) which corresponds to the second half of the lower bounding expression. Note: in the statement of the proposition we simply set $C_2 = \min(C_3, C_4)$ as the used constant.

■

References

Daniel Limon, J Calliess, and Jan Marian Maciejowski. Learning-based nonlinear model predictive control. *IFAC-PapersOnLine*, 50(1):7769–7776, 2017.

M Canale, L Fagiano, and MC Signorile. Nonlinear model predictive control from data: a set membership approach. *International Journal of Robust and Nonlinear Control*, 24(1):123–139, 2014.

Mario Milanese and Carlo Novara. Set membership identification of nonlinear systems. *Automatica*, 40(6): 957–975, 2004.

Gleb Beliakov. Interpolation of lipschitz functions. *Journal of computational and applied mathematics*, 196 (1):20–44, 2006.

Jan-Peter Calliess, Stephen J Roberts, Carl Edward Rasmussen, and Jan Maciejowski. Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control. *Automatica*, 122:109216, 2020.

Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of optimization Theory and Applications*, 79(1):157–181, 1993.

Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch bayesian optimization via local penalization. In *Artificial intelligence and statistics*, pages 648–657. PMLR, 2016.

Cédric Malherbe and Nicolas Vayatis. Global optimization of lipschitz functions. In *International Conference on Machine Learning*, pages 2314–2323. PMLR, 2017.

Stefan Magureanu, Richard Combes, and Alexandre Proutiere. Lipschitz bandits: Regret lower bound and optimal algorithms. In *Conference on Learning Theory*, pages 975–999. PMLR, 2014.

Ankush Chakrabarty, Devesh K Jha, Gregery T Buzzard, Yebin Wang, and Kyriakos G Vamvoudakis. Safe approximate dynamic programming via kernelized lipschitz estimation. *IEEE transactions on neural networks and learning systems*, 32(1):405–419, 2020.

Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *arXiv preprint arXiv:1805.10965*, 2018.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George J Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *arXiv preprint arXiv:1906.04893*, 2019.

RG Strongin. On the convergence of an algorithm for finding a global extremum. *Eng. Cybernetics*, 11: 549–555, 1973.

GR Wood and BP Zhang. Estimation of the lipschitz constant of a function. *Journal of Global Optimization*, 8(1):91–103, 1996.

Yaroslav D Sergeyev. An information global optimization algorithm with local tuning. *SIAM Journal on Optimization*, 5(4):858–870, 1995.

Roman Strongin, Konstantin Barkalov, and Semen Bevzuk. Acceleration of global search by implementing dual estimates for lipschitz constant. In *International Conference on Numerical Computations: Theory and Algorithms*, pages 478–486. Springer, 2019.

Carlo Novara, Lorenzo Fagiano, and Mario Milanese. Direct feedback control design for nonlinear systems. *Automatica*, 49(4):849–860, 2013.

Gleb Beliakov. Monotonicity preserving approximation of multivariate scattered data. *BIT numerical mathematics*, 45(4):653–677, 2005.

Sébastien Bubeck, Gilles Stoltz, and Jia Yuan Yu. Lipschitz bandits without the lipschitz constant. In *International Conference on Algorithmic Learning Theory*, pages 144–158. Springer, 2011.

Jan-Peter Calliess. Lipschitz optimisation for lipschitz interpolation. In *2017 American Control Conference (ACC)*, pages 3141–3146. IEEE, 2017.

François Bachoc, Tom Cesari, and Sébastien Gerchinovitz. Instance-dependent bounds for zeroth-order lipschitz optimization with error certificates. *Advances in Neural Information Processing Systems*, 34: 24180–24192, 2021.

Mohammad Khajenejad, Zeyuan Jin, and Sze Zheng Yong. State and unknown terrain estimation for planetary rovers via interval observers. *Advanced Intelligent Systems*, page 2100040, 2021.

Bruno O Shubert. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388, 1972.

Regina Hunter Mladineo. An algorithm for finding the global maximum of a multimodal, multivariate function. *Mathematical Programming*, 34(2):188–200, 1986.

Dmitri E Kvasov and Yaroslav D Sergeyev. Lipschitz gradients for global optimization in a one-point-based partitioning scheme. *Journal of Computational and Applied Mathematics*, 236(16):4042–4054, 2012.

Danny D'Agostino. An efficient global optimization algorithm with adaptive estimates of the local lipschitz constants. *arXiv preprint arXiv:2211.04129*, 2022.

Ahsan S Alvi, Binxin Ru, Jan Calliess, Stephen J Roberts, and Michael A Osborne. Asynchronous batch bayesian optimisation with improved local penalisation. *arXiv preprint arXiv:1901.10452*, 2019.

Emilio T Maddalena and Colin N Jones. Nsm converges to a k-nn regressor under loose lipschitz estimates. *IEEE Control Systems Letters*, 4(4):880–885, 2020.

Napat Rujeerapaiboon, Daniel Kuhn, and Wolfram Wiesemann. Chebyshev inequalities for products of random variables. *Mathematics of Operations Research*, 43(3):887–918, 2018.

Yao-Liang Yu, Yuxi Li, Dale Schuurmans, and Csaba Szepesvári. A general projection property for distribution families. *Advances in Neural Information Processing Systems*, 22, 2009.

K Ferentios. On tcebycheff's type inequalities. *Trabajos de Estadistica y de Investigacion Operativa*, 33(1): 125, 1982.