STOCHASTIC SAMPLE APPROXIMATIONS OF (LOCAL) MODULI OF CONTINUITY

Anonymous authors
Paper under double-blind review

ABSTRACT

The modulus of local (Lipschitz) continuity of a neural network is a basic quantitative measure of its robustness. Exact mixed-integer methods and relaxation-based semidefinite upper bounds do not scale beyond small networks, while sampling gradient norms at random inputs is widely used as a lower bound, but has received little analysis. We revisit the connection between generalized derivatives and moduli of local continuity for networks definable in an o-minimal structure, showing that any dense-domain selection of the Clarke Jacobian recovers the modulus, and we analyze sampling estimators rigorously: they are almost surely lower bounds and, under an explicit exploration condition that uniform sampling satisfies and that our adaptive variant satisfies once its exploration bonus is bounded below, consistent, asymptotically unbiased, and exact after finitely many samples for ReLU networks; moreover, no adaptive black-box point-query scheme improves upon uniform sampling by more than a constant factor uniformly over all networks. Such lower bounds do not certify robustness; they measure the looseness of certified upper bounds and exhibit concrete sensitive inputs. We then present a non-uniform sampling variant based on upper-confidence-bound policies over an adaptively refined partition of the input domain, whose gains depend on the structure of the gradient field, and evaluate it against LipMIP, LipSDP, and uniform sampling, including on a small transformer to which the former two do not apply.

1 INTRODUCTION

There is a substantial interest in scalable verification of properties of neural networks. Some qualitative properties follow from inspecting the compute graph: a ReLU-based network (Householder, 1941) is not differentiable at the break-points of the activation, but is locally Lipschitz. Quantitative properties, such as the precise modulus of local Lipschitz continuity, are much harder to establish.

The modulus of local continuity (also known as the Lipschitz constant), a key quantitative property of a neural network, is important both for evaluating the robustness (Bastani et al., 2016; Huster et al., 2018; Yang et al., 2022; Zühlke & Kudenko, 2024) of neural networks and for evaluating properties (Mareček et al., 2023; Zhou et al., 2024; Nazarov et al., 2025) of their repeated uses in closed-loop models. Certifying such properties requires an *upper* bound on the modulus. This paper concerns *lower* bounds obtained by sampling. They measure how loose a certified upper bound is, which is how the literature on semidefinite relaxations uses them, they exhibit concrete sensitive inputs, and they remain available for architectures to which exact and relaxation-based methods do not apply. Virmaux & Scaman (2018) showed that estimating the modulus of local Lipschitz continuity is NP-Hard even for a two-layer ReLU-based network. First methods including "AutoLip" and "SeqLip" of Virmaux & Scaman (2018) propagated operator-norm surrogates along the computation graph. Recently, much of the research has focused on two classes of methods. "LipMIP" and its variants, e.g., (Tjeng et al., 2019; Anderson et al., 2020; Zhang et al., 2022; Schwan et al., 2023), utilized mixed-integer programming formulations of the neural network, either directly for ReLU-based networks, or using some piece-wise linear approximation (Gurobi Optimization, LLC). "LipSDP" and its variants, e.g., (Dvijotham et al., 2020; Chen et al., 2020; Latorre et al., 2020; Chen et al., 2021; Fazlyab et al., 2022; Chen, 2022; Lan et al., 2022; Fazlyab et al., 2022; Xue et al., 2022; Wang et al., 2024; Yang et al., 2025; Pauli et al., 2024; Xu & Sivaranjani, 2024; Syed & Hu, 2025), utilize relaxations in the form of semidefinite programming (SDP). State-of-the-art implementations (Lan

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2022; Wang et al., 2024; Yang et al., 2025) scale to shallow networks with hundreds of neurons per hidden layer, substantially less than in many practical applications.

Here, we revisit the connection between generalized derivatives and moduli of local continuity, which is well known at its simplest (Virmaux & Scaman, 2018; Goodfellow, 2018; Weng et al., 2018a; Sridhar et al., 2022): sampling Clarke subgradients at inputs drawn uniformly at random is the traditional lower-bound baseline for both LipSDP and LipMIP; e.g., Wang et al. (2024) sampled 500,000 points. Our contributions are: (i) for networks definable in an o-minimal structure, a proof that any dense-domain selection of the Clarke Jacobian recovers the modulus (Theorem 4.2), with an o-minimal analogue of Warga's theorem (Appendix A); (ii) a statistical analysis of sampling estimators (Appendix B): almost sure lower bounds; under an explicit exploration condition, met by uniform sampling and by our adaptive variant once its bonus is bounded below, consistency, asymptotic unbiasedness, and finite-time exactness for ReLU networks; and a proof that no adaptive black-box point-query scheme improves upon uniform sampling by more than a constant factor uniformly over all networks; (iii) a non-uniform sampling variant (Algorithm 3) based on upper-confidence-bound (UCB) policies over an adaptively refined partition, whose gains, in line with (ii), depend on the structure of the gradient field; (iv) experiments against LipMIP, LipSDP, and uniform sampling, including a transformer to which the former two do not apply.

2 Related Work

We recall the theoretical setup for estimating Lipschitz constants, roughly following the notation of Jordan & Dimakis (2020), and review how the estimation has been done in practice. In particular, we assume $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ are norms on $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively.

Definition 2.1. The **local** (α,β)**-Lipschitz constant** of $f : \mathbb{R}^n \to \mathbb{R}^m$ over $X \subseteq \mathbb{R}^n$ is $L^{(\alpha,\beta)}(f,X) := \sup_{x\neq y\in X} \|f(y)-f(x)\|_\beta/\|x-y\|_\alpha$; if it is finite, f is (α,β)**-Lipschitz** over X.

For C^1 functions on an open convex[1] set X, the well-known starting point is $L^{(\alpha,\beta)}(f,X) = \sup_{x\in X}\|J_f(x)\|_{\alpha,\beta}$, with $\|\cdot\|_{\alpha,\beta}$ the induced matrix norm on $\mathbb{R}^{m\times n}$ (Lemma A.13 extends this to the smooth locus of a definable function). Neural networks with ReLU activations are not smooth but are Lipschitz; where the Jacobian does not exist, one resorts to a set-valued *generalized derivative*, such as the *Clarke Jacobian* J_f^c (Definition A.6), and the formula generalizes as Theorem 1 of Jordan & Dimakis (2020), stated here with the convexity assumption that their proof uses:

Theorem 2.2. *Suppose $f:\mathbb{R}^n\to\mathbb{R}^m$ is (α,β)-Lipschitz over an open convex set X. Then:*

$$L^{(\alpha,\beta)}(f,X) \;=\; \sup_{x\in X, G\in J_f^c(x)} \|G\|_{\alpha,\beta}$$

With convexity, Theorem 2.2 follows for every Lipschitz f from the mean-value theorem for Clarke Jacobians (Clarke, 1990, Proposition 2.6.5) and Rademacher's theorem; Appendix A reproves it for definable f without measure theory.

The modulus of local continuity can be estimated in a number of ways. The most general methods use finite differences between pairs of nearby points. In neural networks, one may leverage (Weng et al., 2018b) frameworks such as PyTorch and TensorFlow to compute generalized derivatives, usually Clarke subgradients. More recent methods specific to neural networks leverage the computation graph of the neural network and either integer programming (Tjeng et al., 2019; Anderson et al., 2020; Zhang et al., 2022; Schwan et al., 2023) with linear relaxations, or polynomial optimization (Chen et al., 2020; 2021) with SDP relaxations (Fazlyab et al., 2022; Wang et al., 2024; Yang et al., 2025; Pauli et al., 2024; Xu & Sivaranjani, 2024; Syed & Hu, 2025). Xue et al. (2022); Pauli et al. (2024); Xu & Sivaranjani (2024) focus on the decomposition of the large SDP variable utilizing some notion of sparsity. There are a variety of other approaches as well (Jordan & Dimakis, 2020; Bonaert et al., 2021; Jafarpour et al., 2022; Avant & Morgansen, 2024), including interval methods and branch and bound without convex relaxations (Bunel et al., 2018; Karg & Lucia, 2020; Wei & Kolter, 2022; Abad Rocamora et al., 2022), but their empirical performance is broadly speaking similar. Finally, Sridhar et al. (2022) develop the ideas of Virmaux & Scaman (2018); Goodfellow (2018); Weng et al. (2018a) using ideas from extreme value theory.

[1]The paper (Jordan & Dimakis, 2020) omits the assumption that X is *convex*, although it is used in the proof of Theorem 1 of Jordan & Dimakis (2020) when considering "$f(x+t(y-x))$" for arbitrary $x,y\in X$.

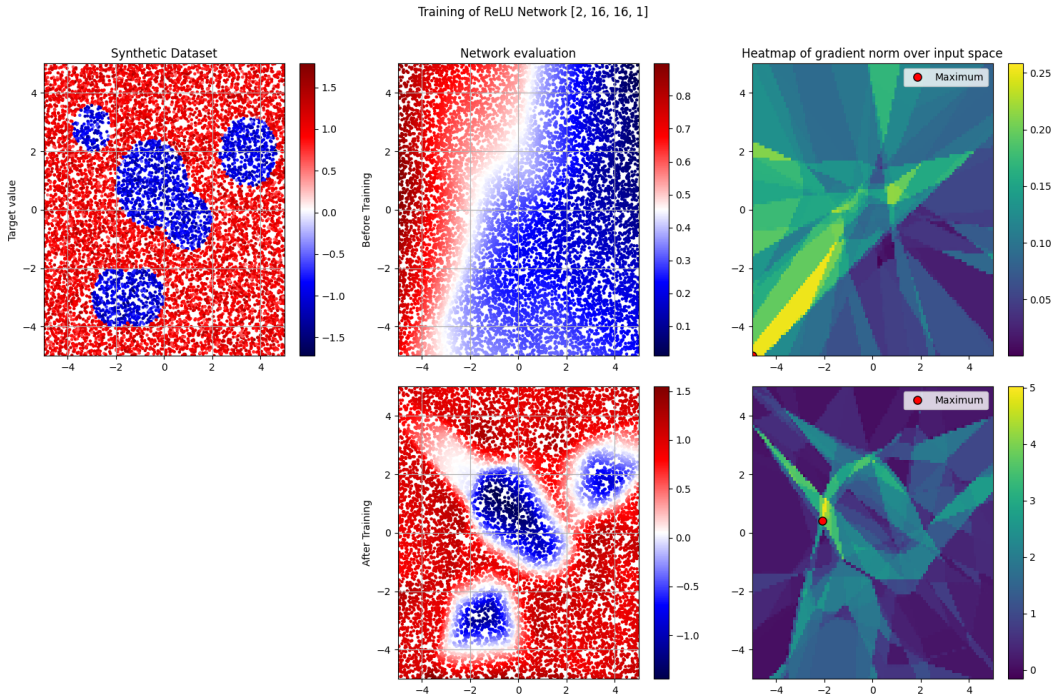


Figure 1: Running example, a ReLU network $[2, 16, 16, 1]$. Left: training data. Middle: the network before (top) and after (bottom) training. Right: norms of Clarke subgradients over the domain before (top) and after (bottom) training; the red dot marks the largest sampled value.

Algorithm 3 below belongs to the family of hierarchical optimistic optimization methods, which combine a tree of nested cells with an upper-confidence index: HOO (Bubeck et al., 2011a), DOO and SOO (Munos, 2011), StoSOO (Valko et al., 2013), extreme bandits (Carpentier & Valko, 2014), DIRECT (Jones et al., 1993; Gablonsky & Kelley, 2001), and the adaptive elimination of Aziz et al. (2018). Their guarantees assume local smoothness of the objective around its maximizers or a known near-optimality dimension, which the gradient-norm field of a ReLU network need not satisfy; we make no such assumption, and Theorem B.19 shows what can then be guaranteed.

More broadly, the estimation of robustness properties (Fazlyab et al., 2022) draws upon a long history of studying neural networks in control theory, e.g., (Psaltis et al., 1988; Parisini & Zoppoli, 1995), and of control-theoretic contributions to machine learning, e.g., (Bunel et al., 2018).

3 Our Approach

We first recall the standard stochastic approximation of the modulus of (local) Lipschitz continuity (Weng et al., 2018b) via the maximum norm of the (generalized) Jacobian $J_f(x)$ over sampled inputs, and then extend it to non-uniform, adaptive sampling.

Remark 3.1. Following common usage for non-smooth ReLU networks, $J_f(x)$ denotes the matrix returned by the differentiation oracle: ideally an element of the Clarke Jacobian for the modulus of local Lipschitz continuity, or of another generalized derivative for other moduli; at a kink, automatic differentiation may return a matrix outside it (Section 4).

3.1 Sampling Clarke Subdifferential

Neural networks built from standard layers are non-smooth, non-convex functions definable in an o-minimal structure (Ioffe, 2008), a class closed under composition that admits a chain rule for the Clarke generalized derivative (Clarke, 1975); see Section 4 and Appendix A. Frameworks such as PyTorch and TensorFlow differentiate the network automatically, operation by operation, so that an

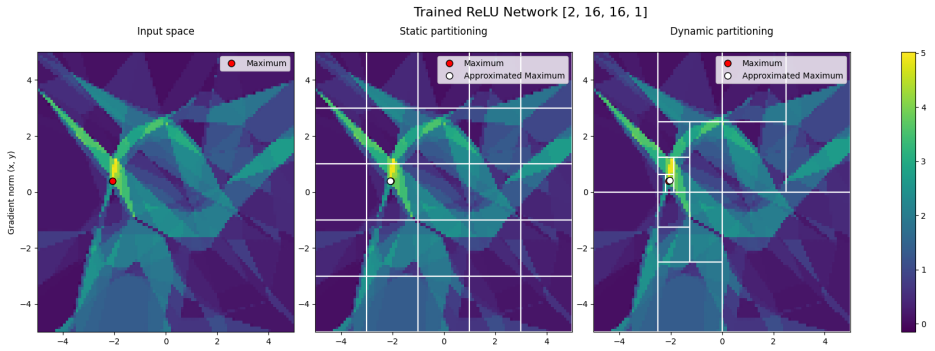


Figure 2: The running example continued. Left: Elements of Clarke subdifferential evaluated on the domain of the trained neural network. Middle: uniform partitioning of the domain. Right: non-uniform partitioning of the domain utilizing the upper-confidence bounds.

element of the Clarke subdifferential (or rather of a conservative set-valued gradient field, cf. Bolte & Pauwels (2021)) is obtained by one call to the autograd library. Taking the maximum over many inputs estimates the modulus of local Lipschitz continuity; see Algorithm 1.

Algorithm 1 Standard Stochastic Approximation, cf. (Weng et al., 2018b)

Input: Neural network $f(x)$ and its bounded open box domain D, number of samples N.
$r \leftarrow 0$
for $1 \leq i \leq N$ **do**
  Select input point $x \in D$ uniformly at random from D
  Evaluate $f(x)$; compute $J_f(x)$ by backpropagation, cf. Remark 3.1
  Set $r \leftarrow \max\{r, \|J_f(x)\|_1\}$
end for
return r

3.2 Generalizations

Algorithm 1 generalizes to other moduli of local continuity, by choosing the norms and the generalized derivative (Example 4.3), and to other sampling schemes. The simplest is stratification: Algorithm 2 (Appendix C) divides the box D into K^d equal sub-boxes and draws N/K^d uniform samples from each (Figure 2). It never increases the probability of a large error (Theorem B.16), but does not help much either (Figure 4).

3.3 Non-uniform Sampling with UCB

Non-uniform sampling, including (adaptive) importance sampling (Kahn, 1950; Robert et al., 1999; Tokdar & Kass, 2010; Bugallo et al., 2017), concentrates samples where the estimate can be improved the most, given what has been sampled so far, without neglecting other regions. In multi-armed bandits (Gittins et al., 2011; Lattimore & Szepesvári, 2020), upper-confidence bound policies (Lai, 1987; Auer et al., 2002) are the classical such scheme. We view the estimation of a modulus of local continuity as an infinity-armed bandit problem (Berry et al., 1997; Wang et al., 2008), an online estimate of the maximum of a distribution, but without the usual assumptions of Lipschitz continuity (Kleinberg et al., 2008; Bubeck et al., 2011b), Bernoulli rewards (Berry et al., 1997; Gong & Sellke, 2023), or a unique optimal arm (De Heide et al., 2021).

Algorithm 3 starts with the undivided box domain D as a single subregion. In iteration t, one sample is drawn uniformly from the subregion with the highest upper-confidence bound (UCB score, or index)

$$\begin{cases} s_{max} + c \cdot \max\{s_\sigma, \sigma_0\} \cdot \sqrt{\frac{\ln(t+1)}{s_n}}, & \text{if } s_n > n_0, \\ \infty, & \text{if } s_n \leq n_0, \end{cases} \tag{1}$$

where s_{max}, s_n, and s_σ are the maximum, the number, and the standard deviation of the values sampled from the subregion so far, $c > 0$ is an exploration constant, and $\sigma_0 \geq 0$ is a floor on the standard deviation. At the first iteration $\geq t_m^k$ after the previous subdivision ($k = 1, 2, \ldots$; roughly at iterations $t_m, t_m^2, t_m^3, \ldots$), before the sample is drawn, a subregion with the highest index is subdivided: its longest side is halved by an axis-aligned hyperplane, so that boxes stay boxes, and both halves start with $s_n = 0$. The threshold n_0 (we use $n_0 = 10$) provides initial samples for each new subregion. The floor $\sigma_0 > 0$ is needed for consistency (Proposition B.13, Remark B.14): without it, a subregion whose first $n_0 + 1$ samples fall into one affine piece of a ReLU network has $s_\sigma = 0$ and may never be revisited. The experiments of Section 5 use $\sigma_0 = 0$; their values remain valid lower bounds (Proposition B.7), but the consistency guarantee does not cover them. As for c, Appendix E shows that overestimating it by orders of magnitude costs little, as Algorithm 3 then degrades gracefully towards Algorithm 1, whereas underestimating it can delay finding the maximizing region; we recommend erring towards large c, e.g., the range of values seen in a short pilot run.

Algorithm 3 Generalized Stochastic Approximation with Non-uniform Partitioning of the Domain

Input: Norms $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$; neural network $f(x)$ and its box domain D; number of samples N; exploration constant c; floor σ_0 and threshold n_0 in (1); subdivision time multiplier t_m.
$r \leftarrow 0$; $t_{\text{subdivision}} \leftarrow t_m$; $S \leftarrow \{D\}$
for $1 \leq i \leq N$ **do**
  if $i \geq t_{\text{subdivision}}$ **then**
    Replace a subregion $s \in S$ maximizing the UCB score (1) by the two halves of s (halving its longest side), each with $s_n = 0$
    $t_{\text{subdivision}} \leftarrow t_{\text{subdivision}} \cdot t_m$
  end if
  Select a subregion $s \in S$ maximizing the UCB score (1)
  Select $x \in s$ uniformly at random; evaluate $f(x)$; compute $\|J_f(x)\|_{\alpha,\beta}$ for the chosen generalized Jacobian by backpropagation (cf. Example 4.3); update s_{max}, s_n, s_σ of s
  Set $r \leftarrow \max\{r, \|J_f(x)\|_{\alpha,\beta}\}$
end for
return r

4 An Analysis

From Theorem 2.2, Algorithms 1–3 always return a lower bound for the Lipschitz constant, provided that "$J_f(x)$" is contained in the Clarke Jacobian $J_f^c(x)$. This proviso is not automatic: at a kink, automatic differentiation can return a matrix outside $J_f^c(x)$ (for $f(x) = \mathrm{relu}(x) - \mathrm{relu}(-x) - x \equiv 0$ it returns -1 at 0 under the convention $\mathrm{relu}'(0) = 0$), but such points are null, so the lower bound holds almost surely under absolutely continuous sampling (Proposition B.7), not surely. In practice, the oracle returns a single matrix, so the question is which single-valued, possibly partially defined, replacements D of J_f^c still recover the constant. Definability covers not only ReLU networks, which are semialgebraic, but also softmax and layer normalization, which are definable in the real exponential field; the transformer of Appendix G is our one experiment outside the semialgebraic setting. Theorem 4.2 answers this for definable networks: any selection of J_f^c on a dense set suffices, e.g., the ordinary Jacobian on the smooth locus. For automatic differentiation, whose output can differ from J_f^c on a null set (Remark 4.5), the statistical analysis in Appendix B relies instead on almost-everywhere agreement (Assumption B.2), which follows from Bolte & Pauwels (2020; 2021); since the sampling laws are absolutely continuous, null sets are never hit.

It has been observed many times (Davis et al., 2020; Bolte & Pauwels, 2021; Bareilles et al., 2025) that nearly all neural networks used in practice are definable in some o-minimal structure. Under this assumption, we describe in Theorem 4.2 below to what extent the role of the Clarke Jacobian in Theorem 2.2 can be replaced with a suitable alternative (see Appendix A for proofs).

We assume throughout: $\mathfrak{R} = (\mathbb{R}; <, +, \cdot, \ldots)$ is an o-minimal expansion of the real field and "definable" means definable in $\mathfrak{R}$ with parameters (the reader may take $\mathfrak{R} = (\mathbb{R}; <, +, \cdot)$, for which definable means semialgebraic); $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ are norms on $\mathbb{R}^n$ and $\mathbb{R}^m$, not necessarily definable;

$X \subseteq \mathbb{R}^n$ is a nonempty definable open convex set; and $f : X \to \mathbb{R}^m$ is a definable (α, β)-Lipschitz function.

We introduce the following provisional terminology:

Definition 4.1. A set-valued map $D : X \rightrightarrows \mathbb{R}^{m\times n}$ is **good** for f if

$$D(x) \subseteq J_f^c(x) \quad (x \in X), \qquad S_D := \{x \in X : D(x) \neq \varnothing\} \text{ is dense in } X.$$

Neither D nor S_D is required to be definable or measurable.

Under these assumptions, we have the following variation of Theorem 2.2:

Theorem 4.2. *If D is good for f, then:*

$$L^{(\alpha,\beta)}(f, X) \;=\; \sup_{x\in X, G\in D(x)} \|G\|_{\alpha,\beta}.$$

Example 4.3. Besides $D = J_f^c$, Theorem 4.2 applies to: the partially defined single-valued map returning $J_f(x)$ at the points where f is Fréchet differentiable or, more generally, C^r-smooth for a given $r \geq 1$; any selection of J_f^c defined on a dense subset of X, in particular one defined almost everywhere; and, when $m = 1$, the Fréchet, limiting, or Clarke subdifferential, e.g., (Bolte et al., 2007).

Note that Theorem 4.2 can fail if we do not assume that f is definable:

Example 4.4. Let $f : \mathbb{R} \to \mathbb{R}$ be a Lipschitz function with Lipschitz constant 1 such that $J_f^c(x) = [-1, 1]$ for every $x \in \mathbb{R}$; such functions occur generically according to Borwein & Wang (1997); Ioffe (2008) although they cannot be definable (as this would violate the Smooth Monotonicity Theorem 7.2.5 from Van den Dries (1998)). Set $D(x) := \{0\}$ for every $x \in \mathbb{R}$; then D is good for f, however $\sup_{x\in\mathbb{R}, G\in D(x)} \|G\|_{\alpha,\beta} = 0 \neq 1$.

As our idealized model for "$J_f(x)$" gets closer to how automatic differentiation (AD) is actually implemented by PyTorch or TensorFlow, then Theorem 4.2 may no longer be true due to sporadic behavior on null sets:

Remark 4.5. Suppose D is either the output of AD as modeled by selection Jacobians (Bolte & Pauwels, 2020), or a conservative mapping for f (Bolte & Pauwels, 2021). Then $D = \{J_f(x)\}$ a.e. on X and so $L^{(\alpha,\beta)}(f, X) \leq \sup_{x\in X, G\in D(x)} \|G\|_{\alpha,\beta}$. However, equality can fail as Bolte & Pauwels (2020; 2021) provide examples where $f \equiv 0$, although $D(x)$ can contain nonzero vectors.

Appendix B establishes the statistical properties of Algorithms 1–3: an almost sure lower bound (Proposition B.7); consistency whenever every subregion keeps being sampled (Theorem B.10); the two results stated informally below; an exact error law for uniform and stratified sampling (Theorem B.16); and a minimax lower bound (Theorem B.19): no adaptive black-box point-query scheme, Algorithm 3 included, improves upon uniform sampling by more than a constant factor uniformly over ReLU networks, so any gain of Algorithm 3 must stem from structure of the gradient field, as in analyses of optimistic optimization under smoothness (Munos, 2011). No independence of the sampled values is assumed (they are dependent under the adaptive selection of Algorithm 3); consistency rests on a conditional Borel–Cantelli argument, not on the law of large numbers. Appendix B also exhibits two explicit failure modes: starvation of a subregion when $\sigma_0 = 0$ (Remark B.14), and four-neuron ReLU networks on which every black-box scheme needs a number of samples inversely proportional to the volume of the maximizing region (Theorem B.19).

Exploration (Proposition B.13, informal). Algorithms 1 and 2 sample every fixed open subset of D infinitely often. So does Algorithm 3, for every $c > 0$ and $t_m > 1$, provided $\sigma_0 > 0$: since subdivisions occur at geometrically spaced times, only $O(\log t)$ subregions exist after t samples, and a subregion that stopped being sampled would see its exploration bonus grow like $c\,\sigma_0\sqrt{\ln t}$ until it is selected again. Hence all three algorithms are consistent, $r_t \uparrow L$ almost surely.

Bias (Proposition B.15, informal). Under persistent exploration, $\mathbb{E}\, r_t \leq L$ for every t and $\mathbb{E}\, r_t \uparrow L$. The bias is strictly positive at every sample size when the set of maximizers of $\|J_f\|$ has measure zero. For piecewise-affine networks, such as ReLU networks, the modulus is attained on a whole open affine piece, so the estimate becomes exact after a finite random number T of samples, with bias at most $L\,\mathbb{P}(T > t)$; for uniform sampling, $\mathbb{P}(T > t) = (1 - p_0)^t$, where p_0 is the volume fraction of that piece, which can be tiny in high dimension.

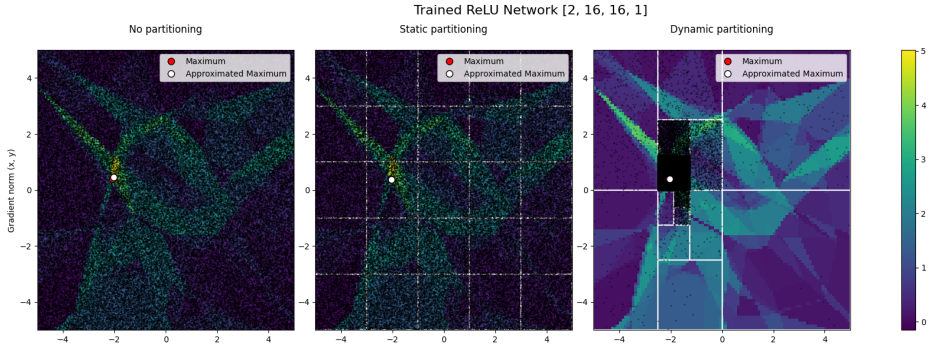


Figure 3: The running example continued. Left: Samples obtained using the standard stochastic approximation (Algorithm 1). Middle: samples obtained using uniform partitioning of the domain (Algorithm 2). Right: samples obtained using non-uniform partitioning of the domain utilizing the upper-confidence bounds (Algorithm 3).

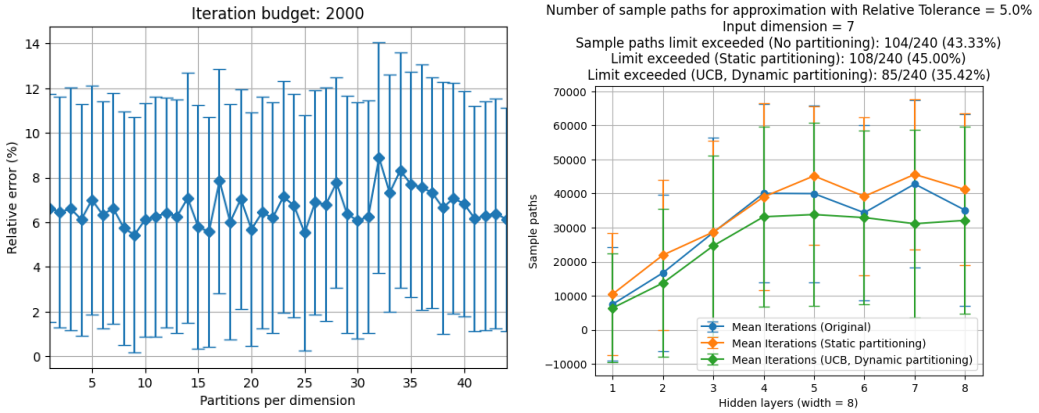


Figure 4: Left: relative error of Algorithm 2 versus the number of subregions (2000 samples per run, spread as evenly as possible over the cells). Right: mean $\pm$ standard deviation of the samples needed for 5% relative error, among runs succeeding within 60000 samples, versus depth, for 30 trained width-8 networks with 7-dimensional inputs per depth. Algorithms 1 and 2 behave alike; Algorithm 3 needs 10–25% fewer samples for depth at least 4 and exceeds the budget in 35% of runs, against 43% and 45%; the standard deviations overlap, so the gain is modest.

5 Experimental Results

As a proof of concept, we implemented our approach in PyTorch (torch.autograd), using ReLU-based neural nets (torch.nn.Linear/torch.nn.ReLU layers). The algorithms need only automatic differentiation and apply unchanged to convolutional, residual, and attention-based networks (Appendix G, on a four-dimensional slice of the input); the obstacle to scale is the input dimension (Theorem B.16), not the architecture. Throughout, we estimate the local $(\ell_\infty, |\cdot|)$-Lipschitz constant of the scalar $\langle c, f\rangle$ over an axis-aligned box D, i.e., $\sup_{x\in D} \|\nabla\langle c, f\rangle(x)\|_1$, as in LipMIP. The box is $[-5, 5]^2$ for the running example (Figures 1-3), $[-25, 25]^2$ for the left plot of Figure 4, $[0, 1]^2$ for Figure 5, $[-1, 1]^d$ for the synthetic nets of Table 3 (Appendix F), and the pixel range $[0, 1]^{784}$ for binary MNIST (Tables 1 and 3). We use $c = 1$ for scalar output nets, $c = (1, -1)$, the logit difference, for binary MNIST, and $c = (1, 0, -1)$ for the three-output nets of Figure 5. To describe the architecture, we use notation such as $[2, w_1, w_2, 1]$ to indicate we are using a standard multilayer perceptron with ReLU activation and two hidden layers of widths w_1, w_2. For the purposes of visualization, we often focus on the two-dimensional input. Synthetic datasets (Figure 1, left) assign values with mean -1.0 inside randomly placed hyperspheres and mean 1.0 elsewhere in the hypercube; training and evaluation details are in Appendix D.

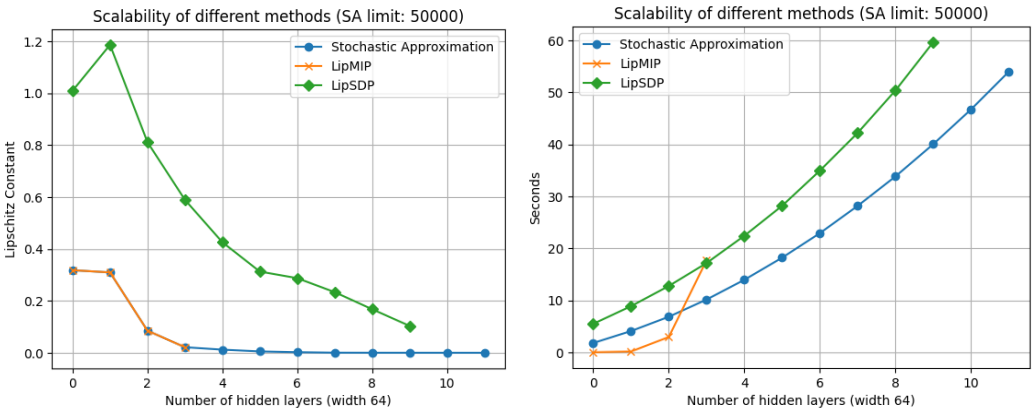


Figure 5: Comparison against LipMIP and LipSDP on untrained networks of width 64. Left: the modulus decreases sharply with depth. The sampled lower bound (50000 samples) coincides with the exact LipMIP value up to depth 3, beyond which LipMIP does not finish in 60 seconds; the certified LipSDP upper bound exceeds both by a wide margin, so the two bracket the unknown modulus for depth above 3. Right: within 60 seconds, Algorithm 3 with 50000 samples reaches depth 11 (704 neurons), LipSDP with Mosek depth 9 (576 neurons), and LipMIP with Gurobi depth 3 (192 neurons).

Table 1: Binary MNIST, following the benchmarking procedure of Jordan & Dimakis (2020); LipMIP is exact where it finishes, otherwise, timeout was set to 3 hours, and proved upper bounds (Gurobi's ObjBound) and found best solutions are reported separately. LipSDP bounds the ℓ_2 modulus; multiplied by $\sqrt{784}$ it is a certified upper bound here. Algorithms 1 and 3 ($c = 5$, partition step 2) use 500000 samples/run; values are mean $\pm$ standard deviation over 5 seeds and times are means. Sampled values are lower bounds, hence negative relative errors. n/a: no exact value.

Setup	Method	Value	Time (s)	Relative Error (%)
[784, 8, 8, 2]	LipMIP	262.48	3.6	0
	LipSDP	354.72	10.4	+35.14
	Algorithm 1	262.31 ± 0.00	32.8	-0.06
	Algorithm 3	262.31 ± 0.00	61.6	-0.06
[784, 20, 20, 2]	LipMIP	308.29	42.7	0
	LipSDP	419.20	20.8	+35.98
	Algorithm 1	295.80 ± 8.69	33.2	-4.05
	Algorithm 3	292.25 ± 7.10	69.9	-5.20
[784, 8, 8, 8, 2]	LipMIP	291.78	47.1	0
	LipSDP	393.71	12.0	+34.93
	Algorithm 1	197.74 ± 5.25	38.3	-32.23
	Algorithm 3	195.12 ± 0.00	68.2	-33.13
[784, 8, 8, 8, 8, 8, 2]	LipMIP, proved bound (timeout)	774.73	10800	n/a
	LipMIP, best found (timeout)	348.81	10800	n/a
	LipSDP	507.62	12.9	n/a
	Algorithm 1	164.47 ± 0.00	49.9	n/a
	Algorithm 3	164.47 ± 0.00	82.9	n/a
[784, 20, 20, 20, 20, 20, 2]	LipMIP, proved bound (timeout)	2683.00	10800	n/a
	LipMIP, best found (timeout)	390.37	10800	n/a
	LipSDP	609.93	18.1	n/a
	Algorithm 1	113.64 ± 13.34	49.6	n/a
	Algorithm 3	98.03 ± 8.21	82.5	n/a

Figures 1–3 show the running example [2, 16, 16, 1]; the red dot marking the maximum is within 0.001% of the value computed by LipMIP. Figure 4 summarizes 240 trained networks with a budget

Table 2: Algorithms 1 and 3 on the transformer (50 000 samples, 5 seeds). Samples to 99%: median over the runs that reached 99% of the best value found in all runs (number of such runs in brackets).

Method	Final value	Samples to 99%	Time (s)
Algorithm 1	0.5238 ± 0.0057	26 457 (4/5)	120.8
Algorithm 3	0.5273 ± 0.0002	3 906 (5/5)	122.2

of 60000 samples (right) and 100 runs per partition size on one network with a budget of 2000 samples (left); Figure 5 uses untrained networks.

In Table 1, we evaluate nets trained on binary MNIST (1 versus 7), following Jordan & Dimakis (2020); since LipMIP computes the modulus of $\langle c, f \rangle$, Algorithms 1 and 3 were applied to $\langle c, f \rangle$ as well, each run 5 times with different seeds. For the last two nets, LipMIP did not finish within 3 hours; the values it reports are the solver's best bound at the time limit and the best solution it found at the time limit, and since no exact value is known, relative errors are n/a. Three observations. First, where LipMIP finishes, the sampled values are within 0.1% of the exact modulus on the smallest net, 4–5% below it on the width-20 net, and 32–33% below it on the depth-3 width-8 net, consistent with sampled lower bounds deteriorating as the near-maximal set shrinks (Theorem B.16). Second, Algorithm 3 does not improve upon uniform sampling here: it returns the same value on two nets and a lower mean on three, at 1.7 to 2.1 times the run time, the overhead being the bookkeeping of the partition; zero standard deviations indicate that all five runs located the same affine piece (Proposition B.15). Third, on the two deepest nets LipSDP is a much tighter upper bound than the LipMIP bound at the time limit, while the best solution found by LipMIP is 2.1 and 3.4 times the sampled values; the modulus thus lies in $[348.81, 507.62]$ and $[390.37, 609.93]$, and 500000 samples do not contribute to these brackets. Where LipMIP finishes, LipSDP exceeds the exact modulus by 35–36%. A lower bound is not a certificate, and we do not claim that sampling improves upon LipSDP; the two are complementary.

Table 2 reports the transformer of Appendix G, a character-level GPT-style model with softmax attention and layer normalization, to which LipMIP and LipSDP do not apply; the domain perturbs four coordinates of the last token's embedding, and the hyperparameters of Algorithm 3 were selected on a separate set of seeds. Here Algorithm 3 reaches 99% of the best value found after a median of 3 906 samples, against 26 457 for Algorithm 1, which reaches this level in only 4 of 5 runs, and its final value is higher with a much smaller spread across seeds. Its overhead is only 1.1% of the wall-clock time, as the backward pass through the network dominates the cost of a sample. This is the clearest gain of adaptive sampling in our experiments; it is obtained on a low-dimensional slice of the input, in line with Theorem B.16, and on MNIST in 784 dimensions no such gain appears (Table 1; Appendix F for further comparisons).

6 Conclusions

We have elaborated upon the connections between generalized derivatives and Lipschitz constants in the setting of *tame optimization* (Ioffe, 2008; Bareilles et al., 2025), analyzed rigorously the sampling estimators that serve as the standard lower-bound baseline in the literature, and presented a non-uniform variant, Algorithm 3. Its advantage over uniform sampling cannot be uniform over networks (Theorem B.19); it appears when the domain has both areas of vanishing (sub)gradients (Hochreiter, 1998) and areas of high variance of the subgradients, common (Hanin, 2018) in deeper and wider nets, and it is modest in our ReLU experiments (Table 1 and Appendix F) and clearest on the transformer (Table 2).

Sampling-based estimators, and the theory developed here, apply to transformers, whose softmax attention and layer normalization are definable in an o-minimal structure, whereas LipMIP and LipSDP do not apply to transformers. Since transformers underlie much of the recent work on deep learning, this alone brings the estimation of moduli of continuity closer to practice. Our transformer is small, and its domain is a low-dimensional slice of the input; scaling to modern-day networks remains a challenge for any method whatsoever, and given that computing the modulus is NP-hard even for two-layer ReLU networks (Virmaux & Scaman, 2018), it is likely to remain one.

AI USE STATEMENT

In this work, we used generative AI tools (large language models) to assist in writing proofs, designing experiments, producing and editing notebooks and experimental code, formatting, and suggesting revisions to the text. The authors checked all mathematical statements and proofs line by line, reviewed and tested the AI-generated and AI-edited notebooks and code, ran the experiments, and checked the reported results. The authors take full responsibility for the contents of this submission.

REFERENCES

Elias Abad Rocamora, Mehmet Fatih Sahin, Fanghui Liu, Grigorios Chrysos, and Volkan Cevher. Sound and complete verification of polynomial networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 3517–3529. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/1700ad4e6252e8f2955909f96367b34d-Paper-Conference.pdf.

Ross Anderson, Joey Huchette, Will Ma, Christian Tjandraatmadja, and Juan Pablo Vielma. Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, 183(1):3–39, 2020.

Matthias Aschenbrenner, Lou van den Dries, and Joris van der Hoeven. *Asymptotic differential algebra and model theory of transseries*, volume 195 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, 2017. ISBN 978-0-691-17543-0. doi: 10.1515/9781400885411. URL https://doi.org/10.1515/9781400885411.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002. doi: 10.1023/A:1013689704352. URL https://doi.org/10.1023/A:1013689704352.

Trevor Avant and Kristi A. Morgansen. Analytical bounds on the local lipschitz constants of relu networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):13902–13913, 2024. doi: 10.1109/TNNLS.2023.3273228.

Maryam Aziz, Jesse Anderton, Emilie Kaufmann, and Javed Aslam. Pure exploration in infinitely-armed bandit models with fixed-confidence. In *Algorithmic Learning Theory*, pp. 3–24. PMLR, 2018.

Gilles Bareilles, Allen Gehret, Johannes Aspman, Jana Lepšová, and Jakub Mareček. Deep learning as the disciplined construction of tame objects, 2025. URL https://arxiv.org/abs/2509.18025.

Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. Measuring neural net robustness with constraints. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/980ecd059122ce2e50136bda65c25e07-Paper.pdf.

Donald A Berry, Robert W Chen, Alan Zame, David C Heath, and Larry A Shepp. Bandit problems with infinitely many arms. *The Annals of Statistics*, 25(5):2103–2116, 1997.

Neer Bhardwaj and Lou van den Dries. On the Pila–Wilkie theorem. *Expositiones Mathematicae*, 40(3):495–542, 2022. doi: 10.1016/j.exmath.2022.03.001. URL https://arxiv.org/abs/2010.14046v3. Appendix A; cited numbering as in arXiv:2010.14046v3.

Jérôme Bolte and Edouard Pauwels. A mathematical model for automatic differentiation in machine learning. *Advances in Neural Information Processing Systems*, 33:10809–10819, 2020.

Jérôme Bolte and Edouard Pauwels. Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. *Mathematical Programming*, 188(1):19–51, 2021.

Jérôme Bolte, Aris Daniilidis, Adrian Lewis, and Masahiro Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18(2):556–572, 2007.

Gregory Bonaert, Dimitar I. Dimitrov, Maximilian Baader, and Martin Vechev. Fast and precise certification of transformers. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, PLDI 2021, pp. 466–481, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383912. doi: 10.1145/3453483.3454056. URL https://doi.org/10.1145/3453483.3454056.

J Borwein and Xianfu Wang. Distinct differentiable functions may share the same clarke subdifferential at all points. *Proceedings of the American Mathematical Society*, 125(3):807–813, 1997.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. $\mathcal{X}$-Armed Bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011a.

Sébastien Bubeck, Gilles Stoltz, and Jia Yuan Yu. Lipschitz bandits without the lipschitz constant. In Jyrki Kivinen, Csaba Szepesvári, Esko Ukkonen, and Thomas Zeugmann (eds.), *Algorithmic Learning Theory*, pp. 144–158, Berlin, Heidelberg, 2011b. Springer Berlin Heidelberg. ISBN 978-3-642-24412-4.

Monica F Bugallo, Victor Elvira, Luca Martino, David Luengo, Joaquin Miguez, and Petar M Djuric. Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

Rudy R Bunel, Ilker Turkaslan, Philip Torr, Pushmeet Kohli, and Pawan K Mudigonda. A unified view of piecewise linear neural network verification. *Advances in Neural Information Processing Systems*, 31, 2018.

Alexandra Carpentier and Michal Valko. Extreme bandits. In *Advances in Neural Information Processing Systems*, volume 27, pp. 1089–1097. Curran Associates, Inc., 2014.

Tong Chen. *Robustness verification of neural networks using polynomial optimization*. PhD thesis, Université Paul Sabatier-Toulouse III, 2022.

Tong Chen, Jean B Lasserre, Victor Magron, and Edouard Pauwels. Semialgebraic optimization for lipschitz constants of relu networks. *Advances in Neural Information Processing Systems*, 33:19189–19200, 2020.

Tong Chen, Jean B Lasserre, Victor Magron, and Edouard Pauwels. Semialgebraic representation of monotone deep equilibrium models and applications to certification. *Advances in Neural Information Processing Systems*, 34:27146–27159, 2021.

F. H. Clarke. *Optimization and nonsmooth analysis*, volume 5 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 1990. ISBN 0-89871-256-4. doi: 10.1137/1.9781611971309. URL https://doi.org/10.1137/1.9781611971309.

Frank H Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D Lee. Stochastic subgradient method converges on tame functions. *Foundations of computational mathematics*, 20(1):119–154, 2020.

Rianne De Heide, James Cheshire, Pierre Ménard, and Alexandra Carpentier. Bandits with many optimal arms. *Advances in Neural Information Processing Systems*, 34:22457–22469, 2021.

Krishnamurthy (Dj) Dvijotham, Robert Stanforth, Sven Gowal, Chongli Qin, Soham De, and Pushmeet Kohli. Efficient neural network verification with exactness characterization. In Ryan P. Adams and Vibhav Gogate (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 497–507. PMLR, 22–25 Jul 2020. URL https://proceedings.mlr.press/v115/dvijotham20a.html.

Abbas Edalat. Smooth approximation of Lipschitz maps and their subgradients. *Journal of the ACM*, 69(1): 8:1–8:32, December 2021. doi: 10.1145/3481805. URL https://doi.org/10.1145/3481805.

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

Mahyar Fazlyab, Manfred Morari, and George J. Pappas. Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. *IEEE Transactions on Automatic Control*, 67(1):1–15, 2022. doi: 10.1109/TAC.2020.3046193.

Jorg Maximilian Xaver Gablonsky and Carl Timothy Kelley. *Modifications of the direct algorithm*. PhD thesis, North Carolina State University, Raleigh, North Carolina, 2001.

John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

Evelyn Xiao-Yue Gong and Mark Sellke. Asymptotically optimal quantile pure exploration for infinite-armed bandits. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 18551–18581. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3b3a83a5d86e1d424daefed43d998079-Paper-Conference.pdf.

Ian Goodfellow. Gradient masking causes clever to overestimate adversarial perturbation size, 2018. URL https://arxiv.org/abs/1804.07870.

Gurobi Optimization, LLC. Gurobi machine learning. https://gurobi-machinelearning.readthedocs.io/en/stable/userguide.html.

Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? *Advances in neural information processing systems*, 31, 2018.

Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

Alston S. Householder. A theory of steady-state activity in nerve-fiber networks: I. definitions and preliminary lemmas. *The bulletin of mathematical biophysics*, 3(2):63–69, 1941. doi: 10.1007/BF02478220. URL https://doi.org/10.1007/BF02478220.

Todd Huster, Cho-Yu Jason Chiang, and Ritu Chadha. Limitations of the lipschitz constant as a defense against adversarial examples. In *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings*, pp. 16–29, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-13452-5. doi: 10.1007/978-3-030-13453-2_2. URL https://doi.org/10.1007/978-3-030-13453-2_2.

A. D. Ioffe. An invitation to tame optimization. *SIAM J. Optim.*, 19(4):1894–1917, 2008. ISSN 1052-6234,1095-7189. doi: 10.1137/080722059. URL https://doi.org/10.1137/080722059.

Saber Jafarpour, Matthew Abate, Alexander Davydov, Francesco Bullo, and Samuel Coogan. Robustness certificates for implicit neural networks: A mixed monotone contractive approach. In *Learning for Dynamics and Control Conference*, pp. 917–930. PMLR, 2022.

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

Matt Jordan and Alexandros G Dimakis. Exactly computing the local lipschitz constant of relu networks. *Advances in Neural Information Processing Systems*, 33:7344–7353, 2020.

Herman Kahn. Random sampling (monte carlo) techniques in neutron attenuation problems–ii. *Nucleonics*, 6 (6):60–65, 1950.

Benjamin Karg and Sergio Lucia. Stability and feasibility of neural network-based controllers via output range analysis. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 4947–4954, 2020. doi: 10.1109/CDC42340.2020.9303895.

Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. Blog post, http://karpathy.github.io/2015/05/21/rnn-effectiveness/, 2015. Introduces the Tiny Shakespeare dataset.

Andrej Karpathy. nanoGPT. https://github.com/karpathy/nanoGPT, 2022. GitHub repository.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pp. 681–690, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605580470. doi: 10.1145/1374376.1374475. URL https://doi.org/10.1145/1374376.1374475.

Tze Leung Lai. Adaptive treatment allocation and the multi-armed bandit problem. *The annals of statistics*, pp. 1091–1114, 1987.

Jianglin Lan, Yang Zheng, and Alessio Lomuscio. Tight neural network verification via semidefinite relaxations and linear reformulations. In *AAAI*, pp. 7272–7280, 2022. URL https://doi.org/10.1609/aaai.v36i7.20689.

Fabian Latorre, Paul Rolland, and Volkan Cevher. Lipschitz constant estimation of neural networks via sparse polynomial optimization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJe4_xSFDB.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Jakub Mareček, Michal Roubalik, Ramen Ghosh, Robert N Shorten, and Fabian R Wirth. Predictability and fairness in load aggregation and operations of virtual power plants. *Automatica*, 147:110743, 2023.

Rémi Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/7e889fb76e0e07c11733550f2a6c7a5a-Paper.pdf.

Rodion Nazarov, Anthony Quinn, Robert Shorten, and Jakub Marecek. humancompatible.interconnect: Testing properties of repeated uses of interconnections of ai systems, 2025. URL https://arxiv.org/abs/2507.09626.

Thomas Parisini and Riccardo Zoppoli. A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31(10):1443–1451, 1995.

Patricia Pauli, Dennis Gramlich, and Frank Allgöwer. Lipschitz constant estimation for general neural network architectures using control tools. *arXiv preprint arXiv:2405.01125*, 2024.

D. Psaltis, A. Sideris, and A.A. Yamamura. A multilayered neural network controller. *IEEE Control Systems Magazine*, 8(2):17–21, 1988. doi: 10.1109/37.1868.

Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

Roland Schwan, Colin N Jones, and Daniel Kuhn. Stability verification of neural network controllers using mixed-integer programming. *IEEE Transactions on Automatic Control*, 68(12):7514–7529, 2023.

Kaustubh Sridhar, Oleg Sokolsky, Insup Lee, and James Weimer. Improving neural network robustness via persistency of excitation. In *2022 American Control Conference (ACC)*, pp. 1521–1526, 2022. doi: 10.23919/ACC53348.2022.9867880.

Usman Syed and Bin Hu. Improved scalable lipschitz bounds for deep neural networks, 2025. URL https://arxiv.org/abs/2503.14297.

Vincent Tjeng, Kai Y. Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyGIdiRqtm.

Surya T Tokdar and Robert E Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.

Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic simultaneous optimistic optimization. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 19–27. PMLR, 2013.

Lou Van den Dries. *Tame topology and o-minimal structures*, volume 248. Cambridge university press, 1998.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

Yizao Wang, Jean-Yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. *Advances in Neural Information Processing Systems*, 21, 2008.

Zi Wang, Bin Hu, Aaron J Havens, Alexandre Araujo, Yang Zheng, Yudong Chen, and Somesh Jha. On the scalability and memory efficiency of semidefinite programs for lipschitz constant estimation of neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

J. Warga. Fat homeomorphisms and unbounded derivate containers. *J. Math. Anal. Appl.*, 81(2):545–560, 1981. ISSN 0022-247X. doi: 10.1016/0022-247X(81)90081-0. URL https://doi.org/10.1016/0022-247X(81)90081-0.

Colin Wei and J Zico Kolter. Certified robustness for deep equilibrium models via interval bound propagation. In *International Conference on Learning Representations*, 2022.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Aurelie Lozano, Cho-Jui Hsieh, and Luca Daniel. On extensions of clever: A neural network robustness evaluation algorithm. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1159–1163. IEEE, 2018a.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations*, 2018b. URL https://openreview.net/forum?id=BkUHlMZ0b.

David Williams. *Probability with Martingales*. Cambridge University Press, Cambridge, 1991. doi: 10.1017/CBO9780511813658.

Yuezhu Xu and S Sivaranjani. ECLipse: Efficient compositional lipschitz constant estimation for deep neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=61YYSy078Z.

Anton Xue, Lars Lindemann, Alexander Robey, Hamed Hassani, George J. Pappas, and Rajeev Alur. Chordal sparsity for lipschitz constant estimation of deep neural networks. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 3389–3396, 2022. doi: 10.1109/CDC51059.2022.9993136.

Jianting Yang, Srećko Đurašinović, Jean-Bernard Lasserre, Victor Magron, and Jun Zhao. Verifying properties of binary neural networks using sparse polynomial optimization. *The International Conference on Learning Representations, arXiv preprint arXiv:2405.17049*, 2025.

Zonghan Yang, Tianyu Pang, and Yang Liu. A closer look at the adversarial robustness of deep equilibrium models. *Advances in Neural Information Processing Systems*, 35:10448–10461, 2022.

Huan Zhang, Shiqi Wang, Kaidi Xu, Linyi Li, Bo Li, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. General cutting planes for bound-propagation-based neural network verification. *Advances in neural information processing systems*, 35:1656–1670, 2022.

Quan Zhou, Ramen Ghosh, Robert Shorten, and Jakub Mareček. Closed-loop view of the regulation of ai: Equal impact across repeated interactions. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, pp. 176–181, 2024. doi: 10.1109/ICDEW61823.2024.00029.

Monty-Maximilian Zühlke and Daniel Kudenko. Adversarial robustness of neural networks from the perspective of lipschitz calculus: A survey. *ACM Computing Surveys*, 2024.

A Proof of Theorem 4.2 and tame Jacobian invariance

We first prove Theorem 4.2 with the dense-domain definition of a good map. The proof identifies the Lipschitz constant with the supremum of the ordinary Jacobian norms on the open C^1 locus, where every nonempty submap of the Clarke Jacobian must agree with the ordinary Jacobian. We then prove a separate, stronger pointwise result: for a definable locally Lipschitz map, deleting any set with dense complement leaves the Jacobian cluster sets unchanged, even before taking convex hulls.

The background needed for these two arguments is as follows.

1. *Smooth cell decomposition*: a definable map on an open set is C^1 on a definable open dense subset whose complement has dimension less than that of the domain (Lemma A.5).
2. *O-minimal dimension theory*: zero-dimensional definable sets are finite, full-dimensional definable sets have interior, and the fiber-dimension theorem controls exceptional directions (Lemma A.11).
3. *Elementary finite-dimensional analysis*: continuity of norms, compactness of bounded closed sets, norming linear functionals, the chain rule, and the scalar mean-value theorem.

The pointwise invariance argument additionally uses one-dimensional o-minimality, definable choice, and piecewise continuity, through the Rectangle Lemma A.20. For the o-minimal background, see Bhardwaj & van den Dries (2022, Appendix A, Theorems A.2 and A.14 and Propositions A.5, A.6, and A.10); our general references are Van den Dries (1998) and Appendix B of Aschenbrenner et al. (2017). Neither proof uses Rademacher's theorem, Warga's theorem, measure theory, or (Lebesgue) integration. The classical Warga theorem is recorded only for comparison in Section A.6.

Throughout the appendix, $n, m \geq 1$, and definability is with parameters in a fixed o-minimal expansion $\mathfrak{R}$ of the real field. Unless stated otherwise, $X \subseteq \mathbb{R}^n$ is nonempty and open, and $f : X \to \mathbb{R}^m$. Definability, continuity, local Lipschitzness, and convexity are imposed explicitly in the results where they are needed. In particular, convexity is needed for the global Lipschitz-constant formula, not for the pointwise invariance theorem. The reader may take "definable" to mean "semialgebraic" throughout.

A set-valued map $D : X \rightrightarrows \mathbb{R}^{m\times n}$ assigns to each $x \in X$ a possibly empty set of matrices. Its graph is $\operatorname{gph} D := \{(x, A) : x \in X,\ A \in D(x)\}$; definability of such a map means definability of its graph. All matrices act from $\mathbb{R}^n$ to $\mathbb{R}^m$. We write $[a, b] := \{(1 - t)a + tb : 0 \leq t \leq 1\}$, and, for distinct a, b, write (a, b) for the corresponding open segment.

A.1 Norms and Lipschitz constants

Fix arbitrary norms $\|\cdot\|_\alpha$ on $\mathbb{R}^n$ and $\|\cdot\|_\beta$ on $\mathbb{R}^m$. Neither norm is required to be definable. Recall that a norm is positive definite, absolutely homogeneous, and satisfies the triangle inequality. All norms in finite dimensions induce the usual Euclidean topology. The induced matrix norm is

$$\|A\|_{\alpha,\beta} := \sup_{\|v\|_\alpha \leq 1} \|Av\|_\beta \qquad (A \in \mathbb{R}^{m\times n}).$$

In particular,

$$\|Av\|_\beta \leq \|A\|_{\alpha,\beta}\|v\|_\alpha. \tag{2}$$

For $v \neq 0$, this follows by applying the definition to $v/\|v\|_\alpha$; the case $v = 0$ is immediate.

Lemma A.1. *For any norm on a finite-dimensional real vector space and any nonempty subset E,*

$$\sup_{z\in E} \|z\| = \sup_{z\in \overline{E}} \|z\| = \sup_{z\in \operatorname{conv}(E)} \|z\|,$$

with $+\infty$ allowed.

Proof. The first equality follows from continuity of the norm. If $z = \sum_{i=1}^k \lambda_i z_i$, with $z_i \in E$, $\lambda_i \geq 0$, and $\sum_i \lambda_i = 1$, then

$$\|z\| \leq \sum_{i=1}^k \lambda_i \|z_i\| \leq \sup_{w\in E} \|w\|.$$

Together with $E \subseteq \operatorname{conv}(E)$, this proves the second equality. □

For a linear functional $\ell : \mathbb{R}^m \to \mathbb{R}$, its dual norm is

$$\|\ell\|_{\beta,*} := \sup_{\|w\|_\beta \le 1} |\ell(w)|.$$

Thus $|\ell(w)| \le \|\ell\|_{\beta,*}\|w\|_\beta$. For each nonzero $w \in \mathbb{R}^m$, there is a *norming functional* ℓ with $\|\ell\|_{\beta,*} = 1$ and $\ell(w) = \|w\|_\beta$: extend the functional $tw \mapsto t\|w\|_\beta$ from the line $\mathbb{R}w$ by the finite-dimensional Hahn–Banach theorem. We will use this elementary duality fact to apply the scalar mean-value theorem to vector-valued maps.

Definition A.2. The (α, β)-Lipschitz constant of f on X is

$$L^{(\alpha,\beta)}(f, X) := \sup_{\substack{x,y\in X \\ x\neq y}} \frac{\|f(y) - f(x)\|_\beta}{\|y - x\|_\alpha} \in [0, +\infty].$$

We call f (α, β)-*Lipschitz* when this quantity is finite, and *locally Lipschitz* when every point has a neighborhood in X on which it is Lipschitz. Local Lipschitzness is independent of the choice of finite-dimensional norms and implies continuity.

A.2 Ordinary Jacobians and the smooth locus

Definition A.3. The map f is *Fréchet differentiable* at $x \in X$ if there is a linear map $A : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$\lim_{\substack{y\to x \\ y\neq x}} \frac{\|f(y) - f(x) - A(y - x)\|_\beta}{\|y - x\|_\alpha} = 0.$$

This map is unique, and its matrix in the standard bases is the *ordinary Jacobian* $J_f(x) \in \mathbb{R}^{m\times n}$. We write $\mathrm{Diff}(f)$ for the differentiability locus, and

$$U_f := \mathrm{Diff}^1(f) := \{x \in X : f \text{ is } C^1 \text{ on a neighborhood of } x\}.$$

Thus U_f is open, $U_f \subseteq \mathrm{Diff}(f)$, and J_f is continuous on U_f.

Lemma A.4. *For every $x \in \mathrm{Diff}(f)$,*

$$\|J_f(x)\|_{\alpha,\beta} \le L^{(\alpha,\beta)}(f, X).$$

Proof. The assertion is immediate if the right side is infinite. Otherwise write it as L. For every $v \in \mathbb{R}^n$, openness of X and differentiability at x give

$$J_f(x)v = \lim_{t\to 0} \frac{f(x + tv) - f(x)}{t}.$$

The norm of each difference quotient is at most $L\|v\|_\alpha$. Taking the limit and then the supremum over $\|v\|_\alpha \le 1$ proves the claim. □

Lemma A.5 (The open smooth locus)**.** *If X and f are definable, then $\mathrm{Diff}(f)$, U_f, and $J_f : \mathrm{Diff}(f) \to \mathbb{R}^{m\times n}$ are definable. Moreover,*

$$\dim(X \setminus U_f) < n.$$

In particular, U_f is open and dense in X. No continuity or Lipschitz hypothesis on f is needed here.

Proof. Differentiability and continuity can be expressed by first-order formulas over the real field, using Euclidean norms; hence the indicated loci and derivative map are definable. Apply smooth cell decomposition to the coordinate functions of f, using a common refinement. Every n-dimensional cell in the resulting partition of X is open and lies in U_f. Their complement has dimension less than n, and contains $X \setminus U_f$. See Bhardwaj & van den Dries (2022, Appendix A, Theorem A.14) or Van den Dries (1998, Theorem 7.3.2). □

A.3 THE CLARKE JACOBIAN AND GOOD MAPS

Definition A.6. For $S \subseteq X$, define the *restricted Jacobian cluster set* at $x \in X$ by

$$K_f^S(x) := \left\{ A \in \mathbb{R}^{m\times n} : \begin{array}{l} \text{there are } x_k \in S \cap \mathrm{Diff}(f) \text{ such that} \\ x_k \to x \text{ and } J_f(x_k) \to A \end{array} \right\}, \qquad K_f(x) := K_f^X(x).$$

Only sequences with a finite matrix limit are included; the points x_k are not required to be distinct. For a locally Lipschitz map f, its *Clarke Jacobian* and its version with a set $N \subseteq X$ deleted are

$$J_f^c(x) := \operatorname{conv} K_f(x), \qquad J_f^{c,N}(x) := \operatorname{conv} K_f^{X\setminus N}(x).$$

The latter notation is defined for arbitrary N, not just null sets. These are the usual cluster-set conventions; compare Edalat (2021, Section 4, equation (2)).

Lemma A.7. *Suppose X and f are definable and f is locally Lipschitz. Then $K_f(x)$ is nonempty and compact for every $x \in X$. Consequently, $J_f^c(x)$ is nonempty, convex, and compact.*

Proof. Fix $x \in X$ and a neighborhood $V \subseteq X$ on which f is Lipschitz. Lemma A.4, applied on V, bounds the ordinary Jacobians on $V \cap \mathrm{Diff}(f)$. Choose points in the dense set $U_f \cap V$ tending to x. A convergent subsequence of their bounded Jacobians gives an element of $K_f(x)$.

To see that $K_f(x)$ is closed, let $A_i \in K_f(x)$ and $A_i \to A$. Choose $z_i \in V \cap \mathrm{Diff}(f)$ with

$$\|z_i - x\|_2 < 1/i, \qquad \|J_f(z_i) - A_i\|_{\alpha,\beta} < 1/i.$$

Then $z_i \to x$ and $J_f(z_i) \to A$, so $A \in K_f(x)$. Local boundedness now gives compactness. By Carathéodory's theorem in the mn-dimensional matrix space, every point of $\operatorname{conv} K_f(x)$ is a convex combination of at most $mn+1$ points of $K_f(x)$. This convex hull is therefore a continuous image of a compact simplex times $K_f(x)^{mn+1}$, and is compact. □

Thus no additional closure of the convex hull is needed. This proof uses density of the smooth locus in place of Rademacher's theorem.

Lemma A.8. *For a locally Lipschitz map f, if $x \in \mathrm{Diff}(f)$, then $J_f(x) \in K_f(x) \subseteq J_f^c(x)$. If $x \in U_f$, then*

$$K_f(x) = J_f^c(x) = \{J_f(x)\}.$$

Proof. The constant sequence $x_k = x$ proves the first assertion. For the second, all sufficiently close points lie in a neighborhood on which f is C^1, and continuity of J_f forces every Jacobian limit at x to equal $J_f(x)$. □

Definition A.9 (Dense-domain good map)**.** For a locally Lipschitz map f, a set-valued map $D : X \rightrightarrows \mathbb{R}^{m\times n}$ is *good for* f if

$$D(x) \subseteq J_f^c(x) \quad (x \in X), \qquad S_D := \{x \in X : D(x) \neq \varnothing\} \text{ is dense in } X.$$

Neither D nor S_D is required to be definable or measurable. The values of D need not be closed or convex. Write

$$s(D) := \sup_{\substack{x\in X \\ A\in D(x)}} \|A\|_{\alpha,\beta}.$$

Example A.10. Suppose X and f are definable and f is locally Lipschitz. The full Clarke Jacobian J_f^c is good. So is the ordinary Jacobian, regarded as the singleton-or-empty map

$$\widehat{J}_f(x) := \begin{cases} \{J_f(x)\}, & x \in \mathrm{Diff}(f), \\ \varnothing, & x \notin \mathrm{Diff}(f). \end{cases}$$

For any dense $S \subseteq X$, the map equal to $\{J_f(x)\}$ on $S \cap U_f$ and empty elsewhere is also good. Indeed, $S \cap U_f$ is dense in X, since U_f is open and dense. These examples require no measure theory.

A.4 PROOF OF THE MAIN THEOREM

The geometric input is a finite-intersection statement for line segments. It lets us perturb one endpoint and use the ordinary mean-value theorem along finitely many smooth pieces, rather than integrate through the exceptional set.

Lemma A.11 (Good endpoints)**.** *Let $B \subseteq \mathbb{R}^n$ be definable with $\dim B < n$, and let $a \in \mathbb{R}^n$. Then*

$$\{b \in \mathbb{R}^n : [a, b] \cap B \text{ is finite}\}$$

is dense in $\mathbb{R}^n$.

Proof. If $n = 1$, the set B is finite. Suppose $n \geq 2$ and use the Euclidean radial projection

$$r : B \setminus \{a\} \longrightarrow \mathbb{S}^{n-1}, \qquad r(q) := \frac{q - a}{\|q - a\|_2}.$$

Every fiber lies on a ray and has dimension at most one. The definable set

$$E := \{u \in \mathbb{S}^{n-1} : \dim r^{-1}(u) = 1\}$$

has dimension at most $n - 2$, by the fiber-dimension theorem (Bhardwaj & van den Dries, 2022, Appendix A, Proposition A.6). Hence its cone

$$C := \{a + tu : u \in E,\ t > 0\}$$

has dimension at most $n - 1$ and has empty interior. If $b \notin C$ and $b \neq a$, the ray through b meets $B \setminus \{a\}$ in a definable set of dimension at most zero, which is finite. Adding a does not affect finiteness. The case $b = a$ is immediate. Thus every $b \notin C$ is a good endpoint, and these points are dense. □

Lemma A.12 (Segment estimate)**.** *Suppose f is continuous on $[a, b] \subseteq X$, with $a \neq b$, and Fréchet differentiable at every point of (a, b). If $\|J_f(z)\|_{\alpha,\beta} \leq M < \infty$ for all $z \in (a, b)$, then*

$$\|f(b) - f(a)\|_\beta \leq M\|b - a\|_\alpha.$$

No definability assumption is needed.

Proof. If $f(a) = f(b)$, the conclusion is immediate. Otherwise choose a norming functional ℓ for $f(b) - f(a)$ and apply the scalar mean-value theorem to

$$h(t) := \ell\big(f(a + t(b - a))\big) \qquad (0 \leq t \leq 1).$$

This function is continuous on $[0, 1]$, differentiable on $(0, 1)$, and

$$|h'(t)| = \big|\ell(J_f(a + t(b - a))(b - a))\big| \leq M\|b - a\|_\alpha.$$

Since $h(1) - h(0) = \|f(b) - f(a)\|_\beta$, the required estimate follows. The endpoints need not be differentiability points. □

Lemma A.13 (The smooth locus determines the Lipschitz constant)**.** *Let X be definable and convex, and let f be continuous and definable. Then*

$$L^{(\alpha,\beta)}(f, X) = \sup_{x \in U_f} \|J_f(x)\|_{\alpha,\beta}$$

as an equality in $[0, +\infty]$.

Proof. Write L and M for the left and right sides, respectively. Lemma A.4 gives $M \leq L$. If $M = +\infty$, this proves the result; suppose $M < +\infty$.

Fix distinct $x, y \in X$ and apply Lemma A.11 to $B := X \setminus U_f$ with $a = x$. We can choose $y_k \in X \setminus \{x\}$ tending to y such that $[x, y_k] \cap B$ is finite. Convexity gives $[x, y_k] \subseteq X$. Subdivide this segment at its finitely many intersection points with B. Each resulting open subsegment lies in U_f, so Lemma A.12 applies to each piece. Their lengths add to $\|y_k - x\|_\alpha$, because they are consecutive pieces of a straight segment. Hence

$$\|f(y_k) - f(x)\|_\beta \leq M\|y_k - x\|_\alpha.$$

Letting $k \to \infty$ and using continuity of f gives the same inequality with y in place of y_k. Taking the supremum over distinct x, y yields $L \leq M$. This argument includes $M = 0$ and does not divide by M. □

We can now prove Theorem 4.2, in the following slightly more general form. Local Lipschitzness suffices when an infinite value of the global Lipschitz constant is allowed.

Theorem A.14 (Dense-domain good maps recover the Lipschitz constant)**.** *Let $X \subseteq \mathbb{R}^n$ be nonempty, definable, open, and convex, and let $f : X \to \mathbb{R}^m$ be definable and locally Lipschitz. If D is good for f, then*

$$L^{(\alpha,\beta)}(f, X) = \sup_{\substack{x \in X \\ A \in D(x)}} \|A\|_{\alpha,\beta}. \tag{3}$$

The equality holds in $[0, +\infty]$, for the arbitrary norms fixed above. Under the global Lipschitz hypothesis in the main text, both sides are finite.

Proof. Set $S := S_D$ and $U := U_f$. By Lemma A.13,

$$L := L^{(\alpha,\beta)}(f, X) = \sup_{x \in U} \|J_f(x)\|_{\alpha,\beta}.$$

Since S is dense and U is open, $S \cap U$ is dense in U. For every $x \in S \cap U$, Lemma A.8 gives

$$\varnothing \neq D(x) \subseteq J_f^c(x) = \{J_f(x)\},$$

so $D(x) = \{J_f(x)\}$. Continuity of J_f on U therefore yields

$$\begin{aligned} s(D) &\geq \sup_{x \in S \cap U} \|J_f(x)\|_{\alpha,\beta} \\ &= \sup_{x \in U} \|J_f(x)\|_{\alpha,\beta} = L. \end{aligned}$$

This also settles the case $L = +\infty$. If $L < +\infty$, every ordinary Jacobian has norm at most L by Lemma A.4. The closed convex matrix ball of radius L contains their limits and convex hulls. Thus $\|A\|_{\alpha,\beta} \leq L$ for every $A \in D(x) \subseteq J_f^c(x)$, giving $s(D) \leq L$. □

Corollary A.15. *Under the hypotheses on X and f in Theorem A.14,*

$$L^{(\alpha,\beta)}(f, X) = \sup_{x \in \mathrm{Diff}(f)} \|J_f(x)\|_{\alpha,\beta} = \sup_{\substack{x \in X \\ A \in J_f^c(x)}} \|A\|_{\alpha,\beta}.$$

Proof. Apply Theorem A.14 to $\widehat{J}_f$ and J_f^c, using Example A.10. □

The last equality recovers Theorem 2.2 in the definable setting, with our convention that Jacobians act from $\mathbb{R}^n$ to $\mathbb{R}^m$.

Corollary A.16 (Equivalence of good maps)**.** *Under the same hypotheses, any two good maps D_0, D_1 satisfy $s(D_0) = s(D_1)$.*

Proof. Both quantities equal $L^{(\alpha,\beta)}(f, X)$ by Theorem A.14. □

Remark A.17 (Agreement and the pointwise order). For a good map D, the equality $D(x) = \{J_f(x)\}$ holds on the dense set $S_D \cap U_f$, but need not hold almost everywhere. Pointwise inclusion orders the good maps into a join-semilattice, with join $(D_0 \vee D_1)(x) = D_0(x) \cup D_1(x)$. They do not form a lattice in general, even for definable f: for $f \equiv 0$ on $\mathbb{R}$, let D_0 equal $\{0\}$ on $\mathbb{Q}$ and be empty elsewhere, and let D_1 equal $\{0\}$ on $\mathbb{R} \setminus \mathbb{Q}$ and be empty elsewhere. Both maps are good, but their intersection is everywhere empty, and they have no common lower bound among good maps. The equivalence of their norm suprema does not require an intersection argument.

Remark A.18 (Density and sampling). Theorem A.14 is a deterministic supremum identity. It does not assert that a derivative is available almost surely under random sampling. For example, J_f^c restricted to $\mathbb{Q}^n \cap X$ and made empty elsewhere is good for a definable locally Lipschitz f, but is empty almost surely under an absolutely continuous sampling law. Any availability and exploration assumptions used in a sampling analysis must therefore be stated separately; they are not consequences of dense-domain goodness.

A.5 AN O-MINIMAL WARGA THEOREM

The main theorem used continuity of J_f on U_f only to recover a supremum. We next show that this locus also recovers every ordinary Jacobian limit, including limits based at differentiability points outside U_f. No convexity assumption on X is made in this subsection.

Theorem A.19 (O-minimal Warga theorem: dense-complement invariance)**.** *Let $X \subseteq \mathbb{R}^n$ be nonempty, definable, and open, and let $f : X \to \mathbb{R}^m$ be definable and locally Lipschitz. For every set $N \subseteq X$ whose complement is dense in X, and every $x \in X$,*

$$K_f^{X\setminus N}(x) = K_f(x), \qquad J_f^{c,N}(x) = J_f^c(x). \tag{4}$$

Neither definability nor measurability of N is required. The hypothesis is $\operatorname{int}_X N = \varnothing$, not that N be nowhere dense or null.

We use the name "o-minimal Warga theorem" to emphasize the comparison with the classical theorem in Section A.6, rather than as a priority claim. The proof rests on the following elementary consequence of o-minimality.

Lemma A.20 (Rectangle lemma)**.** *Let $Q \subseteq \mathbb{R}^d$ be a nonempty definable open box and let $\tau > 0$. If finitely many definable sets $E_1, \dots, E_r$ cover $(0,\tau) \times Q$, then some E_i contains $(0,\tau') \times Q'$ for a positive τ' and a nonempty open box $Q' \subseteq Q$.*

Proof. Define

$$A_i := \{v \in Q : \exists a \in (0,\tau)\ \forall t \in (0,a),\ (t,v) \in E_i\}.$$

For a fixed v, the finitely many definable subsets $\{t \in (0,\tau) : (t,v) \in E_i\}$ cover $(0,\tau)$. One-dimensional o-minimality implies that at least one contains an initial interval $(0,a)$. Thus $Q = \bigcup_i A_i$.

The finite-union dimension property implies that some A_i has nonempty interior. On a nonempty open subset V of that interior, definable choice supplies a definable function $a : V \to (0,\tau)$ such that $(0, a(v)) \times \{v\} \subseteq E_i$. After shrinking to a nonempty open subset, cell decomposition makes a continuous. Around one of its points choose an open box $Q' \subseteq V$ and $\tau' > 0$ such that $a(v) > \tau'$ on Q'. Then $(0,\tau') \times Q' \subseteq E_i$. The choice and continuity facts used here are Theorem A.2 and Proposition A.10 of Bhardwaj & van den Dries (2022, Appendix A). □

Lemma A.21 (Approximation of a differentiable Jacobian)**.** *Let X and f be definable, and let $V \subseteq X$ be definable, open, and dense, with $f|_V$ of class C^1. For every $p \in \operatorname{Diff}(f)$,*

$$(p, J_f(p)) \in \overline{\{(q, J_f(q)) : q \in V\}},$$

where closure is relative to $X \times \mathbb{R}^{m\times n}$. Local Lipschitzness is not required.

Proof. Put $A := J_f(p)$ and suppose the asserted inclusion fails. Using the maximum norm on matrix entries, choose $r, \eta > 0$ with $B_2(p,r) \subseteq X$ such that

$$\max_{i,j} |(J_f(q) - A)_{ij}| \geq \eta \qquad (q \in V \cap B_2(p,r)). \tag{5}$$

Take $Q = (1,2)^n$ and choose $\tau > 0$ so small that $p + tv \in B_2(p,r)$ for $0 < t < \tau$ and $v \in Q$. Define

$$H(t,v) := \frac{f(p+tv) - f(p)}{t} - Av.$$

Differentiability at p implies

$$H(t,v) \longrightarrow 0 \quad (t \downarrow 0) \qquad \text{for every fixed } v \in Q. \tag{6}$$

Consider the following finite definable cover of $(0,\tau) \times Q$:

$$E_0 := \{(t,v) : p + tv \notin V\},$$
$$E_{ij}^{\sigma} := \{(t,v) : p + tv \in V,\ \sigma(J_f(p+tv) - A)_{ij} \geq \eta\},$$

where $1 \leq i \leq m$, $1 \leq j \leq n$, and $\sigma \in \{-1, 1\}$. These sets cover by (5), and are definable because the ordinary derivative map is definable.

By Lemma A.20, one member contains $(0, \tau') \times Q'$ for some nonempty open box $Q' \subseteq Q$. This member cannot be E_0: for any fixed $t > 0$, the nonempty open set $p + tQ'$ must meet the dense set V. Thus for some fixed i, j, σ, throughout this rectangle,

$$p + tv \in V, \qquad \sigma(J_f(p + tv) - A)_{ij} \geq \eta.$$

The chain rule gives

$$\frac{\partial(\sigma H_i)}{\partial v_j}(t, v) = \sigma(J_f(p + tv) - A)_{ij} \geq \eta.$$

Choose $a, b \in Q'$ that differ only in their j-th coordinates, with $b_j - a_j = \ell > 0$. For each fixed $t \in (0, \tau')$, the scalar mean-value theorem along the coordinate segment from a to b gives

$$\sigma\big(H_i(t, b) - H_i(t, a)\big) \geq \eta\ell.$$

The left side tends to zero as $t \downarrow 0$ by (6), whereas the right side is positive and independent of t. This contradiction proves the claim. □

The conclusion approximates the entire Jacobian matrix simultaneously. Approximating different rows or columns along unrelated sequences would not suffice.

Proposition A.22 (Dense restrictions have the same derivative graph closure)**.** *Let X and f be definable, with no continuity or Lipschitz assumption on f, and let $S \subseteq X$ be any dense set. For $T \subseteq X$, put*

$$\Gamma_T := \{(q, J_f(q)) : q \in T \cap \mathrm{Diff}(f)\}.$$

Then, with closures relative to $X \times \mathbb{R}^{m \times n}$,

$$\overline{\Gamma_{S \cap U_f}} = \overline{\Gamma_S} = \overline{\Gamma_{U_f}} = \overline{\Gamma_X}. \tag{7}$$

Proof. Lemma A.21, applied with $V = U_f$, gives $\Gamma_X \subseteq \overline{\Gamma_{U_f}}$. The reverse inclusion after closure follows from $\Gamma_{U_f} \subseteq \Gamma_X$.

Since U_f is open, $S \cap U_f$ is dense in U_f. Continuity of J_f on U_f gives $\Gamma_{U_f} \subseteq \overline{\Gamma_{S \cap U_f}}$. Combining these facts with $\Gamma_{S \cap U_f} \subseteq \Gamma_S \subseteq \Gamma_X$ proves all the equalities. □

Proof of Theorem A.19. Apply Proposition A.22 with $S = X \setminus N$. For every $x \in X$, the fiber of $\overline{\Gamma_S}$ over x is exactly $K_f^S(x)$, by the sequential characterization of closure in finite-dimensional spaces. Similarly, the fiber of $\overline{\Gamma_X}$ is $K_f(x)$. Equality of the graph closures gives $K_f^{X \setminus N}(x) = K_f(x)$. Taking convex hulls gives $J_f^{c,N}(x) = J_f^c(x)$. □

Corollary A.23 (Deletion of a lower-dimensional definable set)**.** *Under the hypotheses on X and f in Theorem A.19, if $N \subseteq X$ is definable and $\dim N < n$, then*

$$K_f^{X \setminus N}(x) = K_f(x), \qquad J_f^{c,N}(x) = J_f^c(x) \qquad (x \in X).$$

Proof. Such a set N has empty interior, so $X \setminus N$ is dense. Apply Theorem A.19. □

Example A.24 (Deleting a dense conull set). Take $S = X \cap \mathbb{Q}^n$ and $N = X \setminus S$. Then S is dense, so ordinary Jacobians at the differentiability points in S recover $K_f(x)$ and $J_f^c(x)$ at every $x \in X$. The deleted set N is itself dense and conull. This is a topological recovery statement, not a claim about recovery by random sampling from a continuous distribution.

Remark A.25. Proposition A.22 and the corresponding equality of cluster sets hold for arbitrary definable f. Local Lipschitzness supplies the nonempty compact sets used in standard Clarke theory. Neither this result nor Theorem A.14 requires polynomial boundedness of the o-minimal structure.

A.6 Classical Warga theorem and comparison

For comparison, consider an arbitrary locally Lipschitz map on an open set; neither the map nor its domain is assumed definable. Rademacher's theorem gives differentiability almost everywhere, so the local boundedness and compactness argument in Lemma A.7 still supplies nonempty compact Jacobian cluster sets. In this general setting, Warga's theorem permits deletion of a Lebesgue-null set, rather than every set with dense complement.

Theorem A.26 (Warga's null-set invariance theorem, in Clarke notation)**.** *Let $X \subseteq \mathbb{R}^n$ be open and let $f : X \to \mathbb{R}^m$ be locally Lipschitz. If $N \subseteq X$ has Lebesgue measure zero, then for every $x \in X$,*

$$J_f^{c,N}(x) = J_f^c(x),$$

or, equivalently,

$$J_f^c(x) = \operatorname{conv}\left\{ \lim_{k\to\infty} J_f(x_k) : x_k \to x, \quad x_k \in \operatorname{Diff}(f) \setminus N \right\}.$$

There are no definability assumptions on X, f, or N.

This is the Clarke-Jacobian formulation of Warga (1981, Theorem 4), stated explicitly with that attribution in Edalat (2021, Section 4, equation (2) and Theorem 4.1). It is recorded here without proof and is not used in either tame theorem.

The comparison is a tradeoff in hypotheses: the classical theorem applies to every locally Lipschitz map but deletes only null sets; Theorem A.19 assumes definability of the map but deletes every set with dense complement. The tame theorem also preserves the matrix cluster sets before convexification, rather than merely their convex hulls.

Example A.27 (Why density alone fails without definability). Choose a closed nowhere dense set $C \subseteq [0, 1]$ of positive Lebesgue measure, and define

$$f(t) := \int_0^t \mathbf{1}_C(s)\, ds \qquad (0 < t < 1).$$

For $s < t$, we have $0 \le f(t) - f(s) \le t - s$, so f is 1-Lipschitz. On the dense open set $S := (0, 1)\setminus C$, it is locally constant, hence $f'(s) = 0$ for all $s \in S$. Therefore $K_f^S(x) = \{0\}$ for every $x \in (0, 1)$.

The Lebesgue fundamental theorem of calculus gives $f'(x) = 1$ at almost every point of $C \cap (0, 1)$. At any such point, the constant sequence gives $1 \in K_f(x) \subseteq J_f^c(x)$. Deleting $C \cap (0, 1)$ changes the Clarke Jacobian there, even though its complement is open and dense. The integration used in this counterexample is solely for comparison, not part of either tame proof.

B Statistical properties of the sampling estimators

This appendix makes precise the statistical claims of Section 4 about Algorithms 1–3. All three algorithms return the largest value of a derivative oracle observed at sampled points. We show that this value is almost surely a lower bound on the modulus of continuity (Proposition B.7), that it converges to the modulus whenever the sampling scheme keeps exploring every subregion (Theorem B.10), that the exploration requirement is met by Algorithms 1 and 2 and by Algorithm 3 once its exploration bonus is bounded away from zero (Proposition B.13), and that the estimator may be biased at every sample size, strictly so when the maximizing set is null, but is asymptotically unbiased, and even exact after finitely many samples for piecewise-affine networks (Proposition B.15). For uniform sampling, we give the exact law of the error and rates under a volume-growth condition (Theorem B.16). Finally, we prove a minimax lower bound showing that no adaptive sampling scheme can improve upon uniform sampling by more than a constant factor uniformly over ReLU networks (Theorem B.19). The proofs rely on Appendix A and on elementary probability, in particular on Lévy's extension of the Borel–Cantelli lemma; they do not use the law of large numbers, since a running maximum is not an average.

B.1 SETTING

Throughout this appendix, $X = \prod_{j=1}^{n}(a_j, b_j) \subseteq \mathbb{R}^n$ is a nonempty bounded open box, which is the domain denoted D in Algorithms 1–3, and $f : X \to \mathbb{R}^m$ is definable and (α, β)-Lipschitz, so that

$$L := L^{(\alpha,\beta)}(f, X) < \infty.$$

We write λ for Lebesgue measure on $\mathbb{R}^n$ and $g(x) := \|J_f(x)\|_{\alpha,\beta}$ for $x \in \mathrm{Diff}(f)$, with U_f, $\mathrm{Diff}(f)$, and J_f^c as in Appendix A. By Lemma A.13, applied to the semialgebraic convex set X,

$$L = \sup_{x \in U_f} g(x), \tag{8}$$

and g is continuous on the open set U_f, since J_f is continuous there and the matrix norm is continuous.

Lemma B.1 (Lower-dimensional definable sets are null)**.** *If $B \subseteq \mathbb{R}^n$ is definable and $\dim B < n$, then $\lambda(B) = 0$. In particular $\lambda(X \setminus U_f) = 0$.*

Proof. By cell decomposition, B is a finite union of cells of dimension less than n (Van den Dries, 1998, Chapter 3 and Chapter 4). A cell of dimension less than n in $\mathbb{R}^n$ has at least one coordinate k that is given by a graph, so that the cell is contained in $\{x \in \mathbb{R}^n : x_k = h(x_1, \ldots, x_{k-1})\}$ for a continuous function h defined on the projection of the cell. Every line parallel to the x_k-axis meets this set in at most one point, so it is null by Fubini's theorem. The second assertion follows from Lemma A.5. □

The oracle. The algorithms do not evaluate g directly, but call a subroutine, typically automatic differentiation (AD), that returns a matrix at every input. We model this subroutine by a Borel measurable function $\hat{g} : X \to [0, \infty)$ and impose the following assumption.

Assumption B.2 (Almost-everywhere exact oracle)**.** There is a Lebesgue-null set $N \subseteq X$ such that $X \setminus N \subseteq U_f$ and $\hat{g}(x) = g(x)$ for all $x \in X \setminus N$.

Remark B.3 (Oracles satisfying Assumption B.2). (i) If, at every $x \in X$, the oracle returns $\|G\|_{\alpha,\beta}$ for some $G \in J_f^c(x)$, or more generally for some $G \in D(x)$ where D is good for f and nonempty everywhere (an oracle must supply a value at every sampled input, so a good map that is empty almost everywhere is not an oracle), then $\hat{g} = g$ on U_f by Lemma A.8, and Assumption B.2 holds with $N = X \setminus U_f$, which is null by Lemma B.1. (ii) If the oracle is AD as modeled by selection Jacobians (Bolte & Pauwels, 2020) or by a conservative field (Bolte & Pauwels, 2021), then the output equals $J_f(x)$ outside a Lebesgue-null set N_{AD}, by the main results of those papers, and Assumption B.2 holds with $N = (X \setminus U_f) \cup N_{\mathrm{AD}}$. Thus the sporadic failures of AD on null sets discussed in Section 4 do not affect the statements below, because the sampling laws are absolutely continuous (Lemma B.6).

Adaptive partition schemes. We formalize the common structure of Algorithms 1–3. A *box* is a closed product of nondegenerate compact intervals. *Halving* a box replaces it by the two boxes obtained by cutting a longest side at its midpoint with an axis-aligned hyperplane; ties between longest sides are broken by a fixed rule. A *partition* $\mathcal{P}$ of X is a finite set of boxes with pairwise disjoint interiors whose union is the closure $\overline{X}$; uniform sampling from a closed box is uniform sampling from its interior, so boundaries are immaterial. Boxes created during the run are called *nodes*; the boxes of the current partition are called *leaves*. Let $(\xi_t)_{t \geq 1}$ be independent random variables, uniformly distributed on $[0, 1]$, which serve as the internal randomness of the scheme.

Definition B.4 (Adaptive partition scheme)**.** An adaptive partition scheme produces, for $t = 1, 2, \ldots$, a partition $\mathcal{P}_t$, a leaf $s_t \in \mathcal{P}_t$, and a sample $x_t \in s_t$ as follows. Let $\mathcal{P}_0$ be a fixed finite partition of $\overline{X}$ into boxes, for instance $\{\overline{X}\}$ or a grid of K^n equal boxes, and let $\mathcal{H}_{t-1} := (\xi_1, x_1, \hat{g}(x_1), \ldots, \xi_{t-1}, x_{t-1}, \hat{g}(x_{t-1}))$ be the history before step t. At step t, as a measurable function of $(\mathcal{H}_{t-1}, \xi_t)$, the scheme either keeps $\mathcal{P}_t := \mathcal{P}_{t-1}$ or halves one leaf of $\mathcal{P}_{t-1}$ to obtain $\mathcal{P}_t$; it then selects a leaf $s_t \in \mathcal{P}_t$, again measurably, and draws x_t uniformly from s_t, conditionally independently of everything else. Its output after t samples is

$$r_t := \max_{1 \leq i \leq t} \hat{g}(x_i),$$

which is nondecreasing in t. We write $\mathcal{F}_t$ for the σ-algebra generated by $\mathcal{H}_t$ and $n_s(t)$ for the number of samples drawn from the node s while it was a leaf, up to and including step t. Children start with

$n_s = 0$. Statistics indexed by $t-1$ are those available before the sample of step t is drawn; a leaf created at step t has $n_s(t-1) = 0$.

Algorithm 1 is the scheme with no halvings, $s_t = \overline{X}$. Algorithm 2 starts from the grid $\mathcal{P}_0$ of K^n equal boxes, performs no halvings, and selects leaves in a round-robin order; the joint law of its first N samples, for N a multiple of K^n, is the law of the samples of Algorithm 2, up to their order. Algorithm 3 halves an index-maximizing leaf at the subdivision times and selects an index-maximizing leaf.[2]

Definition B.5 (Persistent exploration)**.** An adaptive partition scheme *explores persistently* if, almost surely,

(E1) all but finitely many of the nodes created during the run are selected at least once while they are leaves, and

(E2) every node that is never halved is selected infinitely often.

B.2 Almost sure lower bound and consistency

Lemma B.6 (Samples avoid null sets)**.** *For any adaptive partition scheme and any Lebesgue-null set $N \subseteq \mathbb{R}^n$, almost surely $x_t \notin N$ for all $t \geq 1$.*

Proof. Given $\mathcal{F}_{t-1}$ and ξ_t, the sample x_t is uniform on the box s_t, so $\mathbb{P}(x_t \in N \mid \mathcal{F}_{t-1}, \xi_t) = \lambda(N \cap s_t)/\lambda(s_t) = 0$. Take expectations and sum over t. □

Proposition B.7 (Almost sure lower bound)**.** *Under Assumption B.2, for any adaptive partition scheme, almost surely $\hat{g}(x_t) = g(x_t) \leq L$ for all t, and hence $r_1 \leq r_2 \leq \cdots \leq L$.*

Proof. By Lemma B.6, almost surely every x_t lies in $X \setminus N \subseteq U_f \subseteq \mathrm{Diff}(f)$, where $\hat{g}(x_t) = g(x_t)$. Lemma A.4 gives $g(x_t) \leq L$. □

Lemma B.8 (Halving shrinks boxes)**.** *Let Q be a box in $\mathbb{R}^n$ with longest side M. After nk successive halvings along nested boxes, the resulting box has all sides at most $2^{-k}M$, hence Euclidean diameter at most $\sqrt{n}\,2^{-k}M$.*

Proof. Sides never grow under halving. At most n sides have length exceeding $M/2$. As long as one of them remains, the current longest side is one of them, and halving it brings it to length at most $M/2$. Hence after at most n halvings every side is at most $M/2$. Iterate k times. □

Lemma B.9 (Hitting lemma)**.** *Let an adaptive partition scheme explore persistently. Then for every nonempty open set $O \subseteq X$, almost surely $x_t \in O$ for infinitely many t.*

Proof. Fix $x^* \in O$ and put $p_t := \mathbb{P}(x_t \in O \mid \mathcal{F}_{t-1}, \xi_t) = \lambda(O \cap s_t)/\lambda(s_t)$. By Lévy's extension of the Borel–Cantelli lemma (Williams, 1991, Section 12.15), applied to the filtration $\mathcal{G}_t := \sigma(\mathcal{H}_t, \xi_{t+1})$, for which s_t is $\mathcal{G}_{t-1}$-measurable and $\{x_t \in O\} \in \mathcal{G}_t$,

$$\{x_t \in O \text{ for infinitely many } t\} = \Big\{\sum_{t \geq 1} p_t = \infty\Big\} \qquad \text{almost surely.} \tag{9}$$

It therefore suffices to show that $\sum_t p_t = \infty$ almost surely.

Define a nested random sequence of nodes containing x^*: let $R_0 := \overline{X}$, and whenever R_k is halved, let R_{k+1} be a child of R_k containing x^*, chosen by a fixed rule when x^* lies on the cutting hyperplane. Two cases occur.

If the sequence is finite, with last member R_K, then R_K is never halved, so by (E2) it is selected infinitely often. Since R_K is a box, it is the closure of its interior, and $x^* \in R_K \cap O$ with O open,

[2] As displayed, Algorithm 3 subdivides before selecting, so every sample is drawn from a leaf. If an implementation draws the sample at a subdivision step from the parent box, whose uniform law is a mixture of the uniform laws of its two children, and counts it for no leaf, the proofs below cover this variant verbatim: they only use the conditional law of x_t and the fact that at most one sample per subdivision is not counted.

so $O \cap \operatorname{int} R_K$ is a nonempty open set and $q := \lambda(O \cap R_K)/\lambda(R_K) > 0$. At every step t at which $s_t = R_K$ we have $p_t = q$, so $\sum_t p_t = \infty$.

If the sequence is infinite, then Lemma B.8 gives $\operatorname{diam} R_k \to 0$, so $R_k \subseteq O$ for all $k \geq k_0$, since $x^* \in R_k$ and O is an open neighborhood of x^*. By (E1), all but finitely many R_k are selected at least once while they are leaves. At every such step t we have $s_t = R_k \subseteq O$ and $p_t = 1$, so again $\sum_t p_t = \infty$.

Both cases hold on the almost sure event on which (E1) and (E2) hold, which proves the claim by (9). □

Theorem B.10 (Consistency)**.** *Under Assumption B.2, for every adaptive partition scheme that explores persistently,*

$$r_t \uparrow L \qquad \text{almost surely as } t \to \infty.$$

Proof. Fix $\varepsilon > 0$ and let $O_\varepsilon := \{x \in U_f : g(x) > L - \varepsilon\}$. This set is open, since U_f is open and g is continuous on U_f, and nonempty by (8). By Lemma B.9, almost surely some x_t lies in O_ε, and by Lemma B.6 this x_t lies outside N, so $\hat{g}(x_t) = g(x_t) > L - \varepsilon$ and $r_{t'} > L - \varepsilon$ for all $t' \geq t$. Hence $\liminf_t r_t \geq L - \varepsilon$ almost surely. Letting $\varepsilon \downarrow 0$ along a countable sequence and combining with Proposition B.7 proves the theorem. □

Remark B.11. Theorem B.10 concerns a running maximum, not an average, so the relevant tool is a conditional Borel–Cantelli argument rather than the law of large numbers. The theorem does not need any regularity of the sampling density: only positive conditional probability of hitting every fixed open set, infinitely often, enters through (E1)–(E2). This is exactly the availability and exploration hypothesis anticipated in Remark A.18.

B.3 Which selection rules explore persistently

We now verify the exploration condition for the three algorithms. For Algorithm 3 we consider the following family of index rules, which contains the rule (1) of the main text whenever $\sigma_0 > 0$.

Definition B.12 (Optimistic index rule)**.** Fix constants $n_0 \in \mathbb{N}$, $0 < \beta \leq B < \infty$, and $t_m > 1$. Each leaf s carries a statistic $\hat{m}_s(t-1)$ of the values $\hat{g}(x_i)$ sampled from s before step t, with $\hat{m}_s(t-1) \in [0, L]$ whenever all these values lie in $[0, L]$, and an exploration bonus $b_s(t)$, computed from the pre-sample count $n_s(t-1)$, with

$$b_s(t) = +\infty \text{ if } n_s(t-1) \leq n_0, \qquad \beta\sqrt{\frac{\ln(t+1)}{n_s(t-1)}} \leq b_s(t) \leq B\sqrt{\frac{\ln(t+1)}{n_s(t-1)}} \text{ if } n_s(t-1) > n_0.$$

At step t the scheme selects a leaf maximizing the index $I_s(t) := \hat{m}_s(t-1) + b_s(t)$, breaking ties measurably. The k-th halving is applied to an index-maximizing leaf at the first step T_k with $T_k \geq t_m^k$ and $T_k > T_{k-1}$.

Algorithm 3 with the index (1) uses $\hat{m}_s = s_{\max}$ and $b_s(t) = c \max\{s_\sigma, \sigma_0\}\sqrt{\ln(t+1)/s_n}$. If s_σ is the standard deviation of values in $[0, L]$, then $s_\sigma \leq L/2$ with the population convention and $s_\sigma \leq L/\sqrt{2} \leq L$ with the divisor $s_n - 1$, so for $\sigma_0 > 0$ this is an instance of Definition B.12 with $\beta = c\sigma_0$ and $B = c \max\{\sigma_0, L\}$.

Proposition B.13 (Exploration)**.** *Under Assumption B.2:*

1. *Algorithm 1 explores persistently.*
2. *Algorithm 2, continued in round-robin order over its K^n cells, explores persistently.*
3. *Every optimistic index rule (Definition B.12) explores persistently. In particular, Algorithm 3 explores persistently for every $c > 0$ and every $t_m > 1$ once its exploration bonus is bounded below as above.*

Consequently, by Theorem B.10, all three algorithms are consistent.

Proof. Parts 1 and 2 are immediate: the single leaf, respectively each of the finitely many cells, is selected infinitely often, and no node created after the initial grid is halved.

For part 3, by Proposition B.7 we may work on the almost sure event on which every observed value lies in $[0, L]$, so that $\hat{m}_s(t-1) \in [0, L]$ for all leaves and times. Let $M(t)$ denote the number of nodes created up to step t, including the $|\mathcal{P}_0|$ initial boxes. Since the k-th halving happens no earlier than step t_m^k, at most $\log_{t_m} t$ halvings occur up to step t, each creating two nodes, so

$$M(t) \le |\mathcal{P}_0| + 2\log_{t_m} t. \tag{10}$$

(E2). Suppose, for a contradiction, that on an event of positive probability some node s is never halved and is selected only finitely often, so that $n_s(t) \le n^*$ for all t, where we may take $n^* \ge n_0 + 1$. Then, for every step t at which s is a leaf,

$$I_s(t) \ge \beta\sqrt{\ln(t+1)/n^*},$$

whether $b_s(t)$ is infinite or not, since $\hat{m}_s(t-1) \ge 0$ and $n_s(t-1) \le n^*$. At each step t the selected leaf s_t satisfies $I_{s_t}(t) \ge I_s(t)$. If $n_{s_t}(t-1) \le n_0$, then $n_{s_t}(t-1) \le n^*$. Otherwise

$$L+B\sqrt{\frac{\ln(t+1)}{n_{s_t}(t-1)}} \;\ge\; I_{s_t}(t) \;\ge\; \beta\sqrt{\frac{\ln(t+1)}{n^*}}, \qquad \text{that is,} \qquad \frac{1}{\sqrt{n_{s_t}(t-1)}} \;\ge\; \frac{\beta}{B\sqrt{n^*}} - \frac{L}{B\sqrt{\ln(t+1)}}.$$

For all $t \ge t_1$, where t_1 is such that $L/\sqrt{\ln(t_1+1)} \le \beta/(2\sqrt{n^*})$, this gives $n_{s_t}(t-1) \le (2B/\beta)^2 n^* =: n^{**}$. Thus, from step t_1 on, a leaf is selected only while its pre-sample count is at most n^{**}, so every node's count stays bounded by $\bar{n} := \max\{t_1, n^{**}+1\}$ forever. Every counted sample increments the count of exactly one node, and at most one sample per halving is not counted, so for $t \ge t_1$,

$$t \;\le\; \sum_{\text{nodes } v} n_v(t) + \log_{t_m} t \;\le\; \bar{n}\, M(t) + \log_{t_m} t \;\le\; \bar{n}\,(|\mathcal{P}_0| + 2\log_{t_m} t) + \log_{t_m} t,$$

by (10). The right side is $O(\log t)$, which contradicts the left side for large t. Hence (E2) holds almost surely.

(E1). A node created at the k-th halving has infinite index until it has been selected $n_0 + 1$ times, and a leaf with infinite index is selected in preference to any leaf with finite index. Just after the k-th halving there are $|\mathcal{P}_0| + k$ leaves, so at most $(n_0+1)(|\mathcal{P}_0|+k)$ selections are needed before every leaf has finite index, provided no further halving intervenes. The $(k+1)$-st halving occurs at step $T_{k+1} \ge t_m^{k+1}$, while $T_k \le \max\{t_m^k + 1, T_{k-1} + 1\} \le t_m^k + k + 1$ by induction, so

$$T_{k+1} - T_k \;\ge\; t_m^k(t_m - 1) - k - 1,$$

which exceeds $(n_0+1)(|\mathcal{P}_0|+k)$ for all large k. For such k, both nodes created at the k-th halving are selected at least $n_0 + 1 \ge 1$ times before step T_{k+1}, which is the earliest step at which either of them could be halved. Hence (E1) holds, with finitely many exceptions. □

Remark B.14 (Why the bonus must be bounded below). With $\sigma_0 = 0$, the index (1) multiplies the exploration bonus by the empirical standard deviation s_σ of the values observed in the subregion, and for a ReLU network g is constant on each affine piece. Consistency can then fail with positive probability, for every $c > 0$. On $X = (0, 1)$ let

$$f(x) = 3\big(x - \tfrac{1}{4}\big)_+ - \big(x - \tfrac{3}{8}\big)_+ + 2\big(x - \tfrac{1}{2}\big)_+,$$

so that $|f'|$ equals 3 on $(\frac{1}{4}, \frac{3}{8})$, 2 on $(\frac{1}{2}, 1)$, and 0 on the rest of $(0, \frac{1}{2})$; thus $L = 3$. Take $n_0 = 10$ and t_m large enough that, after the first halving at $x = \frac{1}{2}$, both children receive their $n_0 + 1$ initial samples before the next scheduled halving. With positive probability all samples of the root and of the left child $[0, \frac{1}{2}]$ miss $(\frac{1}{4}, \frac{3}{8})$. On this event the left leaf has $s_{\max} = s_\sigma = 0$, hence index 0 forever, while the right leaf has $s_{\max} = 2$, and so does every descendant of it, since $g \equiv 2$ on $(\frac{1}{2}, 1)$. Every later selection and every later halving therefore stays in the right subtree, whichever of its leaves is split, and $r_t = 2 < L$ for all large t. The estimator is then not even asymptotically unbiased on this instance. (The argument must track the whole competing subtree, not a single competing leaf, because the leaf with the highest index may itself be halved.) Any positive floor σ_0 restores (E1)–(E2), since the left leaf's bonus then grows like $c\sigma_0\sqrt{\ln t}$; the value of σ_0 affects only finite-sample behavior. The experiments in Section 5 and Appendix E use $\sigma_0 = 0$, so Proposition B.13 and the asymptotic parts of Proposition B.15 do not certify the evaluated variant; the almost sure lower bound of Proposition B.7 does apply to it.

B.4 BIAS AND FINITE-TIME EXACTNESS

Proposition B.15 (Bias). *Let Assumption B.2 hold and let the adaptive partition scheme explore persistently.*

1. *(Asymptotic unbiasedness.)* $\mathbb{E}\, r_t \le L$ *for every* t*, and* $\mathbb{E}\, r_t \uparrow L$ *as* $t \to \infty$*.*
2. *(Bias at every sample size.) If* $\lambda(\{x \in U_f : g(x) = L\}) = 0$*, then* $\mathbb{P}(r_t < L) = 1$ *and hence* $\mathbb{E}\, r_t < L$ *for every* t*.*
3. *(Finite-time exactness.) If* $A := \{x \in U_f : g(x) = L\}$ *has nonempty interior, then almost surely there is a finite random* T *with* $r_t = L$ *for all* $t \ge T$*, and* $0 \le L - \mathbb{E}\, r_t \le L\, \mathbb{P}(T > t) \to 0$*.*

The hypothesis of part 3 holds whenever f *is piecewise affine, that is, affine on each of finitely many open sets with dense union in* X*; this includes every ReLU network with affine layers. If moreover the norms* $\|\cdot\|_\alpha$ *and* $\|\cdot\|_\beta$ *are definable, for instance any* ℓ_p *norm with rational* p *or* $p = \infty$*, then part 3 applies as soon as* $\lambda(A) > 0$*, so parts 2 and 3 form a dichotomy.*

Proof. Part 1: By Proposition B.7, $0 \le r_t \le L$ almost surely, and $r_t \to L$ almost surely by Theorem B.10. The bounded convergence theorem gives $\mathbb{E}\, r_t \to L$, and monotonicity of r_t gives monotonicity of $\mathbb{E}\, r_t$.

Part 2: $r_t = L$ requires $\hat{g}(x_i) = L$ for some $i \le t$, hence $x_i \in N \cup \{x \in U_f : g(x) = L\}$, a null set. By Lemma B.6 this has probability zero. Since $r_t \le L$ almost surely and $r_t < L$ with positive probability, $\mathbb{E}\, r_t < L$.

Part 3: Let $O := \operatorname{int} A$, a nonempty open subset of U_f on which $g = L$. By Lemma B.9 some x_T lies in O, and by Lemma B.6 it lies outside N, so $\hat{g}(x_T) = L$ and $r_t = L$ for all $t \ge T$. The bound on the bias follows from $L - r_t \le L\, \mathbf{1}_{\{T > t\}}$, and $\mathbb{P}(T > t) \to 0$ since $T < \infty$ almost surely.

For the piecewise-affine case, let $P_1, \dots, P_q$ be the open pieces, with Jacobians $G_1, \dots, G_q$. Every $x \in U_f$ is a limit of points of the dense set $\bigcup_i P_i$, and J_f is continuous on U_f, so $J_f(x) \in \{G_1, \dots, G_q\}$. Hence g takes finitely many values on U_f, and by (8) the value L is attained, necessarily at some G_i with $\|G_i\|_{\alpha,\beta} = L$. Then $P_i \subseteq A$, so A has nonempty interior. Finally, if the norms are definable, then A is definable, since $g|_{U_f}$ is then definable by Lemma A.5. If $\lambda(A) > 0$, then $\dim A = n$ by Lemma B.1, and a definable subset of $\mathbb{R}^n$ of dimension n has nonempty interior (Van den Dries, 1998, Chapter 4). □

For uniform sampling, the law of the error is available in closed form.

Theorem B.16 (Exact error law for uniform and stratified sampling). *Let Assumption B.2 hold and define the* volume profile

$$p(\varepsilon) := \frac{\lambda\{x \in X : \hat{g}(x) > L - \varepsilon\}}{\lambda(X)} = \frac{\lambda\{x \in X \setminus N : g(x) > L - \varepsilon\}}{\lambda(X)}, \qquad 0 < \varepsilon \le L.$$

Then $p(\varepsilon) > 0$ *for every* $\varepsilon > 0$*, and for Algorithm 1,*

$$\mathbb{P}(r_t \le L - \varepsilon) = \big(1 - p(\varepsilon)\big)^t, \qquad L - \mathbb{E}\, r_t = \int_0^L \big(1 - p(\varepsilon)\big)^t \, d\varepsilon. \tag{11}$$

For Algorithm 2 with K^n *cells of equal volume and* N *a multiple of* K^n*,*

$$\mathbb{P}(r_N \le L - \varepsilon) = \prod_s \big(1 - p_s(\varepsilon)\big)^{N/K^n} \le \big(1 - p(\varepsilon)\big)^N, \qquad p_s(\varepsilon) := \frac{\lambda\{x \in s : \hat{g}(x) > L - \varepsilon\}}{\lambda(s)}. \tag{12}$$

In particular:

1. *If* $p(\varepsilon) \ge \kappa \varepsilon^\gamma$ *for* $0 < \varepsilon \le \varepsilon_0 \le L$*, with* $\kappa, \gamma > 0$*, then for Algorithm 1*

$$L - \mathbb{E}\, r_t \;\le\; \Gamma\Big(1 + \tfrac{1}{\gamma}\Big)\, (\kappa t)^{-1/\gamma} + L\, e^{-t\, p(\varepsilon_0)}.$$

2. *If* $p_0 := \lambda\{x \in X : \hat{g}(x) = L\}/\lambda(X) > 0$*, which is the case for piecewise-affine* f*, then for Algorithm 1*

$$\mathbb{P}(r_t < L) = (1 - p_0)^t, \qquad L - \mathbb{E}\, r_t \le L\,(1 - p_0)^t.$$

Proof. Positivity of $p(\varepsilon)$: the set $O_\varepsilon = \{x \in U_f : g(x) > L - \varepsilon\}$ is nonempty and open, hence of positive measure, and removing the null set N does not change its measure.

The samples of Algorithm 1 are independent and uniform on X, and $r_t \le L - \varepsilon$ if and only if $\hat{g}(x_i) \le L - \varepsilon$ for all $i \le t$; this gives the first identity in (11). Since $0 \le L - r_t \le L$ almost surely, $\mathbb{E}[L - r_t] = \int_0^L \mathbb{P}(L - r_t \ge \varepsilon)\, d\varepsilon$, which is the second identity.

For Algorithm 2, the samples are independent, with N/K^n of them uniform on each cell s, which gives the product formula. Since the cells have equal volume, $p(\varepsilon) = K^{-n} \sum_s p_s(\varepsilon)$, and the inequality of arithmetic and geometric means gives $\big(\prod_s (1 - p_s(\varepsilon))\big)^{1/K^n} \le 1 - p(\varepsilon)$; raising to the power N yields (12).

Part 1: Using $1 - p \le e^{-p}$ and monotonicity of p,

$$\int_0^L (1 - p(\varepsilon))^t\, d\varepsilon \le \int_0^{\varepsilon_0} e^{-t\kappa\varepsilon^\gamma}\, d\varepsilon + L\, e^{-t\, p(\varepsilon_0)} \le \int_0^{\infty} e^{-t\kappa\varepsilon^\gamma}\, d\varepsilon + L\, e^{-t\, p(\varepsilon_0)},$$

and the substitution $u = t\kappa\varepsilon^\gamma$ evaluates the last integral to $\Gamma(1 + 1/\gamma)(\kappa t)^{-1/\gamma}$.

Part 2: $r_t < L$ if and only if no sample lies in $\{\hat{g} = L\}$, which has relative volume p_0, and $L - r_t \le L\, \mathbf{1}_{\{r_t < L\}}$. For piecewise-affine f, $p_0 > 0$ by the proof of Proposition B.15. □

Remark B.17 (Curse of dimensionality for uniform sampling). If g is C^2 near an interior maximizer $x^\star$ with $g(x^\star) = L$ and negative definite Hessian, then $p(\varepsilon) \ge \kappa\varepsilon^{n/2}$ for small ε, so part 1 of Theorem B.16 gives $L - \mathbb{E}\, r_t = O(t^{-2/n})$, a bound that deteriorates quickly with the input dimension. This is one-sided: one nondegenerate maximizer does not determine the global near-maximal volume. On $X = (-\frac{1}{4}, \frac{5}{4})$ with $g(x) = 1 - x^2(x-1)^4$, the maximizer $x = 0$ is nondegenerate, but the degenerate maximizer $x = 1$ gives $p(\varepsilon) \asymp \varepsilon^{1/4}$, not $\varepsilon^{1/2}$. A two-sided asymptotic $p(\varepsilon) \asymp \varepsilon^{n/2}$, and with it a bias of exact order $t^{-2/n}$, holds when g extends continuously to $\overline{X}$, attains L at finitely many interior points, all nondegenerate, and stays below L by a fixed margin outside their neighborhoods. The exponent γ plays the role of the near-optimality dimension in the analysis of optimistic optimization (Bubeck et al., 2011a; Munos, 2011; Valko et al., 2013). The inequality in (12) shows that uniform stratification never increases the probability of a large error, which agrees with the small differences between Algorithms 1 and 2 in Figure 4.

B.5 A MINIMAX LOWER BOUND FOR ADAPTIVE SAMPLING

We finally address the sense in which non-uniform sampling can, and cannot, improve upon uniform sampling. We consider the largest natural class of sampling schemes and show that on a two-parameter family of ReLU networks with four hidden neurons, no member of the class can find the modulus with fewer samples, up to a constant factor, than Algorithm 1. Improvements of Algorithm 3 over Algorithm 1, such as those in Table 3 and Appendix G, must therefore come from structure of the gradient field of the particular network, and cannot hold uniformly over all networks.

Definition B.18 (Adaptive sampling scheme)**.** An adaptive *black-box point-query* scheme is a sequence of measurable maps π_t, $t \ge 1$, such that $x_t = \pi_t(\mathcal{H}_{t-1}, \xi_t)$, where $\mathcal{H}_{t-1}$ is the history of Definition B.4, possibly enlarged by the values $f(x_i)$ and the full matrices returned by the oracle at x_i for $i < t$, and ξ_t is the internal randomness; the scheme does not read the weights of f, only point queries. The oracle is assumed to be *local* and *Clarke-valued*: its output at x is $\|G\|_{\alpha,\beta}$ for a matrix $G \in J_f^c(x)$ that depends only on the restriction of f to a neighborhood of x.

This class contains every adaptive partition scheme, in particular Algorithms 1–3, as well as deterministic grids, quasi-random sequences, and Bayesian optimization over point queries. It excludes methods that inspect the weights, which in the family below reveal the location of the maximizing slab. For the networks below, J_f^c is a singleton off three hyperplanes, so any Clarke-valued oracle is local there, and this is all the proof uses; AD returns that singleton as well, since the hidden units carrying the perturbation are inactive away from it.

Theorem B.19 (Needle in a haystack)**.** *Let $X = (0, w)^n$, $a > 0$, and $0 < \rho \le w/2$. For $z \in [0, w-\rho]$ define*

$$f_z(x) := a\,\mathrm{relu}(x_1) + a\,h_\rho(x_1 - z), \qquad h_\rho(u) := \mathrm{relu}(u) - 2\,\mathrm{relu}(u - \rho/2) + \mathrm{relu}(u - \rho),$$

a ReLU network with one hidden layer of four neurons. Let $\kappa := \sup_{\|v\|_\alpha \le 1} \|v_1\|_\beta > 0$. Then $L^{(\alpha,\beta)}(f_z, X) = 2a\kappa$ for every z, and $g_z \le a\kappa$ outside the slab $\{x \in X : z < x_1 < z + \rho/2\}$. For every adaptive sampling scheme π and every $t \ge 1$ there is $z \in [0, w - \rho]$ such that

$$\mathbb{P}_{\pi,z}\Big(r_t \le \tfrac{1}{2} L^{(\alpha,\beta)}(f_z, X)\Big) \;\ge\; 1 - \frac{t\rho}{w - \rho}. \tag{13}$$

By contrast, Algorithm 1 satisfies, for every z,

$$\mathbb{P}\big(r_t = L^{(\alpha,\beta)}(f_z, X)\big) \;\ge\; 1 - \Big(1 - \frac{\rho}{2w}\Big)^t \;\ge\; 1 - e^{-t\rho/(2w)}. \tag{14}$$

Consequently, any scheme that achieves relative error below 50% with probability at least $1/2$ for every z needs $t \ge (w - \rho)/(2\rho)$ samples, whereas $t \ge (2w/\rho)\ln 2$ samples suffice for Algorithm 1; the ratio of the two sample sizes is at most $8 \ln 2$.

Proof. On X, the map f_z depends on $x_1 > 0$ only, and $\partial_1 f_z$ equals $2a$ on the slab $S_z := \{z < x_1 < z + \rho/2\}$, equals 0 on $\{z + \rho/2 < x_1 < z + \rho\}$, and equals a elsewhere off the three hyperplanes $x_1 \in \{z, z + \rho/2, z + \rho\}$. The Jacobian is the row vector $\partial_1 f_z\, e_1^\top$, whose induced norm is $|\partial_1 f_z|\,\kappa$. By (8), $L^{(\alpha,\beta)}(f_z, X) = 2a\kappa$, attained exactly on S_z, and $g_z \le a\kappa$ on $U_{f_z} \setminus S_z$. The flat network $f_\flat(x) := a\,\mathrm{relu}(x_1)$ has $g_\flat \equiv a\kappa$ on X, and $f_z = f_\flat$ on the open set $\{x \in X : x_1 \notin [z, z + \rho]\}$.

Fix π and t. Let Z be uniform on $[0, w - \rho]$, independent of $(\xi_i)_{i \ge 1}$. Run π on $f_\flat$ with internal randomness (ξ_i), producing samples $x_1^\flat, x_2^\flat, \ldots$, whose law does not depend on Z, and run π on f_Z with the same (ξ_i), producing $x_1^Z, x_2^Z, \ldots$. Let $\tau := \inf\{i \ge 1 : Z \le x_{i,1}^\flat \le Z + \rho\}$. We claim that $x_i^Z = x_i^\flat$ for all $i \le \tau$. Indeed, $x_1^Z = \pi_1(\xi_1) = x_1^\flat$. If $x_j^Z = x_j^\flat$ for all $j \le i < \tau$, then each $x_j^\flat$ lies in the open set where $f_Z = f_\flat$, so by locality the observations of the two runs at x_j coincide for $j \le i$, and $x_{i+1}^Z = \pi_{i+1}(\mathcal{H}_i, \xi_{i+1}) = x_{i+1}^\flat$. On $\{\tau > t\}$ therefore $r_t^Z = r_t^\flat$. Since $J_{f_\flat}^c(x) = \{a\, e_1^\top\}$ for every $x \in X$, every observation on $f_\flat$ equals $a\kappa$, so $r_t^\flat = a\kappa = \frac{1}{2} L^{(\alpha,\beta)}(f_Z, X)$ surely.

By independence of Z and $(x_i^\flat)_i$, and since an interval of length ρ has $\mathbb{P}(Z \in \cdot)$-measure at most $\rho/(w - \rho)$,

$$\mathbb{P}(\tau \le t) \le \sum_{i=1}^{t} \mathbb{P}\big(x_{i,1}^\flat - \rho \le Z \le x_{i,1}^\flat\big) \le \frac{t\rho}{w - \rho}.$$

Hence $\mathbb{E}_Z\big[\mathbb{P}_{\pi,Z}(r_t \le a\kappa)\big] \ge \mathbb{P}(\tau > t) \ge 1 - t\rho/(w - \rho)$, and some deterministic z satisfies (13).

For (14), Algorithm 1 draws independent uniform samples, each landing in the open slab S_z with probability $\lambda(S_z)/\lambda(X) = \rho/(2w)$, and every sample in S_z observes $2a\kappa$, because $J_{f_z}^c = \{2a\, e_1^\top\}$ there. The sample-size statements follow by solving $t\rho/(w - \rho) \le 1/2$ and $e^{-t\rho/(2w)} \le 1/2$, and $\frac{(2w/\rho)\ln 2}{(w-\rho)/(2\rho)} = 4 \ln 2 \cdot \frac{w}{w-\rho} \le 8 \ln 2$ for $\rho \le w/2$. □

Remark B.20 (Interpretation). Theorem B.19 is the precise form of the optimality statement in Section 4. Within the class of black-box point-query schemes of Definition B.18, uniform sampling is minimax optimal up to the constant $8 \ln 2$ for the worst case over ReLU networks, and no such scheme, including Algorithm 3, can be uniformly better. The theorem is a worst-case statement over the family f_z; it gives no per-instance rate for Algorithm 3 on other networks. On the other hand, Theorem B.16 shows that uniform sampling suffers from a curse of dimensionality governed by the volume profile, and the analyses of optimistic partition-based methods (Munos, 2011; Valko et al., 2013; Carpentier & Valko, 2014) show that hierarchical optimistic sampling improves the exponent when the near-optimal set has favorable structure, such as a small near-optimality dimension, which is not assumed in this paper. This is the regime illustrated in Table 3 and Appendix G. Quantifying the improvement of Algorithm 3 under such structural assumptions, with its specific subdivision schedule and variance-scaled bonus, is left for future work.

C PSEUDOCODE OF ALGORITHM 2

Algorithm 2 is the stratified variant of Algorithm 1 discussed in Section 3: the box D is divided into K^d equal sub-boxes and N/K^d uniform samples are drawn from each. Its exact error law is given in Theorem B.16.

Algorithm 2 Generalized Stochastic Approximation with Uniform Partitioning of the Domain

Input: Norms $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$. Neural network $f(x)$ and its bounded open box domain D, number of samples N, number of divisions per dimensions K.
$r \leftarrow 0$
Partition domain D into subregions $S \leftarrow$ init_subregions(D, K)
for $s \in S$ **do**
  for $1 \leq i \leq N/|S|$ **do**
    Select input point $x \in s$ uniformly at random from subregion $s \subseteq D$.
    Evaluate $f(x)$; compute $\|J_f(x)\|_{\alpha,\beta}$ for a generalized Jacobian corresponding to the chosen notion of local continuity by backpropagation; cf. Example 4.3.
    Set $r \leftarrow \max\{r, \|J_f(x)\|_{\alpha,\beta}\}$
  end for
end for
return r

D EXPERIMENTAL DETAILS

Synthetic data and training (Figures 1–4). Each synthetic dataset places 3 hyperspheres of random radius $0.1 \times r_{domain} \leq r_{sphere} \leq 0.4 \times r_{domain}$ at random centers in the input hypercube; points inside receive values generated at random with mean -1.0, the remainder values with mean 1.0. For each number of hidden layers in the right plot of Figure 4, 30 networks of width 8 with 7-dimensional inputs were trained, each on a new random dataset of 800 points, with torch.nn.MSELoss() as loss, torch.optim.Adam() with learning rate 5×10^{-4}, and 500 epochs. All three algorithms were limited to 60000 evaluations; Algorithm 2 used 2 divisions per dimension (2^7 subregions); Algorithm 3 used $c = 10$, $\sigma_0 = 0$, $n_0 = 10$, and $t_m = 2$. The plot reports the mean and standard deviation of the number of samples needed to reach 5% relative error, over the runs that reached it within the budget; the fraction of runs exceeding the budget is 43% (Algorithm 1), 45% (Algorithm 2), and 35% (Algorithm 3). These conditional means are taken over different subsets of runs for the three algorithms, so they must be read together with the success fractions. For the left plot, Algorithm 2 was run with varying numbers K of divisions per dimension on one trained network [2, 8, 8, 8, 1], with a budget of 2000 samples per run, $\lfloor 2000/K^2 \rfloor$ in each of the K^2 cells (so that the equal-allocation formula of Theorem B.16 applies exactly to the samples used), and 100 runs per value of K.

Visualization. The 2D heat maps of the gradients were generated on a 400^2 grid. The red dot marking the maximum in Figure 1 is within 0.001% of the value computed by LipMIP.

Scalability (Figure 5). The networks are untrained, with hidden layers of width 64. Stochastic approximation and LipSDP were run without parallelization; 5 cores were allocated to the Gurobi engine in LipMIP.

Hardware and code. All experiments were run on a MacBook with an Apple M3 Pro CPU and 18GB of RAM. The code was built on top of the code released by the authors of LipMIP (Jordan & Dimakis, 2020), with the LipSDP implementation of Fazlyab et al. (2019), both MIT licensed.

E SENSITIVITY AND ROBUSTNESS

To study the sensitivity and robustness of the method, we have developed additional experiments in `https://anonymous.4open.science/r/lipssa-code-7B5C`, where the rest of the code can be found too. The experiments were run on a depth-7 neural net with hidden layers 32-wide, trained on the same toy dataset as in the running example (cf. Figure 1).

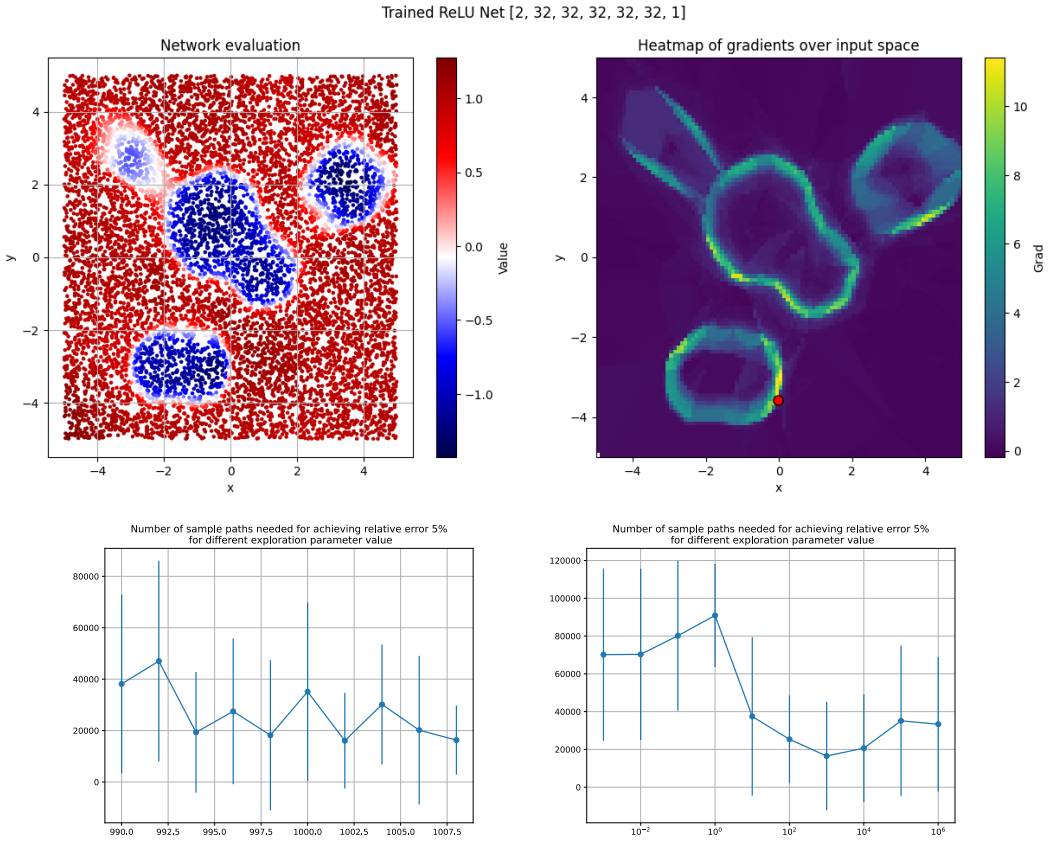


Figure 6: Sensitivity to parameter c of the upper-confidence bound (1) on an example from the running example (cf. Figure 1). Top: the neural network. Bottom left: sensitivity to small perturbations of the optimal parameter c. Bottom right: number of samples required to reach a 5% relative error as a function of the parameter c.

In particular, we focus on hyperparameter c of the upper-confidence bound (1). From Figure 6, it becomes apparent that the optimal value for c in this particular case is quite large, circa 1000. This is perhaps to be expected: it is clear that the model was trained rather hard, possibly even overfitted, which makes the gradient space (plotted on the top right) quite smooth, requiring more exploration. Having said that, it opens up the question of the robustness of the algorithm with respect to the choice of c.

First, for $\sigma_0 > 0$, consistency (Proposition B.13) implies that for any value of c and any relative-error threshold, some number of samples suffices to estimate the modulus within that threshold. For $\sigma_0 = 0$, as used here, this is not guaranteed (Remark B.14), and the following is an empirical observation. For values of c under 10.0, the number of samples required to reach 5% relative-error threshold, sometimes exceeded 100k samples. For values of c in the range from 10 to 1,000,000, the 5% relative-error threshold has reliably been reached within 100k samples.

Second, let us comment on the sensitivity of the number of samples to the parameter c. From the bottom-left subplot in Figure 6, which utilizes a linear scale on the horizontal axis, it is clear that the performance is not particularly sensitive to the perturbation of the exploration parameter c. From the bottom-right subplot in Figure 6, which utilizes a logarithmic scale on the horizontal axis, it is clear that underestimating the optimal value of the exploration parameter c by 3 orders of magnitude can increase the number of samples required to achieve a given relative error substantially. On the other hand, overestimating the optimal value of c even by 3 orders of magnitude is not resulting in major performance degradation, since Algorithm 3 overexplores the input space, similar to Algorithm 1, whose performance is respectable. This asymmetry is the basis of the recommendation in Section 3 to err towards large c.

F Comparison of Algorithms 1 and 3 on trained networks

Table 3: Comparison of Algorithm 1 and Algorithm 3 on nets trained on synthetic data and on binary MNIST: mean $\pm$ standard deviation over 5 seeds, 500000 samples per run.

Setup	Method	Mean Value	Time (s)
Synthetic [3, 32, 32, 32, 1]	Algorithm 1	1.9343 ± 0.0014	36.74 ± 0.14
	Algorithm 3	$\mathbf{1.9370 \pm 0.0000}$	56.71 ± 1.02
Synthetic [4, 64, 64, 64, 64, 1]	Algorithm 1	2.1590 ± 0.0099	42.20 ± 0.05
	Algorithm 3	$\mathbf{2.2219 \pm 0.0019}$	62.59 ± 0.95
Synthetic [8, 64, 64, 64, 64, 1]	Algorithm 1	2.0916 ± 0.0412	42.14 ± 0.69
	Algorithm 3	$\mathbf{2.1223 \pm 0.0396}$	63.42 ± 2.61
Binary MNIST [784, 32, 32, 32, 32, 2]	**Algorithm 1**	$\mathbf{355.0857 \pm 30.8449}$	49.31 ± 0.99
	Algorithm 3	338.6680 ± 22.0273	84.79 ± 1.29

Table 3 compares Algorithms 1 and 3 on low-dimensional and high-dimensional trained neural nets. In the "Synthetic" scenario, the nets were trained to fit points on the function $f(\mathbf{x}) = \sin(\|\mathbf{x}\|_2) + 0.1x_1$, with inputs $\mathbf{x} \in \mathbb{R}^d$ sampled uniformly from a hypercube; the "Binary MNIST" scenario uses a net trained on Binary MNIST. For Algorithm 3, $c = 15$ in Synthetic and $c = 5$ in Binary MNIST, and partition step 2. On the three synthetic nets, Algorithm 3 returns a higher mean value, by 0.1–2.9%, at about 1.5 times the run time; on the 8-dimensional net the difference is within one standard deviation, and we did not compare at equal wall-clock time. Its zero standard deviation on the 3-dimensional net indicates that all five runs located the same affine piece (Proposition B.15). On the MNIST net, with 784 input dimensions, its mean is 4.6% lower, within one standard deviation, at about 1.7 times the run time, consistent with Table 1 in Section 5: with partition step 2 and 500000 samples there are about 19 halvings, so each cell is refined along at most 19 of the 784 coordinates.

G Transformer example (applicability)

LipMIP and LipSDP cannot be applied to architectures with softmax attention and layer normalization. Sampling-based methods apply, since they only require automatic differentiation, and the theory applies as well: softmax and layer normalization are definable in the real exponential field, which is o-minimal (Van den Dries, 1998), so Theorem 4.2 and Appendix B cover this network, our one

experiment outside the semialgebraic setting. On such networks we can only compare Algorithms 1 and 3 with each other. The experiment is illustrative and should be read as a proof of concept rather than as an insight into transformers.

We use a character-level GPT-style transformer in the style of nanoGPT (Karpathy, 2022) (4 layers, 4 attention heads, embedding dimension 64, context length 32; $\approx$ 0.2M parameters) trained on Tiny Shakespeare (Karpathy, 2015). The input $x_0 \in \mathbb{R}^{32\times 64}$ consists of the token and positional embeddings of a 32-character context taken from the validation split. The domain perturbs the first $k = 4$ coordinates of the last token's embedding, $z \in [-1, 1]^4$, and we estimate $\max_z \|\nabla_z \langle c, f(x_0 + z)\rangle\|_1$ for the logits f at the last position with $c = e_0 - e_1$. Both algorithms probed 50 000 samples, and each was run with 5 seeds. For Algorithm 3, the exploration constant $c \in \{1, 5, 15\}$ and the subdivision time multiplier $t_m \in \{2, 1.5\}$ were selected by the mean final value over 5 separate seeds, giving $c = 15$ and $t_m = 1.5$. With these parameters fixed, we then ran a separate evaluation with a different set of 5 seeds; its results are reported in Figure 7 and in Table 2 of Section 5. They show that Algorithm 3 reaches 99% of the best value found in all runs after a median of 3 906 samples, whereas Algorithm 1 needs 26 457 samples and reaches this level in only 4 of 5 runs. The per-sample overhead of Algorithm 3 is 1.1% of the wall-clock time, since the cost of a sample is dominated by the backward pass through the network; the overhead of maintaining the partition thus becomes negligible for larger nets.

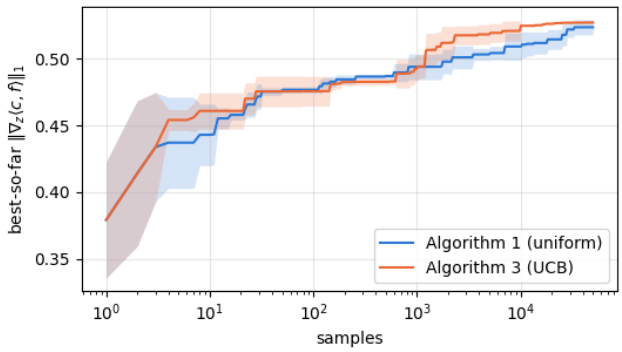


Figure 7: Best-so-far estimate on the transformer, mean $\pm$ std over 5 seeds.