Journal of Machine Learning Research 27 (2026) 1-46 Submitted 6/25; Revised 3/26; Published 5/26

The Sample Complexity of Parameter-Free Stochastic Convex Optimization

Jared Lawrence JPL86@PITT.EDU
Department of Industrial Engineering, University of Pittsburgh
Pittsburgh, PA 15261, USA

Ari Kalinsky AJK245@PITT.EDU
Department of Industrial Engineering, University of Pittsburgh
Pittsburgh, PA 15261, USA

Hannah Bradfield HMB78@PITT.EDU
Department of Industrial Engineering, University of Pittsburgh
Pittsburgh, PA 15261, USA

Yair Carmon YCARMON@TAUEX.TAU.AC.IL
Department of Computer Science, Tel Aviv University
Tel Aviv 6997801, Israel

Oliver Hinder OHINDER@PITT.EDU
Department of Industrial Engineering, University of Pittsburgh
Pittsburgh, PA 15261, USA

Editor: Shiqian Ma

Abstract

We study the sample complexity of stochastic convex optimization when problem parameters such as the distance to optimality and the Lipschitz constant are unknown. We pursue two strategies. First, we develop a reliable model selection method that avoids overfitting to the validation set. This method allows us to generically tune the learning rate of stochastic optimization methods to match the optimal known-parameter sample complexity up to $\log\log$ factors. Second, we develop a regularization-based method that is specialized to the case that only the distance to optimality is unknown. More specifically, it uses norm-regularized empirical risk minimization to estimate the distance to optimality to within a constant factor, allowing known-parameter stochastic optimization methods to achieve optimal sample complexity. This method provides perfect adaptability to unknown distance to optimality, demonstrating a separation between the sample and computational complexity of parameter-free stochastic convex optimization. Combining these two methods allows us to simultaneously adapt to multiple problem structures.

Experiments performing few-shot learning on CIFAR-10 by fine-tuning CLIP models and prompt engineering Gemini to count shapes indicate that our reliable model selection method can help mitigate overfitting to small validation sets.

Keywords: parameter-free optimization, stochastic convex optimization, sample complexity, model selection, avoiding overfitting.

1. Introduction

In machine learning, there is a tension between computational and sample efficiency. If we have unlimited data, then computational efficiency dominates our concerns. For example, when training large language models from scratch, it is typical to only make one pass on the training data with a single set of hyperparameters (Kaplan et al., 2020). On the other hand, if we have limited data, then methods that efficiently use that data are paramount and more computationally intensive approaches become viable. For example, in few-shot learning, it is standard to sweep over many hyperparameters and make multiple data passes (Hu et al., 2022).

The classical theory of stochastic convex optimization assumes that the Lipschitz constant L and distance $R^\star$ from the initial point to the optimum are known. In this setting, there is no trade-off between computational and sample efficiency: stochastic gradient descent (SGD), with step size equal to $R^\star/L\sqrt{T}$, obtains the optimal dimension-free worst-case bounds for sample, gradient oracle, and computational[1] complexity (Agarwal et al., 2012; Nemirovski and Yudin, 1983).

Parameter-free stochastic convex optimization studies the case where problem parameters such as the distance to optimality are unknown. Its computational complexity is almost perfectly understood (McMahan and Orabona, 2014; Cutkosky and Orabona, 2018; Attia and Koren, 2024; Carmon and Hinder, 2024; Khaled and Jin, 2024; Carmon and Hinder, 2022; Jacobsen and Cutkosky, 2022). However, the sample complexity of parameter-free stochastic convex optimization remains to be characterized. In particular, in the canonical setting where we aim to obtain suboptimality guarantees that hold with constant probability and the Lipschitz constant is known but the distance to optimality is unknown, existing lower bounds apply to gradient oracle complexity (i.e., the number of gradient evaluations) but not sample complexity (Carmon and Hinder, 2024, Theorem 2).

A natural approach to addressing the sample complexity of parameter-free optimization is to perform hyperparameter search over different values of the algorithmic parameters to minimize the validation loss. While this is the standard methodology for eliminating hyperparameters, existing literature only provides guarantees for this approach when the loss is bounded (Vapnik, 1999). While some metrics such as accuracy are bounded, other common objectives like cross-entropy loss are not.

Our contributions. In this paper, we

1. Show that *standard model selection*, i.e., minimizing the validation error, can catastrophically overfit (e.g., when tuning learning rates) but enjoys strong guarantees when the population risk is strongly convex.

2. Develop a *generic* reliable model selection method that mitigates this risk of overfitting on small validation sets. The method also obtains the same strong guarantees as standard model selection when population risk is strongly convex. When used for tuning

1. This paper uses the oracle model of computation (Nemirovski and Yudin, 1983). We also include the gradient oracle cost in the computational complexity. Since for SGD the oracle cost, which involves at minimum reading a vector of size of the dimension, is the dominant term in the upper bounds for computational complexity, the oracle lower bounds imply that these computational complexity upper bounds are optimal.

the learning rate, reliable model selection matches the sample complexity of known-parameter stochastic convex optimization up to a $\log\log$ factor in the uncertainty in the distance to optimality.

3. Develop a regularization-based method that matches the sample complexity of *known*-parameter stochastic optimization when the distance to optimality is unknown, removing the $\log\log$ factors present in Contribution 2. The key insight is that norm-regularized empirical risk minimization (ERM) can produce an estimate of the distance to optimality to within a constant factor, which is sufficient for optimal known-parameter stochastic optimization methods to achieve optimal sample complexity. Due to existing lower bounds (Carmon and Hinder, 2024), our result implies a separation between the gradient oracle complexity and sample complexity of parameter-free stochastic optimization. In contrast, no such separation exists for *known*-parameter stochastic optimization. To obtain these results, we develop new concentration inequalities for the sum of i.i.d. vectors and dependent random variables.

4. Combine our regularization-based method with our reliable model selection method to simultaneously adapt to the Euclidean norm, infinity norm, and Manhattan norm for measuring the distance to optimality.

5. Provide few-shot learning and prompt engineering experiments showing that our reliable model selection method can mitigate the risk of overfitting when the validation set is small.

1.1 Related Work

Guarantees for standard model selection. Under the assumption that the noise in function value evaluations is uniformly bounded, Attia and Koren (2024, Theorem 2) provide guarantees on tuning the learning rate of SGD by performing a grid search using standard model selection. However, this assumption fails even for simple setups like logistic regression.

Parameter-free optimization. We survey related results in terms of the Price of Adaptivity (PoA) (Carmon and Hinder, 2024), a concise framework for describing parameter-free bounds. The PoA is the ratio between the suboptimality bound obtained when there is uncertainty in the values of the problem parameters and the optimal worst-case suboptimality with known parameters. In other words, it measures how uncertainty in problem parameters degrades worst-case bounds.

In particular, on L-Lipschitz functions with L known but unknown distance to optimality, the PoA is $O(\sqrt{\ln \rho})$ (McMahan and Orabona, 2014) where ρ is the multiplicative uncertainty in the initial distance to optimality, i.e., the distance to optimality is in the range $[1, \rho]$. Moreover, the sample complexity of this problem matches this upper bound up to constant factors (Carmon and Hinder, 2024). On the other hand, to guarantee the suboptimality is upper bounded with a constant probability, the corresponding PoA is $O(\ln \ln \rho)$ (Carmon and Hinder, 2022). Moreover, a corresponding lower bound shows that the PoA with respect to stochastic gradient oracle complexity cannot be improved beyond $\Omega(\sqrt{\ln \ln \rho})$ (Carmon and Hinder, 2024). Critically for this paper, the latter lower

bound does not preclude the $O(1)$ PoA bound that we show for the sample complexity of parameter-free stochastic optimization (see Theorem 11 and subsequent discussion).

When the Lipschitz constant is also unknown but lies in the range $[1, \ell]$, then the lower bound on the PoA (for constant probability guarantees on the suboptimality) becomes $\Omega(\sqrt{\ln \ln \rho} + \min\{\ell, \rho\}/\sqrt{T})$ (Carmon and Hinder, 2024). This lower bound is matched up to logarithmic factors by a combination of Carmon and Hinder (2022) and Cutkosky (2019).

There are numerous practical implementations of parameter-free stochastic optimization methods (Ivgi et al., 2023; Orabona and Tommasi, 2017; Chen et al., 2022; Kempka et al., 2019; Kreisler et al., 2024; Defazio and Mishchenko, 2023). These papers primarily address large data sets where overfitting is less of a concern. Thus, while our emphasis is on sample efficiency, these papers focus on improving computational efficiency by eliminating the overhead of hyperparameter search.

Cross-validation. The holdout method splits the data into a training set and a validation set. It fits the model on the training set and uses the validation set to choose the final model. A more sophisticated approach is K-fold cross-validation, which evenly splits the data into K folds, and for each fold k, retrains the model on the other $K-1$ folds and holds out the remaining fold. Generalization error is estimated by averaging the error across the K held out folds (Hastie et al., 2009, Chapter 7). Cross-validation requires some form of algorithmic stability to provide theoretical guarantees (Kearns and Ron, 1997), but this is not necessary for the holdout method (see Section 2). In practice, cross-validation is often more sample efficient than the holdout method but because it requires K times more compute, it is not popular for compute-constrained machine learning.

Methods for mitigating overfitting. Blum and Hardt (2015) develop a method for preventing overfitting to leaderboards for bounded objectives such as accuracy. In contrast, our analysis applies to unbounded losses such as cross-entropy. Breiman et al. (1984) proposes the one standard error rule for mitigating overfitting in decision trees. Unfortunately, this rule is restricted to models that can be ordered by a complexity metric and lacks a strong theoretical basis for tuning learning rates.

1.2 Notation

Throughout, we assume $f(x; S)$ is convex in x where S is a random variable with sample space $\mathcal{S}$. For conciseness, we denote $f(x; S_i)$ by $f_i(x)$ where S_i represents the ith sample. Define $F(x) := \mathbb{E}[f(x; S)]$, $F^\star := \inf_{x \in \mathcal{X}} F(x) > -\infty$, $\bar{F}(x) := \frac{1}{n}\sum_{i=1}^n f_i(x)$. Let $\mathcal{X}$ be a closed convex set with $\mathcal{X} \subseteq \mathbb{R}^d$. In most practical cases, $\mathcal{X} = \mathbb{R}^d$. Let $\mathcal{X}^\star := \arg\min_{x \in \mathcal{X}} F(x)$ and $R^\star := \inf_{x \in \mathcal{X}^\star} \|x\|$. If $\mathcal{X}^\star$ is empty, then $R^\star = \infty$. The dual norm is $\|z\|_* = \sup\{z \cdot x : \|x\| \le 1\}$. Let $\boldsymbol{\nabla} h(x)$ be any subgradient of a function $h(x)$ and $\boldsymbol{\nabla}_j h(x)$ be the jth coordinate of $\boldsymbol{\nabla} h(x)$. Define $a \wedge b := \min\{a, b\}$ and $a \vee b := \max\{a, b\}$. For a scalar K, we let $[K] := \{0, 1, \ldots, K\}$ and, with slight abuse of notation, for a vector v we let $[v]_j$ be the jth coordinate of v. Let $\mathbf{0}$ be a vector of zeros. We say the function $h(x)$ is L-Lipschitz if and only if for all $x, x' \in \mathcal{X}$, we have $|h(x) - h(x')| \le L\|x - x'\|$. An equivalent definition is that $\|\boldsymbol{\nabla} h(x)\|_* \le L$ for all $x \in \mathcal{X}$. We say the function f is L-Lipschitz if and only if $h(x) = f(x; S)$ is L-Lipschitz almost surely. We say the function $h : \mathcal{X} \to \mathbb{R}$ is μ-strongly convex if and only if for all $u, v \in \mathcal{X}$ we have $h(u) - h(v) \ge \boldsymbol{\nabla} h(v) \cdot (u - v) + \frac{\mu}{2}\|u - v\|^2$. We say the function f is μ-strongly convex if and only if $h(x) = f(x; S)$ is μ-strongly convex

almost surely. We assume there exists some reference model x_0, a model the user identifies as a reasonable baseline. In the theory (Section 2 and Section 3), for simplicity we set the reference model to be the origin, i.e., $x_0 = \mathbf{0}$. Nonetheless, our results hold for an arbitrary reference model by shifting the origin.

Paper outline. Section 2 analyzes standard model selection and presents our reliable model selection method which can be used for tuning the learning rate in a grid search. Section 3 presents our regularization-based method, which uses norm-regularized ERM to estimate the distance to optimality to within a constant factor, allowing known-parameter stochastic optimization methods to achieve their optimal sample complexity, and combines it with reliable model selection to simultaneously adapt to multiple problem structures. Section 4 provides experiments indicating that our reliable model selection method can mitigate the risk of overfitting.

2. Model Selection Techniques

In this section, we consider the case where there is a finite set of models $x_0, x_1, \ldots, x_K$. In this setting, our goal is to find the best model among these $K + 1$ models. We will also explore applications of these techniques to selecting hyperparameters for stochastic optimization methods.

Section 2.1 studies the performance of standard model selection which chooses the model that minimizes the validation error. In particular, we show that it can effectively tune hyperparameters when strong convexity is present but otherwise may perform poorly compared to parameter-free optimization methods. Section 2.2 studies the performance of a new algorithm called ReliableModelSelection, which obtains good worst-case performance both with and without strong convexity.

2.1 When Does Standard Model Selection Succeed and Fail?

Recall that standard model selection picks the model with the smallest validation error, i.e., $k_{\text{std}} \in \arg\min_{k\in[K]} \bar{F}(x_k)$ where $\bar{F}(x_k) = \frac{1}{n}\sum_{i=1}^{n} f_i(x_k)$. It is well-known that for bounded losses, i.e., $f(x; S) \in [0, 1]$ almost surely for all $x \in \mathcal{X}$, one can show that, using Hoeffding's inequality and a union bound, with probability at least $1 - \delta$ for any $\delta \in (0, 1)$, $F(x_{k_{\text{std}}}) \le \min_{k\in[K]} F(x_k) + \sqrt{2\ln(2(K+1)/\delta)/n}$.

Proposition 1 shows that standard model selection can more effectively optimize hyperparameters when strong convexity is present. For example, if we apply Proposition 1 to (Hazan and Kale, 2014, Theorem 3) with Lipschitz constant L known and set x_k to be the output of their algorithm with strong convexity parameter $\mu_k = \mu_0 e^k$ and if $\mu \in [\mu_0, \mu_K]$ we obtain a suboptimality of at most $O\big(\frac{L^2}{\mu n}\big(\ln\frac{\ln(\mu_K/\mu_0)}{\delta} + \ln\ln n\big)\big)$. Carmon and Hinder (2022, Theorem 4) obtain a similar result with slightly worse sample complexity but a computational complexity that is a log factor better. Adding L to the grid search yields similar guarantees when the Lipschitz constant is also unknown. The proof of Proposition 1 uses strong convexity to argue that solutions, x_k, with better values of the population risk, $F(x)$, are closer to the optimal solution, which due to the assumption that f is Lipschitz ensures $|f(x_k; S) - f(x^\star; S)|$ is small, and this enables higher quality model selection.

Proposition 1 *Suppose that f is L-Lipschitz and F is μ-strongly convex. Then, for all $\delta \in (0,1)$ with probability at least $1-\delta$ we have*

$$F(x_{k_{\text{std}}}) - F^\star \le 2\max\left\{\min_{k\in[K]} F(x_k) - F^\star, \frac{32L^2}{\mu n}\ln\frac{2(K+1)}{\delta}\right\}.$$

Proof Let $\varepsilon_k := F(x_k) - F^\star$, and $k_\star \in \arg\min_{k\in[K]} F(x_k)$. With probability $1-\delta$, for all $k \in [K]$,

$$\left|\varepsilon_k - (\bar{F}(x_k) - \bar{F}(x^\star))\right| \underset{(i)}{\le} 2L\|x_k - x^\star\|\sqrt{\frac{\ln\frac{2(K+1)}{\delta}}{2n}} \underset{(ii)}{\le} 2L\sqrt{\frac{\varepsilon_k \ln\frac{2(K+1)}{\delta}}{\mu n}} \tag{1}$$

where (i) uses a union bound and Hoeffding's inequality (Theorem 12 in Appendix A) with $Z_i = f_i(x_k) - f_i(x^\star)$ and $|Z_i| \le L\|x_k - x^\star\|$ since f is L-Lipschitz, and (ii) uses strong convexity (in particular, Equation (24) in Appendix A). Substituting $k = k_{\text{std}}$ into Equation (1) gives

$$\varepsilon_{k_{\text{std}}} \le \bar{F}(x_{k_{\text{std}}}) - \bar{F}(x^\star) + 2L\sqrt{\frac{\varepsilon_{k_{\text{std}}} \ln\frac{2(K+1)}{\delta}}{\mu n}}. \tag{2}$$

Similarly, for $k = k_\star$ we have

$$\bar{F}(x_{k_\star}) - \bar{F}(x^\star) \le \varepsilon_{k_\star} + 2L\sqrt{\frac{\varepsilon_{k_\star} \ln\frac{2(K+1)}{\delta}}{\mu n}}. \tag{3}$$

Next,

$$\begin{aligned}\varepsilon_{k_{\text{std}}} - \varepsilon_{k_\star} &\underset{(i)}{\le} 2L\sqrt{\frac{\ln\frac{2(K+1)}{\delta}}{\mu n}}\left(\sqrt{\varepsilon_{k_\star}} + \sqrt{\varepsilon_{k_{\text{std}}}}\right) \underset{(ii)}{\le} \sqrt{\frac{32L^2\ln\frac{2(K+1)}{\delta}}{\mu n}\cdot\frac{\varepsilon_{k_{\text{std}}}}{2}} \\ &\le \max\left\{\frac{32L^2\ln\frac{2(K+1)}{\delta}}{\mu n}, \frac{\varepsilon_{k_{\text{std}}}}{2}\right\}\end{aligned}$$

where (i) uses Equation (2), Equation (3) and $\bar{F}(x_{k_{\text{std}}}) \le \bar{F}(x_{k_\star})$, and (ii) uses that $\varepsilon_{k_\star} \le \varepsilon_{k_{\text{std}}}$. Analyzing this inequality in the case that $\frac{32L^2\ln\frac{2(K+1)}{\delta}}{\mu n} \le \frac{\varepsilon_{k_{\text{std}}}}{2}$ and $\frac{32L^2\ln\frac{2(K+1)}{\delta}}{\mu n} > \frac{\varepsilon_{k_{\text{std}}}}{2}$ gives the desired result. ■

Conversely, without strong convexity, standard model selection can perform poorly at tuning learning rates. For a concrete example, consider adaptive SGD (also known as isotropic AdaGrad (Gupta et al., 2017)), which is a scalar variant of AdaGrad (Duchi et al., 2011). While we chose adaptive SGD as an example, similar issues occur for tuning other standard stochastic optimization methods such as regularized ERM or SGD with a fixed step size. The formula for adaptive SGD is

$$u_{t+1} \leftarrow \mathbf{proj}_R\left(u_t - R\frac{\boldsymbol{\nabla} f_{n+t}(u_t)}{\sqrt{\sum_{j=1}^t \|\boldsymbol{\nabla} f_{n+j}(u_j)\|_2^2}}\right),\ \bar{u}_t \leftarrow \frac{1}{t}\sum_{j=1}^t u_j,\ \text{for } t = 1,\dots,n \tag{4}$$

where $u_1 = \mathbf{0}$, $\mathbf{proj}_R(\hat{u}) := \arg\min_{u \in \mathcal{X}: \|u\|_2 \le R} \|u - \hat{u}\|_2$ and $R > 0$ is the learning rate which represents our estimated distance to optimality. We add the constraint $\|u\|_2 \le R$ to ensure the domain is bounded proportionally to the learning rate. To clearly separate the validation set and training set, Equation (4) uses gradient $\boldsymbol{\nabla} f_{n+t}(u_t)$ instead of $\boldsymbol{\nabla} f_t(u_t)$. Thus, we can think of $f_1, \dots, f_n$ as the validation set and $f_{n+1}, \dots, f_{2n}$ as the training set. Denote $\textsc{AdaSGD}(R) = \bar{u}_n$. It is well-known, e.g., Orabona (2019, Theorem 3.9 & 4.14), that assuming f is L-Lipschitz, adaptive SGD starting from $u_1 = \mathbf{0}$, with probability $1 - \delta$ obtains

$$F(\textsc{AdaSGD}(R)) - \min_{x \in \mathcal{X}: \|x\|_2 \le R} F(x) \le O\left(\frac{LR}{\sqrt{n}} \sqrt{\ln 2/\delta}\right) \tag{5}$$

Setting $R = R^\star$ yields the optimal $O(LR^\star \sqrt{\ln 2/\delta}/\sqrt{n})$ suboptimality guarantee. However, this approach relies on $R^\star$ being known.

The standard approach to selecting the learning rate is to run the method across a grid of learning rates and then choose the solution that minimizes the validation loss. Unfortunately, this can lead to poor performance in some situations, as illustrated by Proposition 2. We emphasize that this is not a critique of adaptive SGD, but an example of a broader limitation of standard model selection for hyperparameter tuning in the absence of strong convexity.

Proposition 2 *Assume the validation and training set each consist of $n \ge 3000$ i.i.d. samples. Let $x_k = \textsc{AdaSGD}(\eta_k)$ where $\eta_0, \eta_1, \dots, \eta_K$ are the learning rates we will evaluate. Then, there exists a 1-Lipschitz stochastic convex optimization problem with minimizer at the origin such that with probability at least $1/1000$, $F(x_{k_{\text{std}}}) - F^\star \ge \frac{1}{288\sqrt{n}} \max_{k \in [K]} \eta_k$.*

The proof of Proposition 2 appears in Appendix B. The lower bound is based on the stochastic optimization problem given by $f(x; 0) = |x|$, $f(x; 1) = -x$, and S is a Bernoulli random variable with success probability $q = 1/2 - n^{-0.5}/16$. For this problem, the population objective takes the form

$$F(x) = \begin{cases} -x & x \le 0 \\ \frac{x}{8\sqrt{n}} & x > 0 \end{cases}$$

whereas the validation and training objectives take the form

$$\frac{1}{n}\sum_{i=1}^{n} f_i(x) = \begin{cases} -x & x \le 0 \\ \frac{n - 2\sum_{i=1}^{n} S_i}{n} x & x > 0, \end{cases} \qquad \frac{1}{n}\sum_{i=n+1}^{2n} f_i(x) = \begin{cases} -x & x \le 0 \\ \frac{n - 2\sum_{i=n+1}^{2n} S_i}{n} x & x > 0, \end{cases}$$

respectively. One can show with constant probability that $\sum_{i=n+1}^{2n} S_i \ge n/2 + \Omega(\sqrt{n})$ making the training objective slope negative for $x > 0$. In this case, adaptive gradient descent with learning rate $\max_{k \in [K]} \eta_k$ terminates with $\bar{u}_n \ge \max_{k \in [K]} \eta_k / 36$. Moreover, with constant probability, the validation set will contain $f(x; 1)$ more often than $f(x; 0)$, i.e., $\sum_{i=1}^{n} S_i > n/2$, making the validation objective slope negative for $x > 0$. Consequently $x_{k_{\text{std}}} \ge \max_{k \in [K]} \eta_k / 36$ and thus $F(x_{k_{\text{std}}}) - F^\star = F(x_{k_{\text{std}}}) \ge \frac{1}{288\sqrt{n}} \max_{k \in [K]} \eta_k$.

We emphasize that this lower-bound construction exploits two features of the example: the absence of strong convexity, which permits instability in the training and validation objectives, and the unboundedness of the losses, which allows the suboptimality to grow as the iterates diverge.

In comparison, parameter-free methods (Carmon and Hinder, 2022) in the setting of Proposition 2 (i.e., there is a known Lipschitz constant of 1), can obtain a suboptimality of $O(\ln(\ln(nR^\star/\eta_{\min}))(R^\star+\eta_{\min})/\sqrt{n})$ with constant probability, where $\eta_{\min}$ is a *known* lower bound on $R^\star$. This bound is much less sensitive to uncertainty in the distance to optimality compared with standard model selection. For example, if the distance to optimality is between 1 and ρ then Proposition 2 shows standard model selection will be a factor of $\Omega(\rho)$ worse than if we knew the distance to optimality versus a factor of $O(\ln\ln(n\rho))$ for (Carmon and Hinder, 2022).

2.2 Our Reliable Model Selection Method

Next, we consider how to reliably select high-quality models and avoid the issues that plagued standard model selection for tuning learning rates in Section 2.1. Our method, described in Algorithm 1, requires a list of confidence interval widths $\tau_1, \dots, \tau_K$ that satisfy Condition 1. The algorithm identifies a set of safe models $\mathcal{F}$ over which we minimize the validation error. The algorithmic parameter $\gamma \in [1, \infty)$ controls the size of $\mathcal{F}$, with larger values of γ behaving more like standard model selection. When $\gamma = 1$, the algorithm reduces to minimizing the upper confidence bound on the function value, but this can be overly conservative, heavily favoring models that make predictions similar to the reference model.

Condition 1 *Let $\tau_1, \dots, \tau_K$ be nonnegative scalars such that for some $\delta \in (0,1)$, $\mathbb{P}(\exists k \in \{1, \dots, K\} : \left|F(x_k) - \bar{F}(x_k) - (F(\mathbf{0}) - \bar{F}(\mathbf{0}))\right| > \tau_k) \le \delta$.*

Condition 1 supposes that we have confidence intervals on the closeness of our validation loss to the population loss relative to the reference model, $x_0 = \mathbf{0}$. Depending on the problem at hand, there are multiple ways to generate such confidence intervals. For example, if f is L-Lipschitz and some upper bound $\hat{L} \ge L$ is known, then a result of Maurer and Pontil (2009) (Theorem 14 in Appendix A) with $Z_i = f_i(x_k) - f_i(\mathbf{0})$ gives that with probability at most $\delta_k \in (0,1)$, $\left|F(x_k) - F(\mathbf{0}) - (\bar{F}(x_k) - \bar{F}(\mathbf{0}))\right| > \sqrt{\frac{2\mathbb{V}_k}{n} \ln \frac{4}{\delta_k}} + \frac{14\hat{L}\|x_k\|}{3(n-1)} \ln \frac{4}{\delta_k}$ where

$$\mathbb{V}_k := \frac{1}{n-1} \sum_{i=1}^{n} (f_i(x_k) - f_i(\mathbf{0}) - (\bar{F}(x_k) - \bar{F}(\mathbf{0})))^2$$

is the sample variance of the paired differences $f_i(x_k) - f_i(\mathbf{0})$. Setting $\delta_k = \frac{\delta}{K}$ and applying a union bound shows Condition 1 holds with

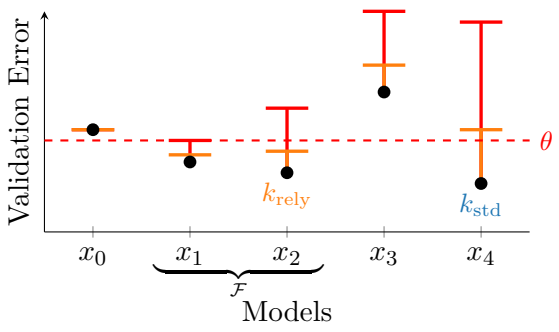
$$\tau_k = \sqrt{\frac{2\tilde{c}\mathbb{V}_k}{n}} + \tilde{c}\frac{14\hat{L}\|x_k\|}{3(n-1)} \text{ where } \tilde{c} := \ln \frac{4K}{\delta}. \tag{6}$$

The following lemma is key to establishing the main guarantees of our algorithm (Theorem 4).

Algorithm 1 Reliable model selection method with illustration

Black dots show the validation errors of candidate models x_0 to x_4. Orange and red intervals depict τ_k and $\gamma\tau_k$, respectively, which are smaller for models with predictions more similar to the reference model, x_0. The threshold θ (red dashed line) is the smallest extended upper bound $\bar{F}(x_k) + \gamma\tau_k$. Models with $\bar{F}(x_k) + \tau_k \le \theta$ form the candidate set $\mathcal{F}$ (in this example $\mathcal{F} = \{1, 2\}$). Among these, x_2 has the lowest error and thus $k_{\text{rely}} = 2$. Standard model selection chooses the lowest validation error across all models, $k_{\text{std}} = 4$.

input A scalar $\gamma \in [1, \infty)$
input Candidate solutions $x_0, \dots, x_K$
input Sample functions $f_1, \dots, f_n$
input $\tau_1, \dots, \tau_K$ satisfying Condition 1.
1: $\tau_0 := 0$
2: $\theta \leftarrow \min_{k \in [K]} \bar{F}(x_k) + \gamma\tau_k$
3: $\mathcal{F} \leftarrow \{k \in [K] : \bar{F}(x_k) + \tau_k \le \theta\}$
4: $k_{\text{rely}} := \arg\min_{k \in \mathcal{F}} \bar{F}(x_k)$
output The chosen index: k_{rely}.



Lemma 3 *With probability at least $1 - \delta$, we have $F(x_{k_{\text{rely}}}) \le F(x_k) + (1 + \gamma)\tau_k$ for all $k \in [K]$.*

Proof For all $k \in [K]$ we have

$$\begin{aligned} 0 &\overset{(i)}{\le} \theta - \bar{F}(x_{k_{\text{rely}}}) - \tau_{k_{\text{rely}}} \overset{(ii)}{\le} \bar{F}(x_k) + \gamma\tau_k - \bar{F}(x_{k_{\text{rely}}}) - \tau_{k_{\text{rely}}} \\ &= \gamma\tau_k + \bar{F}(x_k) - \bar{F}(\mathbf{0}) + \bar{F}(\mathbf{0}) - \bar{F}(x_{k_{\text{rely}}}) - \tau_{k_{\text{rely}}} \overset{(iii)}{\le} (1 + \gamma)\tau_k + F(x_k) - F(x_{k_{\text{rely}}}) \end{aligned}$$

due to (i) $k_{\text{rely}} \in \mathcal{F}$, (ii) the definition of θ, and (iii) Condition 1. ■

Theorem 4 *Suppose that f is L-Lipschitz and τ_k is defined as per Equation (6). Let $\delta \in (0, 1)$ and $\tilde{c} := \ln \frac{4K}{\delta}$. Then, with probability at least $1 - \delta$,*

$$F(x_{k_{\text{rely}}}) \le \min_{k \in [K]} F(x_k) + (1 + \gamma)\left(L\sqrt{\frac{2\tilde{c}}{n-1}} + \frac{14\hat{L}\tilde{c}}{3(n-1)}\right)\|x_k\|. \tag{7}$$

Additionally, if F is μ-strongly convex and $\gamma \ge 3$, then, with probability at least $1 - 2\delta$,

$$F(x_{k_{\text{rely}}}) - F^\star \le 4\left(\min_{k \in [K]} F(x_k) - F^\star\right) + \frac{128}{\mu}\left(2L\sqrt{\frac{2\tilde{c}}{n-1}} + \frac{14\hat{L}\tilde{c}}{3(n-1)}\right)^2. \tag{8}$$

Proof Applying that f is L-Lipschitz to Equation (6) gives

$$\tau_k \le L\|x_k\|\sqrt{\frac{2\tilde{c}}{n-1}} + \frac{14\hat{L}\tilde{c}\|x_k\|}{3(n-1)}.$$

Applying Lemma 3 using this upper bound on τ_k along with a union-bound over the candidates implies Equation (7). It remains to show Equation (8).

Let $\varepsilon_k := F(x_k) - F^\star$, and $k_\star \in \arg\min_{k\in[K]} F(x_k)$. Observe from the definition of $\mathbb{V}_k$ that

$$\begin{aligned}
&\sqrt{\mathbb{V}_{k_\star}} - \sqrt{\mathbb{V}_k} \\
&= \sqrt{\sum_{i=1}^n \frac{(f_i(x_{k_\star}) - \bar{F}(x_{k_\star}) - (f_i(x_0) - \bar{F}(x_0)))^2}{n-1}} - \sqrt{\sum_{i=1}^n \frac{(f_i(x_k) - \bar{F}(x_k) - (f_i(x_0) - \bar{F}(x_0)))^2}{n-1}} \\
&\underset{(i)}{\leq} \sqrt{\sum_{i=1}^n \frac{(f_i(x_{k_\star}) - \bar{F}(x_{k_\star}) - (f_i(x_k) - \bar{F}(x_k)))^2}{n-1}} = \sqrt{\sum_{i=1}^n \frac{(f_i(x_{k_\star}) - f_i(x_k) - (\bar{F}(x_{k_\star}) - \bar{F}(x_k)))^2}{n-1}} \\
&\underset{(ii)}{\leq} 2L\|x_{k_\star} - x_k\|\sqrt{\frac{n}{n-1}}
\end{aligned}$$

where (i) uses the triangle inequality and (ii) uses that f is L-Lipschitz. Thus, by Equation (6),

$$\begin{aligned}
\tau_{k_\star} - \tau_k &= \sqrt{\frac{2\tilde{c}}{n}}\left(\sqrt{\mathbb{V}_{k_\star}} - \sqrt{\mathbb{V}_k}\right) + \frac{14\hat{L}\tilde{c}}{3(n-1)}(\|x_{k_\star}\| - \|x_k\|) \\
&\leq 2L\sqrt{\frac{2\tilde{c}}{n-1}}\|x_{k_\star} - x_k\| + \frac{14\hat{L}\tilde{c}}{3(n-1)}(\|x_{k_\star}\| - \|x_k\|) \\
&\underset{(i)}{\leq} \left(2L\sqrt{\frac{2\tilde{c}}{n-1}} + \frac{14\hat{L}\tilde{c}}{3(n-1)}\right)\|x_{k_\star} - x_k\| \qquad (9)
\end{aligned}$$

where (i) uses the triangle inequality.

Recall that Condition 1 states that, with probability $1-\delta$,

$$\left|F(x_k) - \bar{F}(x_k) - (F(\mathbf{0}) - \bar{F}(\mathbf{0}))\right| \leq \tau_k \quad \forall k \in [K]. \qquad (10)$$

The rest of the proof will proceed in the case that Equation (10) holds.

Let $k' \in \arg\min_{k\in[K]} \bar{F}(x_k) + \gamma\tau_k$. Consider the case that $\bar{F}(x_{k_\star}) + \tau_{k_\star} > \bar{F}(x_{k'}) + \gamma\tau_{k'}$. Then, by Equation (10) with $k = k_\star$ and $k = k'$,

$$\begin{aligned}
0 &< \bar{F}(x_{k_\star}) + \tau_{k_\star} - \bar{F}(x_{k'}) - \gamma\tau_{k'} = \bar{F}(x_{k_\star}) - \bar{F}(\mathbf{0}) + \tau_{k_\star} + \bar{F}(\mathbf{0}) - \bar{F}(x_{k'}) - \gamma\tau_{k'} \\
&\leq F(x_{k_\star}) + 2\tau_{k_\star} - F(x_{k'}) - (\gamma-1)\tau_{k'} = \varepsilon_{k_\star} - \varepsilon_{k'} + 2\tau_{k_\star} - (\gamma-1)\tau_{k'}. \qquad (11)
\end{aligned}$$

Thus,

$$\begin{aligned}
\varepsilon_{k'} - \varepsilon_{k_\star} &\underset{(i)}{\leq} 2\tau_{k_\star} - (\gamma-1)\tau_{k'} \underset{(ii)}{\leq} 2\left(2L\sqrt{\frac{2\tilde{c}}{n-1}} + \tilde{c}\frac{14\hat{L}}{3(n-1)}\right)\|x_{k_\star} - x_{k'}\| - (\gamma-3)\tau_{k'} \\
&\underset{(iii)}{\leq} 2\left(2L\sqrt{\frac{2\tilde{c}}{n-1}} + \tilde{c}\frac{14\hat{L}}{3(n-1)}\right)\left(\sqrt{\frac{2\varepsilon_{k_\star}}{\mu}} + \sqrt{\frac{2\varepsilon_{k'}}{\mu}}\right) \\
&\underset{(iv)}{\leq} 2\left(2L\sqrt{\frac{2\tilde{c}}{n-1}} + \tilde{c}\frac{14\hat{L}}{3(n-1)}\right)2\sqrt{\frac{2\varepsilon_{k'}}{\mu}} = \frac{8}{\sqrt{\mu}}\left(2L\sqrt{\frac{2\tilde{c}}{n-1}} + \tilde{c}\frac{14\hat{L}}{3(n-1)}\right)\sqrt{\frac{\varepsilon_{k'}}{2}} \\
&\underset{(v)}{\leq} \max\left\{\frac{64}{\mu}\left(2L\sqrt{\frac{2\tilde{c}}{n-1}} + \tilde{c}\frac{14\hat{L}}{3(n-1)}\right)^2, \frac{\varepsilon_{k'}}{2}\right\}
\end{aligned}$$

where (i) uses Equation (11), (ii) uses Equation (9), (iii) uses the triangle inequality, Equation (24), and $\gamma \in [3,\infty)$, (iv) uses $\varepsilon_{k_\star} \leq \varepsilon_{k'}$, and (v) uses that $ab \leq \max\{a^2, b^2\}$. Analyzing the latter inequality in the cases where $\frac{\varepsilon_{k'}}{2} > \frac{64}{\mu}\left(2L\sqrt{\frac{2\tilde{c}}{n-1}} + \tilde{c}\frac{14\hat{L}}{3(n-1)}\right)^2$ and $\frac{\varepsilon_{k'}}{2} \leq \frac{64}{\mu}\left(2L\sqrt{\frac{2\tilde{c}}{n-1}} + \tilde{c}\frac{14\hat{L}}{3(n-1)}\right)^2$ yields:

$$\varepsilon_{k'} \leq \max\left\{2\varepsilon_{k_\star}, \varepsilon_{k_\star} + \frac{64}{\mu}\left(2L\sqrt{\frac{2\tilde{c}}{n-1}} + \tilde{c}\frac{14\hat{L}}{3(n-1)}\right)^2\right\}. \tag{12}$$

On the other hand, if $\bar{F}(x_{k_\star}) + \tau_{k_\star} \leq \bar{F}(x_{k'}) + \gamma\tau_{k'}$, i.e., $k_\star \in \mathcal{F}$, then Equation (12) also holds. Thus, it remains to bound $\varepsilon_{k_{\text{rely}}}$ under Equation (12). The proof proceeds exactly as per the proof of Proposition 1 (with k' replacing $k_\star$ and k_{rely} replacing k_{std}) but we include it for completeness. With probability $1-\delta$, for all $k \in [K]$,

$$\left|\varepsilon_k - (\bar{F}(x_k) - \bar{F}(x^\star))\right| \underset{(i)}{\leq} 2L\|x_k - x^\star\|\sqrt{\frac{\ln\frac{2(K+1)}{\delta}}{2n}} \underset{(ii)}{\leq} 2L\sqrt{\frac{\varepsilon_k \ln\frac{4K}{\delta}}{\mu n}} \tag{13}$$

where (i) uses a union bound and Hoeffding's inequality (Theorem 12 in Appendix A) with $Z_i = f_i(x_k) - f_i(x^\star)$ and $|Z_i| \leq L\|x_k - x^\star\|$ by the assumption that f is L-Lipschitz, and (ii) uses Equation (24). Substituting $k = k_{\text{rely}}$ into Equation (13) gives

$$\varepsilon_{k_{\text{rely}}} \leq \bar{F}(x_{k_{\text{rely}}}) - \bar{F}(x^\star) + 2L\sqrt{\frac{\varepsilon_{k_{\text{rely}}} \ln\frac{4K}{\delta}}{\mu n}}. \tag{14}$$

Similarly, for $k = k'$ we have

$$\bar{F}(x_{k'}) - \bar{F}(x^\star) \leq \varepsilon_{k'} + 2L\sqrt{\frac{\varepsilon_{k'} \ln\frac{4K}{\delta}}{\mu n}}. \tag{15}$$

Next,

$$\begin{aligned}\varepsilon_{k_{\text{rely}}} - \varepsilon_{k'} &\underset{(i)}{\leq} 2L\sqrt{\frac{\ln\frac{4K}{\delta}}{\mu n}}\left(\sqrt{\varepsilon_{k'}} + \sqrt{\varepsilon_{k_{\text{rely}}}}\right) \underset{(ii)}{\leq} \sqrt{\frac{32L^2\ln\frac{4K}{\delta}}{\mu n}\cdot\frac{\varepsilon_{k_{\text{rely}}}}{2}} \\ &\leq \max\left\{\frac{32L^2\ln\frac{4K}{\delta}}{\mu n}, \frac{\varepsilon_{k_{\text{rely}}}}{2}\right\}\end{aligned}$$

where (i) uses Equation (14), Equation (15) and $\bar{F}(x_{k_{\text{rely}}}) \leq \bar{F}(x_{k'})$, and (ii) uses that $\varepsilon_{k'} \leq \varepsilon_{k_{\text{rely}}}$. Analyzing this inequality in the case that $\frac{32L^2\ln\frac{4K}{\delta}}{\mu n} \leq \frac{\varepsilon_{k_{\text{rely}}}}{2}$ and $\frac{32L^2\ln\frac{4K}{\delta}}{\mu n} > \frac{\varepsilon_{k_{\text{rely}}}}{2}$ gives

$$\varepsilon_{k_{\text{rely}}} \leq \max\left\{2\varepsilon_{k'}, \varepsilon_{k'} + \frac{32L^2\ln\frac{4K}{\delta}}{\mu n}\right\}. \tag{16}$$

Combining Equation (12) and Equation (16), and taking a union bound over Equation (10) and Equation (13), gives Equation (8). ■

Theorem 4 is the main guarantee for our ReliableModelSelection method. Equation (7) follows immediately from Equation (6) and Lemma 3. To show Equation (8) we establish that there exists a $k \in \mathcal{F}$ with a good objective value, and then we follow the proof of Proposition 1 to show that k_{rely} is almost as good as the best model in $\mathcal{F}$.

We can use Equation (7) to generically obtain sample complexity bounds for parameter-free stochastic optimization. For example, Corollary 5 shows how to obtain such bounds by applying ReliableModelSelection to tune the learning rate for adaptive SGD. To interpret Corollary 5, recall that $f_1, \ldots, f_{2n}$ are i.i.d. samples where $f_1, \ldots, f_n$ is the validation set (i.e., used by ReliableModelSelection) and $f_{n+1}, \ldots f_{2n}$ is the training set, i.e., used by adaptive SGD, see Equation (4).

Corollary 5 *Suppose f is L-Lipschitz with known upper bound $\hat{L} \geq L$, and that $R^\star \in [1, \rho]$ for some known $\rho > 1$. For $k = 1, 2, \ldots, \lceil \ln \rho \rceil$, let $x_k = \text{AdaSGD}(e^k)$ be the output of adaptive SGD, i.e., Equation (4), with learning rate $R_k = e^k$. Then, for any $\delta \in (0, 1)$, Algorithm 1 applied to $x_1, \ldots, x_K$ with $K = \lceil \ln \rho \rceil$ and $\tau_1, \ldots, \tau_K$ defined in Equation (6) achieves, with probability at least $1 - 2\delta$,*

$$F(x_{k_{\text{rely}}}) - F^\star \leq O\left(\sqrt{\ln\frac{\lceil\ln\rho\rceil}{\delta}}\cdot\frac{LR^\star}{\sqrt{n}} + \ln\frac{\lceil\ln\rho\rceil}{\delta}\cdot\frac{\hat{L}R^\star}{n}\right).$$

Proof Since the learning rates $R_k = e^k$ form a geometric grid over $[1, e^{\lceil\ln\rho\rceil}]$, there exists some $k^* \in \{1, \ldots, \lceil\ln\rho\rceil\}$ with $R^\star \leq R_{k^*} \leq e \cdot R^\star$. By Equation (5), for this k^*, with probability $1 - \delta$,

$$F(x_{k^*}) - F^\star \leq O\left(\frac{LR_{k^*}\sqrt{\ln(2/\delta)}}{\sqrt{n}}\right) = O\left(\frac{LR^\star\sqrt{\ln(2/\delta)}}{\sqrt{n}}\right).$$

Applying Equation (7) of Theorem 4 with $k = k^*$ and $\|x_{k^*}\| \le R_{k^*} \le e \cdot R^\star$, and then union-bounding gives that with probability at least $1 - 2\delta$

$$F(x_{k_{\text{rely}}}) - F^\star \le F(x_{k^*}) - F^\star + (1+\gamma)\left(\sqrt{2\tilde{c}}\frac{L\|x_{k^*}\|}{\sqrt{n-1}} + \frac{14\tilde{c}}{3}\cdot\frac{\hat{L}\|x_{k^*}\|}{n-1}\right) \le O\left(\sqrt{\tilde{c}}\frac{LR^\star}{\sqrt{n}} + \tilde{c}\frac{\hat{L}R^\star}{n-1}\right).$$

Since our number of candidates $K = \lceil \ln \rho \rceil$, we have $\tilde{c} = \ln \frac{4\lceil \ln \rho \rceil}{\delta}$. ■

If the upper bound $\hat{L}$ on the Lipschitz constant is within a factor of $\tilde{O}(\sqrt{n})$ of L then the first term dominates and this bound matches the optimal known-parameter sample complexity of $O(LR^\star/\sqrt{n})$ up to log log factors. This sample complexity is slightly better than the sample complexity achieved by Carmon and Hinder (2022) as the log log factors are of lower order. One disadvantage of this approach is that it requires $O(n \ln \rho)$ gradient evaluations thus making its computational complexity worse than Carmon and Hinder (2022). Another disadvantage of this approach is that it requires knowing a range of values that the distance to optimality will lie in, i.e., $[1, \rho]$. This issue will be resolved by our method presented in Section 3.

Finally, Equation (8) shows that Algorithm 1 also automatically adapts to strong convexity, thus preserving the benefits of standard model selection (Proposition 1). However, Proposition 1 is slightly stronger because (i) it does not require an estimate of the Lipschitz constant $\hat{L}$ and (ii) it applies for all $\delta \in (0, 1)$ whereas Algorithm 1 applies for a pre-specified δ.

3. Perfect Adaptivity to Unknown Distance to Optimality

This section develops a method with perfect adaptivity to unknown distance to optimality, matching optimal known-parameter high-probability bounds. Our results hold for p-norms with three values of p: 1, 2, and ∞. While our method requires an estimate of the Lipschitz constant (Assumption 1), we show that for a sample of size n, an estimate within $\tilde{O}(\sqrt{n})$ of the true Lipschitz constant suffices to match optimal known-parameter high-probability bounds.

Assumption 1 (Upper bound on Lipschitz constants) *Suppose that f is differentiable, convex in x, and satisfies $\|\nabla f(x; S)\|_2 \le L \le \hat{L}$ and $|\nabla_j f(x; S)| \le \mathbf{L}_j \le \hat{\mathbf{L}}_j$ almost surely for all $j \le d$ and $x \in \mathcal{X}$, where $\hat{L}$ and $\hat{\mathbf{L}}$ are known.*

Assumption 1 treats f as differentiable only for the sake of simplicity: our results extend to nondifferentiable functions via a smoothing argument, e.g., using the Moreau envelope (Moreau, 1965). Our optimal adaptive method consists of two stages. First, using half our sample, we apply regularized empirical risk minimization (ERM) to find a radius R such that, with high probability, the minimum of F in a ball of radius R is close to its global minimum, and also $R = O(R^\star)$. Second, using the other half of our sample, we apply an off-the-shelf optimal known-parameter stochastic optimization method to approximately minimize F in a ball of radius R. Since R is given to the method, it need not be adaptive to the distance to optimality. Our approach is summarized in Algorithm 2.

Algorithm 2 OptimalAdaptiveMethod

input Sample functions $f_1, \dots, f_{2n}$

input Stochastic optimization algorithm $\mathbf{A}$ mapping a ball radius and n sample functions to an approximate minimizer of F in the ball

1: Compute λ_n from $\Delta_j(x) \coloneqq \frac{1}{n}\sum_{i=1}^n [\nabla f_i(x) - \nabla \bar{F}(x)]_j^2$ (see Table 1)

2: Choose $\hat{x}_{\lambda_n} \in \arg\min_{x\in\mathcal{X}} \frac{1}{n}\sum_{i=1}^n f_i(x) + \lambda_n \|x\|$

3: Set $x_{n,p}^{\text{perfect}} \leftarrow \mathbf{A}(2\|\hat{x}_{\lambda_n}\|; f_{n+1}, \dots, f_{2n})$

output $x_{n,p}^{\text{perfect}}$

Stage 1: Localization via ERM. The key component of our approach is (non-squared) norm regularization. For a given regularization parameter λ_n, consider the following regularized ERM and regularized population risk minimizers:

$$\hat{x}_{\lambda_n} \in \arg\min_{x\in\mathcal{X}} \left\{\bar{F}(x) + \lambda_n\|x\|\right\} \text{ and } x^\star_{\lambda_n} \in \arg\min_{x\in\mathcal{X}} \left\{F(x) + \lambda_n\|x\|\right\}.$$

To obtain our results, it will suffice to choose λ_n such that the following condition holds.

Condition 2 *Let f be differentiable. Suppose $\delta \in (0,1)$ and $\lambda_n > 0$ are (known) constants such that $\mathbb{P}\big(\big\|\nabla \bar{F}(x^\star_\zeta) - \nabla F(x^\star_\zeta)\big\|_* > \lambda_n/2\big) \le \delta$ for $\zeta \in \{\frac{\lambda_n}{3}, 3\lambda_n\}$.*

The following lemma is the key implication of Condition 2.

Lemma 6 *If Condition 2 holds then $\mathbb{P}(\|x^\star_{3\lambda_n}\| \le 2\|\hat{x}_{\lambda_n}\| \le 14\|x^\star_{\lambda_n/3}\|) \ge 1 - 2\delta$.*

Proof The remainder of this proof will assume that

$$\big\|\nabla \bar{F}(x^\star_{\lambda_n/3}) - \nabla F(x^\star_{\lambda_n/3})\big\|_* \le \lambda_n/2 \tag{17a}$$

$$\big\|\nabla \bar{F}(x^\star_{3\lambda_n}) - \nabla F(x^\star_{3\lambda_n})\big\|_* \le \lambda_n/2 \tag{17b}$$

and under this assumption we will show that $\|x^\star_{3\lambda_n}\| \le 2\|\hat{x}_{\lambda_n}\| \le 14\|x^\star_{\lambda_n/3}\|$. Proving this implication suffices to prove the lemma because Condition 2 and a union bound implies Equation (17) holds with probability $1 - 2\delta$.

We will find the following well-known fact useful (Oden and Kikuchi, 1980, Theorem 1.5.1, Section 1.5): if H is convex and differentiable, $\alpha > 0$, and $z \in \arg\min_{x\in\mathcal{X}}\{H(x) + \alpha\|x\|\}$, then for any $y \in \mathcal{X}$,

$$\nabla H(z) \cdot (y - z) \ge \alpha(\|z\| - \|y\|). \tag{18}$$

First, we will show that Equation (17a) implies

$$\|\hat{x}_{\lambda_n}\| \le 7\|x^\star_{\lambda_n/3}\|. \tag{19}$$

Let $q \coloneqq \nabla \bar{F}(x^\star_{\lambda_n/3}) - \nabla F(x^\star_{\lambda_n/3})$. By (i) definition of $\hat{x}_{\lambda_n}$, (ii) convexity of $\bar{F}$, (iii) Equation (18) applied with $H = F$, $z = x^\star_{\lambda_n/3}$, $y = \hat{x}_{\lambda_n}$, and $\alpha = \lambda_n/3$, and (iv) $\|q\|_* \le \lambda_n/2$

from Equation (17a), Hölder's inequality, and the triangle inequality, we have

$$\begin{aligned}
\bar{F}(x^\star_{\lambda_n/3}) + \lambda_n\|x^\star_{\lambda_n/3}\| &\underset{(i)}{\geq} \bar{F}(\hat{x}_{\lambda_n}) + \lambda_n\|\hat{x}_{\lambda_n}\| \\
&\underset{(ii)}{\geq} \bar{F}(x^\star_{\lambda_n/3}) + \boldsymbol{\nabla}\bar{F}(x^\star_{\lambda_n/3})\cdot(\hat{x}_{\lambda_n} - x^\star_{\lambda_n/3}) + \lambda_n\|\hat{x}_{\lambda_n}\| \\
&= \bar{F}(x^\star_{\lambda_n/3}) + \boldsymbol{\nabla}F(x^\star_{\lambda_n/3})\cdot(\hat{x}_{\lambda_n} - x^\star_{\lambda_n/3}) + q\cdot(\hat{x}_{\lambda_n} - x^\star_{\lambda_n/3}) + \lambda_n\|\hat{x}_{\lambda_n}\| \\
&\underset{(iii)}{\geq} \bar{F}(x^\star_{\lambda_n/3}) + \frac{\lambda_n}{3}\left(\|x^\star_{\lambda_n/3}\| - \|\hat{x}_{\lambda_n}\|\right) + q\cdot(\hat{x}_{\lambda_n} - x^\star_{\lambda_n/3}) + \lambda_n\|\hat{x}_{\lambda_n}\| \\
&\underset{(iv)}{\geq} \bar{F}(x^\star_{\lambda_n/3}) + \frac{\lambda_n}{3}(\|x^\star_{\lambda_n/3}\| - \|\hat{x}_{\lambda_n}\|) - \frac{\lambda_n}{2}(\|\hat{x}_{\lambda_n}\| + \|x^\star_{\lambda_n/3}\|) + \lambda_n\|\hat{x}_{\lambda_n}\|.
\end{aligned}$$

Rearranging gives $\|\hat{x}_{\lambda_n}\| \leq 7\|x^\star_{\lambda_n/3}\|$, which implies Equation (19).

The remainder of the proof will show that Equation (17b) implies

$$\|x^\star_{3\lambda_n}\| \leq 2\|\hat{x}_{\lambda_n}\|. \tag{20}$$

Let $s := \boldsymbol{\nabla}\bar{F}(x^\star_{3\lambda_n}) - \boldsymbol{\nabla}F(x^\star_{3\lambda_n})$. By (i) convexity of $\bar{F}$, (ii) Equation (18) applied with $H = F$, $z = x^\star_{3\lambda_n}$, $y = \hat{x}_{\lambda_n}$, and $\alpha = 3\lambda_n$, and (iii) convexity of $\bar{F}$, we get

$$\begin{aligned}
\bar{F}(\hat{x}_{\lambda_n}) + 3\lambda_n\|\hat{x}_{\lambda_n}\| &\underset{(i)}{\geq} \bar{F}(x^\star_{3\lambda_n}) + \boldsymbol{\nabla}\bar{F}(x^\star_{3\lambda_n})\cdot(\hat{x}_{\lambda_n} - x^\star_{3\lambda_n}) + 3\lambda_n\|\hat{x}_{\lambda_n}\| \\
&= \bar{F}(x^\star_{3\lambda_n}) + \boldsymbol{\nabla}F(x^\star_{3\lambda_n})\cdot(\hat{x}_{\lambda_n} - x^\star_{3\lambda_n}) + s\cdot(\hat{x}_{\lambda_n} - x^\star_{3\lambda_n}) + 3\lambda_n\|\hat{x}_{\lambda_n}\| \\
&\underset{(ii)}{\geq} \bar{F}(x^\star_{3\lambda_n}) + 3\lambda_n\|x^\star_{3\lambda_n}\| + s\cdot(\hat{x}_{\lambda_n} - x^\star_{3\lambda_n}) \\
&\underset{(iii)}{\geq} \bar{F}(\hat{x}_{\lambda_n}) + (\boldsymbol{\nabla}\bar{F}(\hat{x}_{\lambda_n}) - s)\cdot(x^\star_{3\lambda_n} - \hat{x}_{\lambda_n}) + 3\lambda_n\|x^\star_{3\lambda_n}\|
\end{aligned}$$

Next, by (i) rearranging the previous display, (ii) Equation (18) applied with $H = \bar{F}$, $z = \hat{x}_{\lambda_n}$, $y = x^\star_{3\lambda_n}$, and $\alpha = \lambda_n$, and (iii) Hölder's inequality, the triangle inequality, and $\|s\|_* \leq \lambda_n/2$ by Equation (17b), we get

$$\begin{aligned}
\|\hat{x}_{\lambda_n}\| &\underset{(i)}{\geq} \|x^\star_{3\lambda_n}\| + \frac{(\boldsymbol{\nabla}\bar{F}(\hat{x}_{\lambda_n}) - s)\cdot(x^\star_{3\lambda_n} - \hat{x}_{\lambda_n})}{3\lambda_n} \\
&\underset{(ii)}{\geq} \|x^\star_{3\lambda_n}\| + \frac{\|\hat{x}_{\lambda_n}\| - \|x^\star_{3\lambda_n}\|}{3} - \frac{s\cdot(x^\star_{3\lambda_n} - \hat{x}_{\lambda_n})}{3\lambda_n} \\
&\underset{(iii)}{\geq} \|x^\star_{3\lambda_n}\| + \frac{\|\hat{x}_{\lambda_n}\| - \|x^\star_{3\lambda_n}\|}{3} - \frac{\|\hat{x}_{\lambda_n}\| + \|x^\star_{3\lambda_n}\|}{6}.
\end{aligned}$$

Rearranging yields $\|x^\star_{3\lambda_n}\| \leq \frac{5}{3}\|\hat{x}_{\lambda_n}\| \leq 2\|\hat{x}_{\lambda_n}\|$, which proves Equation (20). Combining this with Equation (19) proves the claim. ∎

Let us see why the bound $\|x^\star_{3\lambda_n}\| \leq 2\|\hat{x}_{\lambda_n}\| \leq 14\|x^\star_{\lambda_n/3}\|$ establishes that $R = 2\|\hat{x}_{\lambda_n}\|$ is a valid output for the first stage of our method. First, since $\|x^\star_{3\lambda_n}\| \leq R$ we have

$$\min_{x\in\mathcal{X}:\|x\|\leq R} F(x) \leq F(x^\star_{3\lambda_n}) \leq F^\star + 3\lambda_n(\|x^\star\| - \|x^\star_{3\lambda_n}\|) \leq F^\star + 3\lambda_n R^\star, \tag{21}$$

where the second inequality is due to the definition of $x^\star_{3\lambda_n}$ as a minimizer of $x \mapsto F(x) + 3\lambda_n\|x\|$. Second, we have $R \le 14\|x^\star_{\lambda_n/3}\| \le 14R^\star$, implying that $R = O(R^\star)$ as required.

It remains to find values of λ_n such that Condition 2 holds and the optimality gap $3\lambda_n\|x^\star\|$ is sufficiently small. When L is known exactly, this is straightforward: standard concentration inequalities for the sum of bounded vectors (Howard et al., 2020, Corollary 10a) imply that Condition 2 holds for $\lambda_n = O(L\sqrt{\ln(1/\delta)/n})$. However, when we only have bounds on the Lipschitz constant as in Assumption 1, we must estimate λ_n from the empirical gradient variance. The following lemma provides novel empirical vector concentration bounds[2] which establish Condition 2 when substituting $V_i = \boldsymbol{\nabla} f_i(x)$, $\bar{V} = \boldsymbol{\nabla}\bar{F}(x)$, $C_j = \hat{\mathbf{L}}_j$ and $C = \hat{L}$. Table 1 summarizes the resulting choices of λ_n.

Lemma 7 *Let $V_1, \dots, V_n$ be a sequence of i.i.d. random vectors in $\mathbb{R}^d$. Define $\nu := \mathbb{E}[V_i]$, and $\bar{V} := \frac{1}{n}\sum_{i=1}^n V_i$. Then:*

1. *If $\|V_i\|_2 \le C$ almost surely where C is a constant, then for all $\delta \in (0,1)$,*
$$\mathbb{P}\Big(\|\bar{V} - \nu\|_2 > \frac{2\sqrt{\sum_{i=1}^n \|V_i - \bar{V}\|_2^2 \ln\frac{6}{\delta}}}{n} + \frac{10C\ln\frac{6}{\delta}}{n-1}\Big) \le \delta.$$
2. *If $|[V_i]_j| \le C_j$ for all $j \in [d]$ almost surely where C_j is a constant, then for all $\delta \in (0,1)$,*
   (a) $\mathbb{P}\Big(\|\bar{V} - \nu\|_\infty > \max_{j\in[d]} \sqrt{\frac{2\sum_{i=1}^n [V_i - \bar{V}]_j^2 \ln\frac{4d}{\delta}}{n(n-1)}} + \frac{14C_j\ln\frac{4d}{\delta}}{3(n-1)}\Big) \le \delta,$
   (b) $\mathbb{P}\Big(\|\bar{V} - \nu\|_1 > \sum_{j=1}^d \frac{9}{4}\sqrt{\frac{2\sum_{i=1}^n [V_i - \bar{V}]_j^2 \ln\frac{18}{\delta}}{n(n-1)}} + \frac{24C_j\ln\frac{18}{\delta}}{n-1}\Big) \le \delta.$

The proof of Lemma 7 appears in Appendix C.1. Lemma 7.1 applies the ideas of Maurer and Pontil (2009) to replace the population variance in Howard et al. (2020, Corollary 10b) with the sample variance. Lemma 7.2a is a straightforward application of a union bound to standard concentration inequalities. Lemma 7.2b is a novel concentration bound of independent interest. For example, if one coordinate of V_i is a Rademacher random variable and the others are zero almost surely, then Lemma 7.2b shows that with constant probability $\|\bar{V}\|_1 \le O(\sqrt{1/n})$ but Lemma 7.1 only shows $\|\bar{V}\|_2 \le O(\sqrt{1/n})$. Manole and Ramdas (2023, Corollary 23) provides a similar result, but it includes an undesirable dimension-dependence (through the covering number). The proof of Lemma 7.2b hinges on a new lemma which provides a high-probability bound on the sum of *dependent* random variables. In particular, the proof of Lemma 7.2b applies this lemma to $X_j = \big|[\nu - \bar{V}]_j\big|$ and uses standard techniques to bound X_j in terms of the variance of $[V_i]_j$.

Lemma 8 is tighter than the typical union bound approach which incurs an unnecessary logarithmic dependence on d. The crux of standard concentration bounds is upper bounding $\mathbb{E}[e^{t\sum_{j=1}^d X_j}]$ using that $\mathbb{E}[e^{t\sum_{j=1}^d X_j}] = \Pi_{j=1}^d \mathbb{E}[e^{tX_j}]$ where this equality uses independence of $X_1, \dots, X_d$. However, Lemma 8 cannot use this equality because independence is not assumed. The key insight for Lemma 8 is that due to Jensen's inequality: $\mathbb{E}[e^{t\sum_{j=1}^d X_j}] \le \sum_{j=1}^d w_j \mathbb{E}[e^{tX_j/w_j}]$ where w is a carefully chosen vector from the unit simplex. Moreover, since $\mathbb{E}[e^{tX_j/w_j}]$ only involves a single random variable, it is straightforward to bound.

2. We note in passing that Lemma 7 yields that $\mathbb{P}\big(\big\|\boldsymbol{\nabla}\bar{F}(x) - \boldsymbol{\nabla}F(x)\big\|_* > \lambda_n\big) \le \delta$ for *any* fixed $x \in \mathcal{X}$, but our results only require it for the two values used in Condition 2.

Lemma 8 (Dependent-sum lemma) *Let $X_1, \ldots, X_d$ be (possibly dependent) random variables, let a_j, b_j be nonnegative constants for all $j \in [d]$, and let $c \geq 1$. Suppose that for all $j \in [d]$ and $\delta \in (0,1)$, $\mathbb{P}\left(X_j \geq a_j\sqrt{\ln(c/\delta)} + b_j \ln(c/\delta)\right) \leq \delta$. Then, for all $\delta \in (0,1)$, the probability that $\sum_{j=1}^d X_j \geq \frac{9}{4}\sum_{j=1}^d \left(a_j\sqrt{\ln(6c/\delta)} + 2b_j \ln(6c/\delta)\right)$ is at most δ.*

Proof Let t and w_j be nonnegative constants with $\sum_{j=1}^d w_j = 1$ (their exact values will be selected later). For any $t > 0$, Markov's inequality implies that

$$\mathbb{P}\left(\sum_{j=1}^d X_j \geq \gamma\right) = \mathbb{P}\left(\exp\left(t\sum_{j=1}^d X_j\right) \geq \exp(t\gamma)\right) \leq \exp(-t\gamma)\mathbb{E}\left[\exp\left(t\sum_{j=1}^d X_j\right)\right].$$

Since $\sum_{j=1}^d w_j = 1$, by convexity of the exponential function, Jensen's inequality implies that

$$\exp\left(t\sum_{j=1}^d X_j\right) = \exp\left(t\sum_{j=1}^d w_j\frac{X_j}{w_j}\right) \leq \sum_{j=1}^d w_j \exp\left(t\frac{X_j}{w_j}\right).$$

Taking expectations, using the fact that w_j are constant yields

$$\mathbb{E}\left[\exp\left(t\sum_{j=1}^d X_j\right)\right] \leq \sum_{j=1}^d w_j\mathbb{E}\left[\exp\left(\frac{tX_j}{w_j}\right)\right].$$

Thus,

$$\mathbb{P}\left(\sum_{j=1}^d X_j \geq \gamma\right) \leq \exp(-t\gamma)\sum_{j=1}^d w_j\mathbb{E}\left[\exp\left(\frac{tX_j}{w_j}\right)\right]. \tag{22}$$

It remains to bound $\mathbb{E}[\exp(tX_j/w_j)]$. By Lemma 15 (Appendix A) we have

$$\mathbb{P}\left(X_j \geq a_j\sqrt{\ln(c/\delta)} + b_j\ln(c/\delta)\right) \leq \delta \implies \mathbb{P}\left(X_j \geq x\right) \leq c\exp\left(-\frac{x^2}{a_j^2 + 2b_j x}\right).$$

Let $Z_j = \exp(tX_j/w_j)$ then

$$\mathbb{P}(Z_j \geq z_j) = \mathbb{P}(w_j\ln(Z_j)/t \geq w_j\ln(z_j)/t) = \mathbb{P}(X_j \geq w_j\ln(z_j)/t).$$

It follows that

$$\mathbb{E}[Z_j] = \int_0^\infty \mathbb{P}(Z_j \geq z_j)\,dz_j = \int_0^\infty \mathbb{P}(X_j \geq w_j\ln(z_j)/t)\,dz_j \leq 1 + c\int_1^\infty \exp\left(-\frac{\ln(z_j)^2\frac{w_j^2}{t^2}}{a_j^2 + \frac{2b_jw_j}{t}\ln(z_j)}\right)dz_j.$$

Substituting $z_j = \exp(tx_j/w_j)$ gives

$$\mathbb{E}[Z_j] \leq 1 + c\int_0^\infty h(x_j)\,dx_j = 1 + c\left(\int_0^{\hat{x}_j} h(x_j)\,dx_j + \int_{\hat{x}_j}^\infty h(x_j)\,dx_j\right)$$

where $h(x_j) := \frac{t}{w_j}\exp\left(\frac{tx_j}{w_j} - \frac{x_j^2}{a_j^2+2b_jx_j}\right)$ and $\hat{x}_j = \frac{3a_j^2}{2b_j}$. We will now bound each of these terms. First,

$$\int_0^{\hat{x}_j} h(x_j)\,dx_j \le \int_0^{\hat{x}_j} \frac{t}{w_j}\exp\left(\frac{tx_j}{w_j} - \frac{1}{4}\left(\frac{x_j}{a_j}\right)^2\right)dx_j \le 2\sqrt{\pi}\cdot\frac{ta_j}{w_j}\exp\left(\frac{t^2a_j^2}{w_j^2}\right).$$

Second, if

$$\frac{t}{w_j}\cdot\frac{\Gamma}{\tau} \le \frac{1}{2b_j} \text{ and } \tau \in (0, 3\Gamma/4) \tag{23}$$

with τ and Γ to be chosen later to satisfy these requirements, then by (i) the fact that $x_j/(a_j^2+2b_jx_j) = x_j/(2\hat{x}_jb_j/3+2b_jx_j) \ge \frac{1}{2b_j(1/3+1)} = \frac{3}{8b_j}$, (ii) Equation (23), and (iii) using $\tau \in (0, 3\Gamma/4)$ we get

$$\begin{aligned}\int_{\hat{x}_j}^{\infty} h(x_j)\,dx_j &\underset{(i)}{\le} \int_{\hat{x}_j}^{\infty}\frac{t}{w_j}\exp\left(x_j\left(\frac{t}{w_j}-\frac{3}{8b_j}\right)\right)dx_j \underset{(ii)}{\le} \int_{\hat{x}_j}^{\infty}\frac{t}{w_j}\exp\left(\left(\frac{4\tau-3\Gamma}{4\tau}\right)\frac{tx_j}{w_j}\right)dx_j \\ &\le \int_0^{\infty}\frac{t}{w_j}\exp\left(\left(\frac{4\tau-3\Gamma}{4\tau}\right)\frac{tx_j}{w_j}\right)dx_j \underset{(iii)}{=} \frac{4\tau}{3\Gamma-4\tau}.\end{aligned}$$

Combining and substituting into Equation (22) gives

$$\mathbb{P}\left(\sum_{j=1}^d X_j \ge \gamma\right) \le \exp(-\gamma t)\sum_{j=1}^d w_j\mathbb{E}[Z_j]\left(1+\frac{4\tau c}{3\Gamma-4\tau}\right)\exp(-\gamma t) + 2c\sqrt{\pi}\sum_{j=1}^d ta_j\exp\left(\frac{t^2a_j^2}{w_j^2}-\gamma t\right).$$

Setting $\gamma = \Gamma\sum_{j=1}^d a_j\sqrt{\ln(6c/\delta)} + 2b_j\ln(6c/\delta)$ with $\Gamma = \frac{9}{4}$, $t = \tau\frac{\ln(6c/\delta)}{\gamma}$ with $\tau = 1.44$ and $w_j = \Gamma\frac{a_j\sqrt{\ln(6c/\delta)}+2b_j\ln(6c/\delta)}{\gamma}$ ensures we satisfy (23) and $\sum_{j=1}^d w_j = 1$. Substituting these values into our bound on $\mathbb{P}\left(\sum_{j=1}^d X_j \ge \gamma\right)$ gives

$$\begin{aligned}\mathbb{P}\left(\sum_{j=1}^d X_j \ge \gamma\right) &\le \left(1+\frac{4\tau c}{3\Gamma-4\tau}\right)\exp(-\tau\ln(6c/\delta)) + 2c\sqrt{\pi}\frac{\tau}{\Gamma}\sqrt{\ln(6c/\delta)}\exp\left(\left(\frac{\tau^2}{\Gamma^2}-\tau\right)\ln(6c/\delta)\right) \\ &= \left(1+\frac{4\tau c}{3\Gamma-4\tau}\right)\left(\frac{\delta}{6c}\right)^{\tau} + 2c\sqrt{\pi}\frac{\tau}{\Gamma}\sqrt{\ln(6c/\delta)}\left(\frac{\delta}{6c}\right)^{\tau-\frac{\tau^2}{\Gamma^2}} \\ &\le (1+5.82c)\left(\frac{\delta}{6c}\right)^{1.44} + 2.27c\sqrt{\ln(6c/\delta)}\left(\frac{\delta}{6c}\right)^{1.0304} \\ &\le \delta\end{aligned}$$

where the last inequality follows because the preceding right-hand side divided by δ, call it $g_c(\delta)$, is at most 1 on $(0,1)$ for every $c \ge 1$. Term-by-term comparison gives $g_c(\delta) \le g_1(\delta/c)$ as the first-term ratio simplifies to $(1+5.82c)/(6.82c) \le 1$ and the second-term ratio is exactly 1. Since $\delta/c \in (0,1)$, it suffices that $g_1 \le 1$ on $(0,1)$, which holds because g_1 has a unique interior local maximum ≈ 0.93 (at $\delta \approx 5.4\times10^{-7}$) and $g_1(1) \approx 0.996 < 1$. ■

While the values of λ_n calculated in Table 1 facilitate our sample complexity results, they involve a supremum over $\mathcal{X}$ which might be impractical to calculate. An alternative, more practical approach uses our ReliableModelSelection method to grid search over λ_n. This approach still requires no knowledge of the distance to optimality, but increases the sample complexity by a $\log\log$ factor in our uncertainty in the Lipschitz constant; see Appendix C.3 for details.

Stage 2: Constrained stochastic optimization. Having reduced the problem to minimization of $F(x)$ in a ball of known radius R, our next and final step is to find an approximate minimizer in that ball. It is natural to hope that the ERM solution $\hat{x}_{\lambda_n}$, which is in the ball by definition, is a suitable approximate minimizer. Unfortunately, even in the Euclidean case with known Lipschitz constant, the best available high probability guarantees for ERM have an additional $\log n$ factor (Bousquet et al., 2020). Therefore, we consider a generic stochastic optimization algorithm $\mathbf{A}$ that takes in a ball radius $R > 0$ and sample functions[3] $f_{n+1}, \dots, f_{2n}$, and outputs a point $\mathbf{A}(R; f_{n+1}, \dots, f_{2n}) \in \{x \in \mathcal{X} : \|x\| \le R\}$. Assumption 2 parameterizes the approximation guarantee we require of $\mathbf{A}$.

Assumption 2 *The algorithm $\mathbf{A}$ satisfies Assumption 2 with parameter $\phi_n > 0$ if, for a given $\delta > 0$ and all $R > 0$, $\mathbf{A}(R; f_{n+1}, \dots, f_{2n}) \in \{x \in \mathcal{X} : \|x\| \le R\}$ almost surely and*

$$\mathbb{P}\Big(F(\mathbf{A}(R; f_{n+1}, \dots, f_{2n})) - \min_{x \in \mathcal{X}: \|x\| \le R} F(x) \le \phi_n R\Big) \ge 1 - \delta.$$

With Assumption 2, we immediately obtain a generic error bound for Algorithm 2.

Lemma 9 *Let $x_{n,p}^{\text{perfect}}$ be the output of Algorithm 2. If λ_n satisfies Condition 2 and algorithm $\mathbf{A}$ satisfies Assumption 2 with parameter ϕ_n then, with probability at least $1 - 3\delta$, $F(x_{n,p}^{\text{perfect}}) - F^\star \le (14\phi_n + 3\lambda_n)R^\star$ and $\|x_{n,p}^{\text{perfect}}\| \le 14R^\star$.*

Proof Let $R = 2\|\hat{x}_{\lambda_n}\|$. By Lemma 6, this lemma's premise, and a union bound, it follows that, with probability at least $1-3\delta$, both $\|x^\star_{3\lambda_n}\| \le R \le 14\|x^\star_{\lambda_n/3}\|$ and $F(\mathbf{A}(R; f_{n+1}, \dots, f_{2n})) \le \phi_n R + \min_{x \in \mathcal{X}: \|x\| \le R} F(x)$. Upper bounding the latter inequality using (21) and $R \le 14\|x^\star_{\lambda_n/3}\| \le 14R^\star$ yields our desired suboptimality guarantee. The bound $\|x_{n,p}^{\text{perfect}}\| \le 14R^\star$ follows from $\|x_{n,p}^{\text{perfect}}\| \le R$. ■

It remains to instantiate $\mathbf{A}$ and ϕ_n using specific algorithms; we match each p-norm we consider with a different stochastic optimization method, each attaining the minimax optimal high probability bound within a known ball of the corresponding norm, without requiring any knowledge of the Lipschitz constants. For the Euclidean ($p = 2$) case, we use AdaSGD. For $p = 1$ we use entropic mirror descent (i.e., mirror descent with KL divergence) (Beck and Teboulle, 2003; Nemirovski and Yudin, 1983) with adaptive step sizes (Orabona, 2019), which we denote AdaEMD. For $p = \infty$, we use AdaGrad (Duchi et al., 2011; McMahan and Streeter, 2010). Table 1 gives the value of ϕ_n corresponding to

3. Since in Algorithm 2 the radius R depends on the first n sample functions, we apply $\mathbf{A}$ to n additional, fresh samples.

each p, and we provide full details in Appendix C.2, leading to the following result. We focus on $p \in \{1, 2, \infty\}$ because these are the most widely studied geometries in stochastic convex optimization (Beck and Teboulle, 2003; Nemirovski and Yudin, 1983; Duchi et al., 2011), each admitting a minimax-optimal parameter-free algorithm.

p-norm	$\lambda_{n,p}$	$\mathbf{A}_p$	$\phi_{n,p}$
2	$\frac{4\sqrt{\ln\frac{6}{\delta}}}{\sqrt{n}} \sup_{x\in\mathcal{X}} \sqrt{\sum_{j=1}^d \Delta_j(x)} + \frac{20\hat{L}\ln\frac{6}{\delta}}{n-1}$	AdaSGD	$O\Big(\frac{L\sqrt{\ln\frac{1}{\delta}}}{\sqrt{n}}\Big)$
1	$\frac{2\sqrt{2\ln\frac{4d}{\delta}}}{\sqrt{n-1}} \sup_{j\in[d],x\in\mathcal{X}} \sqrt{\Delta_j(x)} + \frac{28\|\hat{\mathbf{L}}\|_\infty \ln\frac{4d}{\delta}}{3(n-1)}$	AdaEMD	$O\Big(\frac{\|\mathbf{L}\|_\infty\sqrt{\ln\frac{d}{\delta}}}{\sqrt{n}}\Big)$
∞	$\frac{9\sqrt{2\ln\frac{18}{\delta}}}{2\sqrt{n-1}} \sup_{x\in\mathcal{X}} \sum_{j=1}^d \sqrt{\Delta_j(x)} + \frac{48\|\hat{\mathbf{L}}\|_1 \ln\frac{18}{\delta}}{n-1}$	AdaGrad	$O\Big(\frac{\|\mathbf{L}\|_1\sqrt{\ln\frac{1}{\delta}}}{\sqrt{n}}\Big)$

Table 1: Settings used in Theorem 10, where rows correspond to different norms (given in the first column) and $\hat{\mathbf{L}}$ is defined in Assumption 1. In the remaining columns, $\lambda_{n,p}$ is the regularization parameter satisfying Condition 2 (for $\Delta_j(x) \coloneqq \frac{1}{n}\sum_{i=1}^n [\nabla f_i(x) - \nabla \bar{F}(x)]_j^2$), algorithm $\mathbf{A}$ is an optimal stochastic optimization method in a given norm ball, and $\phi_{n,p}$ is the error coefficient for which Assumption 2 holds.

Theorem 10 *Suppose that $\delta \in (0,1)$ and Assumption 1 holds. Then, for $p \in \{1,2,\infty\}$, the output $x_{n,p}^{\text{perfect}}$ of Algorithm 2 using $\lambda_{n,p}, \mathbf{A}_p$ and $\phi_{n,p}$ given in Table 1 satisfies, with probability $\geq 1 - 3\delta$,*

$$F(x_{n,p}^{\text{perfect}}) - F^\star \leq (14\phi_{n,p} + 3\lambda_{n,p})R^\star \quad \text{and} \quad \|x_{n,p}^{\text{perfect}}\| \leq 14R^\star.$$

3.1 Simultaneously Adapting to Multiple Problem Structures

Finally, we combine our ReliableModelSelection method (Algorithm 1) with Theorem 10 to simultaneously adapt to multiple problem structures. In particular, we run Algorithm 2 with $p \in \{2,1,\infty\}$ from Table 1 and then use ReliableModelSelection to choose the best output. The proof and algorithm appear in Appendix C.4.

Theorem 11 *Suppose Assumption 1 holds. Additionally, assume that $\delta \in (0, 1/2]$, $\hat{L} \leq L\sqrt{n/\ln(1/\delta)}$ and $\hat{\mathbf{L}}_j \leq \mathbf{L}_j\sqrt{n/\ln(d/\delta)}$ for all $j \leq d$. Then, there exists a parameter-free algorithm that samples $3n$ functions and returns z such that, with probability at least $1-\delta$,*

$$F(z) - F^\star \leq \min_{x^\star\in\mathcal{X}^\star} O\left(\frac{L\|x^\star\|_2\sqrt{\ln\frac{1}{\delta}}}{\sqrt{n}} \bigwedge \frac{\|\mathbf{L}\|_\infty\|x^\star\|_1\sqrt{\ln\frac{d}{\delta}}}{\sqrt{n}} \bigwedge \frac{\|\mathbf{L}\|_1\|x^\star\|_\infty\sqrt{\ln\frac{1}{\delta}}}{\sqrt{n}}\right).$$

Theorem 11 shows that we can—without prior knowledge of $R^\star$—match, up to constant factors, the sample complexity lower bounds for stochastic convex optimization[4] with a

4. The lower bounds for $p \in \{2,\infty\}$ are given by a one-dimensional construction that covers both cases (Carmon and Hinder, 2024, Proposition 1b). The lower bound for the $p = 1$ norm is implied by Carmon

known distance to optimality across three standard geometries. A limitation of Theorem 11 is that it requires an upper bound on the Lipschitz constants that is tight up to a factor of $\tilde{O}(\sqrt{n})$. This is unavoidable due to a lower bound on the sample complexity of parameter-free stochastic optimization (Carmon and Hinder, 2024, Theorem 3).

4. Experiments with ReliableModelSelection

While the primary contribution of this paper is theoretical, we provide two experiments to demonstrate that our ReliableModelSelection method can be useful in practice. This section provides an overview of them; complete details are provided in Appendix E. For both experiments we run our ReliableModelSelection method (Algorithm 1) with $\gamma = 3$ and $\tau_k = \frac{1}{2}(\sqrt{\mathbb{V}_k/n} + M(x_k)/n)$ where $M(x)$ satisfies $|f(x;S) - f(x_0;S)| \leq M(x)$, $\forall S \in \mathcal{S}$, $x \in \mathcal{X}$. This choice of τ_k is inspired by $\tau_k = \sqrt{2\tilde{c}\mathbb{V}_k/n} + 14\tilde{c}M(x_k)/(3n)$ where $\tilde{c} := \ln \frac{4K}{\delta}$ which is a more conservative choice satisfying[5] Condition 1. The function $M(x)$ is tailored to each of our experiments.

Few-shot learning with CIFAR-10 and CLIP. We fine-tune the last layer of a zero-shot CLIP model (Radford et al., 2021) with cross-entropy loss on CIFAR-10 (Krizhevsky, 2009) using AdaGrad with batch size of $\min\{0.5 \times \text{train set size}, 40\}$. We search over hyperparameters in a grid over the number of epochs (10, 20, 30, or 40), the learning rate ($\eta \in \{4^{-10}, 4^{-9}, \ldots, 4^5\}$), and two calibration parameters ($\omega_1 \in \{0.33, 0.66, 1\}$ and $\omega_2 \in \{0.5, 1.0, 2.0\}$) which allow us to consider the combination of weights $\omega_2((1-\omega_1)x + \omega_1 x_0)$ where x is one of the trained model weights. This gives a total of $1 + 4 \times 16 \times 3 \times 3 = 577$ hyperparameter combinations including the zero-shot model. We use the same number of shots for the training and validation sets. We apply the ReliableModelSelection method to minimize the cross-entropy loss on the validation set with $M(x) := 2\|x - x_0\|_{2,\infty}$ where x_k is the weight matrix corresponding to each hyperparameter combination and x_0 is the initial zero-shot model; Appendix D justifies this choice of $M(x)$. The left plot in Figure 1 shows that, when the validation set is small (with 32 shots or fewer), then standard model selection is worse than the zero-shot model, while our method successfully improves upon it; when the validation set is large (64 shots or more), both methods exhibit almost identical performance.

Prompt engineering Gemini. We ask `gemini-1.5-flash-002` (Gemini Team et al., 2024), using 40 different prompts, to count the number of shapes in an image. The task is to select the prompt that yields the lowest absolute error between the LLM-reported and actual number of shapes. We randomly generated 5,000 images containing between 1 and 12 shapes while varying other characteristics such as color, size, and background. We project all predictions to the set $[0, 12]$ and thus set $M(x) := 12$. As the reference model, we used the best, third-best, 20th percentile, median, and worst prompts as evaluated on all 5,000 images. The right plot in Figure 1 shows that for a good choice of reference model, ReliableModelSelection outperforms standard model selection, especially with

and Hinder (2024, Proposition 1b) for $d < 1/\delta$ and by Levy and Duchi (2019, Theorem 3) and Markov's inequality for $d \geq 1/\delta$.

5. By a union bound and Theorem 14 in Appendix A with $Z_i = f_i(x_k) - f_i(x_0)$, $a = -M(x_k)$, and $b = M(x_k)$.

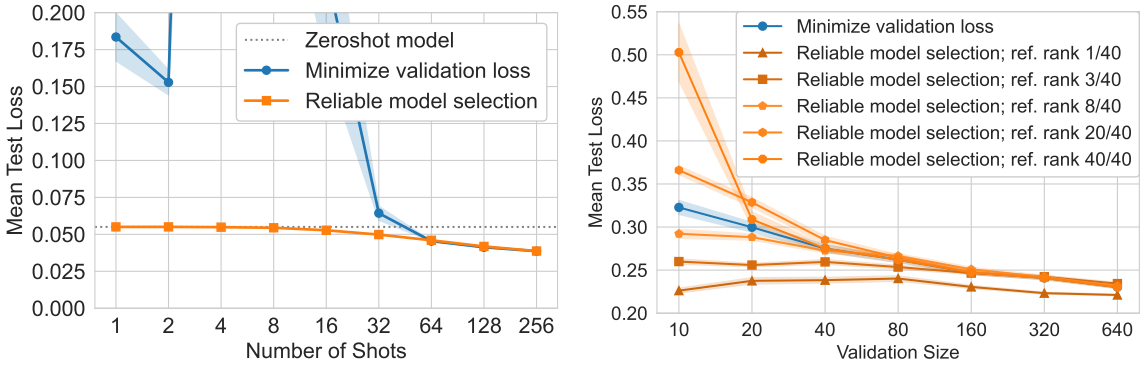


Figure 1: **Left:** CLIP model fine-tuning of the last layer for ViT-L/14 starting from the zero-shot weights on CIFAR-10. **Right:** Prompt engineering a large language model, `gemini-1.5-flash-002`, on the task of counting the number of shapes in images. Both experiments are based on 200 runs, each of which resamples the training and validation sets. Shaded regions represent one standard error.

smaller validation sizes; the reverse holds for a poor reference model. When the validation set is sufficiently large, both methods perform similarly, regardless of the choice of reference model.

Limitations. Our method depends on both knowledge of the problem structure to compute $M(x)$ and a high-quality reference model, which is often not possible. Also, our approach is likely only useful for small validation sets because overfitting is rare for large validation sets (Roelofs et al., 2019; Recht et al., 2019; Yadav and Bottou, 2019; Miller et al., 2020).

Acknowledgments

We thank Aaditya Ramdas, Cristobal Guzman, and John Duchi for helpful input. We also thank Itai Kreisler and Akif Khan for their helpful feedback on the paper.

This work was supported by the NSF-BSF program, under NSF grant #2239527 and BSF grant #2022663. OH acknowledges support from AFOSR grant #FA9550-23-1-0242. YC acknowledges support from the Israeli Science Foundation (ISF) grant no. 2486/21, the Alon Fellowship, and the Adelis Foundation. HB and AK were supported by the Allias/Holzman Undergraduate Research Award from the Department of Industrial Engineering at the University of Pittsburgh. HB was also supported by an REU supplement to NSF grant #2239527. This research was supported in part by the University of Pittsburgh Center for Research Computing and Data, RRID:SCR_022735, through the resources provided. Specifically, this work used the H2P cluster, which is supported by NSF award number OAC-2117681.

Appendix A. Well-Known Results

For Theorem 12, Theorem 13, and Theorem 14, let $Z, Z_1, \ldots, Z_n$ be i.i.d. random variables with values in $[a, b]$ where a and b are constants. Let $\bar{Z}_n = \frac{1}{n}\sum_{i=1}^n Z_i$. Assume $\delta \in (0, 1)$.

Theorem 12 (Hoeffding's inequality (Hoeffding, 1963)) *With probability at least* $1-\delta$*,*

$$\left|\mathbb{E}Z - \bar{Z}_n\right| \leq (b-a)\sqrt{\frac{\ln 2/\delta}{2n}}.$$

Theorem 13 (Bennett's inequality (Bennett, 1962)) *Let* σ^2 *be the variance of* Z*. With probability at least* $1-\delta$*,*

$$\left|\mathbb{E}Z - \bar{Z}_n\right| \leq \sigma\sqrt{\frac{2\ln 2/\delta}{n}} + \frac{(b-a)\ln 2/\delta}{3n}.$$

Theorem 14 (Theorem 4 of Maurer and Pontil (2009)) *With probability at least* $1-\delta$*,*

$$\mathbb{E}Z - \bar{Z}_n \leq \sqrt{\frac{2s_n^2 \ln 2/\delta}{n}} + \frac{7(b-a)\ln 2/\delta}{3(n-1)}$$

with the empirical variance $s_n^2 = \frac{1}{n-1}\sum_{i=1}^n (Z_i - \bar{Z}_n)^2$*.*

Note that by a union bound, Theorem 14 immediately implies that with probability at least $1-\delta$

$$\left|\mathbb{E}Z - \bar{Z}_n\right| \leq \sqrt{\frac{2s_n^2 \ln 4/\delta}{n}} + \frac{7(b-a)\ln 4/\delta}{3(n-1)}.$$

Lemma 15 (Tail Inversion) *Let* X *be a random variable. Let* $a, b > 0$ *and* $c \geq 1$*. For all* $x \geq 0$ *and* $\delta \in (0, 1)$*,*

$$\mathbb{P}\left(X \geq a\sqrt{\ln(c/\delta)} + b\ln(c/\delta)\right) \leq \delta \iff \mathbb{P}(X \geq x) \leq c\exp\left(-\frac{2x^2}{a^2 + 2bx + a\sqrt{a^2+4bx}}\right).$$

Consequently, the following simplified bounds hold

(i) If $\mathbb{P}(X \geq x) \leq c\exp\left(-\frac{x^2}{a^2+bx}\right)$*, then* $\mathbb{P}\left(X \geq a\sqrt{\ln(c/\delta)} + b\ln(c/\delta)\right) \leq \delta$*.*

(ii) If $\mathbb{P}\left(X \geq a\sqrt{\ln(c/\delta)} + b\ln(c/\delta)\right) \leq \delta$*, then* $\mathbb{P}(X \geq x) \leq c\exp\left(-\frac{x^2}{a^2+2bx}\right)$*.*

Proof For $x > 0$, solving the quadratic $x = a\sqrt{\ln(c/\delta)} + b\ln(c/\delta)$ for $\sqrt{\ln(c/\delta)}$ yields

$$\sqrt{\ln(c/\delta)} = \frac{-a + \sqrt{a^2+4bx}}{2b} = \frac{2x}{a + \sqrt{a^2+4bx}}$$

Squaring both sides gives

$$\ln(c/\delta) = \frac{2x^2}{a^2 + 2bx + a\sqrt{a^2+4bx}}$$

Rearranging this directly yields $\delta = c\exp\left(-\frac{2x^2}{a^2+2bx+a\sqrt{a^2+4bx}}\right)$, which establishes the equivalence.

To prove the simplified bounds, we bound the denominator $a^2+2bx+a\sqrt{a^2+4bx}$ inside the exponential. For (i), observing that $a \le \sqrt{a^2+4bx}$ lower-bounds the denominator by $2(a^2+bx)$ yields the result. For (ii), using the inequality $\sqrt{1+z} \le 1+z/2$ for $z = 4bx/a^2$, we have $a\sqrt{a^2+4bx} \le a^2+2bx$. This upper-bounds the denominator by $2(a^2+2bx)$, completing the proof. ■

Lemma 16 *If $F : \mathcal{X} \to \mathbb{R}$ is μ-strongly convex, then*

$$\frac{\mu}{2}\|x-x^\star\|^2 \le F(x) - F(x^\star) \quad \forall x \in \mathcal{X} \tag{24}$$

where $x^\star$ is the unique minimizer of F.

Proof If $F : \mathcal{X} \to \mathbb{R}$ is μ-strongly convex, then

$$\frac{\mu}{2}\|u-v\|^2 + \boldsymbol{\nabla} F(v)\cdot(u-v) \le F(u) - F(v) \quad \forall u, v \in \mathcal{X}.$$

Using $v = x^\star$ and $u = x$ where $x^\star$ is the unique minimizer of F gives the required bound. ■

Appendix B. Proof of Proposition 2

Let $S_1, S_2, \ldots, S_{2n}$ be independent Bernoulli random variables with success probability of $q = \frac{1}{2} - \frac{1}{16\sqrt{n}}$ with

$$\inf_{x\in\mathcal{X}} \mathbb{E}[f(x;S)] \text{ where } f(x;0) = |x|, f(x;1) = -x.$$

We will also let $\eta_{\max} = \max_{k\in[K]} \eta_k$.

Lemma 17 *If $n \ge 3000$ then*

$$\mathbb{P}\left(\sum_{i=1}^n S_i > \frac{n}{2}\right) \ge 0.44 \tag{25a}$$

and

$$\mathbb{P}\left(\sum_{i=1}^n S_i > \frac{n}{2} + \frac{4}{7}\sqrt{2n}\right) \ge 0.03. \tag{25b}$$

Proof Define $Y = \frac{1}{\sigma\sqrt{n}}\sum_{i=1}^n S_i - \frac{q\sqrt{n}}{\sigma}$ where $\sigma^2 = q(1-q)$ is the variance of a Bernoulli random variable. The Berry-Esseen theorem states that

$$|\mathbb{P}(Y \le y) - \Phi(y)| \le \frac{Cm_3}{\sigma^3\sqrt{n}} \tag{26}$$

where Φ is the CDF for the normal distribution, $C = 0.5129$ (Korolev and Shevtsova, 2010), σ is the standard deviation of S_i, and $m_3 = \mathbb{E}[|S_i - q|^3]$. Since $n \geq 3000$ we get

$$\sigma^2 = q(1-q) = \left(\frac{1}{2} - \frac{1}{16\sqrt{n}}\right)\left(\frac{1}{2} + \frac{1}{16\sqrt{n}}\right) = \frac{1}{4} - \frac{1}{256n} \geq 0.4999^2$$

and

$$m_3 = q(1-q)^3 + (1-q)q^3 \leq \max\{q^3, (1-q)^3\} \leq \left(\frac{1}{2} + \frac{1}{16\sqrt{n}}\right)^3 \leq 0.13.$$

It follows that

$$\frac{Cm_3}{\sigma^3\sqrt{n}} \leq \frac{0.5129 \times 0.13}{0.4999^3\sqrt{3000}} \leq 0.01. \tag{27}$$

Thus,

$$\begin{aligned}
\mathbb{P}\left(\sum_{i=1}^n S_i > \frac{n}{2} + \frac{4}{7}\sqrt{2n}\right) &= \mathbb{P}\left(\frac{1}{\sigma\sqrt{n}}\sum_{i=1}^n S_i - \frac{q\sqrt{n}}{\sigma} > \sqrt{n}\frac{\frac{1}{2} - q}{\sigma} + \frac{4\sqrt{2}}{7\sigma}\right) \\
&= \mathbb{P}\left(Y > \frac{1}{16\sigma} + \frac{4\sqrt{2}}{7\sigma}\right) \\
&\geq \mathbb{P}\left(Y > \frac{1}{16 \times 0.4999} + \frac{4\sqrt{2}}{7 \times 0.4999}\right) \\
&= 1 - \mathbb{P}\left(Y \leq \frac{1}{16 \times 0.4999} + \frac{4\sqrt{2}}{7 \times 0.4999}\right) \\
&\geq 1 - \Phi\left(\frac{1}{16 \times 0.4999} + \frac{4\sqrt{2}}{7 \times 0.4999}\right) - \frac{Cm_3}{\sigma^3\sqrt{n}} \geq 0.04 - 0.01 = 0.03
\end{aligned}$$

where the first inequality uses that $\sigma \geq 0.4999$, the second inequality uses Equation (26), and the last inequality uses Equation (27). Similarly,

$$\begin{aligned}
\mathbb{P}\left(\sum_{i=1}^n S_i > \frac{n}{2}\right) = \mathbb{P}\left(Y > \frac{1}{16\sigma}\right) \geq \mathbb{P}\left(Y > \frac{1}{16 \times 0.4999}\right) &= 1 - \mathbb{P}\left(Y \leq \frac{1}{16 \times 0.4999}\right) \\
&\geq 1 - \Phi\left(\frac{1}{16 \times 0.4999}\right) - \frac{Cm_3}{\sigma^3\sqrt{n}} \geq 0.45 - 0.01 \geq 0.44.
\end{aligned}$$

■

Lemma 18 *If $n \geq 3000$ then $\mathbb{P}\left(\frac{\eta_{\max}}{36} < \textsc{AdaSGD}(\eta_{\max})\right) \geq 0.003$.*

Proof Let $W = \sum_{i=n+1}^{2n} S_i$ and denote $\textsc{AdaSGD}(\eta_{\max})$ by $\bar{u}_n$. By Orabona (2019, Theorem 4.14 and Theorem 3.1) we have

$$\mathbb{E}\left[\sum_{i=n+1}^{2n} f_i(\bar{u}_n) - f_i(\eta_{\max}) \,\middle|\, W = w\right] \leq \eta_{\max}\sqrt{2n}.$$

Applying Markov's inequality yields

$$\mathbb{P}\left(\sum_{i=n+1}^{2n} f_i(\bar{u}_n) - f_i(\eta_{\max}) \geq \frac{10}{9}\eta_{\max}\sqrt{2n} \,\middle|\, W = w\right) \leq \frac{9}{10}$$

$$\implies \mathbb{P}\left(\sum_{i=n+1}^{2n} f_i(\bar{u}_n) - f_i(\eta_{\max}) < \frac{10}{9}\eta_{\max}\sqrt{2n} \,\middle|\, W = w\right) \geq \frac{1}{10}.$$

Observe that

$$\sum_{i=n+1}^{2n} f_i(\bar{u}_n) - f_i(\eta_{\max}) = (n-w)|\bar{u}_n| - w\bar{u}_n - (n-w)\eta_{\max} + w\eta_{\max} \geq (n-2w)(\bar{u}_n - \eta_{\max}).$$

Thus, if $2w - n \geq \frac{8}{7}\sqrt{2n} > 0$ then

$$\begin{aligned}
\frac{1}{10} &\leq \mathbb{P}\left(\sum_{i=n+1}^{2n} f_i(\bar{u}_n) - f_i(\eta_{\max}) < \frac{10\eta_{\max}\sqrt{2n}}{9} \,\middle|\, W = w\right) \\
&\leq \mathbb{P}\left((n-2w)(\bar{u}_n - \eta_{\max}) < \frac{10\eta_{\max}\sqrt{2n}}{9} \,\middle|\, W = w\right) \\
&= \mathbb{P}\left(\bar{u}_n > \eta_{\max}\left(1 - \frac{10\sqrt{2n}}{9(2w-n)}\right) \,\middle|\, W = w\right) \leq \mathbb{P}\left(\bar{u}_n > \frac{\eta_{\max}}{36} \,\middle|\, W = w\right).
\end{aligned}$$

Thus, we get

$$\begin{aligned}
\mathbb{P}\left(\bar{u}_n > \frac{\eta_{\max}}{36}\right) &\geq \sum_{w=\lceil \frac{n}{2} + \frac{4}{7}\sqrt{2n} \rceil}^{n} \mathbb{P}\left(\bar{u}_n > \frac{\eta_{\max}}{36} \,\middle|\, W = w\right) \mathbb{P}(W = w) \\
&\geq \frac{1}{10} \sum_{w=\lceil \frac{n}{2} + \frac{4}{7}\sqrt{2n} \rceil}^{n} \mathbb{P}(W = w) \geq \frac{1}{10}\mathbb{P}\left(W > \frac{n}{2} + \frac{4}{7}\sqrt{2n}\right).
\end{aligned}$$

Applying Equation (25b) we get the desired result. ■

Proof [Proof of Proposition 2] If $x \leq 0$ then

$$\frac{1}{n}\sum_{i=1}^{n} f_i(x) = \frac{1}{n}\left(\sum_{i=1}^{n}(1-S_i)|x| - \sum_{i=1}^{n} S_i x\right) = \frac{1}{n}\left(\sum_{i=1}^{n}(S_i - 1)x - \sum_{i=1}^{n} S_i x\right) = -x \geq 0.$$

On the other hand, if $x \geq 0$ then

$$\frac{1}{n}\sum_{i=1}^{n} f_i(x) = \frac{1}{n}\left(\sum_{i=1}^{n}(1-S_i)|x| - \sum_{i=1}^{n} S_i x\right) = \frac{1}{n}\left(\sum_{i=1}^{n}(1-S_i)x - \sum_{i=1}^{n} S_i x\right) = \frac{x}{n}\left(n - 2\sum_{i=1}^{n} S_i\right).$$

Therefore,

$$\sum_{i=1}^{n} S_i > \frac{n}{2} \iff n - 2\sum_{i=1}^{n} S_i < 0 \implies x_{k_{\text{std}}} = \max_{k\in[K]} x_k. \tag{28}$$

Thus

$$\begin{aligned}
\mathbb{P}\left(\frac{\eta_{\max}}{36} < x_{k_{\text{std}}}\right) &\geq \mathbb{P}\left(\frac{\eta_{\max}}{36} < \textsc{AdaSGD}(\eta_{\max}), x_{k_{\text{std}}} = \max_{k\in[K]} x_k\right) \\
&\underset{(i)}{\geq} \mathbb{P}\left(\frac{\eta_{\max}}{36} < \textsc{AdaSGD}(\eta_{\max}), \sum_{i=1}^{n} S_i > \frac{n}{2}\right) \\
&\underset{(ii)}{\geq} \mathbb{P}\left(\frac{\eta_{\max}}{36} < \textsc{AdaSGD}(\eta_{\max})\right)\mathbb{P}\left(\sum_{i=1}^{n} S_i > \frac{n}{2}\right) \underset{(iii)}{\geq} 0.44 \times 0.003 \\
&\geq \frac{1}{1000}.
\end{aligned}$$

where (i) uses (28), (ii) uses that the training set and validation set are independent, and (iii) uses Equation (25a) and Lemma 18. Moreover, if $x_{k_{\text{std}}} > \frac{\eta_{\max}}{36}$ then since by definition $F(x) = (1-q)|x| - qx$ and $q = \frac{1}{2} - \frac{1}{16\sqrt{n}}$ we get

$$\begin{aligned}
F(x_{k_{\text{std}}}) &= (1-q)|x_{k_{\text{std}}}| - qx_{k_{\text{std}}} \geq (1-q)x_{k_{\text{std}}} - qx_{k_{\text{std}}} = (1-2q)x_{k_{\text{std}}} \\
&= \frac{x_{k_{\text{std}}}}{8\sqrt{n}} \geq \frac{\eta_{\max}}{288\sqrt{n}}\ .
\end{aligned}$$

■

Appendix C. Supplementary Material for Section 3

This appendix contains supplementary details for Section 3, particularly establishing the concentration inequalities in Lemma 7, providing more details on Table 1, grid searching over λ_n, and providing the proof of Theorem 11 and the associated algorithm.

C.1 Proof of Lemma 7

C.1.1 Proof of Lemma 7.1

Lemma 19 *Let $V_1, \ldots, V_n$ be a sequence of i.i.d. random vectors in $\mathbb{R}^d$ and let $\nu := \mathbb{E}[V_i]$. Let $H > 0$ be a constant. If $\|V_i - \nu\|_2 \leq H$ almost surely, then for all $n \in \mathbb{N}$,*

$$\mathbb{P}\left(\left\|\frac{1}{n}\sum_{i=1}^{n}(V_i - \nu)\right\|_2 \geq \sqrt{\frac{2\ln(2/\delta)\mathbb{E}\|V_i - \nu\|_2^2}{n}} + \frac{2H\ln(2/\delta)}{3n}\right) \leq \delta$$

Proof This result follows from Corollary 10b of (Howard et al., 2020) setting $Y_t = \frac{1}{n}\sum_{i=1}^{t}(V_i - \nu)$, $\Psi(\cdot) = \|\cdot\|_2$, and $c = H/n$. The selection of $\Psi(\cdot) = \|\cdot\|_2$ yields $D_\star = 1$. Set $m = \mathbb{E}\|V_i -$

$\nu\|_2^2/n = \sum_{i=1}^n \mathbb{E}\left\|(V_i-\nu)/n\right\|_2^2 \geq \sum_{i=1}^t \mathbb{E}\left\|(V_i-\nu)/n\right\|_2^2 = \sum_{i=1}^t \mathbb{E}_{i-1}\left\|(V_i-\nu)/n\right\|_2^2 = V_t$ for all $t \leq n$, then $D^2_\star \mathfrak{s}_P\left(\frac{z}{m}\right)\cdot(V_t - m) \leq 0$. Therefore, for any x we have

$$\mathbb{P}\left(\max_{t\leq n}\left\|\frac{1}{n}\sum_{i=1}^t (V_i-\nu)\right\|_2 \geq x\right) \leq 2\exp\left(-\frac{x^2}{2m+(2/3n)Hx}\right).$$

By Lemma 15 with $a=\sqrt{2m}$, $b=(2/3n)H$, and $c=2$, the following statement holds,

$$\mathbb{P}\left(\max_{t\leq n}\left\|\frac{1}{n}\sum_{i=1}^t (V_i-\nu)\right\|_2 \geq \sqrt{2m\ln(2/\delta)} + \frac{2H\ln(2/\delta)}{3n}\right) \leq \delta.$$

■

Lemma 20 *Let $Z, Z_1, \ldots, Z_n$ be i.i.d. random variables in $[0,H]$ almost surely where $H>0$ is constant. Then with probability at least $1-\delta$, $\mathbb{E}Z \leq \frac{2}{n}\sum_{i=1}^n Z_i + \frac{13H\ln(2/\delta)}{3(n-1)}$.*

Proof We can bound the sample variance using that (i) $\bar{Z}$ minimizes the empirical sum of squares where $\bar{Z} = \frac{1}{n}\sum_{i=1}^n Z_i$ and (ii) $Z^2 \leq HZ$,

$$\frac{1}{n-1}\sum_{i=1}^n \left(Z_i - \bar{Z}\right)^2 \underset{(i)}{\leq} \frac{1}{n-1}\sum_{i=1}^n Z_i^2 \underset{(ii)}{\leq} \frac{H}{n-1}\sum_{i=1}^n Z_i.$$

Combining this with Theorem 14 yields that with probability at least $1-\delta$

$$\mathbb{E}Z \leq \frac{1}{n}\sum_{i=1}^n Z_i + \sqrt{\frac{2H\ln(2/\delta)\sum_{i=1}^n Z_i}{n(n-1)}} + \frac{(7/3)H\ln(2/\delta)}{n-1}. \tag{29}$$

We will simplify Equation (29) by analyzing two cases. If $\frac{1}{n}\sum_{i=1}^n Z_i \leq \frac{2H\ln(2/\delta)}{n-1}$, then Equation (29) becomes

$$\mathbb{E}Z \leq \frac{1}{n}\sum_{i=1}^n Z_i + \frac{13H\ln(2/\delta)}{3(n-1)}.$$

If $\frac{1}{n}\sum_{i=1}^n Z_i > \frac{2H\ln(2/\delta)}{n-1}$, then Equation (29) becomes

$$\mathbb{E}Z < \frac{1}{n}\sum_{i=1}^n Z_i + \sqrt{\frac{1}{n^2}\left(\sum_{i=1}^n Z_i\right)^2} + \frac{(7/3)H\ln(2/\delta)}{n-1} = \frac{2}{n}\sum_{i=1}^n Z_i + \frac{(7/3)H\ln(2/\delta)}{n-1}.$$

Combining these two cases yields our desired inequality. ■

Lemma 21 *Let $V_1, \ldots, V_n$ be a sequence of i.i.d. random vectors in $\mathbb{R}^d$ and let $\nu := \mathbb{E}[V_i]$, and $\bar{V} := \frac{1}{n}\sum_{i=1}^n V_i$ for $n \geq 1$. Let $H>0$ be a constant. If $\|V_i - \nu\|_2 \leq H$ almost surely, then for all $n \in \mathbb{N}$, with probability $1-\delta$, $\sum_{i=1}^n \|V_i-\nu\|_2^2 \leq \sum_{i=1}^n \|V_i - \bar{V}\|_2^2 + 2H^2\ln(2/\delta)$.*

Proof First, note that by adding and subtracting $\bar{V}$ and getting that the summation of the cross term is 0,

$$\begin{aligned}\sum_{i=1}^{n}\|V_i-\nu\|_2^2 &= \sum_{i=1}^{n}\|V_i-\bar{V}+\bar{V}-\nu\|_2^2 = \sum_{i=1}^{n}(\|V_i-\bar{V}\|_2^2+2(V_i-\bar{V})^\top(\bar{V}-\nu)+\|\bar{V}-\nu\|_2^2)\\ &= n\|\bar{V}-\nu\|_2^2+\sum_{i=1}^{n}\|V_i-\bar{V}\|_2^2+2(\bar{V}-\nu)\left(-n\bar{V}+\sum_{i=1}^{n}V_i\right)\\ &= n\|\bar{V}-\nu\|_2^2+\sum_{i=1}^{n}\|V_i-\bar{V}\|_2^2. \end{aligned} \tag{30}$$

Then by Corollary 10a of Howard et al. (2020) with $Y_t = \frac{1}{n}\sum_{i=1}^{t}(V_i-\nu)$, $c_t = H/n$, $\Psi(\cdot) = \|\cdot\|_2$, $D=1$, $m = H^2/n$, $x = \sqrt{(m/2)\ln 2/\delta}$, we get that with probability $1-\delta$ that

$$\|\bar{V}-\nu\|_2 \le H\sqrt{\frac{2\ln(2/\delta)}{n}}. \tag{31}$$

Combining Equation (31) and Equation (30) gives

$$\sum_{i=1}^{n}\|V_i-\nu\|_2^2 = n\|\bar{V}-\nu\|_2^2+\sum_{i=1}^{n}\|V_i-\bar{V}\|_2^2 \le 2H^2\ln(2/\delta)+\sum_{i=1}^{n}\|V_i-\bar{V}\|_2^2$$

as desired. ■

Proof [Proof of Lemma 7.1] Using Lemma 19 with $H = 2C$, as Lemma 7.1 assumes $\|V_i\|_2 \le C$ so $\|V_i-\nu\|_2 \le 2C$, implies that with probability at most $\delta/3$ that

$$\left\|\bar{V}-\nu\right\|_2 \ge \sqrt{\frac{2\ln(6/\delta)\mathbb{E}\|V_i-\nu\|_2^2}{n}}+\frac{4C\ln(6/\delta)}{3n}.$$

By Lemma 20, with $Z_i = \|V_i-\nu\|_2^2$, and therefore $H = 4C^2$, we get that with probability at most $\delta/3$ that

$$\mathbb{E}\|V_i-\nu\|_2^2 > \frac{2}{n}\sum_{i=1}^{n}\|V_i-\nu\|_2^2+\frac{52C^2\ln(6/\delta)}{3(n-1)}.$$

Lemma 21 implies that with probability at most $\delta/3$ that

$$\sum_{i=1}^{n}\|V_i-\nu\|_2^2 > \sum_{i=1}^{n}\|V_i-\bar{V}\|_2^2+8C^2\ln(6/\delta).$$

Applying a union bound to these three statements, all three complementary inequalities hold simultaneously with probability at least $1-\delta$. Chaining these bounds together, and

using the triangle inequality, we obtain with probability at least $1-\delta$ that

$$\begin{aligned}
\left\|\bar{V}-\nu\right\|_2 &\leq \sqrt{\frac{2\ln(6/\delta)\mathbb{E}\|V_i-\nu\|_2^2}{n}}+\frac{4C\ln(6/\delta)}{3n} \\
&\leq \sqrt{\frac{2\ln(6/\delta)}{n}\cdot\left(\frac{2}{n}\sum_{i=1}^{n}\|V_i-\nu\|_2^2+\frac{52C^2\ln(6/\delta)}{3(n-1)}\right)}+\frac{4C\ln(6/\delta)}{3n} \\
&\leq \sqrt{\frac{2\ln(6/\delta)}{n}\cdot\left(\frac{2}{n}\left(\sum_{i=1}^{n}\|V_i-\bar{V}\|_2^2+8C^2\ln(6/\delta)\right)+\frac{52C^2\ln(6/\delta)}{3(n-1)}\right)}+\frac{4C\ln(6/\delta)}{3n} \\
&\leq \frac{2\sqrt{\ln(6/\delta)\sum_{i=1}^{n}\|V_i-\bar{V}\|_2^2}}{n}+\frac{10C\ln(6/\delta)}{n-1}
\end{aligned}$$

as desired. ■

C.1.2 Proof of Lemma 7.2a

Proof The result follows by Theorem 14 with $a=-C_j$, $b=C_j$, $Z_i=[V_i]_j$ for each $j\in[d]$, and then applying a union bound. ■

C.1.3 Proof of Lemma 7.2b

Proof For each coordinate $j\in[d]$ let the sample variance be $s_j^2:=\frac{1}{n-1}\sum_{i=1}^{n}[V_i-\bar{V}]_j^2$ and the population variance be $\sigma_j^2:=\frac{1}{n}\sum_{i=1}^{n}[V_i-\mathbb{E}[V_i]]_j^2$. Theorem 13 with $a=-C_j$, $b=C_j$, $Z_i=[V_i]_j$ for each $j\in[d]$ implies that

$$\mathbb{P}\left(\left|[\nu-\bar{V}_n]_j\right|>\sigma_j\sqrt{\frac{2\ln 2/\delta}{n}}+\frac{2C_j\ln 2/\delta}{3n}\right)\leq\delta.$$

Applying Lemma 8 with $X_j=\left|[\nu-\bar{V}_n]_j\right|$ to this inequality implies that

$$\mathbb{P}\left(\sum_{j=1}^{d}\left|[\nu-\bar{V}_n]_j\right|>\frac{9}{4}\sum_{j=1}^{d}\sigma_j\sqrt{\frac{2\ln 12/\delta}{n}}+\frac{4C_j\ln 12/\delta}{3n}\right)\leq\delta. \tag{32}$$

Equation (3) of Maurer and Pontil (2009) implies that

$$\mathbb{P}\left(\sigma_j-s_j>2C_j\sqrt{\frac{2\ln(1/\delta)}{n-1}}\right)\leq\delta.$$

This inequality implies that, for some fixed $\varepsilon>0$,

$$\mathbb{P}\left(\sigma_j-s_j-\varepsilon\geq 2C_j\sqrt{\frac{2\ln(1/\delta)}{n-1}}\right)\leq\delta.$$

Applying Lemma 8 to this inequality with $X_j = \sigma_j - s_j - \varepsilon$ implies that

$$\mathbb{P}\left(\sum_{j=1}^{d} \sigma_j - s_j > \frac{9}{2}\sqrt{\frac{2\ln(6/\delta)}{n-1}}\sum_{j=1}^{d} C_j\right) \leq \delta. \tag{33}$$

Taking a union bound over Equation (32) with $\frac{2\delta}{3}$ and Equation (33) with $\frac{\delta}{3}$ gives the result. ■

C.2 Explanation of Table 1

Note that the λ_n values are found by substituting $V_i = \boldsymbol{\nabla} f_i(x)$ into Lemma 7 and employing Assumption 1.

The remainder of this section defines AdaSGD$(R; f_{n+1}, \ldots, f_{2n})$, AdaEMD$(R; f_{n+1}, \ldots, f_{2n})$ and AdaGrad$(R; f_{n+1}, \ldots, f_{2n})$, and recaps their known complexities. In particular, AdaSGD$(R; f_{n+1}, \ldots, f_{2n}) = \bar{u}_n$ where $\bar{u}_n$ is computed from Equation (4) starting from $u_1 = \mathbf{0}$. Orabona (2019, Theorem 3.9 & 4.14) prove that with probability at least $1 - \delta$

$$F(\text{AdaSGD}(R; f_{n+1}, \ldots, f_{2n})) - \min_{x \in \mathcal{X}: \|x\|_2 \leq R} F(x) \leq O\left(\frac{LR}{\sqrt{n}}\sqrt{\ln 1/\delta}\right). \tag{34}$$

Next, we consider AdaEMD$(R; f_{n+1}, \ldots, f_{2n})$. Without loss of generality assume $\mathcal{X} \subseteq \{x \in \mathbb{R}^d : x \geq \mathbf{0}\}$, since we can always decompose $x = x_+ - x_-$ where each component of x_+ and x_- is greater than or equal to zero. The update equations for mirror descent starting from $u_1 = \mathbf{0}$ for $t = 1, \ldots, n$ are

$$\begin{aligned}
\bar{u}_t &\leftarrow \frac{1}{t}\sum_{l=1}^{t} u_l \\
u_{t+1} &\in \operatorname*{arg\,min}_{u \in \mathcal{X}: \|u\|_1 \leq R} \left\{\eta_t \boldsymbol{\nabla} f_{n+t}(u_t)^\top u + D_\Phi(u, u_t)\right\} \\
\eta_t &\leftarrow \frac{R\sqrt{2}}{2\sqrt{\sum_{l=1}^{t} \|\boldsymbol{\nabla} f_{n+l}(u_l)\|_\infty^2}}
\end{aligned}$$

where $\Phi(x) := \sum_{j=1}^{d} [x]_j \ln [x]_j$ and D_Φ is the Bregman divergence (Beck and Teboulle, 2003). Setting AdaEMD$(R; f_{n+1}, \ldots, f_{2n}) = \bar{u}_n$, gives (Orabona, 2019, Theorem 6.11 & 3.14) that with probability $1 - \delta$,

$$F(\text{AdaEMD}(R; f_{n+1}, \ldots, f_{2n})) - \min_{x \in \mathcal{X}: \|x\|_1 \leq R} F(x) \leq O\left(\frac{\|\mathbf{L}\|_\infty R}{\sqrt{n}}\sqrt{\ln d/\delta}\right). \tag{35}$$

Finally, let $\textsc{AdaGrad}(R; f_{n+1}, \dots, f_{2n}) = \bar{u}_n$ where for $t = 1, \dots, n$,

$$\begin{aligned}
\bar{u}_t &\leftarrow \frac{1}{t}\sum_{i=1}^{t} u_i \\
u_{t+1} &\leftarrow \operatorname*{arg\,min}_{u \in \mathcal{X}: \|u\|_\infty \le R} \|u_t - R G_t^{-1/2} \boldsymbol{\nabla} f_{n+t}(u_t) - u\|_{G_t^{1/2}} \\
G_{t+1} &\leftarrow G_t + \mathbf{diag}(\boldsymbol{\nabla} f_{n+t}(u_t))^2
\end{aligned}$$

$G_0 = \mathbf{0}$, and $\|v\|_H := \sqrt{v^\top H v}$. From (Gupta et al., 2017, Section 3.2) and (Orabona, 2019, Theorem 3.14), with probability $1 - \delta$ this obtains the suboptimality guarantee:

$$F(\textsc{AdaGrad}(R; f_{n+1}, \dots, f_{2n})) - \min_{x \in \mathcal{X}: \|x\|_\infty \le R} F(x) \le O\left(\frac{\|\mathbf{L}\|_1 R}{\sqrt{n}} \sqrt{\ln 1/\delta}\right). \tag{36}$$

C.3 Grid Searching over λ_n to Eliminate the Intractable Calculation in Table 1

This section explains how we can grid search over λ_n using our ReliableModelSelection method to avoid the intractable calculations for λ_n currently present in Table 1.

Define,

$$\ell_p := \begin{cases} \hat{L} & \text{if } p = 2 \\ \|\hat{\mathbf{L}}\|_\infty & \text{if } p = 1 \\ \|\hat{\mathbf{L}}\|_1 & \text{if } p = \infty, \end{cases} \qquad l_p := \begin{cases} L & \text{if } p = 2 \\ \|\mathbf{L}\|_\infty & \text{if } p = 1 \\ \|\mathbf{L}\|_1 & \text{if } p = \infty, \end{cases}$$

$$\hat{\lambda}_{n,p}^{(0)} := \begin{cases} \frac{4\sqrt{\ln \frac{6}{\delta}}}{\sqrt{n}} + \frac{20 \ln \frac{6}{\delta}}{n-1} & \text{if } p = 2 \\ \frac{2\sqrt{2 \ln \frac{4d}{\delta}}}{\sqrt{n-1}} + \frac{28 \ln \frac{4d}{\delta}}{3(n-1)} & \text{if } p = 1 \\ \frac{9\sqrt{2 \ln \frac{18}{\delta}}}{2\sqrt{n-1}} + \frac{48 \ln \frac{18}{\delta}}{n-1} & \text{if } p = \infty, \end{cases}$$

and

$$\hat{\lambda}_{n,p}^{(k)} := e^k \hat{\lambda}_{n,p}^{(0)} \text{ for } k = 1, \dots, K_p \text{ where } K_p := 1 \vee \lceil \ln \ell_p \rceil.$$

Let $\hat{x}_{k,p}$ be the output of Algorithm 2 with norm and algorithm specified in Table 1 but regularizer $\hat{\lambda}_{n,p}^{(k)}$ instead of $\lambda_{n,p}$, then due to Theorem 10 we get with probability $1 - 3\delta$ that for all $\hat{\lambda}_{n,p}^{(k)} \ge \lambda_{n,p}$ that

$$F(\hat{x}_{k,p}) - F^\star \le (14\phi_{n,p} + 3\hat{\lambda}_{n,p}^{(k)}) R^\star \quad \text{and} \quad \|\hat{x}_{k,p}\| \le 14 R^\star.$$

Applying this inequality for $\hat{\lambda}_{n,p}^{(k)} \in [\lambda_{n,p}, e\lambda_{n,p}]$ with

$$\begin{aligned}
\tau_{k,p} &= \sqrt{\frac{2\tilde{c}_p \mathbb{V}_k}{n}} + \tilde{c}_p \frac{14 \ell_p \|x_k\|}{3(n-1)} \\
\tilde{c}_p &:= \ln \frac{4K_p}{\delta}.
\end{aligned}$$

gives by Equation (7) and a union bound that, with probability $1-4\delta$,

$$F(x_{k_{\text{rely}},p}) - F^\star \le (14\phi_{n,p} + 3e\lambda_{n,p})R^\star + 14(1+\gamma)\left(\sqrt{2\tilde{c}_p}\frac{l_p R^\star}{\sqrt{n-1}} + \tilde{c}_p\frac{14\ell_p R^\star}{3(n-1)}\right).$$

Substituting the values for each of these terms and fixing γ to be a constant gives

$$F(x_{k_{\text{rely}},p}) - F^\star = \begin{cases} O\left(R^\star l_p\sqrt{\frac{\ln(\delta^{-1}\ln\ell_p)}{n-1}} + \frac{\ell_p R^\star \ln\left(\delta^{-1}\ln\ell_p\right)}{n-1}\right) & \text{if } p \in \{2,\infty\} \\ O\left(R^\star l_1\sqrt{\frac{\ln(d\,\delta^{-1}\ln\ell_1)}{n-1}} + \frac{\ell_1 R^\star \ln\left(d\,\delta^{-1}\ln\ell_1\right)}{n-1}\right) & \text{if } p = 1. \end{cases}$$

as desired. Note this approach uses K_p calls to an empirical risk minimization oracle and the appropriate algorithm from Table 1.

C.4 Proof of Theorem 11

Algorithm 3 Simultaneously adapting to multiple problem structures

input Sample functions $f_1, \dots, f_{3n}$
input Maximum failure probability $\delta > 0$
input Lipschitz estimates $\hat{L}$ and $\hat{\mathbf{L}}$
  for $k = 1, 2, 3$ **do**
    Compute $x_k = \mathbf{A}(2\|\hat{x}_{\lambda_n}\|; f_{n+1}, \dots, f_{2n})$ using algorithm $\mathbf{A}$, norm $\|\cdot\|$ and regularizer λ_n from kth row of Table 1 where $\hat{x}_{\lambda_n} \in \arg\min_{x\in\mathcal{X}} \bar{F}(x) + \lambda_n\|x\|$.
    $\mathbb{V}_k = \frac{1}{n-1}\sum_{i=1}^n (f_i(x_k) - f_i(\mathbf{0}) - (\bar{F}(x_k) - \bar{F}(\mathbf{0})))^2$
    $\tau_k \leftarrow \sqrt{\frac{2\mathbb{V}_k}{n}\ln\frac{12}{\delta}} + \frac{14}{3}\cdot\frac{\ln\frac{12}{\delta}}{n-1}(\hat{L}\|x_k\|_2 \wedge \|\hat{\mathbf{L}}\|_\infty\|x_k\|_1 \wedge \|\hat{\mathbf{L}}\|_1\|x_k\|_\infty)$
  end for
output $z \leftarrow x_{k_{\text{rely}}}$ from applying ReliableModelSelection to $\{(x_k, \tau_k)\}_{k=1}^3$ on $f_{2n+1}, \dots, f_{3n}$ with fixed $\gamma \in [1, \infty)$.

Proof We use Algorithm 3 to obtain the desired result.

Let $Z_{k,i} := F(x_k) - f_i(x_k) - (F(\mathbf{0}) - f_i(\mathbf{0}))$ and $q_k := \hat{L}\|x_k\|_2 \wedge \|\hat{\mathbf{L}}\|_\infty\|x_k\|_1 \wedge \|\hat{\mathbf{L}}\|_1\|x_k\|_\infty$. Note that almost surely $|Z_{k,i}| \le q_k$. Thus, by Theorem 14 and a union bound we deduce that Condition 1 holds with the τ_k value specified in Algorithm 3. Thus, Lemma 3 implies that with probability $1-\delta$

$$F(x_{k_{\text{rely}}}) - F^\star \le \min_{k\in[K]} F(x_k) - F^\star + (1+\gamma)\tau_k. \tag{37}$$

Moreover, by definition of τ_k and the assumption that $L \le \hat{L} \le L\sqrt{n/\ln(1/\delta)} \implies \ln(1/\delta) \le n$ we get

$$\tau_k = \sqrt{\frac{2\mathbb{V}_k}{n}\ln\frac{12}{\delta}} + \frac{14}{3}\cdot\frac{\ln\frac{12}{\delta}}{n-1}q_k \le \sqrt{\frac{2\mathbb{V}_k}{n}\ln\frac{12}{\delta}} + \frac{14}{3}\cdot\sqrt{\frac{n}{n-1}}\cdot\sqrt{\frac{\ln(12)\ln\frac{12}{\delta}}{n-1}}q_k.$$

By Assumption 1 and from Lemma 6 applied separately for each $k \in \{1,2,3\}$ and combined via a union bound, with probability $1-6\delta$ we have $\|x_1\|_2 \le 14\|x^\star\|_2$, $\|x_2\|_1 \le$

$14\|x^\star\|_1$ and $\|x_3\|_\infty \leq 14\|x^\star\|_\infty$ for all $x^\star \in \mathcal{X}^\star$. Thus, for all $x^\star \in \mathcal{X}^\star$ we get with probability $1 - 6\delta$ that

$$\tau_1 \leq O\left(\frac{L\|x^\star\|_2}{\sqrt{n}}\sqrt{\ln 1/\delta}\right) \tag{38a}$$

$$\tau_2 \leq O\left(\frac{\|\mathbf{L}\|_\infty\|x^\star\|_1}{\sqrt{n}}\sqrt{\ln 1/\delta}\right) \tag{38b}$$

$$\tau_3 \leq O\left(\frac{\|\mathbf{L}\|_1\|x^\star\|_\infty}{\sqrt{n}}\sqrt{\ln 1/\delta}\right). \tag{38c}$$

Theorem 10, a union bound, $\hat{L} \leq L\sqrt{n/\ln(1/\delta)}$, and $\hat{\mathbf{L}}_j \leq \mathbf{L}_j\sqrt{n/\ln(d/\delta)}$ implies that, with probability $1 - 9\delta$, for all $x^\star \in \mathcal{X}^\star$

$$F(x_1) - F^\star \leq O\left(\frac{L\|x^\star\|_2}{\sqrt{n}}\sqrt{\ln 1/\delta}\right)$$
$$F(x_2) - F^\star \leq O\left(\frac{\|\mathbf{L}\|_\infty\|x^\star\|_1}{\sqrt{n}}\sqrt{\ln d/\delta}\right)$$
$$F(x_3) - F^\star \leq O\left(\frac{\|\mathbf{L}\|_1\|x^\star\|_\infty}{\sqrt{n}}\sqrt{\ln 1/\delta}\right)$$

Combining these inequalities with Equation (37) and Equation (38) using a union bound gives the result with probability $1 - 10\delta$. ■

Appendix D. Justifying the Norm for the Experiments

In this section, we show that for multivariate logistic regression problems with normalized features (as is the case for zero-shot CLIP models (Radford et al., 2021, Section 3.1.2)), every sample $f(\cdot; S)$ is 2-Lipschitz with respect to the $\|\cdot\|_{2,\infty}$ norm (Equation (39)).

Define the softmax function as

$$\mathbf{p}_l(z) := \frac{e^{[z]_l}}{\sum_{c=1}^C e^{[z]_c}} \quad \mathbf{p}(z) := \begin{pmatrix} \mathbf{p}_1(z) \\ \vdots \\ \mathbf{p}_C(z) \end{pmatrix}$$

where C is the number of classes and z is a vector of scores. In this section, with slight abuse of notation, we will let S be a random vector of size $m+1$ with S_0 denoting the first element and $S_{1:m}$ the remaining elements. For a matrix D, let D_c be its cth row. Let $\mathbf{e}_c$ be the vector with a one in the cth position and zeros everywhere else.

Proposition 22 *Let C be the number of classes in the classification problem and m the number of features in a multivariate logistic regression problem with $\mathcal{X} = \mathbb{R}^{C\times m}$,*

$$f(X; S) = -\log(\mathbf{p}_{S_0}(XS_{1:m})), \quad \mathcal{S} = \{S \in \mathbb{R}^{m+1} : S_0 \in \{1, \ldots, C\} \text{ and } \|S_{1:m}\|_2 \leq 1\}.$$

Then,

$$\sup_{S\in\mathcal{S}} \left|f(X; S) - f(X'; S)\right| \leq 2\|X - X'\|_{2,\infty} \quad \forall X, X' \in \mathbb{R}^{C\times m}. \tag{39}$$

Proof Since $\nabla f(X;S) = (\mathbf{p}(XS_{1:m}) - \mathbf{e}_{S_0})S_{1:m}^\top$ we get that

$$\|\nabla f(X;S)\|_{2,1} = \|(\mathbf{p}(XS_{1:m}) - \mathbf{e}_{S_0})S_{1:m}^\top\|_{2,1} \le \|\mathbf{p}(XS_{1:m}) - \mathbf{e}_{S_0}\|_1 \|S_{1:m}\|_2 \le 2.$$

Since $\|\cdot\|_{2,\infty}$ is dual to $\|\cdot\|_{2,1}$, we deduce $f(\cdot;S)$ is 2-Lipschitz with respect to the $\|\cdot\|_{2,\infty}$ norm. ■

Appendix E. Further Details of Experiments

Costs of running experiments. We estimate the total cost of running our experiments for this paper (including unused and failed experiments), at current cloud computing prices, to be well under $1,000 including purchasing tokens for Gemini.

Assets used. For the few-shot learning experiments with CIFAR-10 (Krizhevsky, 2009) and CLIP we used the pandas (McKinney et al., 2011), PyTorch (Paszke et al., 2019), and OpenCLIP (Ilharco et al., 2021) packages with the ViT-L/14-quickgelu model (Dosovitskiy et al., 2021; Radford et al., 2021) pretrained on the DFN-2B data set (Fang et al., 2024). In the shape-counting experiment, for image generation, prompting `Gemini-1.5-Flash`, and processing the results, we used the matplotlib (Hunter, 2007), google.generativeai (Gemini Team et al., 2024), pandas (McKinney et al., 2011), and word2number (Nagpal, 2014) packages; for running the model selection methods, we used tidyverse (Wickham et al., 2019).

Additional plots. We provide some additional plots to supplement Figure 1 in the body of the paper. Figure 2 provides two additional plots for the few-shot learning experiments. The left plot in Figure 2 shows that the performance gap between reliable and standard model selection in the left plot of Figure 1 is primarily driven by poor tail performance. The right plot in Figure 2 shows that reliable model selection is also reasonable at obtaining a good error rate, even though it is applied to minimizing cross-entropy loss. Figure 3 shows the 95th percentile for the prompt engineering experiments. This shows a similar pattern to the right plot of Figure 1.

More details for prompt engineering experiment. Shape-counting images were systematically generated with the following criteria:

- Each image contains between 1 and 12 shapes, sampled uniformly.
- Each shape belongs to one of six categories, sampled uniformly (ellipse, rectangle, triangle, hexagon, star, pentagon).
- Each shape is defined by a bounding box whose height and width are independently sampled uniformly between 10% and 30% of the image width.
- Shape coordinates are sampled uniformly at random, skipping placements that would cause bounding boxes to overlap or extend beyond the image boundaries.
- Each shape is filled with one of eight colors, sampled uniformly (red, blue, green, purple, orange, yellow, pink, cyan).

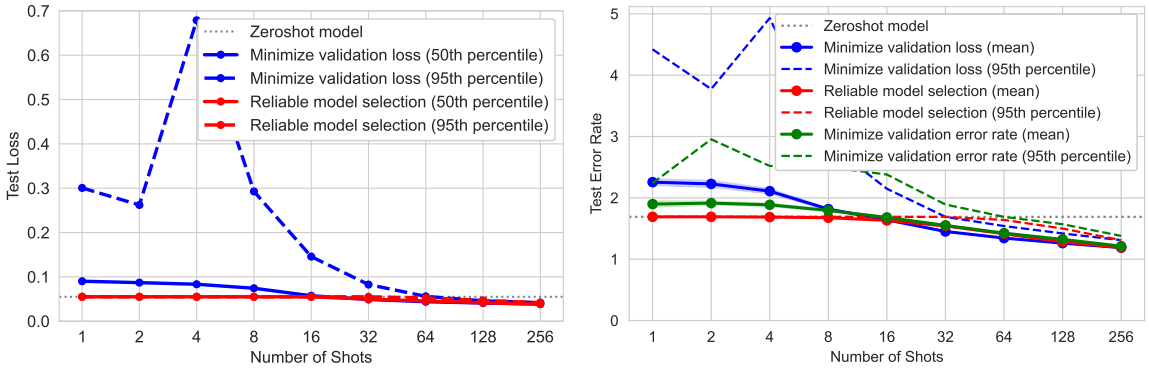


Figure 2: Additional plots for the CLIP experiments. The percentiles are over the runs. For the right plot with error rates, reliable model selection is applied to the validation loss (i.e., cross-entropy).

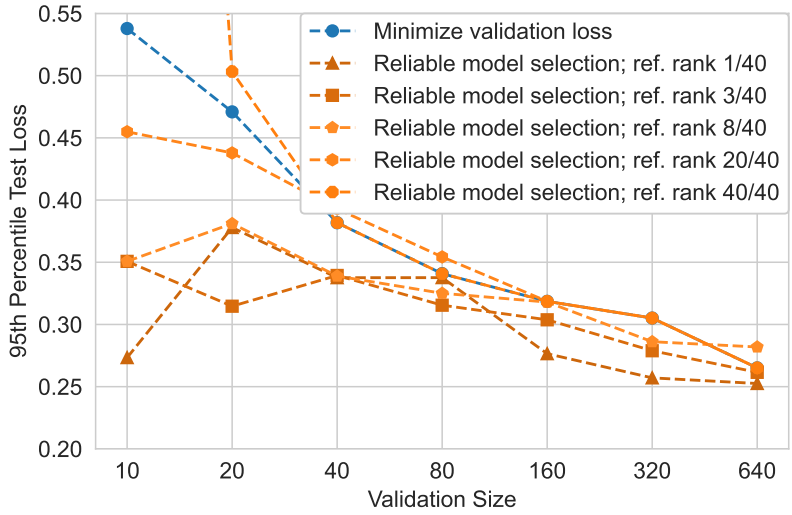


Figure 3: Additional plot for the prompt engineering experiments. The percentiles are over the runs.

- The background of each image is sampled uniformly from 15 preset options: a children's ball pit, a tropical beach, blurry lights, a building, purple wavy lines, a dog, tulips, wavy glowing triangles, fading horizontal lines, distorted colorblocks, a Persian rug, a telescope's image of distant galaxies, a black and white spiral, van Gogh's *Starry Night*, and an image of static white noise.

- Each image is square with background images rescaled to fit.

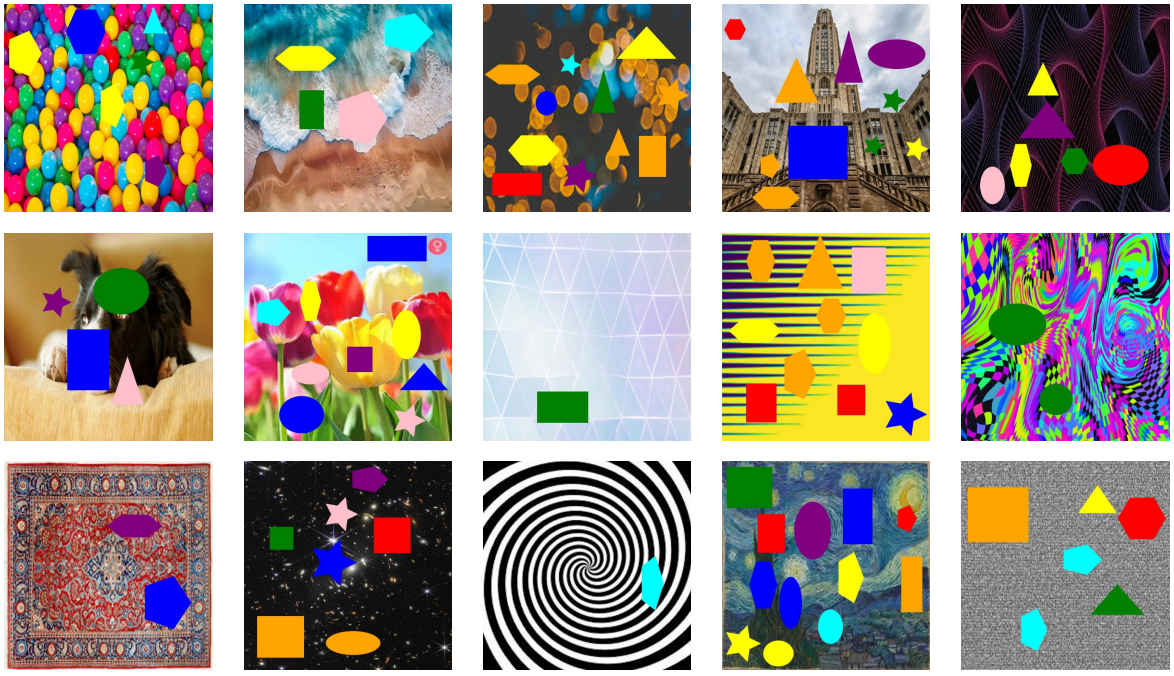


Figure 4: A sample of images, generated by the outlined process, that were presented to Gemini showcasing all 15 backgrounds.

We evaluate 40 distinct textual prompts on each generated image using `Gemini-1.5-Flash` at temperature 0. All images are sent with the same square sizing at 64 DPI resolution. Numerical responses are extracted from the LLM's output as follows:

1. Remove all asterisks (`*`) left as formatting artifacts.
2. Return the last token directly if it is purely numeric.
3. Isolate content within braces $\{\dots\}$ and within square brackets $[\dots]$, returning 0 if a hyphenated range is detected.
4. Perform a reverse search through each remaining word (what we have isolated in the previous step, or the entire output if an isolation was not made), first checking for numeric types and then attempting to extract numbers in word form. The first successful conversion is returned, or 0 otherwise.
5. Clip all extracted values outside the range $[0, 12]$ to the nearest endpoint.

This numerical extraction is compared to the true number of shapes in each image to determine the error, in this case absolute error.

Table 2: Prompts ordered by performance on all $5,000$ examples.

Rank	**Prompt**	**Test Error**
1	Analyze the image carefully to identify and count all visible shapes from the specified list, distinguishing them from any background patterns or non-relevant objects. First, scan the entire image systematically, noting each shape's outline and geometric properties. Then, categorize each detected shape by type (e.g., square vs. rectangle) while verifying no overlaps or partial occlusions are causing miscounting. Finally, tally the total number of valid shapes, ensuring no duplicates or background artifacts are included. Provide the final count as a numeric value between 1 and 12 enclosed in `\mbox{}`	0.2174
2	The image provided to you contains various shapes. How many shapes are present? Please show your work and put your final answer inside the brackets of `\mbox{}`. Think carefully, this problem can be tricky!	0.2440
3	Count the geometric shapes (e.g., circles, triangles, squares). Put your answer inside the brackets of `\mbox{}`.	0.2504
4	The image provided to you contains various shapes. How many shapes are present? Please show your work and put your final answer inside the brackets of `\mbox{}`	0.2518

Continued on next page

Table 2: Prompts ordered by performance on all 5,000 examples.

Rank	Prompt	Test Error
5	The image provided to you contains various shapes. How many shapes are present? Please show your work and put your final answer as `\mbox{#number of shapes}`	0.2544
6	The image has shapes. How many? You need to use this: `\mbox{#number of shapes}`.	0.2728
7	Count the shapes, put your answer inside the brackets of `\mbox{}`.	0.2734
8	Look at the image and count the shapes. Once you've finished, provide your answer in this format: `\mbox{}`.	0.2822
9	Please determine how many shapes are present in the image provided to you. Note that the background should not be included in the count. Once you've calculated the total, express your answer in this format: `\mbox{#total shapes}`.	0.3004
10	You will receive an image that displays a diverse array of geometric shapes, which may include hexagons, ovals, triangles, and rectangles. Your task is to carefully and meticulously count the total number of distinct shapes that appear within the image. It is important to note that these shapes can vary significantly in size, be located in different areas of the image, and may also differ in color and orientation. Additionally, be aware that the background of the image might feature a gradient pattern that visually resembles a circular shape. This gradient, however, should not be counted as one of the shapes for the purposes of your analysis. Once you have completed your count and are confident in your assessment, please ensure that you report your final answer explicitly by placing it inside the brackets of the `\mbox{}` command.	0.3098
11	The image provided to you contains various shapes. How many shapes are present? You MUST be in the following format: `\mbox{#number of shapes}`	0.3140
12	in the pic count the shapes. put the answer inside the brackets of `\mbox{}`	0.3146
13	The image provided to you contains various shapes. How many shapes are present? You ABSOLUTELY MUST be in the following format: `\mbox{#number of shapes}`	0.3180
14	Please count the number of shapes that appear in the following image. Be aware that the background should not be counted in the calculations. Once you have your final answer, put your answer in the following format: `\mbox{#number of shapes}`.	0.3222

Continued on next page

Table 2: Prompts ordered by performance on all 5,000 examples.

Rank	Prompt	Test Error
15	Count the shapes. After counting the total number of shapes, recount the number to make sure that the final answer is correct, and then put your answer inside the brackets of `\mbox{}`.	0.3360
16	Count the shapes. After counting the total number of shapes, double-check your answer for accuracy, and then put your answer inside the brackets of `\mbox{}`.	0.3374
17	Count the shapes seen in the image. Put your answer inside the brackets of `\mbox{}`. For example, if there are 3 circles, 2 triangles, and 1 square, the total count is 6. Put your answer as `\mbox{6}`.	0.3442
18	Count the shapes. First, identify all individual shapes (e.g., circle, square, triangle). Second, count each shape and add them together. Third, put your answer for the total count inside the brackets of `\mbox{}`.	0.3504
19	The image provided contains various shapes such as triangles, squares, and circles. Add the total number of shapes that can be seen in the picture. After adding the number of shapes, include your final answer in the following format: `\mbox{}`.	0.3774
20	Analyze the image carefully. Identify and count all distinct shapes present in the image, considering their individual characteristics, such as size, color, and orientation. Be meticulous in ensuring that each unique shape is accounted for. Provide your answer inside the brackets of `\mbox{}`.	0.3914
21	The image provided to you contains several shapes, including triangles, squares, and circles. Count all the visible shapes in the picture and calculate the total number. Once you have the total, present your final answer in this format: `\mbox{}`.	0.4208
22	Identify and count all the shapes in the image (e.g., circles, squares, triangles). Begin by identifying each distinct shape. Then, tally the total number of shapes. Finally, place your total count inside the brackets of `\mbox{}`.	0.4358
23	Count the shapes that appear in the image. List all the shapes you observe (e.g., circle, triangle, square), count them to determine the total, and then put the answer inside the brackets of `\mbox{}`.	0.4540
24	Count the number of distinct shapes in the image with care and caution. There are 1-12 shapes generated. You will be graded on the L1 distance of your answer to the ground truth. Report your answers in the brackets of `\mbox{}`.	0.4546

Continued on next page

Table 2: Prompts ordered by performance on all 5,000 examples.

Rank	Prompt	Test Error
25	Count only the shapes (e.g., circles, triangles, squares) and exclude any lines or patterns. Put your answer inside the brackets of `\mbox{}`.	0.4694
26	Count the number of distinct shapes in the image with care and caution. There are 1-12 shapes generated. If you are unable to count the number of shapes from the image, choose the final number to be somewhere in the middle. Report your answers in the brackets of `\mbox{}`.	0.4914
27	Analyze the attached image and count the number of distinct shapes present. The image may contain between 1 and 12 shapes, including rectangles, squares, circles, ovals, hexagons, pentagons, and stars. Be sure to disregard any background elements or patterns that are not one of the specified shapes. Provide a breakdown of the count for each shape type (e.g., "3 circles, 2 squares, 1 star") if possible, and then provide the total number of shapes. Finally, record the total number of shapes within the curly braces of the LaTeX command `\mbox{}`.	0.5020
28	Please recognize and count the different shapes in the image, and categorize them by type. Put the final number of shapes in the image in the format: `\mbox{}`.	0.5060
29	Count all complete, non-overlapping shapes visible and put your answer inside the brackets of `\mbox{}`.	0.5094
30	Thoroughly inspect the image and identify all the distinct shapes. List the shapes by their type (e.g., circles, squares, triangles), then count the total number of shapes. Finally, present your total count in the following format: `\mbox{}`.	0.5232
31	Please identify and count the different shapes in the image, and list them by type. Put the final number of shapes in the image in the format: `\mbox{}`.	0.5234
32	Count the number of distinct shapes in the image with care and caution. There are 1-12 shapes generated. If you are unable to count the number of shapes from the image, choose the final number to be 6. Report your answers in the brackets of `\mbox{}`.	0.5370
33	Carefully examine the image and identify all the distinct shapes. List the shapes by their type (e.g., circles, squares, triangles), then count the total number of shapes. Finally, present your total count in the following format: `\mbox{}`.	0.5590

Continued on next page

Table 2: Prompts ordered by performance on all 5,000 examples.

Rank	Prompt	Test Error
34	Examine this image carefully and help me count the total number of distinct geometric shapes present. First, scan the image systematically from left to right, top to bottom, identifying each instance of rectangles, squares, circles, ovals, hexagons, pentagons, and stars. Ignore any background patterns or decorative elements that aren't complete shapes. For similar shapes that overlap or touch, count them as separate shapes if you can clearly distinguish their individual boundaries. Keep a running tally for each shape type you encounter, then sum them up for the total count. Express your final answer using the format `\mbox{n}` where n is the total number of shapes you've counted. Before giving your final answer, double-check your count to ensure you haven't missed any shapes or counted any twice.	0.6262
35	Count the number of distinct shapes in the image with care and caution. There are 1-6 shapes generated. Report your answers in the brackets of `\mbox{}`.	0.8400
36	Count the number of distinct shapes in the image with care and caution. There are 1-24 shapes generated. Report your answers in the brackets of `\mbox{}`.	0.9718
37	Count the number of distinct shapes in the image with care and caution. There are 1-12 shapes generated. Report your answers in the brackets of `\mbox{}`.	1.0892
38	The image you've received contains several geometric shapes of various types (e.g., circles, squares, triangles, etc.). Please identify and count each type of shapes separately. After counting, please show the breakdown of your findings (e.g., how many circles, squares, triangles, etc.). Finally, provide the total number of shapes in the format: `\mbox{#number of shapes}`. Ensure that you answer accounts for each shape present, and please explain how you identified and counted them.	1.1234
39	See the image and count shapes; put answer inside `\mbox{}`	1.1276
40	Count the number of distinct shapes in the image with care and caution. There are 1-100 shapes generated. Report your answers in the brackets of `\mbox{}`.	1.6580

References

Alekh Agarwal, Peter L Bartlett, Pradeep Ravikumar, and Martin J Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex op-

timization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012.

Amit Attia and Tomer Koren. How free is parameter-free stochastic optimization? In *International Conference on Machine Learning (ICML)*, 2024.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

George Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57(297):33–45, 1962.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *International Conference on Machine Learning*, pages 1006–1014. PMLR, 2015.

Olivier Bousquet, Yegor Klochkov, and Nikita Zhivotovskiy. Sharper bounds for uniformly stable algorithms. In *Conference on Learning Theory*, pages 610–626. PMLR, 2020.

Leo Breiman, Jerome Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Chapman and Hall, New York, 1984.

Yair Carmon and Oliver Hinder. Making SGD parameter-free. In *Conference on Learning Theory (COLT)*, 2022.

Yair Carmon and Oliver Hinder. The price of adaptivity in stochastic convex optimization. *arXiv:2402.10898*, 2024.

Keyi Chen, John Langford, and Francesco Orabona. Better parameter-free stochastic optimization with ODE updates for coin-betting. In *AAAI Conference on Artificial Intelligence*, 2022.

Ashok Cutkosky. Artificial constraints and hints for unbounded online learning. In *Conference on Learning Theory (COLT)*, 2019.

Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. In *Conference On Learning Theory*, pages 1493–1529. PMLR, 2018.

Aaron Defazio and Konstantin Mishchenko. Learning-rate-free learning by D-adaptation. In *International Conference on Machine Learning (ICML)*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T Toshev, and Vaishaal Shankar. Data filtering networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=KAk6ngZ09F`.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024.

Vineet Gupta, Tomer Koren, and Yoram Singer. A unified approach to adaptive regularization in online and stochastic optimization. *arXiv:1706.06569*, 2017.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition, 2009.

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, 15 (1):2489–2512, 2014.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Steven R Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform chernoff bounds via nonnegative supermartingales. *Probability Surveys*, 17:257–317, 2020.

Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9068–9077, 2022.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL `https://doi.org/10.5281/zenodo.5143773`. If you use this software, please cite it as below.

Maor Ivgi, Oliver Hinder, and Yair Carmon. DoG is SGD's best friend: A parameter-free dynamic step size schedule. In *International Conference on Machine Learning (ICML)*, 2023.

Andrew Jacobsen and Ashok Cutkosky. Parameter-free mirror descent. In *Conference on Learning Theory (COLT)*, 2022.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv:2001.08361*, 2020.

Michael Kearns and Dana Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 152–162, 1997.

Michal Kempka, Wojciech Kotlowski, and Manfred K Warmuth. Adaptive scale-invariant online algorithms for learning linear models. In *International Conference on Machine Learning (ICML)*, 2019.

Ahmed Khaled and Chi Jin. Tuning-free stochastic optimization. In *International Conference on Machine Learning (ICML)*, 2024.

V Yu Korolev and Irina G Shevtsova. On the upper bound for the absolute constant in the berry–esseen inequality. *Theory of Probability & Its Applications*, 54(4):638–658, 2010.

Itai Kreisler, Maor Ivgi, Oliver Hinder, and Yair Carmon. Accelerated parameter-free stochastic optimization. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 3257–3324. PMLR, 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Daniel Levy and John C Duchi. Necessary and sufficient geometries for gradient methods. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Tudor Manole and Aaditya Ramdas. Martingale methods for sequential estimation of convex functionals and divergences. *IEEE Transactions on Information Theory*, 69(7):4641–4658, 2023.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample variance penalization. *Conference on Learning Theory (COLT)*, 2009.

Wes McKinney et al. pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9):1–9, 2011.

H Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximations. In *Conference on Learning Theory (COLT)*, 2014.

H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. *Conference on Learning Theory (COLT)*, 2010.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. The effect of natural distribution shift on question answering models. In *International conference on machine learning*, pages 6905–6916. PMLR, 2020.

Jean-Jacques Moreau. Proximity and duality in a hilbertian space. *Bulletin of the Mathematical Society of France*, 93:273–299, 1965.

Akshay Nagpal. word2number, 2014. URL `https://github.com/akshaynagpal/w2n`.

Arkadi Nemirovski and David Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, New York, 1983.

John Tinsley Oden and Noboru Kikuchi. Theory of variational inequalities with applications to problems of flow through porous media. *International Journal of Engineering Science*, 18(10):1173–1284, 1980.

Francesco Orabona. A modern introduction to online learning. *arXiv:1912.13213v7*, 2019.

Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

Rebecca Roelofs, Vaishaal Shankar, Benjamin Recht, Sara Fridovich-Keil, Moritz Hardt, John Miller, and Ludwig Schmidt. A meta-analysis of overfitting in machine learning. *Advances in neural information processing systems*, 32, 2019.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi: 10.21105/joss.01686.

Chhavi Yadav and Léon Bottou. Cold case: The lost MNIST digits. *Advances in neural information processing systems*, 32, 2019.