A Generalization Result for Convergence in Learning-to-Optimize

Michael Sucker michael.sucker@math.uni-tuebingen.de
Department of Mathematics
University of Tübingen
Tübingen, Germany

Peter Ochs ochs@cs.uni-saarland.de
Department of Mathematics and Computer Science
Saarland University
Saarbrücken, Germany

Abstract

Convergence in learning-to-optimize is hardly studied, because conventional convergence guarantees in optimization are based on geometric arguments, which cannot be applied easily to learned algorithms. Thus, we develop a probabilistic framework that resembles deterministic optimization and allows for transferring geometric arguments into learning-to-optimize. Our main theorem is a generalization result for parametric classes of potentially non-smooth, non-convex loss functions and establishes the convergence of learned optimization algorithms to stationary points with high probability. This can be seen as a statistical counterpart to the use of geometric safeguards to ensure convergence. To the best of our knowledge, we are the first to prove convergence of optimization algorithms in such a probabilistic framework.

1 Introduction

Proving convergence in learning-to-optimize is a hard problem. This is due to the fact that the problem instances are *functions*, which cannot be observed globally. Rather, the region explored during training is strongly influenced by the chosen initialization and the maximal number of iterations. This spatial restriction is important for the theoretical analysis:

It typically prevents the usage of both limits and the mathematical argument of induction.

As convergence is inherently linked to the notion of limits, this subtlety prevents proving convergence of the learned algorithm. A way to mitigate this problem rather easily is the usage of safeguards: The update step of the algorithm is restricted to such an extent that it can be analyzed similar to a hand-crafted algorithm, that is, the trajectory created by the algorithm satisfies certain properties, for example, a sufficient descent in each iteration, *independently* of the training. Yet, this comes at a price: Not only the analysis of the learned algorithm, also its performance is restricted and eventually similar to hand-crafted algorithms. At that point, however, justifying the additional effort required for learning becomes difficult. Nevertheless, there is an important detail:

These constraints are just a *safeguard* to enforce the desired properties of the trajectory.

Intuitively, whenever the safeguard is "activated", the paradigm and performance of learning-to-optimize is alleviated or even extinguished. Moreover, from a mathematical point of view, these safeguards are sufficient, however, not necessary, that is, they are overly restrictive. Therefore, if we can guarantee, in another way, that the trajectories created by the algorithm have the desired properties, we do not need the constraints anymore. This can be achieved statistically by means of generalization. Thus, in this paper, we combine a general convergence result from variational analysis with a PAC-Bayesian generalization theorem to show that the properties of the trajectory, which are needed to derive convergence of the algorithm, actually generalize to unseen problems. This results in our main theorem, which is applicable in a possible non-smooth non-convex optimization setup, and lower bounds the probability to observe a trajectory, generated by the learned algorithm, that converges to a stationary point of the loss function.

2 Related Work

This work draws on the fields of learning-to-optimize, the PAC-Bayesian learning approach, and convergence results based on the Kurdyka-Łojasiewicz property. Since we use existing bounds, we discuss the PAC-Bayesian approach only briefly, and for learning-to-optimize we focus on approaches that provide some theoretical guarantees. For an introduction to learning-to-optimize, Chen et al. (2021) provide a good overview about the variety of approaches. Similarly, for the PAC-Bayesian approach, good introductory references are given by Guedj (2019) and Alquier (2021).

Broader Context. Since optimization is an integral part of machine learning, learning-to-optimize has significant overlap with meta-learning ("learning-to-learn") and AutoML. Meta-learning is a subset of learning-to-optimize, because it is restricted to machine learning applications (Vilalta and Drissi, 2002; Hospedales et al., 2021), while learning-to-optimize applies to optimization in general. On the other hand, AutoML refers to automating all steps necessary to create a machine learning application (Yao et al., 2018; Hutter et al., 2019; He et al., 2021), which also includes the choice of an optimization procedure.

Learning-to-Optimize with Guarantees. To date, learned optimization methods show impressive performance, yet lack theoretical guarantees (Chen et al., 2021). However, in some applications convergence guarantees are indispensable: It was shown that learning-based methods might fail to reconstruct the crucial details in a medical image (Moeller et al., 2019). In the same work, the authors prove convergence of their learned method by restricting the update to descent directions. Similar safeguarding techniques were employed by Prémont-Schwarz et al. (2022) and Heaton et al. (2023). The basic idea is to constrain the learned object in such a way that known convergence results are applicable, and it has been applied successfully for different schemes and under different assumptions (Sreehari et al., 2016; Chan et al., 2016; Teodoro et al., 2017; Tirer and Giryes, 2018; Buzzard et al., 2018; Ryu et al., 2019; Sun et al., 2019; Terris et al., 2021; Cohen et al., 2021). Similarly, by restricting to a specific ODE, Xie et al. (2024) leverage the convergence of a continuous-time trajectory and the stability of a discretization scheme to provide convergence results for their

algorithm. A major advantage of these "constrained" methods is the fact that the number of iterations is not restricted a priori and that, often, some convergence guarantees can be provided. A major drawback, however, is their severe restriction: Typically, the update-step has to satisfy certain geometric properties, and the results only apply to specific algorithms and/or problems. Another approach, pioneered by Gregor and LeCun (2010), is unrolling, which limits the number of iterations, yet can be applied to any iterative algorithm. Here, the IHT algorithm is studied by Xin et al. (2016) while Chen et al. (2018) consider the unrolled ISTA. However, in the theoretical analysis of unrolled algorithms, the notion of convergence itself is difficult, and one rather has to consider the generalization performance: This has been done by means of Rademacher complexity (Chen et al., 2020), by using a stability analysis Kobler et al. (2020), or in terms of PAC-Bayesian generalization guarantees (Sucker and Ochs, 2023; Sucker et al., 2024). Recently, generalization guarantees based on the whole trajectory of the algorithm, for example, the expected time to reach the stopping criterion, have been proposed (Sucker and Ochs, 2024). The main drawback of generalization guarantees is their reliance on a specific distribution. To solve this, another line of work studies the design of learned optimization algorithms and their training, and how it affects the possible guarantees (Wichrowska et al., 2017; Metz et al., 2019, 2022). Here, Liu et al. (2023) identify common properties of basic optimization algorithms and propose a math-inspired architecture. Similarly, Castera and Ochs (2024) analyze widely used optimization algorithms, extract common geometric properties from them, and provide design-principles for learning-to-optimize. *We present a novel approach that allows to deduce the convergence of a generic learned algorithm with high probability. It is based on a generalization result, that is, it relies on an underlying distribution, and can be seen as a statistical counterpart to the use of geometric safeguards. The methodology is widely applicable and it does not restrict the design of the algorithm.*

PAC-Bayesian Generalization Bounds. The PAC-Bayesian framework allows for giving high probability bounds on the risk. The key ingredient is a change-of-measure inequality, which determines the divergence or distance in the resulting bound. While most bounds involve the Kullback–Leibler divergence as measure of proximity (McAllester, 2003a,b; Seeger, 2002; Langford and Shawe-Taylor, 2002; Catoni, 2004, 2007; Germain et al., 2009), more recently other divergences have been used (Honorio and Jaakkola, 2014; London, 2017; Bégin et al., 2016; Alquier and Guedj, 2018; Ohnishi and Honorio, 2021; Amit et al., 2022; Haddouche and Guedj, 2023). In doing so, the PAC-bound relates the true risk to other terms such as the empirical risk. Yet, it does not directly say anything about the absolute numbers. Therefore, one typically aims to minimize the provided upper bound (Langford and Caruana, 2001; Dziugaite and Roy, 2017; Pérez-Ortiz et al., 2021; Thiemann et al., 2017). Nevertheless, a known difficulty in PAC-Bayesian learning is the choice of the prior distribution, which strongly influences the performance of the learned models and the theoretical guarantees (Catoni, 2004; Dziugaite et al., 2021; Pérez-Ortiz et al., 2021). In part, this is due to the fact that the divergence term can dominate the bound, such that the posterior is close to the prior. Especially, this applies to the Kullback-Leibler divergence. This lead to the idea of choosing a data- or distribution-dependent prior (Seeger, 2002; Parrado-Hernández et al., 2012; Lever et al., 2013; Dziugaite and Roy, 2018; Pérez-Ortiz et al., 2021). By using an independent subset of the data set, also the prior gets optimized

to yield a good performance. *We use one of the PAC-Bayesian generalization theorems provided by Sucker and Ochs (2024) (which involves the Kullback-Leibler divergence) to show that the needed properties of the trajectory of the learned algorithm generalize to unseen problems from the same distribution, and, for the experiments, we use a data-dependent prior.*

The Kurdyka-Łojasiewicz inequality. Single-point convergence of the trajectory of an algorithm is a challenging problem. For example, Absil et al. (2005) show that this might fail even for simple algorithms like gradient descent on infinitely-often differentiable functions. Further, they show that a remedy is provided by the *Łojasiewicz inequality*, which holds for real analytic functions (Bierstone and Milman, 1988). In fact, in the special case of the *Polyak-Łojasiewicz inequality*, gradient descent converges linearly (Polyak, 1963). The large class of tame functions or definable functions excludes many pathological failure cases, and extensions of the Łojasiewicz inequality to smooth definable functions are provided by Kurdyka (1998). Similarly, extensions to the nonsmooth subanalytic or definable setting are shown by Bolte et al. (2007b), Bolte et al. (2007a), and Attouch and Bolte (2009), which yields the *Kurdyka–Łojasiewicz inequality*. It is important to note that these are not just theoretical constructs: Most functions in practice are definable and, thus, satisfy the Kurdyka–Łojasiewicz inequality automatically. Using this, several algorithms have been shown to converge even for nonconvex functions (Attouch and Bolte, 2009; Attouch et al., 2010, 2013; Bolte et al., 2014; Ochs et al., 2014; Ochs, 2019). *We employ the abstract convergence theorem for descent methods provided by Attouch et al. (2013) together with the PAC-Bayesian generalization result to derive our main theorem. To our knowledge, this is a novel combination of two distinct approaches, which allows for deriving new and general convergence results for generic learned optimization algorithms.*

3 Preliminaries and Assumptions

Notation. We write generic sets in type-writer font, for example, $\mathsf{A} \subset \mathbb{R}^d$. Given a metric space X, $\mathsf{B}_\varepsilon(x)$ denotes the open ball around $x \in X$ with radius $\varepsilon > 0$. We assume every metric space to be endowed with the metric topology and corresponding Borel σ-field $\mathcal{B}(X)$. Similarly, given a product space $X \times Y$, the product σ-algebra is denoted by $\mathcal{B}(X) \otimes \mathcal{B}(Y)$. We consider the space $\mathbb{R}^d$ with Euclidean norm $\|\cdot\|$ and, for notational simplicity, abbreviate $S := \mathbb{R}^d$ and $P := \mathbb{R}^p$. The space of sequences in S is denoted by $S^{\mathbb{N}_0}$. Here, we reserve the notation $\xi = (\xi^{(n)})_{n\in\mathbb{N}_0}$ for the sequence generated by the algorithm. We endow $S^{\mathbb{N}_0}$ with the product σ-algebra, which is the smallest σ-algebra, such that all canonical projections $\mathscr{X}_i : S^{\mathbb{N}_0} \to S$, $(x_n)_{n\in\mathbb{N}_0} \mapsto x_i$, are measurable. For notions from non-smooth analysis, we follow Rockafellar and Wets (2009). For convenience of the reader, we have collected details of these definitions in Appendix A. In short, a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is called *proper*, if $f(x) < +\infty$ for at least one $x \in \mathbb{R}^d$, and we denote its *effective domain* by $\operatorname{dom} f$. Further, it is called *lower semi-continuous*, if $\liminf_{x\to\bar{x}} f(x) \geq f(\bar{x})$ for all $\bar{x} \in \mathbb{R}^d$. Furthermore, for $x \in \operatorname{dom} f$, $\partial f(x)$ denotes the (limiting) *subdifferential* of f at x. Similarly, for $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$, $\partial_x f(x, y)$ denotes the (partial) subdifferential of $f(\cdot, y)$ at x. In general, $\partial f : \mathbb{R}^d \rightrightarrows \mathbb{R}^d$ is a set-valued mapping, and we denote its *domain* and *graph* by $\operatorname{dom} \partial f$ and $\operatorname{gph} \partial f$, respectively. Here, a set-valued mapping $S : \mathbb{R}^k \rightrightarrows \mathbb{R}^l$ is said to be *outer semi-continuous* at $\bar{x}$, if $\limsup_{x\to\bar{x}} S(x) \subset S(\bar{x})$, where the *outer limit* is defined

as $\limsup_{x\to\bar{x}} S(x) = \{u \mid \exists x^{(k)} \to \bar{x},\ \exists u^{(k)} \to u \text{ with } u^{(k)} \in S(x^{(k)})\}$. Finally, the space of measures on X is denoted by $\mathcal{M}(X)$, and all probability measures that are absolutely continuous w.r.t. a reference measure $\mu \in \mathcal{M}(X)$ are denote by $\mathcal{P}(\mu) := \{\nu \in \mathcal{M}(X) : \nu \ll \mu \text{ and } \nu[X] = 1\}$. Here, the Kullback-Leiber divergence between two measures μ and ν is defined as $D_{\mathrm{KL}}(\nu \parallel \mu) = \nu[\log(f)] = \int_X \log(f(x))\ \nu(dx)$, if $\nu \ll \mu$ with density f, and $+\infty$ otherwise.
To analyze the trajectory $\xi = (\xi^{(n)})_{n\in\mathbb{N}_0}$ generated by a learned algorithm, Sucker and Ochs (2024) introduce a Markovian model for learning-to-optimize, which we want to use in the following. It is based on the following two assumptions. For this, please recall that a *Polish space* is a separable topological space that admits a complete metrization.

Assumption 1. *The state space $(S, \mathcal{B}(S), \mathbb{P}_{\mathscr{I}})$, the parameter space $(P, \mathcal{B}(P), \mathbb{P}_{\mathscr{P}})$, the hyperparameter space $(H, \mathcal{B}(H), \mathbb{P}_{\mathscr{H}})$, and the randomization space $(R, \mathcal{B}(R), \mathbb{P}_{\mathscr{R}})$ are Polish probability spaces.*

We use the special cases $S = \mathbb{R}^d$ and $P = \mathbb{R}^p$. Here, one could also consider a (finite-dimensional) state space S that *encompasses* the space of the optimization variable, that is, $S = \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$.

Remark 3.1. *The countable product of Polish spaces is Polish, and the product σ-algebra and the Borel σ-algebra coincide (Kallenberg, 2021, Lemma 1.2). Hence, for all considered spaces in the following, the Borel and the product σ-algebra coincide.*

Assumption 2. *The (possibly extended-valued)* loss function $\ell : S \times P \to [0, \infty]$ *and the* algorithmic update $\mathcal{A} : H \times P \times S \times R \to S$ *are both measurable.*

Remark 3.2. *The extended-valued real numbers $\mathbb{R} \cup \{\pm\infty\}$ are topologically isomorphic to the compact interval $[-1, 1]$ and we have the equality $\mathcal{B}(\mathbb{R} \cup \{\pm\infty\})|_{\mathbb{R}} = \mathcal{B}(\mathbb{R})$. Further, every $\mathcal{B}(\mathbb{R})$-measurable function can be considered as a extended-valued measurable function (Klenke, 2013, p.37-38).*

Starting from $\xi^{(0)} \sim \mathbb{P}_{\mathscr{I}}$, the algorithm generates a sequence of iterates $\xi = (\xi^{(n)})_{n\in\mathbb{N}_0}$ as follows:

$$\xi^{(n+1)} = \mathcal{A}(\alpha, \theta, \xi^{(n)}, \eta^{(n+1)}), \quad n \geq 0\,,$$

where the *hyperparameters* $\alpha \in H$ allow for adjusting the algorithm, the *parameters* $\theta \in P$ specify the loss function $\ell(\cdot, \theta)$ the algorithm is applied to, and $\eta^{(n+1)} \in R$ models the *(internal) randomness* of the algorithm. Based on this, Sucker and Ochs (2024) construct a suitable probability space $(\Omega, \mathcal{F}, \mathbb{P})$ that describes the joint distribution over hyperparameters α, parameters θ, and corresponding trajectories ξ generated by $\mathcal{A}(\alpha, \theta, \cdot, \cdot)$. Then, by leveraging a well-known result by Catoni (2007), they show that properties of the trajectories ξ, *encoded as sets* $\mathsf{A} \in \mathcal{B}(P) \otimes \mathcal{B}(S^{\mathbb{N}_0})$, generalize in a PAC-Bayesian way: On an intuitive level, the probability to observe a problem instance $\ell(\cdot, \theta)$ and a corresponding trajectory ξ generated by the algorithm $\mathcal{A}(\alpha, \theta, \cdot, \cdot)$, which satisfies the properties encoded in A, can be bounded based on empirical estimates on the i.i.d. data set $\mathscr{P}_{[N]}$ of N parameters. Note that, by considering the complementary event A^c, this can also be turned into a lower bound.

Theorem 3.3. *Let $\mathsf{A} \subset P \times S^{\mathbb{N}_0}$ be measurable. Then, for $\lambda \in (0, \infty)$, it holds that:*

$$\mathbb{P}_{\mathscr{P}_{[N]}}\Big\{\forall \rho \in \mathcal{P}(\mathbb{P}_{\mathscr{H}}) \;:\; \rho[\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}\}] \leq \Phi^{-1}_{\frac{\lambda}{N}}\Big(\frac{1}{N}\sum_{n=1}^{N} \rho\left[\mathbb{P}_{(\mathscr{P}_n,\xi_n)|\mathscr{H},\mathscr{P}_n}\{\mathsf{A}\}\right] + \frac{D_{\mathrm{KL}}(\rho \parallel \mathbb{P}_{\mathscr{H}}) + \log\left(\frac{1}{\varepsilon}\right)}{\lambda}\Big)\Big\} \geq 1 - \varepsilon\,,$$

where $\Phi_a^{-1}(p) := \frac{1-\exp(-ap)}{1-\exp(-a)}$.

Here, $\mathbb{P}_{\mathscr{H}}$ is the so-called *prior* over hyperparameters, every $\rho \in \mathcal{P}(\mathbb{P}_{\mathscr{H}})$ is called a *posterior*, and $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}$ is the conditional distribution of the parameters with corresponding trajectory, given the hyperparameters. In this paper, we apply Theorem 3.3 to the set of sequences that converge to a stationary point of ℓ.

Definition 3.4. *Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be proper. A point $x \in S$ is called* stationary *for f, if $0 \in \partial f(x)$.*

For this, we make the following additional assumption:

Assumption 3. *The function $\ell : S \times P \to [0, \infty]$ is proper, lower semi-continuous, and continuous on* dom ℓ. *Furthermore, the map $(x, \theta) \mapsto \partial_x \ell(x, \theta)$ is outer semi-continuous.*

Finally, our main result also relies on the following well-known convergence theorem by Attouch et al. (2013, Theorem 2.9). It shows that convergence is a consequence of some general properties of the trajectory, which can be observed during training, or, even more so, for which the algorithm can be trained.

Theorem 3.5. *Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be a proper lower semi-continuous function that is bounded from below. Further, suppose that $(x^{(n)})_{n\in\mathbb{N}_0} \subset \mathbb{R}^d$ is a sequence satisfying the following property: There exists positive scalars a and b such that:*

(i) Sufficient-decrease condition*: For each $k \in \mathbb{N}_0$, $f(x^{(k+1)}) + a\|x^{(k+1)} - x^{(k)}\|^2 \leq f(x^{(k)})$.*

(ii) Relative-error condition*: For each $k \in \mathbb{N}_0$, there exists $v^{(k+1)} \in \partial f(x^{(k+1)})$, such that $\|v^{(k+1)}\| \leq b\|x^{(k+1)} - x^{(k)}\|$.*

(iii) Continuity condition*: For any convergent subsequence $x^{(k_j)} \overset{j\to\infty}{\to} \hat{x}$, we have $f(x^{(k_j)}) \overset{j\to\infty}{\to} f(\hat{x})$.*

If, additionally, the sequence $(x^{(n)})_{n\in\mathbb{N}_0}$ is bounded and f is a Kurdyka-Łojasiewicz function, then $(x^{(n)})_{n\in\mathbb{N}_0}$ converges to a stationary point of f.

Remark 3.6. *(i) The function $\ell(\cdot, \theta)$ is proper, continuous on its domain, and bounded from below. Hence, for $(x^{(n)})_{n\in\mathbb{N}_0} \subset$ dom $\ell(\cdot, \theta)$, the continuity condition holds. Thus, we have to make sure that the sufficient-decrease condition, the relative-error condition, and the boundedness do hold.*

(ii) Actually, Theorem 2.9 of Attouch et al. (2013) is stated slightly different: They assume existence of a convergent subsequence instead of boundedness. Yet, the boundedness assumption implies existence and is standard (Bolte et al., 2014).

4 Theoretical Results

In this section, we combine Theorem 3.3 with Theorem 3.5 to get a generalization result for the convergence of learned algorithms to stationary points. In doing so, we bring together advanced tools from learning theory and optimization. We show that the probability to observe a parameter θ and a corresponding trajectory ξ, which converges to a stationary point of $\ell(\cdot, \theta)$, generalizes. For this, we formulate the sufficient-descent condition, the relative-error condition, and the boundedness assumption as measurable sets in $P \times S^{\mathbb{N}_0}$, such that their intersection is exactly the sequences satisfying the properties of Theorem 3.5.

4.1 Measurability

For this, we first encode the needed properties of the trajectory as sets in $P \times S^{\mathbb{N}_0}$. Then, to apply Theorem 3.3, we show that they are actually measurable w.r.t. $\mathcal{B}(P) \otimes \mathcal{B}(S^{\mathbb{N}_0})$. Hence, denote the (parametric) set of stationary points of ℓ by

$$\mathsf{A}_{\text{stat}} := \{(\theta, x) \in P \times S \ : \ 0 \in \partial_x \ell(x, \theta)\}\,.$$

Then, the section $\mathsf{A}_{\text{stat},\theta} := \{x \in S \ : \ (\theta, x) \in \mathsf{A}_{\text{stat}}\}$ is the set of stationary points of $\ell(\cdot, \theta)$.

Lemma 4.1. *Suppose that Assumptions 1, 2 and 3 hold. Define the (parametric) set of sequences that converge to a stationary point of ℓ as*

$$\mathsf{A}_{\text{conv}} := \{(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in P \times S^{\mathbb{N}_0} \ : \ \exists x^* \in \mathsf{A}_{\text{stat},\theta} \ s.t. \ \lim_{n\to\infty} \|x^{(n)} - x^*\| = 0\}\,.$$

Then A_{conv} is measurable.

Proof. The idea of the proof is to show that A_{conv} can be written as *countable* intersection/union of measurable sets. This is possible, because the considered spaces are Polish, that is, they have a countable, dense subset, and they are complete, that is, limits of Cauchy sequences are inside the space. The details of the proof are provided in Appendix B. □

Lemma 4.2. *Assume that Assumptions 1 and 2 hold. Define the (parametric) set of sequences that satisfy the* sufficient-descent condition *as*

$$\begin{aligned}\mathsf{A}_{\text{desc}} := \Big\{&(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in P \times S^{\mathbb{N}_0} : (x^{(n)})_{n\in\mathbb{N}_0} \subset \operatorname{dom} \ell(\cdot, \theta) \text{ and } \exists a > 0 \text{ s.t. } \forall k \in \mathbb{N}_0 \\ &\ell(x^{(k+1)}, \theta) + a\|x^{(k+1)} - x^{(k)}\|^2 \leq \ell(x^{(k)}, \theta)\Big\}\,.\end{aligned}$$

Then A_{desc} is measurable.

Proof. Since $\mathbb{Q}$ is dense in $\mathbb{R}$, we can restrict to $a \in (0, \infty) \cap \mathbb{Q} =: \mathbb{Q}_+$. Then A_{desc} can be written as

$$\left(\bigcup_{a\in\mathbb{Q}_+} \bigcap_{k\in\mathbb{N}_0} \mathsf{A}_{a,k}\right) \cap \left(\bigcap_{k\in\mathbb{N}_0} \Big\{\ell(x^{(k)}, \theta) < \infty\Big\}\right)\,,$$

where $\mathsf{A}_{a,k}$ is given by:

$$\Big\{\ell(x^{(k+1)}, \theta) + a\|x^{(k+1)} - x^{(k)}\|^2 \leq \ell(x^{(k)}, \theta)\Big\}\,.$$

Since σ-algebras are stable under countable unions/intersection, it suffices to show that the sets $\{\ell(x^{(k)}, \theta) < \infty\}$ and $\mathsf{A}_{a,k}$ are measurable. Here, the set $\{\ell(x^{(k)}, \theta) < \infty\}$ can be written as:

$$\left\{\ell(x^{(k)}, \theta) < \infty\right\} = \left(\ell \circ \Phi \circ (id, \mathscr{X}_k)\right)^{-1} [0, \infty),$$

where $\Phi : P \times S \to S \times P$ just interchanges the coordinates (which is measurable), and id is the identity on P. Since $[0, \infty)$ is a measurable set and ℓ is assumed to be measurable, we have that $\{\ell(x^{(k)}, \theta) < \infty\}$ is meaurable for each $k \in \mathbb{N}_0$. To show that $\mathsf{A}_{a,k}$ is measurable, we define the function $g_a : (\text{dom } \ell)^2 \to \mathbb{R}$ through $((x_1, \theta_1), (x_2, \theta_2)) \mapsto \ell(x_2, \theta_2) - \ell(x_1, \theta_1) + a\|x_2 - x_1\|^2$. Then, g_a is measurable and $\mathsf{A}_{a,k}$ can be written as:

$$\begin{aligned}
\mathsf{A}_{a,k} &= \{g_a(x_k, \theta, x_{k+1}, \theta) \leq 0\} \\
&= \{(g_a \circ (\mathscr{X}_k, id, \mathscr{X}_{k+1}, id) \circ \iota)(\theta, (x_n)_{n \in \mathbb{N}_0}) \leq 0\} \\
&= (g_a \circ (\mathscr{X}_k, id, \mathscr{X}_{k+1}, id) \circ \iota))^{-1} (-\infty, 0],
\end{aligned}$$

where $\iota : P \times S^{\mathbb{N}_0} \to (S^{\mathbb{N}_0} \times P)^2$ is the diagonal inclusion $(\theta, x) \mapsto ((x, \theta), (x, \theta)$, which is measurable w.r.t. to the product-σ-algebra on $(S^{\mathbb{N}_0} \times P)^2$, since $\iota^{-1}(\mathsf{B}_1 \times \mathsf{B}_2) = \mathsf{B}_1 \cap \mathsf{B}_2$. Thus, the set $\mathsf{A}_{a,k}$ is measurable, which concludes the proof. □

We proceed with the relative error condition. It involves a union over all subgradients, and therefore might not be measurable. Hence, we have to restrict to subgradients given through a *measurable selection*, that is, a measurable function $v : \text{dom } \partial_x \ell \to S$, such that $v(x, \theta) \in \partial_x \ell(x, \theta)$ for every $(x, \theta) \in \text{dom } \partial_x \ell$. Under the given assumptions, its existence is guaranteed by Corollary C.3 in Appendix C.

Lemma 4.3. *Suppose that Assumptions 1, 2 and 3 hold. Define the (parametric) set of sequences that satisfy the* relative-error condition *as*

$$\begin{aligned}
\mathsf{A}_{\text{err}} := \Big\{ &(\theta, (x^{(n)})_{n \in \mathbb{N}_0}) \in P \times S^{\mathbb{N}_0} : (\theta, x^{(k)}) \in \text{dom } \partial_x \ell \ \forall k \in \mathbb{N}_0 \text{ and } \exists b > 0 \text{ s.t. } \forall k \in \mathbb{N}_0 \\
&\|v(x^{(k+1)}, \theta)\| \leq b\|x^{(k+1)} - x^{(k)}\| \Big\}.
\end{aligned}$$

Then A_{err} *is measurable.*

Proof. This follows similar to the proof of Lemma 4.2 with

$$g_b((x_1, \theta_1), (x_2, \theta_2)) = \|v(x_2, \theta_2)\| - b\|x_2 - x_1\|.$$

The details are given in Appendix C. □

Lemma 4.4. *Assume that Assumption 1 holds. Define the set of* bounded sequences *as:*

$$\tilde{\mathsf{A}}_{\text{bound}} = \left\{(x^{(n)})_{n \in \mathbb{N}_0} \in S^{\mathbb{N}_0} \ : \ \exists c \geq 0 \text{ s.t. } \|x^{(k)}\| \leq c \ \forall k \in \mathbb{N}_0\right\}.$$

Then $\mathsf{A}_{\text{bound}} := P \times \tilde{\mathsf{A}}_{\text{bound}}$ *is measurable.*

Proof. This follows as before with $g_c(x) := \|x\|$. The details are given in Appendix D. □

4.2 Convergence to stationary points

We are now in a position to derive our main result.

Corollary 4.5. *Suppose that Assumptions 1, 2, and 3 hold. Furthermore, assume that $\ell(\cdot,\theta)$ is a Kurdyka-Łojasiewicz function for every $\theta \in P$. Then the sets $\mathsf{A}_{\mathrm{desc}} \cap \mathsf{A}_{\mathrm{err}} \cap \mathsf{A}_{\mathrm{bound}} \subset P \times S^{\mathbb{N}_0}$ and $\mathsf{A}_{\mathrm{conv}} \subset P \times S^{\mathbb{N}_0}$ are measurable, and it holds that:*

$$\mathsf{A}_{\mathrm{desc}} \cap \mathsf{A}_{\mathrm{err}} \cap \mathsf{A}_{\mathrm{bound}} \subset \mathsf{A}_{\mathrm{conv}}\,.$$

Proof. Take $(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in \mathsf{A}_{\mathrm{desc}} \cap \mathsf{A}_{\mathrm{err}} \cap \mathsf{A}_{\mathrm{bound}}$. Then, $(x^{(n)})_{n\in\mathbb{N}_0}$ satisfies the sufficient-descent condition and the relative-error condition for $\ell(\cdot,\theta)$, and the sequence $(x^{(n)})_{n\in\mathbb{N}_0}$ stays bounded. Further, $(x^{(n)})_{n\in\mathbb{N}_0}$ also satisfies the continuity condition, since we have $(x^{(n)})_{n\in\mathbb{N}_0} \subset \operatorname{dom}\ell(\cdot,\theta)$ and ℓ is continuous on its domain. Thus, Theorem 3.5 implies that $(x^{(n)})_{n\in\mathbb{N}_0}$ converges to a stationary point of $\ell(\cdot,\theta)$. Hence, there exists $x^* \in \mathsf{A}_{\mathrm{stat},\theta}$, such that $\lim_{n\to\infty}\|x^{(n)} - x^*\| = 0$. Therefore, $(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in \mathsf{A}_{\mathrm{conv}}$. □

In particular, if μ is a (probability) measure on $P \times S^{\mathbb{N}_0}$, for example, $\mu = \mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}=\alpha}$ for a given $\alpha \in H$, by the monotonicity of measures it holds that:

$$\mu\{\mathsf{A}_{\mathrm{desc}} \cap \mathsf{A}_{\mathrm{err}} \cap \mathsf{A}_{\mathrm{bound}}\} \le \mu\{\mathsf{A}_{\mathrm{conv}}\}\,.$$

This idea yields our main theorem:

Theorem 4.6. *Suppose that Assumptions 1, 2, and 3 hold. Further, assume that $\ell(\cdot,\theta)$ is a Kurdyka–Łojasiewicz function for every $\theta \in P$. Abbreviate $\mathsf{A} := \mathsf{A}_{\mathrm{desc}} \cap \mathsf{A}_{\mathrm{err}} \cap \mathsf{A}_{\mathrm{bound}}$. Then, for $\lambda \in (0,\infty)$, it holds that:*

$$\mathbb{P}_{\mathscr{P}_{[N]}}\Big\{\forall\rho \in \mathcal{P}(\mathbb{P}_{\mathscr{H}}) \;:\; \rho[\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}_{\mathrm{conv}}\}] \ge 1 - \Phi^{-1}_{\frac{\lambda}{N}}\Big(\frac{1}{N}\sum_{n=1}^{N}\rho\left[\mathbb{P}_{(\mathscr{P}_n,\xi_n)|\mathscr{H},\mathscr{P}_n}\{\mathsf{A}^c\}\right] + \frac{D_{\mathrm{KL}}(\rho \parallel \mathbb{P}_{\mathscr{H}}) + \log\left(\frac{1}{\varepsilon}\right)}{\lambda}\Big)\Big\} \ge 1-\varepsilon\,.$$

Proof. By taking the complementary events in Corollary 4.5, we have $\mathbb{P}_{\mathscr{H}}$-a.s.:

$$\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}^c\} \ge 1 - \mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}_{\mathrm{conv}}\}\,.$$

By Theorem 3.3, for any measurable set $\mathsf{B} \subset P \times S^{\mathbb{N}_0}$ and $\lambda \in (0,\infty)$, we have:

$$\mathbb{P}_{\mathscr{P}_{[N]}}\Big\{\forall\rho \in \mathcal{P}(\mathbb{P}_{\mathscr{H}}) \;:\; \rho[\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{B}\}] \le \Phi^{-1}_{\frac{\lambda}{N}}\Big(\frac{1}{N}\sum_{n=1}^{N}\rho\left[\mathbb{P}_{(\mathscr{P}_n,\xi_n)|\mathscr{H},\mathscr{P}_n}\{\mathsf{B}\}\right] + \frac{D_{\mathrm{KL}}(\rho \parallel \mathbb{P}_{\mathscr{H}}) + \log\left(\frac{1}{\varepsilon}\right)}{\lambda}\Big)\Big\} \ge 1-\varepsilon\,.$$

Hence, using $\mathsf{B} := \mathsf{A}^c$, inserting the inequality above, and rearranging the terms yields the result. □

Remark 4.7. *Actually, the lower bound holds for $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}\}$. Further, since we do not know $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}_{\mathrm{conv}} \setminus \mathsf{A}\}$, we do not know how tight this bound is for $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}_{\mathrm{conv}}\}$.*

5 Experiments

In this section, we conduct two experiments: The strongly convex and smooth problem of minimizing quadratic functions with varying strong convexity, varying smoothness, and varying right-hand side, and the non-smooth non-convex problem of training a neural network on different data sets.

5.1 Quadratic Problems

First, we train the algorithm $\mathcal{A}$ to solve quadratic problems. Thus, each optimization problem $\ell(\cdot, \theta)$ is of the form

$$\min_{x \in \mathbb{R}^d} \frac{1}{2} \|Ax - b\|^2, \quad A \in \mathbb{R}^{d \times d},\ b \in \mathbb{R}^d,$$

such that the parameters are given by $\theta = (A, b) \in \mathbb{R}^{d^2+d} =: P$, and the optimization variable is $x \in \mathbb{R}^d$ ($d = 200$). The strong-convexity and smoothness constants of ℓ are sampled randomly in the intervals $[m_-, m_+], [L_-, L_+] \subset (0, +\infty)$, and we define the matrix A_j, $j = 1, ..., N$, as a *diagonal matrix* with entries $a^j_{ii} = \sqrt{m_j} + i(\sqrt{L_j} - \sqrt{m_j})/d$, $i = 1, ..., d$. In principle, this is a severe restriction. However, we do not use this knowledge explicitly in the design of our algorithm $\mathcal{A}$, that is, if the algorithm "finds" this structure during learning by itself, it can leverage on it. Like this, the given class of functions is L_+-smooth and m_--strongly convex, such that we use *heavy-ball with friction* (HBF) (Polyak, 1964) as worst-case optimal baseline. Its update is given by $x^{(n+1)} = x^{(n)} - \beta_1 \nabla f(x^{(n)}) + \beta_2 \left(x^{(n)} - x^{(n-1)}\right)$, where the optimal worst-case convergence rate is attained for $\beta_1 = \left(\frac{2}{\sqrt{L_+} + \sqrt{\mu_-}}\right)^2, \beta_2 = \left(\frac{\sqrt{L_+} - \sqrt{\mu_-}}{\sqrt{L_+} + \sqrt{\mu_-}}\right)^2$ (Nesterov, 2018). Similarly, the learned algorithm $\mathcal{A}$ performs an update of the form $\xi^{(n+1)} = \xi^{(n)} + \beta^{(n)} d^{(n)}$, where $\beta^{(n)}$ and $d^{(n)}$ are predicted by separate blocks of a neural network. Here, we stress that the update is not constrained in any way. For more details on the architecture we refer to Appendix E. Since the functions are smooth and strongly convex, they satisfy all the needed properties of Theorem 3.5, and the continuity condition does hold. Hence, we only have to check the sufficient-descent condition and the relative-error condition. Here, obviously, it is impossible in practice to check them for all $k \in \mathbb{N}_0$. Thus, we restrict to $n_{\text{train}} = 500$ iterations. *Assuming to be given* a measurable selection $v(\bar{x}, \bar{\theta}) \in \partial_x \ell(\bar{x}, \bar{\theta})$, the relative-error condition is trivially satisfied with $b := \max_{k \leq n_{\text{train}}} \{\|v(\xi^{(k)}, \theta)\|\} / \min_{k \leq n_{\text{train}}} \|\xi^{(k)} - \xi^{(k-1)}\|$, such that we only have to check the sufficient-descent condition during training. Finally, we consider a trajectory ξ to be converged, if the loss is smaller than 10^{-16}. For more details about the actual training of the algorithm we refer to Appendix F. The results of this experiment are shown in Figure 1: The left plot shows the distance to the minimizer x^*_θ over the iterations, where HBF is shown in blue and the learned algorithm in pink. We can see that the learned algorithm outperforms HBF by several orders of magnitude. The right plot shows the estimated probabilities $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}\}$ (orange), $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}_{\text{conv}}\}$ (purple), and the PAC-bound (dark orange) on 250 test sets of size $N = 250$. We can see that the PAC-bound is quite tight for $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}\}$, while there is a substantial gap $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}_{\text{conv}} \setminus \mathsf{A}\}$, such that it is not

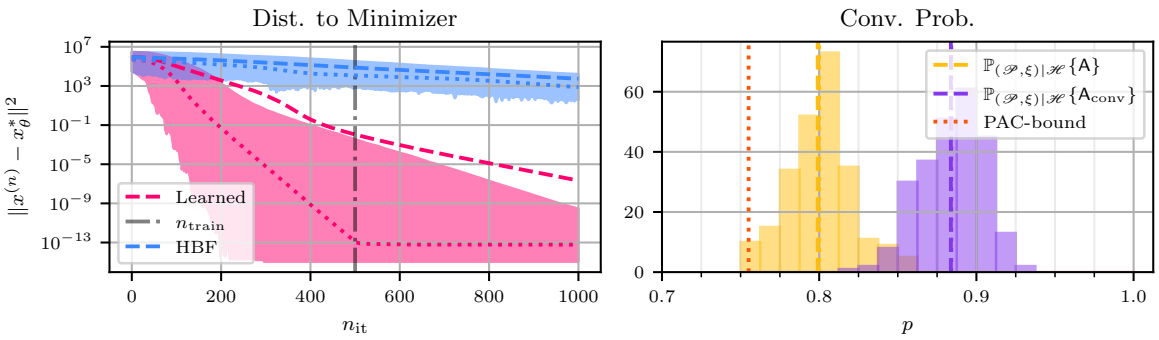

Figure 1: Solving quadratic problems: The left figure shows the distance to the minimizer over the iterations, where HBF is shown in blue and the learned algorithm in pink. The mean and median are shown as dashed and dotted lines, respectively, while the shaded region represents the test data up to the quantile $q = 0.95$. One can see that the learned algorithm converges way faster than HBF. The right plot shows the estimates (dashed lines) for $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}\}$ (orange), $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}_{\text{conv}}\}$ (purple), and the PAC-bound (dark orange). One can see that the predicted chain of inequalities $1 - \Phi^{-1}(...) \leq \mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}\} \leq \mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}_{\text{conv}}\}$ does hold true.

really tight for $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}_{\text{conv}}\}$. Nevertheless, it *guarantees* that the learned algorithm will converge in about 75% of the test problems.

5.2 Training a Neural Network

As second experiment, we train the algorithm $\mathcal{A}$ to train a neural network $\mathtt{N}$ on a simple regression problem. Thus, the algorithm $\mathcal{A}$ predicts parameters $\beta \in \mathbb{R}^p$, such that $\mathtt{N}(\beta, \cdot)$ estimates a function $g : \mathbb{R} \to \mathbb{R}$ from noisy observations $y_{i,j} = g_i(x_j) + \varepsilon_{i,j}$, $i = 1, ..., N$, $j = 1, ..., K$ $(K = 50)$, with $\varepsilon_{i,j} \overset{iid}{\sim} \mathcal{N}(0, 1)$. Here, we use the mean square error as loss for the neural network, and for $\mathtt{N}$ we use a fully-connected two layer neural network with ReLU-activation functions. Then, by using the data sets as parameters, that is, $P = \mathbb{R}^{K\times 2}$ and $\theta_i = \{(x_{i,j}, y_{i,j})\}_{j=1}^K$, the loss functions for the algorithm are given by $\ell(\beta, \theta_i) := \frac{1}{K}\sum_{j=1}^K (\mathtt{N}(\beta, x_{i,j}) - y_{i,j})^2$, which are non-smooth non-convex in β. Here, the input x is transformed into the vector $(x, x^2, ..., x^5)$, such that the parameters $\beta \in \mathbb{R}^p$ are given by the weights $A_1 \in \mathbb{R}^{50\times 5}, A_2 \in \mathbb{R}^{1\times 50}$ and biases $b_1 \in \mathbb{R}^{50}, b_2 \in \mathbb{R}$ of the two fully-connected layers. Thus, the optimization space is of dimension $p = 351$. For the functions g_i we use polynomials of degree $d = 5$, where we sample the coefficients $(c_{i,0}, ..., c_{i,5})$ uniformly in $[-5, 5]$. Similarly, we sample the points $\{x_{i,j}\}_{j=1}^K$ uniformly in $[-2, 2]$. As baseline we use Adam (Kingma and Ba, 2015) as it is implemented in PyTorch (Paszke et al., 2019), and we tune its step-size with a simple grid search over 100 values in $[10^{-4}, 10^{-2}]$, such that its performance is best for the given $n_{\text{train}} = 250$ iterations. This yields the value $\kappa = 0.008$. Note that we use Adam in the "full-batch setting" here, while, originally, it was introduced for the stochastic case. On the other hand, the learned algorithm performs the update

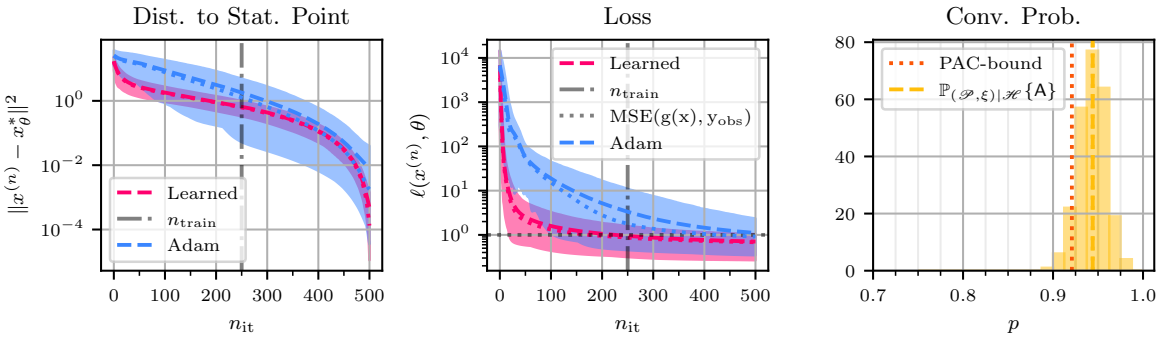

Figure 2: Training a neural network: The left figure shows the distance to the estimated stationary point and the figure in the middle shows the loss. Here, Adam is shown in blue and the learned algorithm in pink. The mean and median are shown as dashed and dotted lines, respectively, while the shaded region represents the test data up to the quantile $q = 0.95$. One can see that the learned algorithm minimizes the loss faster than Adam, and it seems to converge to a stationary point. The right plot shows the estimate (dashed line) for $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}\}$ (orange) and the PAC-bound (dark orange).

$x^{(n+1)} = x^{(n)} + d^{(n)}/\sqrt{n}$, where $d^{(n)}$ is predicted by a neural network. Again, we stress that $d^{(n)}$ is not constrained in any way. For more details on the architecture, we refer to Appendix G. As we cannot access the stationary points directly, we approximate them by running gradient descent for $5 \cdot 10^4$ iterations with a step-size of $1 \cdot 10^{-6}$, starting for each problem and algorithm from the last iterate ($n = 500$). Similarly, we cannot estimate the convergence probability in this case, only the probability for the event A. The results of this experiment are shown in Figure 2: The left plot shows the distance to the (approximated) stationary point and the plot in the middle shows the loss. Here, Adam is shown in blue, while the learned algorithm is shown in pink. Finally, the right plot shows the estimate for $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}\}$ and the predicted PAC-bound. We can see that the learned algorithm does indeed seem to converge to a stationary point and it minimizes the loss faster than Adam. Further, the PAC-bound is quite tight, and it *guarantees* that the learned algorithm will converge in about 92% of the problems.

6 Conclusion

We presented a generalization result for the convergence of learned optimization algorithms to stationary points of the loss function. This approach can be seen as being complementary to the use of safeguarding: Instead of imposing the needed properties, we show that the probability to observe a trajectory, which has these properties, generalizes to unseen problems. While it is theoretically sound, practically this approach has at least four drawbacks: First, and foremost, one simply *cannot* observe the whole trajectory in practice. Thus, one will only get an approximation to this result, that is, one can only check whether the used conditions do hold up to a certain number of iterations. Nevertheless, by using sufficiently

many iterations, one could guarantee that the algorithm gets sufficiently close to a stationary point. Secondly, instead of checking the conditions that were used here, one could also just try to observe the final result by, for example, observing the norm of the gradient. However, when checking the proposed conditions, one is guaranteed to get arbitrary close to a stationary point, while, when just observing the gradient norm, one could end up with a small gradient norm that is arbitrary far away from a stationary point. Especially, this applies in the non-smooth setting, where the subdifferential does not necessarily tell anything about the distance to a stationary point, or to applications where one simply cannot access stationary points during training. Thirdly, for now, training the algorithm in such a way that it actually does satisfy the proposed properties on a majority of problems is quite difficult and time-consuming. Lastly, Theorem 3.5 is not well-suited for stochastic optimization. Nevertheless, Theorem 3.3 and the overall methodology can also be used for deriving convergence results for stochastic algorithms.

Appendix A. Missing Definitions

The following definitions can be found in the book of Rockafellar and Wets (2009). A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\pm\infty\}$ is called *proper*, if $f(x) < +\infty$ for at least one point $x \in \mathbb{R}^d$ and $f(x) > -\infty$ for all $x \in \mathbb{R}^d$. In this case, the *effective domain* of f is the set

$$\text{dom } f := \{x \in \mathbb{R}^d \ : \ f(x) < +\infty\}\,.$$

Similarly, for a set-valued mapping $S : X \rightrightarrows U$ the *graph* is defined as

$$\text{gph } S := \{(x, u) \in X \times U \ : \ u \in S(x)\}\,,$$

while its *domain* is defined as

$$\text{dom } S := \{x \in X \ : \ S(x) \neq \emptyset\}\,.$$

The *outer limit* of a set-valued map $S : \mathbb{R}^d \rightrightarrows \mathbb{R}^d$ is defined as:

$$\limsup_{x \to \bar{x}} \ S(x) := \left\{u \mid \exists x^{(k)} \to \bar{x},\ \exists u^{(k)} \to u \text{ with } u^{(k)} \in S(x^{(k)})\right\}\,.$$

Based on this, S is said to be *outer semi-continuous* at $\bar{x}$, if

$$\limsup_{x \to \bar{x}} \ S(x) \subset S(\bar{x})\,.$$

Definition A.1. *Consider a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\pm\infty\}$ and a point $\bar{x}$ with $f(\bar{x})$ finite. For a vector $v \in \mathbb{R}^d$, one says that*

(i) v is a regular subgradient *of f at $\bar{x}$, if*

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x}\rangle + o\left(\|x - \bar{x}\|\right)\,.$$

The set of regular subgradients of f at $\bar{x}$, denoted by $\hat{\partial} f(\bar{x})$, is called the regular subdifferential *of f at $\bar{x}$.*

(ii) v is a (general) subgradient *of f at $\bar{x}$, if there are sequences $x^{(n)} \to \bar{x}$ and $v^{(n)} \to v$ with $f(x^{(n)}) \to f(\bar{x})$ and $v^{(n)} \in \hat{\partial} f(x^{(n)})$. The set of subgradients of f at $\bar{x}$, denoted by $\partial f(\bar{x})$, is called the (limiting)* subdifferential *of f at $\bar{x}$.*

Finally, the following definition can be found in Attouch et al. (2013, Definition 2.4, p.7).

Definition A.2. *a) The function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is said to have the* Kurdyka-Łojasiewicz property *at $\bar{x} \in \text{dom } \partial f$, if there exist $\eta \in (0, +\infty]$, a neighborhood U of $\bar{x}$, and a continuous concave function $\varphi : [0, \eta) \to [0, \infty)$, such that*

(i) $\varphi(0) = 0$,
(ii) φ is $\mathcal{C}^1$ on $(0, \eta)$,
(iii) for all $s \in (0, \eta)$, $\varphi'(s) > 0$,
(iv) for all x in $U \cap \{f(\bar{x}) < f < f(\bar{x}) + \eta\}$, the Kurdyka-Łojasiewicz inequality holds

$$\varphi'\left(f(x) - f(\bar{x})\right) \cdot \text{dist}(0, \partial f(x)) \geq 1\,.$$

b) Proper lower semi-continuous functions which satisfy the Kurdyka-Łojasiewicz property at each point of dom ∂f *are called* Kurdyka-Łojasiewicz functions.

Appendix B. Proof of Lemma 4.1

Lemma B.1. *Suppose Assumption 3 holds. Then,* A_{stat} *is closed.*

Proof. Take $(\theta^{(n)}, x^{(n)})_{n\in\mathbb{N}} \subset \mathsf{A}_{\text{stat}}$ with $(\theta^{(n)}, x^{(n)}) \to (\bar{\theta}, \bar{x}) \in P \times S$. We need to show that $(\bar{\theta}, \bar{x}) \in \mathsf{A}_{\text{stat}}$. Since $(x, \theta) \mapsto \partial_x \ell(x, \theta)$ is outer semi-continuous, we have:

$$\limsup_{(x,\theta)\to(\bar{x},\bar{\theta})} \partial_x \ell(x, \theta) \subset \partial_x \ell(\bar{x}, \bar{\theta}) \,.$$

By definition of the outer limit, this is the same as:

$$\left\{ u \in S \mid \exists (x^{(k)}, \theta^{(k)}) \to (\bar{x}, \bar{\theta}),\ \exists u^{(k)} \to u \text{ with } u^{(k)} \in \partial_x \ell(x^{(k)}, \theta^{(k)}) \right\} \subset \partial_x \ell(\bar{x}, \bar{\theta}) \,.$$

In particular, we have that $(\theta^{(n)}, x^{(n)})_{n\in\mathbb{N}} \to (\bar{x}, \bar{\theta})$, and it holds $0 \in \partial_x \ell(\theta^{(n)}, x^{(n)})$ for all $n \in \mathbb{N}$. Thus, setting $u^{(n)} := 0$ for all $n \in \mathbb{N}$ and $u := 0$, we conclude that $0 \in \partial_x \ell(\bar{x}, \bar{\theta})$. Hence, $(\bar{\theta}, \bar{x}) \in \mathsf{A}_{\text{stat}}$, and A_{stat} is closed. □

Now, we can prove Lemma 4.1:

Proof. To show measurability of A_{conv}, we adopt the notation of the *limes inferior* for sets from probability theory: If d is a metric on $P \times S$ and $\varepsilon > 0$, define the set

$$\{\mathsf{B}_\varepsilon(\theta, x) \text{ ult.}\} := \left\{ (\theta', x^{(k)}) \in \mathsf{B}_\varepsilon(\theta, x) \text{ ult.} \right\} := \bigcup_{n\in\mathbb{N}_0} \bigcap_{k\geq n} \left\{ (\theta', x^{(k)}) \in \mathsf{B}_\varepsilon(\theta, x) \right\} .$$

Here, $\left\{ (\theta', x^{(k)}) \in \mathsf{B}_\varepsilon(\theta, x) \right\}$ is a short-hand notation for $\{(\theta', (x^{(n)})_{n\in\mathbb{N}_0}) \in P \times S^{\mathbb{N}_0} : (\theta', x^{(k)}) \in \mathsf{B}_\varepsilon(\theta, x)\}$. Thus, $\{\mathsf{B}_\varepsilon(\theta, x) \text{ ult.}\}$ is the (parametric) set of all sequences in S that *ultimately* lie in the ball with radius ε around (θ, x). Note that $\{\mathsf{B}_\varepsilon(\theta, x) \text{ ult.}\}$ is measurable w.r.t. to the product σ-algebra on $P \times S^{\mathbb{N}_0}$, since it is the countable union/intersection of measurable sets, where $\{(\theta', x^{(k)}) \in \mathsf{B}_\varepsilon(\theta, x)\}$ is measurable, since it can be written as $\{d((\theta', x^{(k)}), (\theta, x)) < \varepsilon\} = (g \circ (id, \mathscr{X}_k))^{-1} [0, \varepsilon)$. Here, id is the identity on P, and $g(\theta', x') := d((\theta', x'), (\theta, x))$ is continuous.
Since the proof does not get more complicated by considering a general Polish space P instead of $\mathbb{R}^p$, we prove the result in this more general setting: If A_{stat} is empty, we get that $\mathsf{A}_{\text{conv}} = \emptyset$, which is measurable. Hence, w.l.o.g. assume that $\mathsf{A}_{\text{stat}} \neq \emptyset$. Since P is Polish, there is a complete metric d_P and a countable dense subset $\hat{P} \subset P$. Then, we have that $d := d_P + \|\cdot - \cdot\|$ is a complete metric on $P \times S$, which metrizes the product topology, and therefore yields the same product σ-algebra. Furthermore, the set $\mathcal{D} := \hat{P} \times \mathbb{Q}^d$ is dense in $P \times S$. We claim that:

$$\mathsf{A}_{\text{conv}} = \bigcap_{k\in\mathbb{N}} \bigcup_{\substack{(\theta,x)\in\mathcal{D} \\ \mathsf{A}_{\text{stat}} \cap \mathsf{B}_{1/k}(\theta,x) \neq \emptyset}} \left\{ \mathsf{B}_{1/k}(\theta, x) \text{ ult.} \right\} =: \mathsf{C} \,.$$

If this equality holds, A_{conv} is measurable as a countable intersection/union of measurable sets. Thus, it remains to show the equality $\mathsf{A}_{\text{conv}} = \mathsf{C}$, which we do by showing both inclusions. Therefore, first, take $(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in \mathsf{A}_{\text{conv}}$. Then there exists $x^* \in S$, such that $(\theta, x^*) \in \mathsf{A}_{\text{stat}}$ and $\lim_{n\to\infty} \|x^{(n)} - x^*\| = 0$. Hence, for any $k \in \mathbb{N}$, there exists $N_k \in \mathbb{N}$,

such that $x^{(n)} \in \mathsf{B}_{1/3k}(x^*)$ for all $n \geq N_k$. Now, take $(\theta_k, x_k) \in \mathcal{D}$, such that $\theta_k \in \mathsf{B}_{1/3k}(\theta)$ and $x_k \in \mathsf{B}_{1/3k}(x^*)$, which exists, since $\mathcal{D}$ is dense. Then, for all $n \geq N_k$ we have:

$$d((\theta, x^{(n)}), (\theta_k, x_k)) = d_P(\theta, \theta_k) + \|x^{(n)} - x_k\| \leq d_P(\theta, \theta_k) + \|x^{(n)} - x^*\| + \|x^* - x_k\| < \frac{1}{k}\,,$$

that is, $(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in \{\mathsf{B}_{1/k}(\theta_k, x_k) \text{ ult.}\}$. Further, we have:

$$d((\theta, x^*), (\theta_k, x_k)) < \frac{2}{3k} < \frac{1}{k}\,.$$

Hence, $(\theta_k, x_k) \in \mathcal{D}$ with $\mathsf{A}_{\text{stat}} \cap \mathsf{B}_{1/k}(\theta_k, x_k) \neq \emptyset$. Since such a tuple $(\theta_k, x_k) \in \mathcal{D}$ can be found for any $k \in \mathbb{N}$, we get:

$$(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in \bigcup_{\substack{(\theta', x)\in\mathcal{D} \\ \mathsf{A}_{\text{stat}}\cap\mathsf{B}_{1/k}(\theta', x)\neq\emptyset}} \{\mathsf{B}_{1/k}(\theta', x) \text{ ult.}\}, \quad \forall k \in \mathbb{N}\,.$$

Then, however, this implies $(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in \mathsf{C}$, which shows the inclusion $\mathsf{A}_{\text{conv}} \subset \mathsf{C}$. Now, conversely, let $(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in \mathsf{C}$. Then, for every $k \in \mathbb{N}$ there exists $(\theta_k, x_k) \in \mathcal{D}$ with $\mathsf{A}_{\text{stat}} \cap \mathsf{B}_{1/k}(\theta_k, x_k) \neq \emptyset$, and a $N_k \in \mathbb{N}$, such that

$$(\theta, x^{(n)}) \in \mathsf{B}_{1/k}(\theta_k, x_k), \quad \forall n \geq N_k\,.$$

The resulting sequence of midpoints $(\theta_k, x_k)_{k\in\mathbb{N}}$ is Cauchy in $P \times S$, because: For $k, l \in \mathbb{N}$, we have that $(\theta, x^{(n)}) \in \mathsf{B}_{1/k}(\theta_k, x_k)$ for all $n \geq N_k$, and $(\theta, x^{(n)}) \in \mathsf{B}_{1/l}(\theta_l, x_l)$ for all $n \geq N_l$. Thus, for $n \geq N := \max\{N_k, N_l\}$, we get $(\theta, x^{(n)}) \in \mathsf{B}_{1/k}(\theta_k, x_k) \cap \mathsf{B}_{1/l}(\theta_l, x_l)$, which allows for the following bound:

$$\begin{aligned} d((\theta_k, x_k), (\theta_l, x_l)) &\leq d((\theta_k, x_k), (\theta, x^{(N_k)})) + d((\theta, x^{(N_k)}), (\theta, x^{(N)})) \\ &\quad + d((\theta, x^{(N)}), (\theta, x^{(N_l)})) + d((\theta, x^{(N_l)}), (\theta_l, x_l)) \\ &\leq \frac{1}{k} + \frac{2}{k} + \frac{2}{l} + \frac{1}{l} \leq \frac{3}{k} + \frac{3}{l} \overset{k,l\to\infty}{\to} 0\,. \end{aligned}$$

Hence, by completeness of $P \times S$, the sequence $(\theta_k, x_k)_{k\in\mathbb{N}_0}$ has a limit (θ^*, x^*) in $P \times S$. First, we show that $\theta^* = \theta$: Since $(\theta, x^{(N_k)}) \in \mathsf{B}_{1/k}(\theta_k, x_k)$ for all $k \in \mathbb{N}$, we have by continuity of the metric:

$$d_P(\theta, \theta^*) = \lim_{k\to\infty} d_P(\theta, \theta_k) \leq \lim_{k\to\infty} d((\theta, x^{(N_k)}), (\theta_k, x_k)) \leq \lim_{k\to\infty} \frac{1}{k} = 0\,.$$

Thus, actually, $(\theta_k, x_k) \to (\theta, x^*)$. Second, we show that $(\theta, x^*) \in \mathsf{A}_{\text{stat}}$, that is, $x^* \in \mathsf{A}_{\text{stat},\theta}$: Assume the contrary, that is, $(\theta, x^*) \in \mathsf{A}^c_{\text{stat}}$. By Lemma B.1, the set A_{stat} is closed. Thus, its complement $\mathsf{A}^c_{\text{stat}}$ is open, and there exists $\varepsilon > 0$ with $\mathsf{B}_\varepsilon(\theta, x^*) \subset \mathsf{A}^c_{\text{stat}}$, that is, $\mathsf{B}_\varepsilon(\theta, x^*) \cap \mathsf{A}_{\text{stat}} = \emptyset$. Since $(\theta_k, x_k) \to (\theta, x^*)$, there exists $N \in \mathbb{N}$, such that $d((\theta_k, x_k), (\theta, x^*)) < \frac{\varepsilon}{3}$ for all $k \geq N$. Then, however, taking $k \geq N$ with $\frac{1}{k} < \frac{\varepsilon}{3}$, we conclude that

$$\mathsf{B}_{1/k}(\theta_k, x_k) \cap \mathsf{A}_{\text{stat}} = \emptyset\,.$$

By definition of the sequence $(\theta_k, x_k)_{k\in\mathbb{N}}$, this is a contradiction. Hence, we have $(\theta, x^*) \in \mathsf{A}_{\text{stat}}$, and it remains to show that also the sequence $(x^{(n)})_{n\in\mathbb{N}_0}$ converges to x^*. For this, assume the contrary again. Then there exists an $\varepsilon > 0$ with the property that for all $N \in \mathbb{N}$, one can find a $\tilde{n} \geq N$, such that $\|x^{(\tilde{n})} - x^*\| \geq \varepsilon$. Now, choose $k \in \mathbb{N}$ large enough, such that $\|x_k - x^*\| \leq \frac{\varepsilon}{3}$ and $\frac{1}{k} < \frac{\varepsilon}{3}$. Then, since $(\theta, x^{(n)}) \in \mathcal{B}_{1/k}(\theta_k, x_k)$ for all $n \geq N_k$, we have for all $n \geq N_k$:

$$\|x^{(n)} - x^*\| \leq \|x^{(n)} - x_k\| + \|x_k - x^*\| \leq \frac{2\varepsilon}{3} < \varepsilon\,.$$

Again, this is a contradiction and such an $\varepsilon > 0$ cannot exists. Thus, $(x^{(n)})$ converges to $x^* \in \mathsf{A}_{\text{stat},\theta}$, and we have $(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in \mathsf{A}_{\text{conv}}$, which concludes the proof. □

Appendix C. Existence of Measurable Selection and Proof of Lemma 4.3

Definition C.1. *A set-valued mapping $S : T \rightrightarrows \mathbb{R}^d$ is* measurable*, if for every open set $\mathsf{O} \subset \mathbb{R}^d$ the set $S^{-1}(\mathsf{O}) \subset T$ is measurable. In particular,* dom S *must be measurable.*

Lemma C.2. *Suppose Assumption 3 holds. Then $(x, \theta) \mapsto \partial_x \ell(x, \theta)$ is closed-valued and measurable.*

Proof. Since $\partial_x \ell(x, \theta)$ is the subdifferential of $\ell(\cdot, \theta)$ at x, by Rockafellar and Wets (2009, Theorem 8.6, p.302), we have that, for every $\bar{\theta} \in P$, and every $\bar{x} \in$ dom $\ell(\cdot, \bar{\theta})$, the set $\partial_x \ell(\bar{x}, \bar{\theta})$ is closed. Hence, we have that $\partial_x \ell(\bar{x}, \bar{\theta})$ is closed for every $(\bar{x}, \bar{\theta}) \in$ dom ℓ. Further, for $(\bar{x}, \bar{\theta}) \notin$ dom ℓ, we have $\partial_x \ell(\bar{x}, \bar{\theta}) = \emptyset$, which is closed, too. Therefore, $(\bar{x}, \bar{\theta}) \mapsto \partial_x \ell(\bar{x}, \bar{\theta})$ is closed-valued. Finally, since $(\bar{x}, \bar{\theta}) \mapsto \partial_x \ell(\bar{x}, \bar{\theta})$ is also outer semi-continuous, Rockafellar and Wets (2009, Exercise 14.9, p.649) implies that $\partial_x \ell$ is measurable w.r.t. $\mathcal{B}(S \times P)$. □

Corollary C.3. *Suppose Assumption 3 holds. Then there exists a measurable selection for $\partial_x \ell$, that is, a measurable map $v :$ dom $\partial_x \ell \to S$, such that $v(x, \theta) \in \partial_x \ell(x, \theta)$ for all $(x, \theta) \in S \times P$.*

Proof. By Lemma C.2, the map $(x, \theta) \mapsto \partial_x \ell(x, \theta)$ is closed-valued and measurable. Hence, the result follows directly from Rockafellar and Wets (2009, Corollary 14.6, p.647). □

Now, we can prove Lemma 4.3:

Proof. Since we can again restrict to $b \in \mathbb{Q} \cap (0, \infty) =: \mathbb{Q}_+$, A_{err} can be written as:

$$\mathsf{A}_{\text{err}} = \left(\bigcup_{b\in\mathbb{Q}_+} \bigcap_{k\in\mathbb{N}_0} \mathsf{B}_{b,k} \right) \cap \left(\bigcap_{k\in\mathbb{N}_0} \{(x^{(k)}, \theta) \in \text{dom } \partial_x \ell\} \right)\,,$$

where $\mathsf{B}_{b,k}$ is given by:

$$\mathsf{B}_{b,k} := \left\{ (\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \in P \times S^{\mathbb{N}_0} \;:\; \|v(x^{(k+1)}, \theta)\| \leq b\|x^{(k+1)} - x^{(k)}\| \right\}.$$

Hence, since σ-algebras are stable under countable unions/intersections, we only have to show measurability of the sets $\mathsf{B}_{b,k}$ and $\{(\theta, x^{(k)}) \in \text{dom } \partial_x \ell\}$. Here, it holds that:

$$\{(x^{(k)}, \theta) \in \text{dom } \partial_x \ell\} = (\mathscr{X}_k, id)^{-1} (\text{dom } \partial_x \ell)\,,$$

where id is the identity on P. By Lemma C.2, dom $\partial_x \ell$ is measurable, such that $\{(\theta, x^{(k)}) \in \text{dom } \partial_x \ell\}$ is measurable for each $k \in \mathbb{N}_0$. Thus, it remains to show the measurability of the set $\mathsf{B}_{b,k}$. For this, introduce the function $g_b : (\text{dom } \partial_x \ell)^2 \to \mathbb{R},\ ((x_1, \theta_1), (x_2, \theta_2)) \mapsto \|v(x_2, \theta_2)\| - b\|x_2 - x_1\|$. Since v is measurable, and the norm is continuous, we have that g_b is measurable. With this, we can write the set $\mathsf{B}_{b,k}$ as:

$$\begin{aligned}\mathsf{B}_{b,k} &= \{g_b(x^{(k)}, \theta, x^{(k+1)}, \theta) \leq 0\} \\ &= \{(g_b \circ (\mathscr{X}_k, id, \mathscr{X}_{k+1}, id) \circ \iota)\,(\theta, (x^{(n)})_{n\in\mathbb{N}_0}) \leq 0\} \\ &= (g_b \circ (\mathscr{X}_k, id, \mathscr{X}_{k+1}, id) \circ \iota)^{-1}\,(-\infty, 0]\,,\end{aligned}$$

where $\iota : P \times S^{\mathbb{N}_0} \to (S^{\mathbb{N}_0} \times P)^2$ is the diagonal inclusion $(x_1, x_2) \mapsto ((x_2, x_1), (x_2, x_1))$, which again is measurable. Thus, $\mathsf{B}_{b,k}$ is measurable for each $k \in \mathbb{N}_0$ and $b \in \mathbb{Q}_+$, which concludes the proof. □

Appendix D. Proof of Lemma 4.4

By definition of the product σ-algebra on $P \times S^{\mathbb{N}_0}$, it suffices to show that $\tilde{\mathsf{A}}_{\text{bound}}$ is measurable. Then, as it suffices to consider $c \in [0, \infty) \cap \mathbb{Q} =: \mathbb{Q}_+$, one can write $\tilde{\mathsf{A}}_{\text{bound}}$ as:

$$\tilde{\mathsf{A}}_{\text{bound}} = \bigcup_{c\in\mathbb{Q}_+} \bigcap_{k\in\mathbb{N}_0} \underbrace{\{(x^{(n)})_{n\in\mathbb{N}_0} \in S^{\mathbb{N}_0} \ :\ \|x^{(k)}\| \leq c\}}_{=:\mathsf{C}_{c,k}}.$$

Thus, by the properties of a σ-algebra, it suffices to show that the sets $\mathsf{C}_{c,k}$ with $c \in \mathbb{Q}_+$ and $k \in \mathbb{N}_0$ are measurable. By defining $g(x) = \|x\|$, this follows directly from the identity $\mathsf{C}_{c,k} = (g \circ \mathscr{X}_k)^{-1}\,[0, c]$.

Appendix E. Architecture of the Algorithm for Quadratic Problems

The algorithmic update is adopted from Sucker and Ochs (2024) and consists of two blocks:

1) The first block consists of 1×1-convolutional layers with ReLU-activation functions and computes the update direction $d^{(n)}$. As features, we use the normalized gradient $d_1^{(n)} := \frac{\nabla \ell(x^{(n)}, \theta)}{\|\nabla \ell(x^{(n)}, \theta)\|}$, the normalized momentum term $d_2^{(n)} := \frac{x^{(n)} - x^{(n-1)}}{\|x^{(n)} - x^{(n-1)}\|}$, and their coordinate-wise product $d_1^{(n)} \odot d_2^{(n)}$. The normalization is done to stabilize the training.

2) The second block consists of linear layers with ReLU-activation functions and computes the step-size $\beta^{(n)}$. As features, we use the (logarithmically transformed) gradient norm $s_1^{(n)} := \log\big(1 + \|\nabla \ell(x^{(n)}, \theta)\|\big)$, the (logarithmically transformed) norm of the momentum term $s_2^{(n)} := \log\big(1 + \|x^{(n)} - x^{(n-1)}\|\big)$, and the current and previous (logarithmically transformed) losses $s_3^{(n)} := \log\big(1 + \ell(x^{(n)}, \theta)\big)$, $s_4^{(n)} := \log\big(1 + \ell(x^{(n-1)}, \theta)\big)$. Again, the logarithmic scaling is done to stabilize training. Here, the term "+1" is added to map zero onto zero.

Importantly, we want to stress that the algorithmic update is not constrained in any way: the algorithm just predicts a direction and a step-size, and we do not enforce them to have any specific properties.

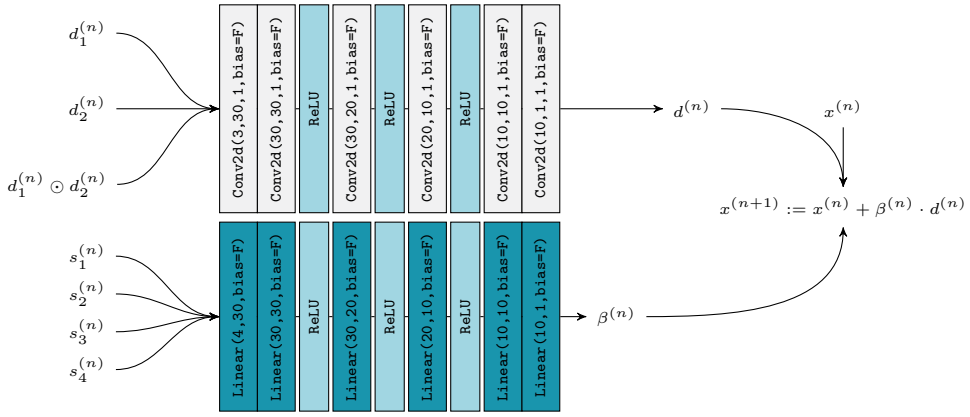

Figure 3: Update step of $\mathcal{A}$: The directions $d_1^{(n)}$, $d_2^{(n)}$ and $d_1^{(n)} \odot d_2^{(n)}$ are inserted as different channels into the `Conv2d`-block, which performs 1×1 "convolutions", that is, the algorithm acts coordinate-wise on the input. The scales $s_1^{(n)}$, ..., $s_4^{(t)}$ get transformed separately by the fully-connected block.

Appendix F. Training of the Algorithm

For training, we mainly use the procedure proposed (and described in detail) by Sucker et al. (2024); Sucker and Ochs (2024). For completeness, we briefly summarize it here: In the outer loop, we sample a loss-function $\ell(\cdot, \theta)$ randomly from the training set. Then, in the inner loop, we train the algorithm on this loss-function with ℓ_{train} given by

$$\ell_{\text{train}}(\alpha, \theta, \xi^{(n)}) = \mathbb{1}\{\ell(\xi^{(n)}, \theta) > 0\} \frac{\ell(\xi^{(n+1)}, \theta)}{\ell(\xi^{(n)}, \theta)} \cdot \mathbb{1}_{\mathsf{C}^c}(\xi^{(n)}, \theta)\,,$$

where $\mathsf{C} := \{(\theta, x) \in P \times S \;:\; \ell(x, \theta) < 10^{-16}\}$ is the convergence set. That is, in each iteration the algorithm computes a new point and observes the loss ℓ_{train}, which is used to update its hyperparameters. We run this procedure for $150 \cdot 10^3$ iterations. This yields hyperparameters $\alpha^{(0)} \in H$, such that $\mathcal{A}(\alpha^{(0)}, \cdot, \cdot)$ has a good performance. However, typically, it is not a descent method yet, that is, $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}=\alpha^{(0)}}\{\mathsf{A}\}$ is small, such that the PAC-bound would be useless. Therefore, we employ the probabilistic constraining procedure proposed (and described in detail) by Sucker et al. (2024) in a progressive way: Starting from $\alpha^{(0)}$, we try to find a sequence of hyperparameters $\alpha^{(1)}, \alpha^{(2)}, ...$, such that

$$\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}=\alpha^{(0)}}\{\mathsf{A}\} < \mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}=\alpha^{(1)}}\{\mathsf{A}\} < \mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}=\alpha^{(2)}}\{\mathsf{A}\} < \dots\,.$$

Remark F.1. *The notation $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}\}$ is not entirely correct and is rather to be understood suggestively, as the final prior distribution $\mathbb{P}_{\mathscr{H}}$ is yet to be constructed. However, we think that it is easier to understand this way and therefore allow for this inaccuracy.*

For this, we test the probabilistic constraint every 1000 iterations, that is: Given $\alpha^{(i)}$, we train the algorithm (as before) for another 1000 iterations, which yields a candidate $\tilde{\alpha}^{(i+1)}$. If $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}=\alpha^{(i)}}\{\mathsf{A}\} < \mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}=\tilde{\alpha}^{(i+1)}}\{\mathsf{A}\}$, we accept $\alpha^{(i+1)} := \tilde{\alpha}^{(i+1)}$, otherwise

we reject it and start again from $\alpha^{(i)}$. This finally yields some hyperparameters α_0 that have a good performance and such that $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}=\alpha_0}\{\mathsf{A}\}$ is large enough (here: about 90%). Then, starting from α_0, we construct the *actual* discrete prior distribution $\mathbb{P}_{\mathscr{H}}$ over points $\alpha_1, ..., \alpha_{n_{\text{sample}}} \in H$, by a sampling procedure. Finally, we perform the (closed-form) PAC-Bayesian optimization step, which yields the posterior $\rho^* \in \mathcal{P}(\mathbb{P}_{\mathscr{H}})$. In the end, for simplicity, we set the hyperparameters to

$$\alpha^* = \underset{i=1,...,n_{\text{sample}}}{\arg\max} \; \rho^*\{\alpha_i\}\,.$$

For the construction of the prior, we use $N_{\text{prior}} = 500$ functions, for the probabilistic constraint we use $N_{\text{val}} = 500$ functions, and for the PAC-Bayesian optimization step we use $N_{\text{train}} = 250$ functions, all of which are sampled i.i.d., that is, the data sets are independent of each other.

Remark F.2. *Training the algorithm to yield a good performance is comparably easy. On the other hand, turning it into an algorithm, such that $\mathbb{P}_{(\mathscr{P},\xi)|\mathscr{H}}\{\mathsf{A}\}$ is large enough (in our case: a descent method without enforcing it geometrically) is challenging and, unfortunately, not guaranteed to work. Nevertheless, it is key to get a meaningful guarantee.*

Appendix G. Architecture of the Algorithm for Training the Neural Network

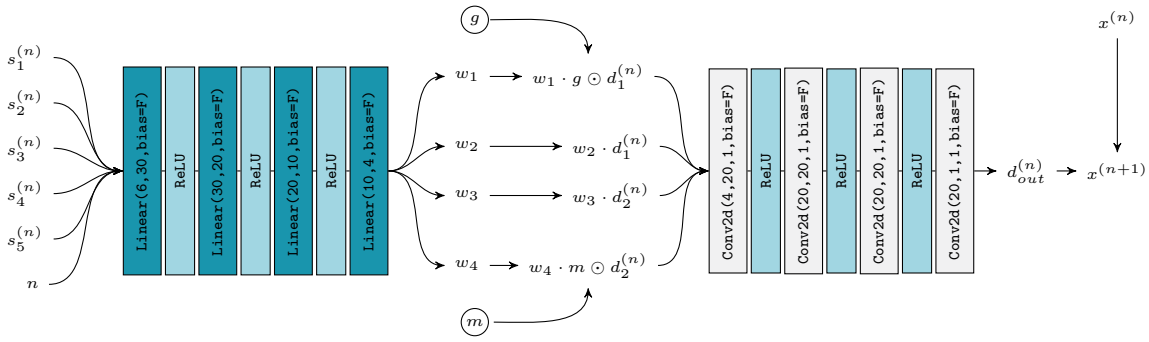

Figure 4: Algorithmic update for training the neural network: Based on the given six features, the first block computes four weights $w_1, ..., w_4$, which are used to perform a weighting of the different directions $g \odot d_1^{(n)}$, $d_1^{(n)}$, $d_2^{(n)}$, $m \odot d_2^{(n)}$, which are used in the second block. This second block consists of a 1x1-convolutional blocks, which compute an update direction $d_{out}^{(n)}$. Then, we update $x^{(n+1)} := x^{(n)} + d_{out}^{(n)}/\sqrt{n}$.

The algorithmic update is adopted from Sucker and Ochs (2024) and consists of two blocks:

1) The first block consists of linear layers with ReLU-activation functions and computes four weights $w_1, ..., w_4$. As features, we use the (logarithmically transformed) gradient norm $s_1^{(n)} := \log\big(1 + \|\nabla \ell(x^{(n)}, \theta)\|\big)$, the (logarithmically transformed) norm of the

momentum term $s_2^{(n)} := \log\left(1 + \|x^{(n)} - x^{(n-1)}\|\right)$, the difference between the current and previous loss $s_3^{(n)} := \ell(x^{(n)}, \theta) - \ell(x^{(n-1)}, \theta)$, the scalarproduct between the (normalized) gradient and the (normalized) momentum term $s_4^{(n)}$, the maximal absolute value of the coordinates of the gradient $s_5^{(n)}$, and the iteration counter n.

2) The second block consists of 1×1-convolutional layers with ReLU-activation functions and computes the update direction $d_{\text{out}}^{(n)}$. As features, we use the normalized gradient $d_1^{(n)} := \frac{\nabla \ell(x^{(n)}, \theta)}{\|\nabla \ell(x^{(n)}, \theta)\|}$, the normalized momentum term $d_2^{(n)} := \frac{x^{(n)} - x^{(n-1)}}{\|x^{(n)} - x^{(n-1)}\|}$, and their "preconditioned" versions $g \odot d_1^{(n)}$ and $m \odot d_2^{(n)}$, where the weights $m, d \in \mathbb{R}^d$ are learned, too.

Again, we want to stress that the algorithmic update is not constrained in any way: the algorithm just predicts a direction, and we do not enforce them to have any specific properties.

References

Pierre-Antoine Absil, Robert Mahony, and Ben Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization*, 16(2):531–547, 2005.

Pierre Alquier. User-friendly introduction to PAC-Bayes bounds. *arXiv preprint arXiv:2110.11216*, 2021.

Pierre Alquier and Benjamin Guedj. Simpler PAC-Bayesian bounds for hostile data. *Machine Learning*, 107(5):887–902, 2018.

Ron Amit, Baruch Epstein, Shay Moran, and Ron Meir. Integral probability metrics PAC-Bayes bounds. *Advances in Neural Information Processing Systems*, 35:3123–3136, 2022.

Hedy Attouch and Jérôme Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116:5–16, 2009.

Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality. *Mathematics of operations research*, 35(2):438–457, 2010.

Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1):91–129, 2013.

Luc Bégin, Pascal Germain, François Laviolette, and Jean-Francis Roy. PAC-Bayesian bounds based on the Rényi divergence. In *Artificial Intelligence and Statistics*, pages 435–444. PMLR, 2016.

Edward Bierstone and Pierre D Milman. Semianalytic and subanalytic sets. *Publications Mathématiques de l'IHÉS*, 67:5–42, 1988.

Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007a.

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

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1): 459–494, 2014.

Gregery T Buzzard, Stanley H Chan, Suhas Sreehari, and Charles A Bouman. Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium. *SIAM Journal on Imaging Sciences*, 11(3):2001–2020, 2018.

Camille Castera and Peter Ochs. From learning to optimize to learning optimization algorithms. *arXiv preprint arXiv:2405.18222*, 2024.

Olivier Catoni. *Statistical learning theory and stochastic optimization: Ecole d'Eté de Probabilités de Saint-Flour, XXXI-2001*, volume 1851. Springer Science & Business Media, 2004.

Olivier Catoni. PAC-Bayesian supervised classification: The thermodynamics of statistical learning. *Lecture Notes-Monograph Series*, 56:i–163, 2007.

Stanley H Chan, Xiran Wang, and Omar A Elgendy. Plug-and-play ADMM for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2016.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *arXiv preprint arXiv:2103.12828*, 2021.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. *Advances in Neural Information Processing Systems*, 31, 2018.

Xinshi Chen, Yufei Zhang, Christoph Reisinger, and Le Song. Understanding deep architecture with reasoning layer. *Advances in Neural Information Processing Systems*, 33: 1240–1252, 2020.

Regev Cohen, Michael Elad, and Peyman Milanfar. Regularization by denoising via fixed-point projection. *SIAM Journal on Imaging Sciences*, 14(3):1374–1406, 2021.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

Gintare Karolina Dziugaite and Daniel M Roy. Data-dependent PAC-Bayes priors via differential privacy. *Advances in neural information processing systems*, 31, 2018.

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel Roy. On the role of data in PAC-Bayes bounds. In *International Conference on Artificial Intelligence and Statistics*, pages 604–612. PMLR, 2021.

Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 353–360, 2009.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.

Benjamin Guedj. A primer on PAC-Bayesian learning. In *Proceedings of the second congress of the French Mathematical Society*, volume 33, 2019.

Maxime Haddouche and Benjamin Guedj. Wasserstein PAC-Bayes learning: Exploiting optimisation guarantees to explain generalisation, 2023.

Xin He, Kaiyong Zhao, and Xiaowen Chu. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.

Howard Heaton, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Safeguarded learned convex optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7848–7855, 2023.

Jean Honorio and Tommi Jaakkola. Tight bounds for the expected risk of linear classifiers and PAC-Bayes finite-sample guarantees. In *Artificial Intelligence and Statistics*, pages 384–392. PMLR, 2014.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

O. Kallenberg. *Foundations of Modern Probability*. Probability theory and stochastic modelling. Springer, 2021. ISBN 9783030618728.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Achim Klenke. *Wahrscheinlichkeitstheorie*. Springer Spektrum, 2013.

Erich Kobler, Alexander Effland, Karl Kunisch, and Thomas Pock. Total deep variation: A stable regularizer for inverse problems. *arXiv preprint arXiv:2006.08789*, 2020.

Krzysztof Kurdyka. On gradients of functions definable in o-minimal structures. In *Annales de l'institut Fourier*, volume 48, pages 769–783, 1998.

John Langford and Rich Caruana. (Not) bounding the true error. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

John Langford and John Shawe-Taylor. PAC-Bayes and margins. *Advances in neural information processing systems*, 15, 2002.

Guy Lever, François Laviolette, and John Shawe-Taylor. Tighter PAC-Bayes bounds through distribution-dependent priors. *Theoretical Computer Science*, 473:4–28, 2013.

Jialin Liu, Xiaohan Chen, Zhangyang Wang, Wotao Yin, and HanQin Cai. Towards constituting mathematical structures for learning to optimize. In *International Conference on Machine Learning*, pages 21426–21449. PMLR, 2023.

Ben London. A PAC-Bayesian analysis of randomized learning with application to stochastic gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017.

David McAllester. Simplified PAC-Bayesian margin bounds. In *Learning theory and Kernel machines*, pages 203–215. Springer, 2003a.

David McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51(1):5–21, 2003b.

Luke Metz, Niru Maheswaranathan, Jeremy Nixon, Daniel Freeman, and Jascha Sohl-Dickstein. Understanding and correcting pathologies in the training of learned optimizers. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4556–4565. PMLR, 2019.

Luke Metz, C Daniel Freeman, James Harrison, Niru Maheswaranathan, and Jascha Sohl-Dickstein. Practical tradeoffs between memory, compute, and performance in learned optimizers. In *Conference on Lifelong Learning Agents*, pages 142–164. PMLR, 2022.

Michael Moeller, Thomas Mollenhoff, and Daniel Cremers. Controlling neural networks via energy dissipation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3256–3265, 2019.

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

Peter Ochs. Unifying abstract inexact convergence theorems and block coordinate variable metric ipiano. *SIAM Journal on Optimization*, 29(1):541–570, 2019.

Peter Ochs, Yunjin Chen, Thomas Brox, and Thomas Pock. ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

Yuki Ohnishi and Jean Honorio. Novel change of measure inequalities with applications to PAC-Bayesian bounds and Monte Carlo estimation. In *International Conference on Artificial Intelligence and Statistics*, pages 1711–1719. PMLR, 2021.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. PAC-Bayes bounds with data dependent priors. *Journal of Machine Learning Research*, 13(1):3507–3531, 2012.

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*, volume 32. Curran Associates, Inc., 2019.

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22(227): 1–40, 2021.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

B.T. Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

Isabeau Prémont-Schwarz, Jaroslav Vítků, and Jan Feyereisl. A simple guard for learned optimizers. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 17910–17925. PMLR, 2022.

R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Plug-and-play methods provably converge with properly trained denoisers. In *International Conference on Machine Learning*, pages 5546–5557. PMLR, 2019.

Matthias Seeger. PAC-Bayesian generalisation error bounds for Gaussian process classification. *Journal of Machine Learning Research*, 3:233–269, 2002.

Suhas Sreehari, S Venkat Venkatakrishnan, Brendt Wohlberg, Gregery T Buzzard, Lawrence F Drummy, Jeffrey P Simmons, and Charles A Bouman. Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging*, 2(4):408–423, 2016.

Michael Sucker and Peter Ochs. Pac-bayesian learning of optimization algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 8145–8164. PMLR, 2023.

Michael Sucker and Peter Ochs. A Markovian model for learning-to-optimize. *arXiv preprint arXiv:2408.11629*, 2024.

Michael Sucker, Jalal Fadili, and Peter Ochs. Learning-to-optimize with PAC-Bayesian guarantees: Theoretical considerations and practical implementation. *arXiv preprint arXiv:2404.03290*, 2024.

Yu Sun, Brendt Wohlberg, and Ulugbek S Kamilov. An online plug-and-play algorithm for regularized image reconstruction. *IEEE Transactions on Computational Imaging*, 5(3): 395–408, 2019.

Afonso M Teodoro, José M Bioucas-Dias, and Mário AT Figueiredo. Scene-adapted plug-and-play algorithm with convergence guarantees. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2017.

Matthieu Terris, Audrey Repetti, Jean-Christophe Pesquet, and Yves Wiaux. Enhanced convergent pnp algorithms for image restoration. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1684–1688. IEEE, 2021.

Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin. A strongly quasiconvex PAC-Bayesian bound. In *International Conference on Algorithmic Learning Theory*, pages 466–492. PMLR, 2017.

Tom Tirer and Raja Giryes. Image restoration by iterative denoising and backward projections. *IEEE Transactions on Image Processing*, 28(3):1220–1234, 2018.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18:77–95, 2002.

Olga Wichrowska, Niru Maheswaranathan, Matthew W Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando Freitas, and Jascha Sohl-Dickstein. Learned optimizers that scale and generalize. In *International conference on machine learning*, pages 3751–3760. PMLR, 2017.

Zhonglin Xie, Wotao Yin, and Zaiwen Wen. Ode-based learning to optimize. *arXiv preprint arXiv:2406.02006*, 2024.

Bo Xin, Yizhou Wang, Wen Gao, David Wipf, and Baoyuan Wang. Maximal sparsity with deep networks? *Advances in Neural Information Processing Systems*, 29, 2016.

Quanming Yao, Mengshuo Wang, Yuqiang Chen, Wenyuan Dai, Yu-Feng Li, Wei-Wei Tu, Qiang Yang, and Yang Yu. Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint arXiv:1810.13306*, 2018.