# How Implicit Regularization of ReLU Neural Networks Characterizes the Learned Function

## Part I: the 1-D Case of Two Layers with Random First Layer

# HOW IMPLICIT REGULARIZATION OF NEURAL NETWORKS AFFECTS THE LEARNED FUNCTION - PART I

JAKOB HEISS, JOSEF TEICHMANN AND HANNA WUTTE

Abstract. Today, various forms of neural networks are trained to perform approximation tasks in many fields. However, the solutions obtained are not fully understood. Empirical results suggest that typical training algorithms favor regularized solutions. These observations motivate us to analyze properties of the solutions found by gradient descent initialized close to zero, that is frequently employed to perform the training task. As a starting point, we consider one dimensional (shallow) ReLU neural networks in which weights are chosen randomly and only the terminal layer is trained. We show that the resulting solution converges to the smooth spline interpolation of the training data as the number of hidden nodes tends to infinity. Moreover, we derive a correspondence between the early stopped gradient descent and the smoothing spline regression. This might give valuable insight on the properties of the solutions obtained using gradient descent methods in general settings.

## 1. Introduction

Even though neural networks are becoming increasingly popular in supervised learning, their theoretical understanding is still very limited. The most important open questions in the mathematical theory of neural networks nowadays include the following:[1]

I. **Generalization**: Why and under which conditions can neural networks make good predictions of the output for new unseen input data even though they have only been trained on finitely many data points? How does the trained function behave out of sample? How can one get control of over-fitting?

II. **Gradient Descent**: When training neural networks, a typically very high-dimensional non-convex optimization problem is claimed to be solved by (stochastic) gradient descent quite fast. There is relatively good understanding of how this algorithm evolves in long term, in particular seen from the point of view of simulated annealing. However, what happens if the algorithm is stopped early after a realistic number of steps depending on a certain starting point?

III. **Expressiveness:** How expressive are neural networks (with a finite number of nodes)? [31, 3, 16, 22]

IV. **Summary**: What are the advantages and disadvantages of different architectures? What are the advantages and disadvantages of considering neural networks in approximation/prediction tasks compared to other methods such as Random Forests or Kernel-based Gaussian processes? In both theory and applications, it is of great

The authors gratefully acknowledge the support from ETH-foundation. We are very thankful for numerous helpful discussions, feedback, corrections and proof reading—especially to: Lukas Fertl, Peter Mühlbacher, Martin Štefánik, Alexis Stockinger and Jakob Weissteiner.

[1]The literature agrees with questions I–III to be central [29]. Question IV motivates the importance of questions I–III by summarizing them and concluding their implications.

interest to gain a precise understanding of IV, much of which could be achieved by answering I–III.

The goal of this work is to contribute to answering these questions by rigorously proving Theorems 3.8 and 3.17 that almost completely resolve question II (cp. eq. (28)) for the restricted class of wide **r**andomized **s**hallow neural **n**etworks (RSNs) with ReLU activation (i.e., wRRSNs). These answers together with the intuition acquired from sections 1.1 and 1.2 give quite extensive insights to I and thus IV.[2]

The result of this work can be seen in analogy to mean field theory in thermodynamics: like we are understanding the collision behavior of each particle, we understand the training behavior of each neuron[3, 4]. However, due to the extensive number of interactions between particles/neurons the complexity increases in a way that the individual behavior of a particle/neuron does no longer give direct insight into the overall system's behavior. In both cases, taking the limit to infinity allows to precisely derive the system's behavior in terms of interpretable macroscopic laws/theorems (see Theorem 3.8[5]).

### 1.1. The Regression Problem as a basis for Machine Learning.

Throughout this paper, we consider the task of supervised learning, for which the setting is typically introduced as follows.

Let $\mathcal{X}$ respectively $\mathcal{Y}$ be an input and output space. Assume further, we observe a finite number $N \in \mathbb{N}$ of i.i.d. samples $(x_i^{\text{train}}, y_i^{\text{train}}) \in \mathcal{X} \times \mathcal{Y}$ with $i \in \{1, \dots, N\}$ from an *unknown* probability distribution $\mathbb{P}_D$ on $\mathcal{X} \times \mathcal{Y}$. Given an additional realization $(X, Y)(\omega)$ of $(X, Y) \sim \mathbb{P}_D$, for which we can only observe $X(\omega)$ but not $Y(\omega)$, the goal is to make a suitable prediction $\hat{f}(X(\omega))$ of $Y(\omega)$. Thus, for a given cost function $C : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, we are interested in an estimator $\hat{f} : \mathcal{X} \to \mathcal{Y}$ with low risk, i.e., for which the expected cost $\mathbb{E}\left[C\left(\hat{f}(X), Y\right)\right]$ is minimal. However, since $\mathbb{P}_D$ is unknown, this risk cannot be calculated. In supervised machine learning, one hence tries to learn an estimator $\hat{f}$ based on the given training data $(x_i^{\text{train}}, y_i^{\text{train}})_{i \in \{1,\dots,N\}}$. A common heuristic[6] is to minimize a suitable training loss

$$L(f) := \sum_{i=1}^{N} l\left(f(x_i^{\text{train}}), y_i^{\text{train}}\right) \tag{1}$$

for a chosen loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ over a suitable class of functions $\mathcal{H}$, i.e.,

$$\min_{f \in \mathcal{H}} L(f).$$

[2] We also contribute to answering question III within the results marked with a "*": Remark 2.2, Corollary 2.3, Lemma 2.4 and Remark 2.5 in Section 2. These results form an independent story line.

[3] In this work, only *artificial* neural networks are considered. Thus, terms such as 'neurons' and 'neural networks' do not refer to actual biological neurons but rather to their artificial counterparts. The term "node" will be used interchangeable with the term "neuron".

[4] Notation remark: To improve readability of the paper, we use partially transparent (grey) fonts to encourage the reader to skip these details.

[5] Theorem 3.8 results from letting the number of neurons $n$ tend to infinity. In thermodynamics, Brownian motion particle movements or heat equations result from taking the limit of the number of particles to infinity.

[6] Historically, the squared loss $l(\hat{y}, y) := (\hat{y} - y)^2$ has often been used (in the case of regression). In the literature, minimizing the training loss $L$ is motivated sometimes as empirical cost minimization (or empirical risk minimization) if $C \propto l$ and sometimes as maximum log-likelihood method if the logarithm of the density of the noise $Y - \mathbb{E}[Y|X]$ is proportional to $l$.

*Remark* 1.1 (Setting). Throughout this work, we consider $\mathcal{X} = \mathbb{R}^d$ with input dimension $d \in \mathbb{N}$ and $\mathcal{Y} = \mathbb{R}$. In such a setting, we speak of supervised learning and regression interchangeably. Moreover, the (non-negative) loss function $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is generally assumed to be convex and continuously differentiable in the first component (see Assumption 4).[7]

It is important to note that in Section 3.1 we derive this papers main contribution Theorem 3.8 for $d = 1$. In future work, this result will be extended to $d \geq 2$. By contrast, the results presented in Section 3.2 hold true for general input dimension $d \in \mathbb{N}$. However, Theorem 3.17 linking the network resulting from gradient descent to the ridge network (with explicitly regularized parameters) is derived for $l(\hat{y}, y) := (\hat{y} - y)^2$ (Assumption 5; see also Remark 3.18 ).

Historically, linear regression [10, 11, 21] was among the first methods used within supervised learning. Here, one restricts oneself to a tiny subspace of all functions: the space of (affine-)linear functions. This choice indeed favors parsimony: if the number of samples $N$ is larger than the input dimension $d(+1)$ there exists a unique[8] function $\hat{f}$ that fits through the training data best, i.e. minimizes the training loss

$$L\left(\hat{f}\right) := \sum_{i=1}^{N} \left(\hat{f}(x_i^{\text{train}}) - y_i^{\text{train}}\right)^2. \tag{3}$$

Although this approach is still extensively used in real-world applications, the space of linear functions often is not sufficient, as true relations between input and output are mostly more involved if not highly non-linear. Ideally, the class $\mathcal{H}$ would hence be chosen to be more expressive, so as to be able to approximate well these underlying maps from input $X$ onto output $Y$.

As a consequence, the challenge nowadays is to choose the "most desirable" function $\hat{f}$ out of the infinitely many functions with equal training loss $L\left(\hat{f}\right)$. This opens the question to what the mathematical meaning of "most desirable" could be. At least intuitively, engineers have quite specific convictions (also known as *inductive bias*) which functions are not desirable (see Figures 1 and 2). This intuition could be formalized mathematically as

[7] This papers main result, Theorem 3.8, continues to hold true for more general choices of convex and continuously differentiable loss functions $l_i : \mathbb{R} \to \mathbb{R}_{\geq 0}$, $i = 1, \ldots, N$ and

$$L(f) := \sum_{i=1}^{N} l_i\left(f(x_i^{\text{train}})\right), \tag{2}$$

(see Remark A.2 in Appendix A.1).

[8] The solution of a least square linear regression is unique, if there are $d$ linearly independent training data points $x_i^{\text{train}}$ (or $d+1$ affine independent input points $x_i^{\text{train}}$ if an intercept is used). If the training data points are drawn as i.i.d. samples from a distribution that is absolutely continuous with respect to the $d$-dimensional Lebesgue measure, this is almost surely the case, if $d(+1) \leq N$.

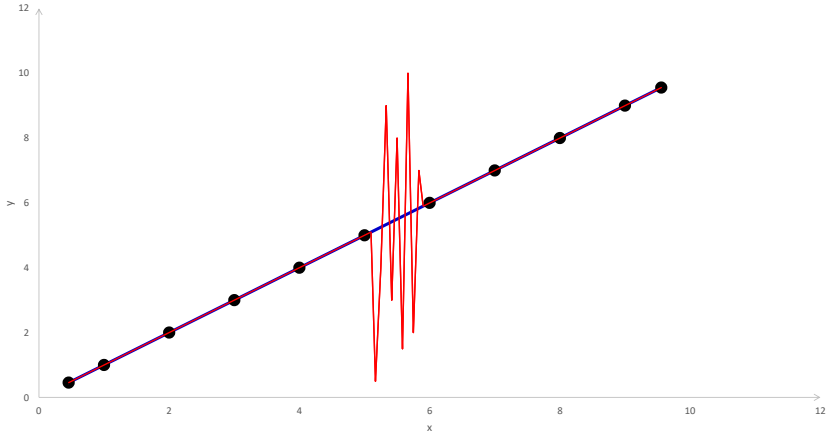


FIGURE 1. Example: Given these $N = 11$ training data points $(x_i^{\text{train}}, y_i^{\text{train}})$ (black dots) there are infinitely many functions $f$ that perfectly fit through the training data and therefore have training loss $L(f) = 0$. The engineer's intuition often tells us that one should prefer the straight blue line over the oscillating red line, even though both functions have zero training loss $L(f) = 0$.

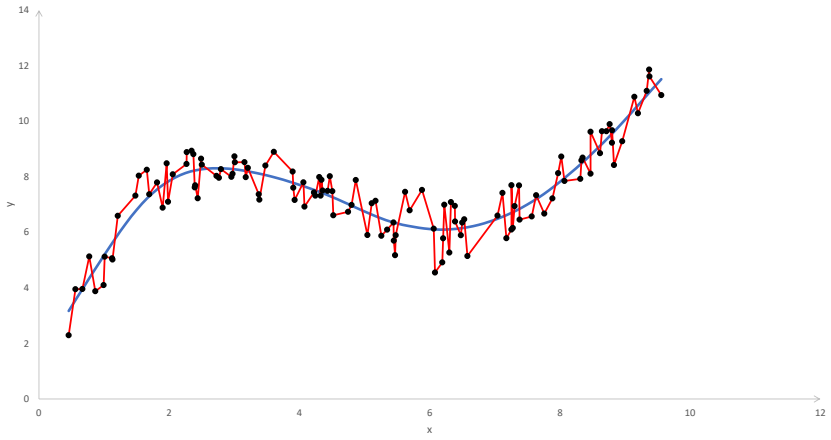


FIGURE 2. Example: Given these $N = 120$ training data points $(x_i^{\text{train}}, y_i^{\text{train}})$ (black dots) there are infinitely many functions $f$ that perfectly fit through the training data and therefore have training loss $L(f) = 0$. For many applications our intuition tells us that we should prefer the smooth blue line $f^{*,\lambda}$ over the oscillating red line, even though the smooth function $f^{*,\lambda}$ results in training loss $L\left(f^{*,\lambda}\right) > 0$.

a Bayesian prior knowledge[9] [5, e.g. page 22].

One approach to capture the engineer's intuition about the prior knowledge is to directly regularize the second derivative of $\hat{f}$. Therefore, in the case of input-dimension $d = 1$, the spline regression [30, 7, 18] is frequently considered in order to choose the function $\hat{f}$ which minimizes a weighted combination of the integrated square of the second derivative and the training loss $L$.

**Definition 1.2** (spline regression)**.** Let $\forall i \in \{1, \dots, N\} : x_i^{\text{train}}, y_i^{\text{train}} \in \mathbb{R}$ and $\lambda \in \mathbb{R}_{>0}$. Then the *(smoothing[10]) regression spline* $f^{*,\lambda} : \mathbb{R} \to \mathbb{R}$ is defined[11] as

$$f^{*,\lambda} \overset{11}{:\in} \underset{f \in \mathcal{C}^2(\mathbb{R})}{\arg\min} \underbrace{\left( \overbrace{\sum_{i=1}^{N} \left( f(x_i^{\text{train}}) - y_i^{\text{train}} \right)^2}^{L(f)=} + \lambda \overbrace{\int_{-\infty}^{\infty} \left( f''(x) \right)^2 dx}^{P^1(f):=} \right)}_{=:F^{\lambda}(f)} \tag{4}$$

and for a given function $g : \mathbb{R} \to \mathbb{R}_{\geq 0}$ the *weighted regression spline* $f_g^{*,\lambda}$ is defined[11] as

$$f_g^{*,\lambda} \overset{11}{:\in} \underset{\substack{f \in \mathcal{C}^2(\mathbb{R}) \\ \text{supp}(f) \subseteq \text{supp}(g)}}{\arg\min} \underbrace{\left( \overbrace{\sum_{i=1}^{N} \left( f(x_i^{\text{train}}) - y_i^{\text{train}} \right)^2}^{L(f)=} + \lambda \overbrace{g(0) \int_{\text{supp}(g)} \frac{\left( f''(x) \right)^2}{g(x)} dx}^{P^g(f):=} \right)}_{=:F^{\lambda,g}(f)} . \tag{5}$$

[9] From the machine learning point of view, one could theoretically formulate this prior knowledge regarding the unknown distribution of $(X, Y)$ on $\mathcal{X} \times \mathcal{Y}$ as a (probability)-measure on the space of all probability measures on $\mathcal{X} \times \mathcal{Y}$. If the prior measure is a probability measure, one can work perfectly rigorously in the framework of classical Bayes law. If the prior measure is not a probability measure, we speak of an improper prior, which can also lead to good results in applications. Consider for instance the very restrictive prior measure that assigns measure 0 to the set of all non-linear functions and weights all linear functions the same. Since this measure assigns $\infty$ to the subspace of all linear functions, it is an improper prior. This improper prior leads to the least-square linear regression in the case of i.i.d. normally distributed noise. The simple intuitive prior knowledge "I am absolutely sure that $f_{\text{True}}$ is linear, but I consider all linear functions as equally likely." is captured quite well by this improper prior and the solution of the corresponding Bayesian problem can be computed quite fast (linear regression). But for most real-world applications, a more realistic intuitive prior knowledge such as "I cannot exclude any function for sure, but I have some vague feeling that $f_{\text{True}}$ is more likely to be a 'simpler', 'smoother' function than a 'heavily oscillating' function." is harder to mathematically formalize and calculating the solution of such Bayesian problems is often not tractable (with today's computational power). Still, Bayesian theory can be considered a very powerful and general abstract theoretical framework without explicitly solving Bayesian problems and even without explicitly writing down priors.

[10] In the literature, the spline regression is often called *(natural) (cubic) smoothing spline*, but in this text $f^{*,\lambda}$ will simply be called regression spline.

[11] We use the notation $a :\in \{s\}$ to define $a$ as the unique element $s$ of the set $\{s\}$ (i.e. $a := s$). So strictly speaking the set after ":∈" should be a singleton—we are using footnotes to indicate under which assumptions uniqueness can be guaranteed. The (weighted) regression spline $f_g^{*,\lambda}$ is uniquely defined (i.e. $\arg\min_f \left( L(f) + P^g(f) \right) = \left\{ f_g^{*,\lambda} \right\}$) if $\exists (i,j) \in \{1, \dots, N\}^2 : x_i^{\text{train}} \neq x_j^{\text{train}}$ and $g(0) \neq 0$ in the case of the weighted regression spline. The "arg min" is defined as the set of all minimizers:

$$\text{(arg min)} \qquad \underset{s \in S}{\arg\min}\, F(s) := \left\{ s \in S \;\middle|\; F(s) = \min_{\tilde{s} \in S} F(\tilde{s}) \right\} = \{\, s \in S \mid \forall \tilde{s} \in S : F(s) \leq F(\tilde{s}) \,\}.$$

The meta parameter $\lambda$ controls the trade-off between low training loss and low squared second derivative. See $f^{*,\lambda}$ in Figure 2 for an example of the regression spline (which corresponds to the weighted regression spline $f_g^{*,\lambda}$ with constant weight $g \equiv c > 0$).

Letting the regularization parameter $\lambda$ tend to zero in (4), one obtains the smooth spline interpolation, i.e. the "smoothest" $\mathcal{C}^2$-function interpolating the observed data.

**Definition 1.3** (spline interpolation)**.** Let $\forall i \in \{1, \dots, N\} : x_i^{\text{train}}, y_i^{\text{train}} \in \mathbb{R}$ and $\lambda \in \mathbb{R}_{>0}$. Then the *(smooth) spline interpolation* $f^{*,0+} : \mathbb{R} \to \mathbb{R}$ is defined[12] as:

$$f^{*,0+} := \lim_{\lambda \to 0+} f^{*,\lambda} \overset{12}{\in} \underset{\substack{f \in \mathcal{C}^2(\mathbb{R}), \\ f(x_i^{\text{train}}) = y_i^{\text{train}} \ \forall i \in \{1,\dots,N\}}}{\arg\min} \left( \int_{-\infty}^{\infty} \left(f''(x)\right)^2 dx \right). \tag{6}$$

The Definitions 1.2 and 1.3 can also be seen as solutions to mathematically defined Bayesian problems [18][13].

1.2. **A paradox of neural networks.** As argued above, within a regression problem one might have an intuition about certain attributes of solution functions $\hat{f}$ that are particularly "desirable". Moreover, these ideas of suitability could be incorporated directly by including certain regularization terms to the learning problem, such as seen in the popular example of the spline regression $f_g^{*,\lambda}$. Surprisingly, however, standard algorithms applied to train neural networks (i.e. gradient descent applied to the training loss $L$) are able to find "desirable" functions $\hat{f}$ *without explicit regularization*. This paradox shall be discussed throughout the present section. In particular, we will demonstrate two severe misassumptions typically made in the classical approach to explain supervised learning using neural networks.

The paradox can be observed for deep [13] as well as for shallow[14] neural networks. This paper resolves the phenomenon rigorously only in the context of (specific) shallow neural networks (cp. Section 3). We start by defining these objects below. Further work is required to extend the results to deep neural networks.[14]

**Definition 1.4** (shallow neural network[14])**.** Let the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ be a non-constant Lipschitz function. Then, a *shallow neural network* is defined as $\mathcal{NN}_\theta : \mathbb{R}^d \to \mathbb{R}$ s.t.

$$\mathcal{NN}_\theta(x) := \sum_{k=1}^{n} w_k \, \sigma\left( b_k + \sum_{j=1}^{d} v_{k,j} x_j \right) + c \quad \forall x \in \mathbb{R}^d,$$

with

- number of neurons $n \in \mathbb{N}$ and input dimension $d \in \mathbb{N}$,

[12] Analogous to footnote 11, the spline interpolation $f^{*,0+}$ is uniquely defined if $\exists (i,j) \in \{1,\dots,N\}^2 : x_i^{\text{train}} \neq x_j^{\text{train}}$. The right-hand side optimization problem in eq. (6) has a unique minimizer $f^{*,0+}$.

[13] More precisely, Definitions 1.2 and 1.3 can be seen as limits of Bayesian problems [18, p. 502]. Definitions 1.2 and 1.3 cannot be solutions of a classical Bayesian problem with a *proper* prior (cp. footnote 9 on page 5, [18, eq. (4.1) on p. 501] and [33]).

[14] In recent literature it has become fashionable to call *shallow neural networks* "simple deep neural networks" or "two-layer (deep) neural networks" [12, Section 1.1 p. 3]. These three terms all are reasonable, since such a network consists of three layers of neurons (input→hidden→output), therefore it has two layers of weights and biases $((v,b) \to (w,c))$ and thus one hidden layer of neurons. Throughout this paper, we use the classical notion of "shallow neural networks" to describe these objects. Within the current section as well as in Section 4, we will express the desire to extend our theory to deep neural networks. This can alternatively be read as extending the theory to "even deeper neural networks".

- weights $v_k \in \mathbb{R}^d$, $w_k \in \mathbb{R}$, $k = 1, \dots, n$ and
- biases $c \in \mathbb{R}$, $b_k \in \mathbb{R}$, $k = 1, \dots, n$.

Weights and biases are collected in

$$\theta := (w, b, v, c) \in \Theta := \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n \times d} \times \mathbb{R}.$$

**Paradox 1.** The paradox of how the training of neural networks leads to solution functions that are surprisingly sensible from a Bayesian perspective (summarized in Figure 3) consists of two parts:

1. In the literature it is often claimed that the goal of training a neural network is to find parameters

$$\theta^* \in \operatorname*{arg\,min}_{\theta \in \Theta} L\left(\mathcal{NN}_\theta\right), \tag{7}$$

such that the corresponding neural network $\hat{f} := \mathcal{NN}_{\theta^*}$ fits through the training data as good as possible (where goodness of fit is characterized by the choice of loss $L$ ).

However, such an optimal neural network $\mathcal{NN}_{\theta^*}$ might have bad generalization properties. First, if the number of hidden neurons $n \geq N$ is larger or equal than the number of training data points $N$, there are infinitely many (7)-optimizing shallow neural networks $\mathcal{NN}_{\theta^*}$ that generalize arbitrarily badly[15], even if there were only zero noise $\varepsilon_i = 0$ on the training data.
Second, if $n \leq N - 2$, then $\mathcal{NN}_{\theta^*}$ can be unique, but $\mathcal{NN}_{\theta^*}$ might still overfit to the noise on the training data (see Figure 4). As a consequence of the universal approximation theorems [8, 15], we have that large neural networks $\mathcal{NN}_{\theta^*}$ (or any other universally approximating class of functions) can potentially behave arbitrarily badly (as, for instance, in Figure 1) in-between the training data $x_i^{\text{train}}$ while keeping the training loss arbitrarily low, i.e. $L\left(\mathcal{NN}_{\theta^*}\right) \leq \epsilon$, exactly because of their universal approximation properties. (If a very small number of neurons $n \ll \frac{N}{d}$ were chosen, over-fitting of $\mathcal{NN}_{\theta^*}$ would not pose such a severe problem, however, in that case, neural networks would lose their universal approximation property (which is one of their main selling points) and therefore $\mathcal{NN}_{\theta^*}$ could not achieve a low loss $L\left(\mathcal{NN}_{\theta^*}\right)$.)

Paradoxically, however, extremely large (trained) neural networks $\mathcal{NN}_\theta$ typically generalize very well in practice. Indeed, Theorems 3.8 and 3.17 will demonstrate how well neural networks $\mathcal{NN}_\theta$ with an infinite number of neurons behave in between the data.

2. The objective function in optimization problem (7) (in the case of typical activation functions) is a Lebesgue-almost everywhere differentiable function on the finite dimensional $\mathbb{R}$-vector space $\Theta$. Thus, for solving (7), it seems evident not only to most

[15]For ReLU activation functions, one can prove, that for every training data $\left(x_i^{\text{train}}, y_i^{\text{train}}\right)_{i \in \{1,\dots,N\}}$ there exist infinitely many $\mathcal{NN}_{\theta^*}$ such that the $d$-dimensional Lebesgue-measure of the set $\left\{ x \in [-1,1]^d \,\middle|\, |\mathcal{NN}_{\theta^*}(x)| > 9999 \right\}$ is larger than 99% and $L\left(\mathcal{NN}_{\theta^*}\right) = 0$. If $n \geq N - 1$ and $n \geq 2$ also infinitely many solutions exist that generalize arbitrary badly in a bit weaker sense: For every training data $\left(x_i^{\text{train}}, y_i^{\text{train}}\right)_{i \in \{1,\dots,N\}}$ there exist infinitely many $\mathcal{NN}_{\theta^*}$ such that the $d$-dimensional Lebesgue-measure of the set $\left\{ x \in [-1,1]^d \,\middle|\, |\mathcal{NN}_{\theta^*}(x)| > 9999 \right\}$ is larger or equal than 49% and $L\left(\mathcal{NN}_{\theta^*}\right) = 0$. This implies that there exist different global optima $\mathcal{NN}_{\theta^*}$ of $L$ that are arbitrarily far from each other in any $L^p$-norm.

engineers to use a gradient descent algorithm (where the gradient can be calculated via backpropagation algorithm in the case of neural networks). When considering the training loss $L$ , stochastic gradient descent might be as well used.[16]

However, there are no known guarantees that this algorithm converges to a global optimum for a general, typically non-convex optimization problem. Moreover, numerical experiments show that if the algorithm continues for a reasonable time, the solution function obtained is still quite far from being optimal (w.r.t. the target function $L$, that the algorithm claims to try to optimize.) (e.g. Figure 4).

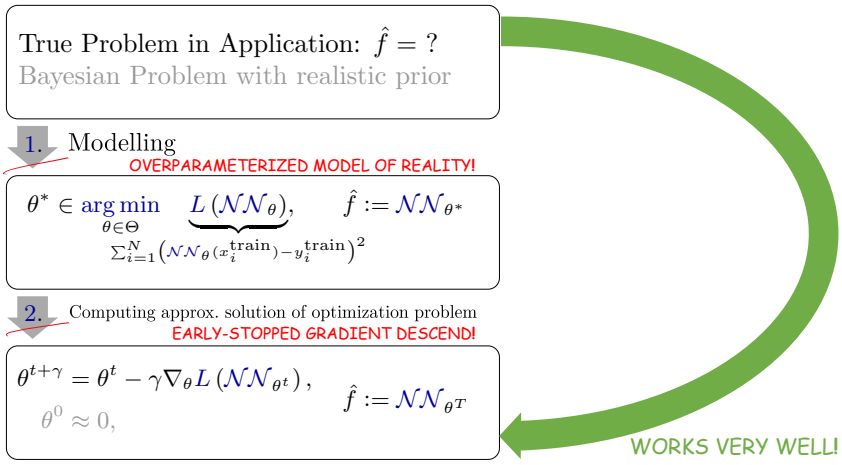


FIGURE 3. Paradox 1: 1. It would not be desirable for neural networks to solely minimize the training loss $L$. 2. The (stochastic) gradient descent algorithm (also known as backpropagation algorithm) typically does not succeed in finding a global optimum. Nevertheless, the algorithm results in functions $\hat{f} = \mathcal{NN}_{\theta^T}$ that are surprisingly useful for a wide range of practical applications.

1.3. **Resolving Paradox 1: Implicit Regularization.** In the following, we like to resolve the paradox described above. Moreover, at the end of this section, a short overview will be given, showing how this work contributes to a better understanding of the aforementioned phenomenon.

Points 1, 2 and the observation that neural networks are very useful in practice can be true at the same time:

As discussed above, an "optimal" network $\mathcal{NN}_{\theta^*}$ would typically perform quite poorly in practice (cp. 1). However, such a network is hardly obtained as a solution from a generic training process involving a gradient descent based algorithm. The reason being that, fortunately, the backpropagation algorithm which was designed to yield trained networks close

[16]The stochastic gradient descent poses immense computational advantages in the case of a very large number $N$ of training observations (cp. item 2. on page 25). Within the present work, stochastic gradient descent can be treated equivalently to ordinary gradient descent as we are considering the regime of constant $\gamma/\tau \equiv T$ with diminishing learning rate $\gamma \to 0$ and $N \in \mathbb{N}$ fixed.

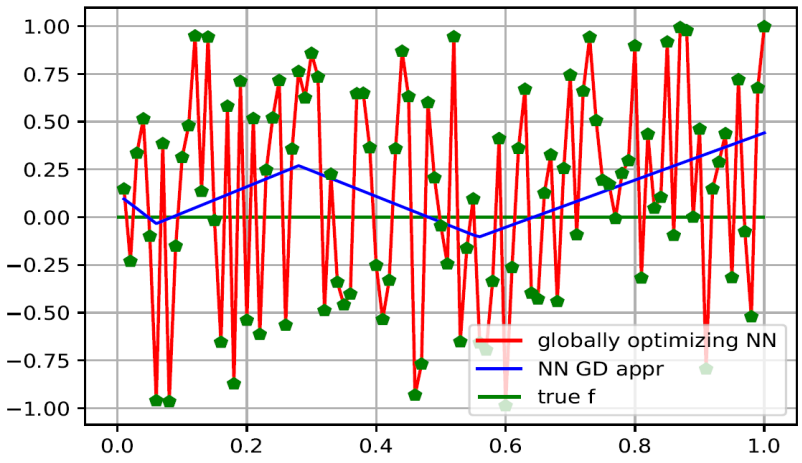


FIGURE 4. Example: Let $N = 100$ training samples $(x_i^{\text{train}}, y_i^{\text{train}})$ be scattered uniformly around the true function $f_{\text{True}} = 0$ and consider a shallow neural network $\mathcal{NN}$ with $n = N = 100$ hidden nodes. After 10000 training epochs of Adam SGD [19] the neural network does not converge to the global optimum $\mathcal{NN}_{\theta^*}$ (red line) with $L(\mathcal{NN}_{\theta^*}) = 0$, but to a more regular function $\mathcal{NN}_{\theta^T}$ (blue line) which is closer to the true function $f_{\text{True}}$.

to $\mathcal{NN}_{\theta^*}$ by minimizing the training loss $L$ does not achieve[17] this goal (cp. 2, i.e. typically $L(\mathcal{NN}_{\theta^T}) \gg L(\mathcal{NN}_{\theta^*})$). Instead, it surprisingly succeeds in reaching a much more desirable objective by not only minimizing the training loss $L$ but also *implicitly*[18] regularizing the problem. Hence, the typically bad generalization property 1 of $\mathcal{NN}_{\theta^*}$ does not contradict the great out-of-sample performance of $\mathcal{NN}_{\theta^T}$, which is observed to be the much more regular.

This phenomenon is known in the literature as "implicit regularization" [27, 26, 23, 20, 32, 29, 12] (also known as "implicit bias"[32]). It demonstrates that questions I and II, i.e. the generalization properties of neural networks and the use of gradient descent-based methods in their training are strongly linked in practice.

In applications, the phenomenon of implicit regularization is frequently observed [14, 24, 27, 26, 23, 20, 29]. Nonetheless, the theory behind it is still largely unexplored [23, 20, 29, 24]. The contribution of this work (summarized in Figure 5) is proving very precisely in which manner the implicit regularization effects occur when training a so-called **r**andomized **s**hallow neural **n**etwork (RSN) (a specific type of neural network with one hidden layer

[17] In the limit of infinite training time $T \to \infty$, the gradient descent method can converge to a global optimum. As we will see in the sequel, even though there typically are infinitely many global optima this limit will be a very specific representative (cp. Definitions 3.3 and 3.7, Theorems 3.8 and 3.17 and eq. (27)). Nonetheless, the training process is typically stopped after a few epochs (with training time $T \ll \infty$). The corresponding solution $\mathcal{NN}_{\theta^T}$ typically satisfies $L(\mathcal{NN}_{\theta^T}) \gg L(\mathcal{NN}_{\theta^*})$ and is much more desirable (cp. Definition 3.5 and eq. (28)).

[18] "*Implicitly*" means that one uses exactly the same algorithm (gradient descent on the training loss $L$ cp. Figure 3) that one would use, if one did not care about regularization, but running the algorithm surprisingly results in a very regular solution function $\mathcal{NN}_{\theta^T}$.

and randomly chosen first-layer parameters—Definition 2.1) with a large number of hidden nodes $n \to \infty$ and ReLU activation (i.e., a wRRSN) using a gradient descent method. As we shall see in the following, for such a network (as a function from $\mathcal{X}$ to $\mathcal{Y}$) the second derivative is implicitly regularized during training. More precisely, we will characterize the solution function obtained in infinite training time for wide networks with a large number of hidden nodes (cp. Definition 3.5 and Theorems 3.8 and 3.17). In a typical setting, this limit is very close to a regression spline $f^{*,\lambda}$, whose theory is highly understood [30, 7, 18].

*Remark* 1.5 ($P$-Functional). In supervised learning, $P$-regularized loss minimization models, i.e.,

$$f^{*,P,\lambda} \in \operatorname*{arg\,min}_f L(f) + \lambda P(f),$$

are typically quite easy to interpret and have nice theoretical properties (e.g. Definition 1.2). Each of these models is fully characterized by its regularizing functional $P\colon \mathcal{Y}^{\mathcal{X}} \to \bar{\mathbb{R}}$ (e.g. $P = P^g$ in the case of weighted smooting spline regression $f_g^{*,\lambda}$).[19, 20] Our key finding is that other supervised learning algorithms (such as standard neural network algorithms) that are typically not considered as $P$-regularized loss minimization, in fact are equivalent to $P$-regularized loss minimization with a specific $P$-functional (i.e. $\mathcal{NN}_{\theta^{\text{result}}} \approx f^{*,P,\lambda}$). We believe that the framework of $P$-regularized loss minimization could be very well suited to understand and compare the behavior of many different standard methods in supervised learning (in particular neural networks). Whether or not a certain $P$-functional (or an equivalent algorithm) leads to functions $f^{*,P,\lambda}$ that generalize well, depends on one's prior[21] belief. The goal of this work will not be to determine how *well* certain types of neural networks generalize in general situations (this is not possible without assumptions on the data generating process—i.e. $\mathbb{P}_D$). Instead, the main Theorem 3.8 expresses *how* a certain neural network $\mathcal{RN}^{*,\bar{\lambda}} \approx \mathcal{RN}_{w^T}$ behaves, by showing its equivalence to a certain $P$-regularized loss minimizer $f_{g,\pm}^{*,\lambda} \approx f^{*,\lambda}$ characterized by a certain $P$-functional $P_\pm^g$ (see Definition 3.5). The long-term goal of this line of research is to describe the learning-behavior of every neural network configuration with its own $P$-functional (see Figure 5), such that one can choose a suitable configuration based on one's prior belief.

Within this paper, we state two main theorems that jointly lead to the desired characterization of the solution function obtained in the limit.

---

[19] The letter "$P$" can be motivated by the fact that the $P$-functional ***penalizes*** less regular functions $f \in \mathcal{Y}^{\mathcal{X}}$, assigning to them a large value of the ***penalty*** $P(f)$. Moreover, it expresses a certain ***prior belief***[21] of which types of functions should be preferred in the supervised learning task. Metaphorically speaking the $P$-functional could in some sense be seen as the "psyche" of a particular type of neural network. (I.e. the $P$-functional enables us to easily conclude how the experiences $(x_i^{\text{train}}, y_i^{\text{train}})$ a neural network $\mathcal{NN}$ encounters during training, effect its future behaviour $\hat{f}(x) = \mathcal{NN}_{\theta^T}(x)$ for any future situation $x \in \mathcal{X}$. This would be a typical question asked in *psychology* in the case of biological neural networks. Note that different architectures (e.g. different activation functions or different number of layers) can lead to a different *psyche*/character within this analogy.)

[20] Instead of restricting the definition of $P$ and the optimization problem $\min_f L(f) + \lambda P(f)$ to a certain subspace (e.g. $\mathcal{C}^2$) one can also define $P(f) := \infty$ for all functions outside the subspace.

[21] $P$ should not be directly interpreted as the prior distribution on the function space. However, some $P$-functionals have the property that $f^{*,P,\lambda} \in \operatorname{arg\,min}_f L(f) + \lambda P(f)$ is equal to the Bayesian a posteriori mean with respect to some Bayesian prior distribution (see e.g. [18]).

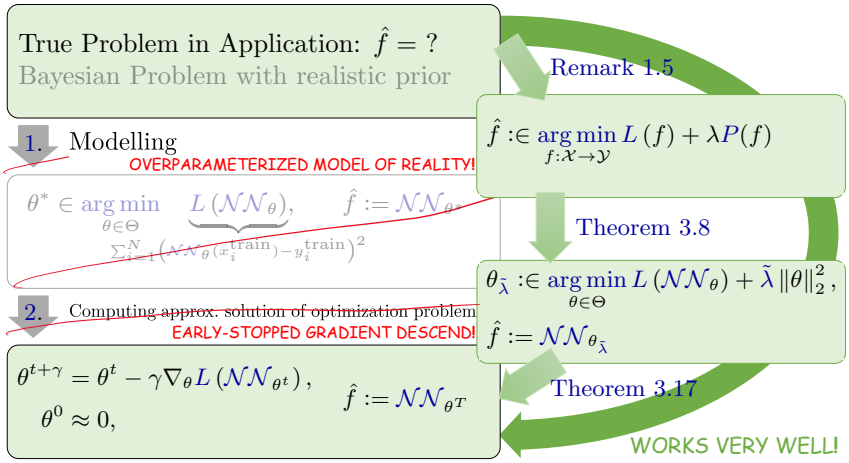


FIGURE 5. Solution of Paradox 1: The (early-stopped) (stochastic) gradient descent algorithm on $L$ (w.r.t. the trainable terminal-layer weights) does not solely minimize $L$—instead it minimizes a regularized optimization problem much more accurately, when initialized close to zero $\theta^0 \approx 0$ (Theorem 3.17). The line of research starting with this paper describes this regularization macroscopically on the function space in terms of a $P$-functional (see Remark 1.5). (Theorem 3.8 reveals the very easy to interpret $P$-functional $P_\pm^g$ (Definition 3.5) in the case of a wide 1-dimensional network of the form $\mathcal{RN}$ (Definition 2.1). Other types of neural networks correspond to different $P$-functionals that will be shown in future work.).

- Theorem 3.17 connects the **r**andomized **s**hallow neural **n**etwork (RSN) obtained by performing ordinary gradient descent initialized close to zero to train the parameters without any explicit regularization to the one obtained from an implicit ridge regularization of the weights. (This theorem builds on very similar results that are well known in the literature [4, 9, 29, 12].)
- Theorem 3.8 shows how the training of the wRRSN's weights via ridge regularization results in the (slightly adapted) spline regularization of the learned network function if the number of neurons $n \to \infty$. This theorem is the main contribution of this work.

Understanding the training of neural networks and, in particular, their frequently astonishing generalization properties has been at the center of interest in many recent works. Without aiming to be exhaustive, we give a brief overview of existing results most related to the the present paper.

- There are a number of works that discuss implicit regularization on the weight space (comparable to Theorem 3.17) [4, 32, 29, 12][22]. However, within these works it is mostly not explained how these effects translate to implicit regularization on the function space. As an exception within the framework of classification, [32,

[22][32, 29] focus on classification (exponential loss) and in [4, 12] regression problems (with least square training loss $L$) are considered.

29] give insight about the margins between the classes, which is a property of the learned function. These papers provide a precise and quite complete mathematical understanding of linear neural networks without any hidden layers. The theorems in these papers that deal with neural networks with one (or more) hidden layers serve as a basis for arguments why an implicit regularization effect can exist on a qualitative level, but not on a precise quantitative level (especially when non-linear activation functions $\sigma$ are considered).

- Contrary to the above, this paper's main contribution, Theorem 3.8, explains the implicit regularization effects on the function space. In that regard, the results presented in [24, 20, 23] are more closely related.
  - in [23], the implicit regularization effects that happen when fully training a shallow neural network $\mathcal{NN}$ with non-linear ReLU activation function $\sigma = \max(0, \cdot)$ are studied on a qualitative level in the context of classification (cross entropy loss over the softmax as a training loss). In said work, the notion "pseudo-smooth" [23, e.g. p. 4] is used, but a quantitative mathematical analysis of the pseudo-smoothness is missing.
  - Similarly in [24] (by Google Brain), the implicit regularization for a fully trained shallow neural network $\mathcal{NN}$ with non-linear ReLU activation functions $\sigma = \max(0, \cdot)$ is discussed. In the context of regression (using an arbitrary differentiable loss function) the main goal of [24] is to explain the macroscopic behavior of the learned neural network function $\mathcal{NN}_{\theta^T}$, i.e. its generalization properties in between the training data. Within this work, a very rich qualitative understanding of $\mathcal{NN}_{\theta^T}$ as well as very helpful visualizations are provided, however, there is no mention of a precise quantitative formula. Hence, a complete macroscopic characterization of the learned function is not given. In contrast, within the present paper, we provide a precise quantitative macroscopic formula (Definition 3.5) that characterizes trained wRRSNs $\mathcal{RN}$. Thus, the present paper provides a quite complete understanding of wRRSNs $\mathcal{RN}$. In near future work, we intend to present results that characterize in which sense a fully trained network $\mathcal{NN}_{\theta^T}$ is macroscopically optimal (cp. item iii in Section 4).
  - The implicit regularization effects in the training of deep neural networks with non-linear ReLU activation functions $\sigma = \max(0, \cdot)$ are studied in [20]. Therein, it is stated that the learned function interpolates "almost linearly" between samples. This behavior is related to a low (in the case of ReLUs distributional) second derivative which corresponds to the notion of "gradient gaps" introduced in [20].
- [17] gives an exact characterisation of the limiting function by proving an equivalence between neural networks and kernel methods (Gaussian Processes) under quite general assumptions. At the moment, the neural tangent kernel theory introduced in [17] probably is the most general well-developed theory about the macroscopic behavior of wide deep neural networks. Apart from the fact that neural tangent theory is much further developed at the moment, both the $P$-functional theory and the neural tangent kernel theory have their advantages and disadvantages that will be compared in future work.
- Recently, there has been growing interest in analyzing the convergence behavior of the gradient descent algorithm in the training of infinitely wide (shallow and deep)

neural networks ([17], [6], [25]). Moreover, in these works, conditions for convergence to global optima are discussed.

- In an earlier work, the relation between (possibly multivariate versions of) spline interpolation and network structures was analyzed. The paper [28] nicely motivates the reasonability of approximation tasks including general regularizing terms that control the approximating function's derivatives. It is shown that the solution to the spline interpolation problem 1.3 can be explicitly represented as an element of an $N$-dimensional subspace (where $N$ is the number of data points at hand) of the space of smooth functions, a basis of which is given by certain Green functions corresponding to the optimization problem. Based on that observation, a so-called regularization network that implements the smooth spline interpolation using the basis functions as activation functions is defined. However, this result does not treat implicit regularization effects but rather explicitly implements the desired regularization in the form of a network structure.

The remainder of this paper is structured as follows. In Section 2, we begin by defining the specific type of neural network $\mathcal{RN}$ considered in the subsequent analyses: *1-dimensional* ***w****ide* ***R****eLU* ***r****andomized*[23] ***s****hallow neural* ***n****etworks* (wRRSNs) (9). Moreover, we discuss the expressiveness of the function class of such RSNs and give further definitions that are central to the understanding of the main Theorems 3.8 and 3.17.

Thereafter, in Sections 3.1 and 3.2, Theorems 3.8 and 3.17 are formulated and discussed. The corresponding proofs are to be found in Appendix A. Finally, in Section 4 the implications of these results are summarized in eqs. (27) and (28). Moreover, therein, we give a brief outlook on planned future work.

## 2. Randomized Shallow Neural Networks (*RSN*s)

Within this section, we like to introduce the notion of **r**andomized **s**hallow neural **n**etwork (RSN), a specific kind of artificial neural network with one hidden layer, that we consider for our analyses.

**Definition 2.1** (RSN)**.** Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space, and the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ Lipschitz continuous and non-constant. Then a ***r****andomized* ***s****hallow neural* ***n****etwork (RSN)* is defined as $\mathcal{RN}_{w,\omega} : \mathbb{R}^d \to \mathbb{R}$ s.t.

$$\mathcal{RN}_{w,\omega}(x) := \sum_{k=1}^{n} w_k \, \sigma\left( b_k(\omega) + \sum_{j=1}^{d} v_{k,j}(\omega) x_j \right) \qquad \forall \omega \in \Omega \; \forall x \in \mathbb{R}^d \tag{8}$$

with[24]

- number of neurons $n \in \mathbb{N}$ and input dimension $d \in \mathbb{N}$,
- trainable weights $w_k \in \mathbb{R}$, $k = 1, \ldots, n$,

[23] The most striking property of this type of network is that the first layer is chosen randomly and not trained, i.e. after random initialization only the terminal layer is trained. One might expect that this randomness decreases the regularity of the learned function, but in fact the effect is quite the opposite: as we will thoroughly discuss, the learned function will be especially smooth because of this randomness, where smoothness is understood as minimizing the integrated squared second derivative; cp. Theorem 3.8)

[24] One could include an additional bias $c \in \mathbb{R}$ to the last layer too, but in the limit $n \to \infty$ this last-layer bias $c$ does not change the behavior of the trained network-functions $\mathcal{RN}_{w^T}$ or $\mathcal{RN}^{*,\tilde{\lambda}}$. In Figures 6–8 this last layer bias $c$ was included in the training.

- non-trainable random biases $b_k : (\Omega, \Sigma) \to (\mathbb{R}, \mathfrak{B})$ i.i.d. real-valued random variables $k = 1, \dots, n$,
- non-trainable random weights $v_k : (\Omega, \Sigma) \to (\mathbb{R}^d, \mathfrak{B}^d)$ i.i.d. $\mathbb{R}^d$-valued random variables $k = 1, \dots, n$.

*Remark 2.2 (further notation). Throughout this paper, $\mathbb{P}_{\#} f$ denotes the push-forward measure of $\mathbb{P}$ under the map $f$. Moreover, we frequently use the notation $\mu := \mathbb{P}_{\#}(b, v)$ for denoting the distribution of a random first-layer parameter vector $(b, v) : \Omega \to \mathbb{R}^{d+1}$ corresponding to an RSN $\mathcal{RN}_w$ and write $\lambda^d$ for the Lebesgue measure on $\mathbb{R}^d$. We further introduce the map $\psi_{(b,v)} : \Omega \times \mathbb{R}^d \to \mathbb{R}^n$, with $\psi_{(b,v)} : (\omega, x) \mapsto \psi_{(b,v)(\omega)}(x)$ s.t. $\psi_{(b,v)(\omega)}(x)_k = \sigma\left(b_k(\omega) + \sum_{j=1}^d v_{k,j}(\omega) x_j\right)$ for any $k = 1, \dots, n$, mapping the input to an RSN's hidden layer. We call $\operatorname{range}(\psi_{(b,v)}) := \bigcup_{\omega \in \Omega} \operatorname{range}(\psi_{(b,v)(\omega)}) \subseteq \mathbb{R}^n$ the latent space of an RSN.

Before describing in detail the implicit regularization effects obtained by applying gradient descent methods to train the last layer of such an RSN in Section 3, we elaborate on the expressiveness* (question III) of RSNs.

*The class of RSNs might be interesting in supervised learning due to a number of reasons. First, as a corollary to any of the much-cited universal approximation theorems, randomized shallow networks are what we call *universal in probability*: Building on the results of [15, 8] and later [22], we obtain that any real-valued continuous function on a compact subset of $\mathbb{R}^d$ can be arbitrarily well approximated by an RSN with arbitrarily high probability. This result holds under relatively weak assumptions on the activation function and probability distribution of first-layer weights and biases and is given below in Corollary 2.3.

*Second, given any set of (distinct) observations $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \dots, N$, $N \in \mathbb{N}$, if the induced measure on the latent space is zero on sets of lower codimension, then, almost surely, there exists an RSN that precisely interpolates these data. In other words, for suitable choices of randomness in the first layer, with probability one the class of randomized shallow networks contains representatives whose parameters are optimal solutions to (7). More precisely, we have Lemma 2.4.

***Corollary 2.3** (Universal in probability)**.** *Let $X \subset \mathbb{R}^d$ be compact and $f \in C(X, \mathbb{R})$. Furthermore, let $\mathcal{RN}_w$ be as in Definition 2.1, with weights $v_k$ and biases $b_k$, $k = 1, \dots, n$ i.i.d. according to $\mu := \mathbb{P}_{\#}(b, v)$ with $\mu \gg \lambda^{d+1}$. Then, under mild conditions on the activation function (e.g. $\sigma$ non-polynomial [22])*

$$\forall \epsilon \in \mathbb{R}_+, \lim_{n \to \infty} \mu^n \left(\exists w \in \mathbb{R}^n : ||\mathcal{RN}_w - f||_\infty > \epsilon\right) = 0.$$

*Here, $\mu^n$ denotes the $n$-fold product measure of $\mu$.*

*Proof.* The proof of Corollary 2.3 is formulated in Appendix A.3. □

*Remark 2.2, Corollary 2.3, Lemma 2.4, Remark 2.5 and the text in-between (marked with a "*") form an independent story line dealing with question III about the expressiveness. If these results are skipped, one can still understand the main story line and the main Theorems 3.8 and 3.17.

***Lemma 2.4** (Almost sure interpolation)**.** *Let distinct observations $(x_i^{train}, y_i^{train}) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \dots, N$ be given. Then, any (perfectly trained[25]) RSN $\mathcal{RN}_w$ with $n \geq N$ hidden nodes such that $\mathbb{P}_{\#}(\psi_{(b,v)}(x_i^{train}))[A] = 0$ for any $A \subseteq \text{range}(\psi_{(b,v)})$ of codimension less than $n$ and $i = 1, \dots, N$, almost surely interpolates the data, i.e.*

$$\mathbb{P}\left[\exists w^* \in \mathbb{R}^n : \mathcal{RN}_{w^*}(x_i^{train}) = y_i^{train}, \quad \forall i = 1, \dots, N\right] = 1.$$

*Proof.* The proof of Lemma 2.4 is formulated in Appendix A.3. □

**Remark* 2.5. In Lemma 2.4 we required random features of the latent space $\psi_{(b,v)}(x_i)$, $i = 1, \dots, N$ to follow a distribution on $\mathbb{R}^n$ that puts zero mass on sets of lower codimension. A setting which is rather usual in applications and for which this condition is satisfied would for instance consist in taking $\mathbb{P}_{\#}(b, v) \ll \lambda^{d+1}$ and $\sigma : \mathbb{R} \to (0, 1)$, $\sigma(x) = \exp(x)/(1+\exp(x))$.[26]

By Lemma 2.4 and Corollary 2.3, the function class of RSNs is expressive enough to qualify as a suitable architecture within the framework of supervised learning. At the same time, these results raise the question I, if RSNs generalize badly to unseen data, because of over-parametrization and over-fitting (see Paradox 1-1.). Our main Theorems 3.8 and 3.17 are dealing with question I by providing a certain understanding of the implicit regularization effects that occur when training a specific kind of RSN: As we will show in the sequel, training the last layer of a **w**ide (i.e. $n \to \infty$), **R**eLU-activated **RSN** (wRRSN) using gradient descent initialized close to zero corresponds to solving a smooting spline regression. Note, that this result does not depend on the number of data points $N$ used in the training and thus holds true for any finite number of observations $N \in \mathbb{N}$. The main assumptions we require to hold are made precise in Assumption 1 below.

**Assumption 1.** Using the notation from Definition 2.1:

a) The activation function $\sigma(\cdot) = \max(0, \cdot)$ is ReLU.[27]
b) The distribution of the quotient $\xi_k := \frac{-b_k}{v_k}$ has a probability density function $g_\xi$ with respect to the Lebesgue measure.[28]
c) The input dimension $d = 1$.[29]

Under these assumptions, eq. (8) simplifies to

$$\mathcal{RN}_w(x) = \sum_{k=1}^{n} w_k \max(0, b_k + v_k x) \quad \forall x \in \mathbb{R}\,. \tag{9}$$

[25]Since the optimization problem is convex in the last-layer weighs $w$, the gradient descent actually converges to a global minimum. Hence, under the conditions of Lemma 2.4, the statement can be refined to:

$$\mathbb{P}\left[\lim_{T\to\infty} \mathcal{RN}_{w^T}(x_i^{\text{train}}) = y_i^{\text{train}}, \quad \forall i = 1, \dots, N\right] = 1.$$

[26]For ReLU activation functions almost sure interpolation is often not the case with finite $n < \infty$, but the probability of perfect interpolation converges to one when the number of neurons $n \to \infty$ tends to infinity.

[27]In future work we want to derive other $P$-functionals for other activation functions instead of the **re**ctified **l**inear **u**nits (ReLU)

[28]Assumption 1b) holds for any distribution typically used in practice. Moreover, it implies that $\mathbb{P}[v_k = 0] = 0 \quad \forall k \in \{1, \dots, n\}$. Note that Assumption 1b) is required in order to exclude certain degenerate cases of RSNs such as those with constant weights and biases $w_k, b_k, k = 1, \dots, n$, and could in fact be weakened.

[29]In part II we are going to generalize the result to arbitrary input dimension $d \in \mathbb{N}$.

We henceforth require Assumption 1 to be in place. For later uses, we further introduce the notions of kink positions corresponding to a one-dimensional RSN with ReLU activation and their density function.

**Definition 2.6** (kink positions $\xi$)**.** The *kink positions* $\xi_k := \frac{-b_k}{v_k}$ are defined using the notation of Definition 2.1 under the Assumption 1.

**Definition 2.7** (kink position density $g_\xi$)**.** The *probability density function* $g_\xi : \mathbb{R} \to \mathbb{R}_{\geq 0}$ *of the kink position* $\xi_k := \frac{-b_k}{v_k}$ is defined in the setting of Definition 2.6.

## 3. Main Theorems

We now proceed to show that a standard gradient descent method applied to optimize the (trainable) parameters $w$ of an **w**ide **R**eLU **r**andomized **s**hallow neural **n**etwork (wRRSN) $\mathcal{RN}$, implicitly minimizes the second derivative of the solution function $\mathcal{RN}_{w^T}$. That is, in the many particle (i.e. neurons) limit ($n \to \infty$) and as training time $T \to \infty$ tends to infinity, the solution found by the gradient descent algorithm $\mathcal{RN}_{w^T}$ converges to a slightly adapted smooth spline interpolation $f^{*,0+}_{g,\pm} \approx f^{*,0+}$, if initialized $w^0 \approx 0$ close to zero.

Our result follows by two separate observations. First, note that training a wide RSN in essence reduces to solving a (random) kernelized linear regression in high dimensions (over-parameterized). We obtain in Theorem 3.17 that training an RSN up to infinity (initialized at zero $w^0 = 0$) leads to the same solution as performing ridge regression (Definition 3.2) with diminishing regularization to tune the parameters of the RSN's terminal layer. Note, that the results in Section 3.2 hold for a general input dimension $d \in \mathbb{N}$ and any fixed number of neurons in the hidden layer $n \in \mathbb{N}$.

Second, in Section 3.1, we relate the RSN $\mathcal{RN}^{*,\tilde{\lambda}}$ with optimal terminal-layer parameters $w^{*,\tilde{\lambda}}$ chosen according to a ridge regression (Definition 3.2) to a smoothing spline $f^{*,\lambda}$ (with certain regularization parameters $\tilde{\lambda} := \lambda n 2g(0)$ and $\lambda \in \mathbb{R}_{>0}$ respectively). More precisely, we show in Theorem 3.8 that as the number of hidden nodes $n$ (i.e. the dimension of the hidden layer) tends to infinity the ridge regularized RSN $\mathcal{RN}^{*,\tilde{\lambda}}$ converges to a slightly adapted smoothing spline $f^{*,\lambda}_{g,\pm}$ in probability with respect to a certain Sobolev norm. Recall that, by Assumption 1, we prove this correspondence for wRRSNs with one-dimensional input.

*Remark* 3.1. The implicit regularization effects we characterize within this paper are of an asymptotic nature. For applications, however, it is interesting to note that, even for finitely many hidden nodes and finite training time, one can bound the distance between the solution obtained by gradient descent and a certain smoothing spline (see also Sections 3.2.1 and 4 for further details). The analysis of such bounds will be thematized in future work.

In the following Sections 3.1 and 3.2 we discuss both observations separately, before combining them to formulate our main conclusion in Section 4. We start by introducing the notions of ridge regularized RSN and minimum norm network.

**Definition 3.2** (ridge regularized RSN)**.** Let $\forall i \in \{1, \dots, N\} : (x_i^{\text{train}}, y_i^{\text{train}}) \in \mathbb{R}^{d+1}$ for some $N, d \in \mathbb{N}$. Furthermore, let $\mathcal{RN}_{w,\omega}$ be a randomized shallow network as introduced in Definition 2.1. The ridge regularized RSN is defined as

$$\mathcal{RN}^{*,\tilde{\lambda}}_{\omega} := \mathcal{RN}_{w^{*,\tilde{\lambda}}(\omega),\omega} \quad \forall \omega \in \Omega\,, \tag{10}$$

with $w^{*,\tilde{\lambda}}(\omega)$ such that

$$w^{*,\tilde{\lambda}}(\omega) :\in \underset{w\in\mathbb{R}^n}{\arg\min} \overbrace{\sum_{i=1}^{N} l(\mathcal{RN}_{w,\omega}(x_i^{\text{train}}), y_i^{\text{train}})}^{L(\mathcal{RN}_{w,\omega})} + \tilde{\lambda}||w||_2^2 \quad \forall\omega\in\Omega\,. \tag{11}$$
$$\underbrace{\qquad\qquad\qquad\qquad}_{F_n^{\tilde{\lambda}}(\mathcal{RN}_{w,\omega})}$$

The ridge regularization is also known as "weight decay", "ridge penalization", "$L^2$ (parameter) regularization" or "Tikhonov regularization" (or "ridge regression", "$\ell_2$ penalty", ...)[13, section 7.1.1 on p. 227].

**Definition 3.3** (minimum norm RSN)**.** Using the notation from Definition 3.2, the *minimum norm*[30] *RSN* is then defined as $\mathcal{RN}^{*,0+} := \mathcal{RN}_{w^{*,0+}}$ with weights

$$w^{*,0+}(\omega) := \lim_{\tilde{\lambda}\to 0+} w^{*,\tilde{\lambda}}(\omega) \quad \forall\omega\in\Omega\,. \tag{12}$$

3.1. **Ridge Regularized RSN → Spline Regularization** ($d=1, \lambda\in\mathbb{R}_{>0}$)**.** Throughout this section we rigorously derive the correspondence between the regression spline $f^{*,\lambda}$ respectively the ridge regularized RSN $\mathcal{RN}^{*,\tilde{\lambda}}$ with penalty parameters $\lambda>0$ and $\tilde{\lambda}>0$. For giving a detailed description of the convergence behavior, we introduce an adapted version of the regression spline, for which we consider a weighted version of the spline penalization restricted to the support of the weighting function and introduce certain "boundary conditions". Depending on the distribution of the random weights $w_k$ and biases $w_b$, the ridge regularized RSN $\mathcal{RN}^{*,\tilde{\lambda}}$ will converge to such a (slightly) adapted version $f_{g,\pm}^{*,\lambda}$ of the classical regression spline $f^{*,\lambda}$.

*Remark* 3.4. For constant $g\equiv g(0)$, one recovers the original spline regression.[31] As we will show in the sequel, the distribution chosen for the kink positions $\xi$ of the $\mathcal{RN}^{*,\tilde{\lambda}}$ to be trained in the approximation task will determine the weighting function of the corresponding $f_{g,\pm}^{*,\lambda}$. The adapted spline hence is a rich concept that nicely displays the impact of the engineer's choices when setting up the network to be trained.

**Definition 3.5** (adapted spline regression)**.** Let $\forall i\in\{1,\dots,N\} : x_i^{\text{train}}, y_i^{\text{train}}\in\mathbb{R}$ and $\lambda\in\mathbb{R}_{>0}$. Then for a given function $g:\mathbb{R}\to\mathbb{R}_{\geq 0}$ the *adapted regression spline* $f_{g,\pm}^{*,\lambda}$ is

[30] Upon all global optima $w^*(\omega)$ of the training loss $L$, the minimum norm RSN $\mathcal{RN}_{w^{*,0+}(\omega),\omega}$ has weights $w^{*,0+}(\omega)$ with minimal norm. In the over-parameterized setting ($n\gg N$) there are infinitely many global optima $\mathcal{RN}_{w^*(\omega),\omega}$ with arbitrary large norm $\|w^*(\omega)\|_2$, but $w^{*,0+}(\omega)$ is always unique. If the number of hidden neurons $n$ is large enough (see Lemma 2.4), $w^{*,0+}$ could be equivalently defined as

$$w^{*,0+}(\omega) :\in \underset{w\in\mathbb{R}^n,\forall i\in\{1,\dots,N\}:\ \mathcal{RN}_{w,\omega}(x_i^{\text{train}})=y_i^{\text{train}}}{\arg\min} \|w\|_2 \quad \forall\omega\in\Omega.$$

[31] This statement holds in the limit $\frac{g}{g(0)}\to 1$. Formally eq. (13) in Definition 3.5 would not have a classical minimizer, if $g$ were constant (see footnote 34), but one could reformulate the definition of $P_{\pm}^{g}$ in Definition 3.5 by replacing the minimum by an infimum to extend Definition 3.5 to arbitrary weighting functions $g$ that do not have finite second momentum or that even have infinite integral like constant $g\equiv g(0)\neq 0$. For typical choices of distribution for the first-layer weights $v_k$ and biases $b_k$, the corresponding weighting function $g$ fulfills the finite second moment condition.

defined[32] as

$$f^{*,\lambda}_{g,\pm} \overset{32}{:\in} \underset{f\in\mathcal{C}^2(\mathbb{R})}{\arg\min} \underbrace{L(f) + \lambda P^g_\pm(f)}_{=:F^{\lambda,g}_\pm(f)}, \tag{13}$$

with

$$P^g_\pm(f) := 2g(0) \min_{\substack{(f_+,f_-)\in\mathcal{T}\\ f=f_++f_-}} \left( \int_{\mathrm{supp}(g)} \frac{\left(f_+''(x)\right)^2}{g(x)} dx + \int_{\mathrm{supp}(g)} \frac{\left(f_-''(x)\right)^2}{g(x)} dx \right),$$

and

$$\begin{aligned}\mathcal{T} := \Big\{ (f_+,f_-) \in \mathcal{C}^2(\mathbb{R})\times\mathcal{C}^2(\mathbb{R}) \Big|\ & \mathrm{supp}(f_+'') \subseteq \mathrm{supp}(g), \mathrm{supp}(f_-'') \subseteq \mathrm{supp}(g),\\ & \lim_{x\to-\infty} f_+(x) = 0, \lim_{x\to-\infty} f_+'(x) = 0,\\ & \lim_{x\to+\infty} f_-(x) = 0, \lim_{x\to+\infty} f_-'(x) = 0 \Big\}.\end{aligned}$$

*Remark* 3.6. If for the weighting function $g$ it holds that $\mathrm{supp}(g)$ is compact (cp. Assumption 2a)), we define

$$C^\ell_g := \min(\mathrm{supp}(g)) \quad \text{and} \quad C^u_g := \max(\mathrm{supp}(g)). \tag{14}$$

Furthermore, in that case, the set $\mathcal{T}$ can be rewritten: From $\mathrm{supp}(f_+'') \subseteq \mathrm{supp}(g)$ it follows that $f_+' \in \mathcal{C}^1(\mathbb{R})$ is constant on $(-\infty, C^\ell_g]$. With $\lim_{x\to-\infty} f_+'(x) = 0$ we obtain that $f_+'(x) = 0\ \forall x \le C^\ell_g$. By the same argument we obtain $f_+(x) = 0\ \forall x \le C^\ell_g$. Moreover, we have that $\exists\, c_+ \in \mathbb{R} : f_+'(x) \equiv c_+$ on $[C^u_g, \infty)$. Analogous derivations lead to $f_-'(x) \equiv c_-\ \forall x \le C^\ell_g$ with $c_- \in \mathbb{R}$ and $f_-(x) = f_-'(x) = 0$ on $[C^u_g, \infty)$. Hence, altogether, we have

$$\begin{aligned}\mathcal{T} = \Big\{ (f_+,f_-) \in \mathcal{C}^2(\mathbb{R})\times\mathcal{C}^2(\mathbb{R}) \Big|\ & \mathrm{supp}(f_+'') \subseteq \mathrm{supp}(g), \mathrm{supp}(f_-'') \subseteq \mathrm{supp}(g),\\ & \forall x \le C^\ell_g : f_+(x) = 0 = f_+'(x),\\ & \forall x \ge C^u_g : f_-(x) = 0 = f_-'(x) \Big\}.\end{aligned}$$

If we assume $\mathrm{supp}(g) = [C^\ell_g, C^u_g]$ we get:

$$\begin{aligned}\mathcal{T} = \Big\{ (f_+,f_-) \in \mathcal{C}^2(\mathbb{R})\times\mathcal{C}^2(\mathbb{R}) \Big|\ & \exists c_-, c_+ \in \mathbb{R} :\\ & \forall x \le C^\ell_g : \left(f_+(x) = 0 = f_+'(x)\ \wedge\ f_-'(x) = c_-\right),\\ & \forall x \ge C^u_g : \left(f_-(x) = 0 = f_-'(x)\ \wedge\ f_+'(x) = c_+\right) \Big\}.\end{aligned}$$

Building on Definition 3.5, we define an adapted version of the smooth spline interpolation.

[32]The adapted regression spline $f^{*,\lambda}_{g,\pm}$ is uniquely defined if $g$ is the probability density function of a distribution with finite first and second moment and $g(0) \neq 0$ (cp. Definition A.3 and footnote 55).

**Definition 3.7** (adapted spline interpolation)**.** Let $\forall i \in \{1,\dots,N\} : x_i^{\text{train}}, y_i^{\text{train}} \in \mathbb{R}$ and $\lambda \in \mathbb{R}_{>0}$. Then the *adapted spline interpolation* $f_{g,\pm}^{*,0+} : \mathbb{R} \to \mathbb{R}$ is defined [33] as:

$$f_{g,\pm}^{*,0+} := \lim_{\lambda \to 0+} f_{g,\pm}^{*,\lambda}. \tag{15}$$

Before stating the core result of this paper's analyses in Theorem 3.8, we like to discuss further assumptions we make therein. These requirements are technicalities that facilitate the proof of Theorem 3.8 and could be weakened (see footnotes 34–37).

**Assumption 2.** Using the notation from Definitions 2.1 and 2.7 the following assumptions extend Assumption 1:

a) The probability density function $g_\xi$ of the kinks $\xi_k$ has compact support $\operatorname{supp}(g_\xi)$. [34]
b) The density $g_\xi|_{\operatorname{supp}(g_\xi)}$ is uniformly continuous on $\operatorname{supp}(g_\xi)$.[35]
c) The reciprocal density $\frac{1}{g_\xi}\Big|_{\operatorname{supp}(g_\xi)}$ is uniformly continuous on $\operatorname{supp}(g_\xi)$.[36]
d) The conditioned distribution $\mathcal{L}(v_k|\xi_k = x)$ of $v_k$ is uniformly continuous in $x$ on $\operatorname{supp}(g_\xi)$. [37]
e) $\mathbb{E}\left[v_k^2\right] < \infty$.[38]

The following technical Assumption 3 makes the result of Theorem 3.8 more readable by referring to the easier Definition 3.5. Without Assumption 3, the Corollary 3.12 would still hold, which is more general than Theorem 3.8, but uses the heavier notation of Definition 3.9.

**Assumption 3.** Using the notation from Definitions 2.1 and 2.7 the following assumptions extend Assumption 1:

[33] Analogous to footnote 32 the spline interpolation $f_{g,\pm}^{*,0+}$ is uniquely defined if $g$ is the probability density function of a distribution with finite first and second moment and if $\exists (i,j) \in \{1,\dots,N\}^2 : x_i^{\text{train}} \neq x_j^{\text{train}}$.

[34] We believe that Assumption 2a) can be weakened quite extensively. However, for applications, it is not too restrictive given that real-world computers anyhow cover a compact range of numbers only. This assumption facilitates our proofs and it assures that a minimum of (30) exists. If one skips Assumption 2a) completely, it could happen that (30) does not have a classical minimum (e.g. $\mathbb{P}[v_k = -1] = \frac{1}{2} = \mathbb{P}[v_k = 1]$ and $b_k \sim$ Cauchy). As a remedy, one could define a weaker concept of minimum being the limit of minimizing sequences which converge to a unique function on every compact set. This also corresponds to the unique point-wise limit of minimizing sequences, which is not a classical minimum, because it does not satisfy all the boundary conditions $\lim_{x\to-\infty} f_+(x) = 0 = \lim_{x\to+\infty} f_-(x)$ anymore. For of this weaker minimum concept, Theorem 3.8 would need to be reformulated at least slightly, in case Assumption 2a) were entirely skipped. This weaker minimum concept can also be seen as the limit of adapted regression splines $f_{g,\pm}^{*,\lambda}$ for truncated $g$ as the range of the truncation tends to $(-\infty,\infty)$. This footnote will not be proved in this paper.

[35] One could think of replacing Assumption 2b) by the weaker assumption that $g_\xi$ is (improper) Riemann-integrable, however, almost all distributions which are typically used in practice satisfy Assumption 2b).

[36] Assumption 2c) implies that $\min_{x\in\operatorname{supp}(g_\xi)} g_\xi > 0$. Similarly to footnote 35, this assumption might be weakened in a way allowing $g_\xi$ to have finitely many jumps and $\min_{x\in\operatorname{supp}(g_\xi)} g_\xi$ to be zero.

[37] Similarly to footnote 35, Assumption 2d) might be attenuated.

[38] Assumption 2e) always holds in typical scenarios. Assumption 2e) together with Assumption 2a) and d) implies that $\mathbb{E}\left[v_k^2|\xi_k = x\right]$ is bounded on $\operatorname{supp}(g_\xi)$.

a) $g_\xi(0) \neq 0$.[39]
b) The distributions of the random weights and biases $v_k$ respectively $b_k$ are symmetric w.r.t the sign, i.e.
   i) $\mathbb{P}[v_k \in E] = \mathbb{P}[v_k \in -E] \quad \forall E \in \mathfrak{B}$ and
   ii) $\mathbb{P}[b_k \in E] = \mathbb{P}[b_k \in -E] \quad \forall E \in \mathfrak{B}$.

**Assumption 4.** The loss function[40] $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is
a) non-negative[41],
b) convex and
c) continuously differentiable[42] (i.e. $l(\cdot, y) \in \mathcal{C}^1(\mathbb{R}) \quad \forall y \in \mathbb{R}$)
in the first component.

**Theorem 3.8** (ridge weight penalty corresponds to adapted spline)**.** *Let $N \in \mathbb{N}$ be a finite number of arbitrary training data $(x_i^{train}, y_i^{train})$. Using the notation from Definitions 2.1, 2.7, 3.2 and 3.5 and let*[43] *$\forall x \in \mathbb{R} : g(x) := g_\xi(x)\mathbb{E}\left[v_k^2 | \xi_k = x\right]^{\frac{1}{2}}$ and $\tilde{\lambda} := \lambda n 2 g(0)$, then, under the Assumptions 1–4, the following statement holds for every compact set $K \subset \mathbb{R}$:*

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left\| \mathcal{RN}^{*,\tilde{\lambda}} - f^{*,\lambda}_{g,\pm} \right\|_{W^{1,\infty}(K)} = 0.^{44} \tag{16}$$

*Proof.* The proof of Theorem 3.8 is formulated in Appendix A.1. □

Without Assumption 3, Theorem 3.8 has to be reformulated to Corollary 3.12. This is done in the rest of this section.

**Definition 3.9** (asymmetric adapted spline regression)**.** Let $\forall i \in \{1, \dots, N\} : x_i^{\text{train}}, y_i^{\text{train}} \in \mathbb{R}$ and $\lambda \in \mathbb{R}_{>0}$. Then for given functions $g_+ : \mathbb{R} \to \mathbb{R}_{\geq 0}$, $g_- : \mathbb{R} \to \mathbb{R}_{\geq 0}$ the *asymmetric adapted regression spline* $f^{*,\lambda}_{g_+,g_-,\pm} := f^{*,\lambda}_{g_+,g_-,+} + f^{*,\lambda}_{g_+,g_-,-} + \gamma^{*,\lambda}_{g_+,g_-}$ is defined[45] as

$$\left(f^{*,\lambda}_{g_+,g_-,+}, f^{*,\lambda}_{g_+,g_-,-}, \gamma^{*,\lambda}_{g_+,g_-}\right) \overset{45}{\in} \underset{(f_+,f_-,\gamma)\in\mathcal{T}_{g_+,g_-}}{\arg\min} \underbrace{\left(L\left(f_+ + f_- + \gamma\right) + \lambda P^{g_+,g_-}(f_+, f_-, \gamma)\right)}_{=: F^{\lambda,g_+,g_-}_{+-}(f_+,f_-,\gamma)}, \tag{17}$$

[39] Assumption 3a) has to be satisfied due to the way Definition 3.5 and Theorem 3.8 are formulated, although the theory could be easily reformulated (see for instance Corollary 3.12) if Assumption 3a) were not satisfied. The theorems presented would hold as well if $g(0)$ were replaced by a fixed value $g(x_{\text{mid}})$ or by e.g. $\frac{1}{2}\int_{-1}^{1} g(x)dx$, however, the results are more easily interpreted if $x_{\text{mid}}$ is located somewhere "in the middle" of the training data. Theorem 3.8 would even hold true if $g(0) := 1$ (see Corollary 3.12 and Definition 3.9).

[40] Actually the main Theorem 3.8 is proven for even more general loss functions $l_i$ in Appendix A.1 (see Remark 3.18, Definition A.1 and Remark A.2)

[41] Assumption 4a) could be weakend—e.g. bounded from below should be sufficient, because w.l.o.g. one could subtract the lower bound.

[42] Assumption 4c) might be weakend to locally Lipschitz.

[43] Since all $v_k$ are identically distributed and all $\xi_k$ are identically distributed as well, the conditioned expectation $\mathbb{E}\left[v_k^2 | \xi_k = x\right]$ does not depend on the choice of $k \in \{1, \dots, n\}$.

[44] Using the definition of the $\mathbb{P}$-lim, equation (16) reads as: $\forall \epsilon \in \mathbb{R}_{>0} : \forall \rho \in (0,1) : \exists n_0 \in \mathbb{N} : \forall n \geq n_0 :$ $\mathbb{P}\left[\left\| \mathcal{RN}^{*,\tilde{\lambda}} - f^{*,\lambda}_{g,\pm} \right\|_{W^{1,\infty}(K)} < \epsilon\right] > \rho$.

[45] The optimization problem (17) should be interpreted such that $\frac{0}{0}$ is replaced by zero (For example, if $\mathbb{P}[v = 0] = 0$ the last fraction should be ignored.). The triple $\left(f^{*,\lambda}_{g_+,g_-,+}, f^{*,\lambda}_{g_+,g_-,-}, \gamma^{*,\lambda}_{g_+,g_-}\right)$ and thus the adapted regression spline $f^{*,\lambda}_{g,\pm}$ is uniquely defined if $g_+$,$g_-$ are probability density functions of distributions with finite first and second moment and if $\exists (i,j) \in \{1, \dots, N\}^2 : x_i^{\text{train}} \neq x_j^{\text{train}}$.

with

$$P^{g_+,g_-}(f_+,f_-,\gamma) := \int_{\text{supp}(g_+)} \frac{\left(f_+''(x)\right)^2}{g_+(x)}dx + \int_{\text{supp}(g_-)} \frac{\left(f_-''(x)\right)^2}{g_-(x)}dx + \frac{\gamma^2}{\mathbb{P}[v_k=0]\,\mathbb{E}\left[\max(0,b)^2\right]},$$

and

$$\mathcal{T}_{g_+,g_-} := \Big\{(f_+,f_-,\gamma)\in\mathcal{C}^2(\mathbb{R})\times\mathcal{C}^2(\mathbb{R})\times\mathbb{R}\,\Big|\,\text{supp}(f_+'')\subseteq\text{supp}(g_+),\text{supp}(f_-'')\subseteq\text{supp}(g_-),\ \lim_{x\to-\infty}f_+(x)=0,\ \lim_{x\to-\infty}f_+'(x)=0,\ \lim_{x\to+\infty}f_-(x)=0,\ \lim_{x\to+\infty}f_-'(x)=0,\ \mathbb{P}[v=0]=0\Rightarrow\gamma=0\Big\}.$$

*Remark* 3.10 (connection to Definition 3.5). If Assumption 3 holds, then

$$2g(0)P^{g_+,g_-}(f_+,f_-,0) = P^{g}_{+-}(f_+,f_-) \tag{18}$$

holds with $g=g_+=g_-$ and connects Definition 3.9 with Definitions 3.5 and A.3.[46]

**Definition 3.11** (conditioned kink position density $g_\xi^+$, $g_\xi^-$)**.** The *conditioned kink position density* $g_\xi^+ : \mathbb{R}\to\mathbb{R}$ of $\xi_k$ conditioned on $v_k>0$ is defined such that $\int_E g_\xi^+(x)dx = \mathbb{P}[\xi_k\in E|v_k>0] \quad \forall E\in\mathfrak{B}$. Analogously, $\int_E g_\xi^-(x)dx = \mathbb{P}[\xi_k\in E|v_k<0], \forall E\in\mathfrak{B}$.

**Corollary 3.12** (generalized Theorem 3.8)**.** *Let $N\in\mathbb{N}$ be a finite number of arbitrary training data $\left(x_i^{train},y_i^{train}\right)$. Using the notation from Definitions 2.1, 3.2, 3.9 and 3.11 and let $\forall x\in\mathbb{R}$ :*

$$g_+(x) := g_\xi^+(x)\mathbb{E}\left[v_k^2|\xi_k=x,v_k>0\right]\mathbb{P}\left[v_k>0\right],$$
$$g_-(x) := g_\xi^-(x)\mathbb{E}\left[v_k^2|\xi_k=x,v_k<0\right]\mathbb{P}\left[v_k<0\right],$$

*and $\tilde{\tilde{\lambda}} := \lambda n$. Then, under the Assumptions 1, 2 and 4, the following statement holds for every compact set $K\subset\mathbb{R}$:*

$$\mathbb{P}\text{-}\lim_{n\to\infty}\left\|\mathcal{RN}^{*,\tilde{\tilde{\lambda}}} - f^{*,\lambda}_{g_+,g_-,\pm}\right\|_{W^{1,\infty}(K)} = 0.^{47} \tag{19}$$

*Proof.* The proof of Corollary 3.12 is analagous to the proof of Theorem 3.8 in Appendix A.1. (The footnotes 56, 58 and 62 on pages 32 and 36 in Appendix A.1 help to understand this analogy.) □

3.2. **RSN and Gradient Descent → Implicit Ridge Regularization** ($d\in\mathbb{N}$)**.** We now move on to derive the relation between the RSN $\mathcal{RN}_{w^T}$ whose terminal-layer parameters is optimized performing gradient descent initialized at zero $w^0=0$ up to a certain time point $T$ on the one hand, and the ridge regularized RSN $\mathcal{RN}^{*,\tilde{\lambda}}$ with penalization parameter $\tilde{\lambda}$ on the other. In particular, we show that in the limit of infinite training time the solution $\mathcal{RN}_{w^\infty}$ obtained from the GD method corresponds to the one resulting by taking the limit $\tilde{\lambda}\to 0$ in the ridge problem (This solution is also referred to as minimum norm solution $\mathcal{RN}^{*,0+}$.). Note again, that this result is well known thanks to the work of i.a. [4, 9, 29, 12]. Within the

[46] This factor $2g(0)$ explains the difference between $\tilde{\lambda}:=\lambda n 2g(0)$ in Theorem 3.8 and $\tilde{\tilde{\lambda}}:=\lambda n$ in Corollary 3.12.

[47] Using the definition of the $\mathbb{P}$-lim, equation (19) reads as: $\forall\epsilon\in\mathbb{R}_{>0}:\forall\rho\in(0,1):\exists n_0\in\mathbb{N}:\forall n\geq n_0:$ $\mathbb{P}\left[\left\|\mathcal{RN}^{*,\tilde{\tilde{\lambda}}} - f^{*,\lambda}_{g_+,g_-,\pm}\right\|_{W^{1,\infty}(K)} < \epsilon\right] > \rho.$

present section, we like to collect the most important findings relating these two solutions within our setting.

Moreover, we will argue that, if suitably transformed, the ridge path mapping $\tilde{\lambda}$ to the optimal parameter corresponds to the GD path mapping training time to the corresponding parameter. Again, this equivalence has been discussed in the existing literature (e.g. [4, 9, 29]). In these works, it is frequently claimed that the GD solution at time $T$ approximately coincides with the ridge solution for $\tilde{\lambda} = 1/T$. We intend to make this relation more precise below (cp. eq. (26)). Within future work we will further analyze the errors arising from that approximate relation (see also Section 4 Item 3.).

Throughout this section, we consider the setting of supervised learning with squared loss, i.e., we require Assumption 5 to hold true. We begin by defining the trained RSNs $\mathcal{RN}_{w^T}$ obtained by pursuing the gradient flow w.r.t. this choice of training loss starting in the origin $w^0{=}0$ in parameter space up to time $T$.

**Assumption 5.** The loss function $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is given by $l(\hat{y}, y) := (\hat{y} - y)^2$.

**Definition 3.13** (time-$T$ solution)**.** Let $\forall i \in \{1, \dots, N\} : (x_i^{\text{train}}, y_i^{\text{train}}) \in \mathbb{R}^{d+1}$ for some $N, d \in \mathbb{N}$ and $\mathcal{RN}_w$ be a **r**andomized **s**hallow neural **n**etwork (RSN) with $n \in \mathbb{N}$ hidden nodes. For any $\omega \in \Omega$ and $T > 0$, the time-$T$ solution to the problem

$$\min_{w \in \mathbb{R}^n} \underbrace{\sum_{i=1}^{N} \left( \mathcal{RN}_{w,\omega}(x_i^{\text{train}}) - y_i^{\text{train}} \right)^2}_{L(\mathcal{RN}_{w,\omega})} \tag{20}$$

is defined as $\mathcal{RN}_{w^T(\omega),\omega}$, with weights $w^T(\omega) \in \mathbb{R}^n$ obtained by taking the gradient flow

$$\begin{aligned} dw^t &= -\nabla_w L\left(\mathcal{RN}_{w^t}\right) dt, \\ w^0 &= 0, \end{aligned} \tag{GD}$$

corresponding to (20) up to time $T$.

*Remark* 3.14. In practice, the weights $w^T$ of the time-$T$ solution as introduced in Definition 3.13 are approximated by taking $\tau := T/\gamma$ steps of size $\gamma > 0$ according to the Euler discretization

$$\begin{aligned} \check{w}^{t+\gamma} &= \check{w}^t - \gamma \nabla_w L(\mathcal{RN}_{\check{w}^t}), \\ \check{w}^0 &= 0, \end{aligned}$$

corresponding to (GD).

Within our setting, which in essence corresponds to a kernelized linear regression with random features, the time-$T$ solution takes an explicit form, as shown in Lemma 3.15.

**Lemma 3.15.** *Let $\forall i \in \{1, \dots, N\} : (x_i^{train}, y_i^{train}) \in \mathbb{R}^{d+1}$ for some $N, d \in \mathbb{N}$ and for any $\omega \in \Omega$, let $\mathcal{RN}_{w,\omega}$ be a randomized shallow network with $n \geq N$ hidden nodes. Define further $X(\omega) \in \mathbb{R}^{N \times n}$ via*

$$X_{i,k}(\omega) := \sigma\left( b_k(\omega) + \sum_{j=1}^{d} v_{k,j}(\omega) x_{i,j}^{train} \right) \quad \forall i \in \{1, \dots, N\} \, \forall k \in \{1, \dots, n\},$$

*where $x_{i,j}^{train}$ denotes the $j^{th}$ component of $x_i^{train}$. For any $T \geq 0$, the weights $w^T(\omega)$ corresponding to the time-$T$ solution $\mathcal{RN}_{w^T(\omega),\omega}$ satisfy*

$$w^T(\omega) = -\exp\left(-2TX^\top(\omega)X(\omega)\right) w^{*,0+}(\omega) + w^{*,0+}(\omega), \tag{21}$$

*with weights $w^{*,0+}(\omega)$ corresponding to the minimum norm network (see Definition 3.3).*

*Proof.* The proof of Lemma 3.15 is formulated in Appendix A.2. □

With the above, the asymptotic behavior of $w^T(\omega)$ is easily analyzed. As Remark 3.16 shows, the time-$T$ parameters $w^T(\omega)$ converge to the minimum norm parameters $w^{*,0+}(\omega)$ (see Definition 3.3). Consequently, the time-$T$ solution converges to the ridge penalized network when choosing the penalization accordingly, as is discussed in Theorem 3.17.

*Remark* 3.16 (limiting solution of gradient descent). By Lemma 3.15, the weights $w^T$ corresponding to the time-$T$ solution converge to the minimum norm solution $w^{*,0+}$ as time tends to infinity—i.e. taking the limit $T \to \infty$ in (21), we have $\lim_{T\to\infty} w^T(\omega) = w^{*,0+}(\omega)$ $\forall \omega \in \Omega$.

*Proof.* The proof of Remark 3.16 is formulated in Appendix A.2. □

**Theorem 3.17.** *Let $\mathcal{RN}_{w^T}$ be the time-$T$ solution and consider for $\tilde{\lambda} = \frac{1}{T}$ the corresponding ridge solution $\mathcal{RN}^{*,\frac{1}{T}}$ (cp. Definitions 3.2 and 3.13). We then have that*

$$\forall \omega \in \Omega: \quad \lim_{T\to\infty} \left\| \mathcal{RN}_\omega^{*,\frac{1}{T}} - \mathcal{RN}_{w^T(\omega),\omega} \right\|_{W^{1,\infty}(K)} = 0. \tag{22}$$

*Proof.* The proof of Theorem 3.17 is formulated in Appendix A.2. □

*Remark* 3.18 (Relaxed requirements on loss function). Without Assumption 5 Theorem 3.17 can still be proven, if instead one requires that $l(\cdot, y_i^{\text{train}}) : \mathbb{R} \to \mathbb{R}_{\geq 0}$ has a unique minimum for every $i = 1 \ldots, N$, where $l$ is given as in Assumption 4. This will be proven in future work.

3.2.1. *Early Stopping.* Moreover, we may use the representation (21) to derive an approximate relation between the weights $w^T$ corresponding to the time-$T$ solution and those obtained by performing a ridge regression with penalization parameter $\tilde{\lambda}$. The idea is to first analyze which singular value is trained most at a given time $T$ in an infinitesimal step along the solution path of $w^T$. In other words, we seek to find $s \geq 0$ that maximizes the gradient w.r.t. time of the singular values corresponding to the matrix exponential characterizing the time-$T$ solution, i.e. we solve

$$\arg\max_{s\geq 0} \nabla_T \exp(-2Ts) = \arg\max_{s\geq 0} -2s\exp(-2Ts).$$

The unique solution is given by

$$s^* = \frac{1}{2T}. \tag{23}$$

In a second step, we compare the closed-form solution of the parameters resulting from a $\tilde{\lambda}$-ridge regression to the time-$T$ solution, which we now consider to be characterized by

$s^*(T)$. To that end, we remark that using the singular value decomposition of the data matrix $X \in \mathbb{R}^{N\times n}$, i.e. $X = U\Sigma V^\top$ with

$$\Sigma = \begin{pmatrix} \operatorname{diag}(\sqrt{s_1}, \dots, \sqrt{s_r}) & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{N\times n},$$

these solutions may be written as

$$w^T = -V \begin{pmatrix} \operatorname{diag}\left(\frac{\exp(-2Ts_1)-1}{\sqrt{s_1}}, \dots, \frac{exp(-2Ts_r)-1}{\sqrt{s_r}}\right) & 0 \\ 0 & 0 \end{pmatrix} U^\top y, \tag{24}$$

$$w_{\tilde\lambda} = V \begin{pmatrix} \operatorname{diag}\left(\frac{\sqrt{s_1}}{s_1+\tilde\lambda}, \dots, \frac{\sqrt{s_r}}{s_r+\tilde\lambda}\right) & 0 \\ 0 & 0 \end{pmatrix} U^\top y. \tag{25}$$

We then arrive at the ridge estimate approximating the time-$T$ solution by comparing eqs. (24) and (25) for the singular value $s^*$, i.e., the one that is most affected by the training at time-point $T$. Hence, we relate the time-$T$ solution to the ridge solution obtained using the penalization parameter

$$\tilde\lambda(T) = \frac{1}{2T(e-1)}. \tag{26}$$

Note that, by the above relation $\tilde\lambda(T)$ still is of order $1/T$ and hence the asymptotic behavior that we characterize in Theorem 3.17 below, is sufficiently captured taking the relation $\tilde\lambda(T) = 1/T$. However, for comparing the early-stopped time-$T$ solution $\mathcal{RN}_{w^T}$ to a ridge regularized RSN $\mathcal{RN}^{*,\tilde\lambda}$ and, as a consequence, to a certain regression spline $f^{*,\lambda} \approx f^{*,\lambda}_{g,\pm}$, we make use of the precise relation (26). See also Section 4 for empirical results, that underline the quality of the fit.

## 4. Conclusion and Future Work

Combining the main Theorems 3.8 and 3.17 finally yields our main result: for a large number of training epochs $\tau = T/\gamma$, the obtained **w**ide (large number of neurons $n$) **R**eLU **r**andomized **s**hallow neural **n**etwork (wRRSN)

$$\mathcal{RN}_{\tilde w^T, \tilde w^0} \overset{\tilde w^0 \to 0}{\approx} \mathcal{RN}_{\tilde w^T} \overset{\gamma\to 0}{\approx} \mathcal{RN}_{w^T} \underset{\text{Theorem 3.17}}{\overset{T\to\infty}{\approx}} \mathcal{RN}^{*,0+} \underset{\text{Theorem 3.8}}{\overset{\mathbb{P}}{\underset{n\to\infty}{\approx}}} f^{*,0+}_{g,\pm} \overset{\frac{g}{g(0)}\to 1}{\approx} f^{*,0+} \tag{27}$$

is very close to the spline interpolation $f^{*,0+}$. Here, the notation $\overset{\to}{\approx}$ corresponds to a mathematically proved exact limit in the very strong[48] Sobolev norm $\|\cdot\|_{W^{1,\infty}(K)}$ (in probability in the case of $\overset{\mathbb{P}}{\underset{n\to\infty}{\approx}}$).

In applications, however, both the number of hidden nodes and training steps are finite. Hence, it is particularly interesting to note that in typical settings for *arbitrary* training time $T \in \mathbb{R}_{>0}$ (including early stopping, i.e. $T \ll \infty$) the same relation approximately holds true. In other words, by taking $T \overset{(26)}{=} \frac{1}{2\tilde\lambda(e-1)}$ and $\tilde\lambda \overset{\text{Th. 3.8}}{=} \lambda n 2g(0)$, we have

[48] Convergence in $\|\cdot\|_{W^{1,\infty}(K)}$ implies uniform convergence on $K$ or convergence in $W^{1,p}(K)$. Even stronger Sobolev convergenve, such as convergence w.r.t. $W^{2,p}$, cannot be shown since $\mathcal{RN}_w \notin W^{2,p}(K)$.

$$\mathcal{RN}_{\check{w}^T,\check{w}^0} \overset{\check{w}^0\approx 0}{\underset{1.}{\approx}} \mathcal{RN}_{\check{w}^T} \overset{\gamma\approx 0}{\underset{2.}{\approx}} \mathcal{RN}_{w^T} \underset{3.}{\approx} \mathcal{RN}^{*,\tilde{\lambda}} \overset{\mathbb{P}\text{ n large}}{\underset{4.}{\approx}} f_{g,\pm}^{*,\lambda} \overset{\substack{\text{standard distrib.}\\ \text{for } v \text{ and } b\\ \text{and } K\subseteq[-1,1]}}{\underset{5.}{\approx}} f^{*,\lambda}, \tag{28}$$

where "$\approx$" represents equality up to a (small) approximation error (that can be strictly larger than zero).

It is planned to give a more detailed description of approximation (28) in future work. To give an outlook, we remark the following.

1. The first approximation should be quite simple but is not focused on within this work.[49] (As only the last layer of $\mathcal{RN}$ is trained, one could just start with $w^0 = 0$)
2. It is of importance to choose the learning rate $\gamma$ rather small.[50] As will be discussed in future work, stochastic gradient descent allows to chose $\gamma$ such that the effective step size per floating point operation is larger. Note, that by the above discussions we have that for an RSN $\mathcal{RN}$ the learning rate $\gamma$ should typically be chosen approximately inversely proportional to the number of neurons $n$. Another interesting insight that we might elaborate on in more detail in upcoming work is that the "approximation error" we get from larger values of $\gamma$ has a very specific structure that allows to some extent to explain it on a macroscopic functional level.
3. Multiple papers assume that the third approximation is quite precise for arbitrary values of $T \in \mathbb{R}_{>0}$ without rigorous proof [4, 9, 29]. We believe that these "approximation errors" which typically are "rather small" but not vanishing could even cancel with the "approximation errors" in 5. to some extent, thus having a positive effect on the convergence. This theory could be part of close future work. 3. would be particularly interesting for real-world applications, since it gives an improved understanding of the solution functions obtained by stopping the GD algorithm *early*.[51]
4. The mathematically precise asymptotic relation is the subject of Theorem 3.8. We refer to future work for quantitative bounds discussing the number of neurons needed to achieve approximation up to a certain accuracy.
5. The adapted regression spline $f_{g,\pm}^{*,\lambda}$ is a macroscopically defined object that already is nice to interpret. Intuitively, it is plausible that $f_{g,\pm}^{*,\lambda}$ is very close to the very desirable $f^{*,\lambda}$ on the $[-1,1]$-cube (and in its close surrounding), if one uses typical[52] distributions for the first-layer weights and biases $v$ and $b$, and if the training data is scaled and shifted to fit into the $[-1,1]$-cube. Additionally, by that same intuition, it follows that if popular rules of thumb such as scaling and shifting the

[49] Lemma A.17 demonstrates, that with increasing $n$ the initial weights $\check{w}^0$ should be chosen closer to zero.

[50] For finite values of $T$ a standard result on Euler discretization can be used. In the limit $T \to \infty$ one can formulate a direct argument that combines items 2. and 3.: $\lim_{T\to\infty} \check{w}^T = w^{*,0+}$, if the learning rate $\gamma < 1/r(X^\top X)$ is smaller than 1 over the spectral radius (largest eigenvalue) of $X^\top X$ [4, p. 4] [12, p. 11].

[51] We note, that it might be more reasonable to chose $\tilde{\lambda} = \frac{se^{-2sT}}{1-e^{-2sT}}$ instead of $\tilde{\lambda} = \frac{1}{T}$, with an appropriate choice of $s$ (cp. eqs. (23) and (26)) to get better approximation bounds. Nonetheless, in Section 3 and eq. (27) we work with the relation $\tilde{\lambda} = \frac{1}{T}$, as it is commonly suggested in literature [4, Section 2.3 on p. 5]. Moreover, in the limit $T \to \infty$ these relations coincide.

[52] For instance, one could choose $b_k, v_k \sim Unif(-c, c)$ i.i.d. uniformly symmetrically distributed or $b_k, v_k \sim \mathcal{N}(0, c)$ i.i.d. normally distributed with zero mean.

data to the $[-1,1]$-cube are broken, one can obtain rather poor approximations $f^{*,\lambda}_{g,\pm}$. Consequently, by providing these insights on the circumstances that would lead to undesirable results, Theorem 3.8 greatly contributes to answering question IV about best practices in machine learning.

4.1. **Empirical results.** As a proof of concept we like to empirically verify the approximate relations (28) discussed above. To that end, we consider the aim of approximating the function $f : \mathbb{R} \to \mathbb{R}, x \mapsto \sin(\pi x)$, given $N = 16$ noisy data points $(x_i, f(x_i) + \epsilon_i) \in \mathbb{R}^2$, where $x_i$, $i = 1, \dots, N$ are equidistant points in the interval $[-1, 1]$ and $\epsilon_i$ are realizations of a centred Gaussian random variable with standard deviation $\texttt{scale} = \texttt{1/8}$. Figure 6 shows a comparison of the solution functions obtained by

a) training an RSN with ReLU activation using a standard implementation of gradient descent with step size $\gamma = 2^{-11}$ for $\tau = 2^{15}$ epochs (resulting in $\mathcal{RN}_{\check{w}^T}$ with $T = \tau\gamma$),
b) training that same RSN using a ridge penalty on the terminal weights with penalization parameter $\tilde{\lambda} = \frac{1}{e-1}\frac{1}{2\tau\gamma}$ according to eq. (26) (resulting in $\mathcal{RN}^{*,\tilde{\lambda}}$) and
c) the spline regression with penalization parameter $\lambda = \frac{\tilde{\lambda}}{n2g(0)}$ (resulting in $f^{*,\lambda}$). (Here, $n$ represents the RSN's number of hidden nodes and the weighting function $g$ is defined in Theorem 3.8.)

The RSN was chosen to consist of $n = 2^{12}$ hidden nodes with first-layer weights and biases sampled from a Uniform distribution on $[-0.05, 0.05]$. Moreover, a last-layer bias was included in the training (cp. Footnote 24).

Within this paper's setting, this experiment corresponds to comparing the time-$T$ solution $\mathcal{RN}_{\check{w}^T}$ for $T = 16$ to the ridge regularized RSN $\mathcal{RN}^{*,\tilde{\lambda}}$ with $\tilde{\lambda} = \frac{1}{e-1}\frac{1}{2T} \approx 0.018$ and the smooth regression spline $f^{*,\lambda}$ with penalization parameter $\lambda \approx 0.014$.
As Figure 6 nicely shows, the three solution functions almost coincide on $[-1, 1]$. This is of particular interest, since the training data typically is scaled to fit the interval $[-1, 1]$.

In certain situations (that will be explained in Appendix B) the adapted regression spline $f^{*,\lambda}_{g,\pm} \approx \mathcal{RN}^{*,\tilde{\lambda}}$ can deviate more from the classical regression spline $f^{*,\lambda}$ as can be seen in Figure 7 far outside the training data. The RSN's architecture could be extended to incorporate a direct affine link onto the output, which, when included in the training process, can make up for the observed difference (see also item ii below). However, as indicated in item 3., this deviation might be an empirical hint of how the errors occurring in the approximation of the ridge regularized RSN $\mathcal{RN}^{*,\tilde{\lambda}}$ by the RSN $\mathcal{RN}_{w^T}$ on the one hand, and the approximation of the regression spline $f^{*,\lambda}$ by the ridge regularized RSN $\mathcal{RN}^{*,\tilde{\lambda}}$ on the other are partially cancelling under certain conditions, such that the fitted RSN $\mathcal{RN}_{w^T}$ in fact is closer to the regression spline $f^{*,\lambda}$ than the ridge regularized RSN $\mathcal{RN}^{*,\tilde{\lambda}}$.

A more detailed view on the trained RSN $\mathcal{RN}_{w^T}$ is given in Figure 8. Therein, we visualize the RSN's (distributional) second derivative at the respective realized kink positions as well as a convoluted version of it using a Gaussian kernel. We observe that on average, the RSN's curvature is evenly spread among neighboring kinks.

4.2. **Future work.** Besides discussing the correspondence of the spline interpolation and an RSN trained using gradient descent for a finite number of nodes and finite training time, we intend to extend the theory in upcoming work as follows:

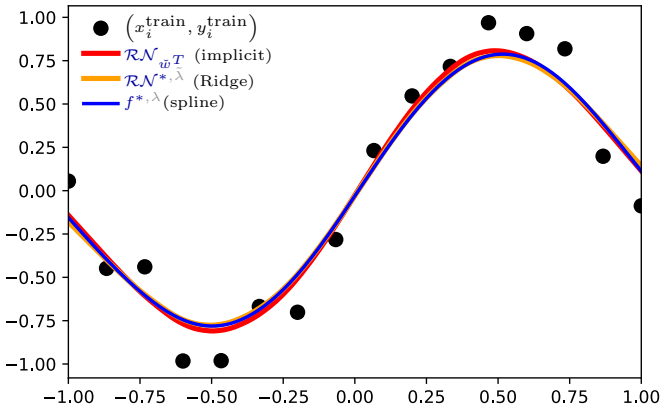


FIGURE 6. Comparison of the solution functions obtained from performing gradient descent (red line) and ridge regularization (yellow line) to train an RSN to the spline regression (blue line) with parameters chosen as suggested by eq. (26) and Theorem 3.8.

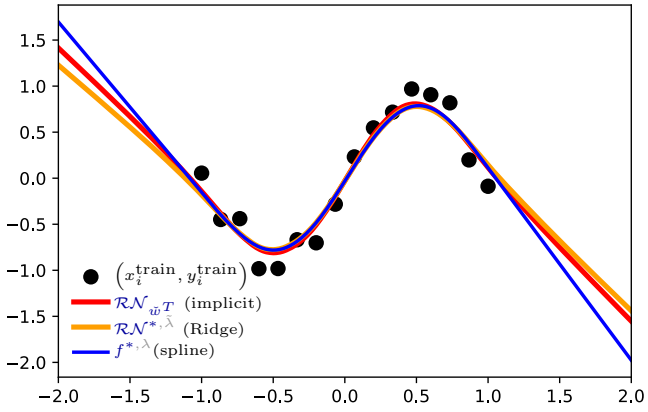


FIGURE 7. Large scale comparison of the solution functions as in Figure 6. Outside the training data, the trained RSN $\mathcal{RN}_{\tilde{w}^T}$ ranges in between the ridge regularized RSN $\mathcal{RN}^{*,\tilde{\lambda}}$ and the regression spline $f^{*,\lambda}$.

i. Generalizing to multidimensional input in $\mathcal{X} = \mathbb{R}^d$ (see part II).[53]

[53]Since we will publish these theorems very soon, it would be a waste of resources if multiple people work on it independently. If you are working on similar results, it makes sense to collaborate—if you want to do so, please contact one of the authors.

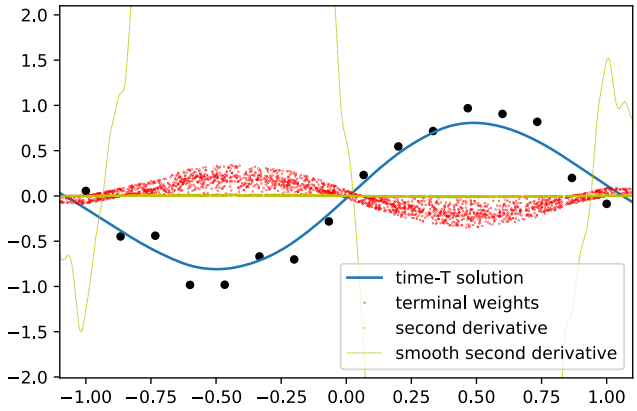


FIGURE 8. The trained RSN $\mathcal{RN}_{w^T}$ (blue line) and its (distributional) second derivative $\sum_{k=1}^{n} v_k w_k \delta_{\xi_k}$ (yellow dots) at the respective realized kink positions and a smoothed version of it. The smooth second derivative was obtained from a convolution using a Gaussian kernel. It nicely captures the trained RSN's curvature. Moreover, the values of the terminal layer's weights $w_k$ at the respective kink positions $\xi_k$ are given (red dots).

ii. With the insights gained from Theorem 3.8, possibilities arise how to save computational time, memory and energy consumption by replacing certain groups of neurons by other algorithms (or simply by adding certain direct connections from input to the output skipping the hidden layer). This can also offer other advantages[54]. Theorem 3.8 and its proof inspire to choose special types of randomness for the first-layer weights and biases. Naturally, we are interested to find out whether these choices provide advantages in the training of such $\mathcal{RN}$ or other architectures.[53]
iii. Proving convergence to a differently regularized function (which is optimal with respect to another $P$-functional) in the case of ordinary training of both layers of $\mathcal{NN}$ instead of only training the last layer (see part III).[53]
iv. Generalization do deep neural networks with more hidden layers (e.g. deep convolutional neural networks). The long-term goal of this line of research is to find a $P$-functional (or another easy to interpret macroscopic description) for each type of neural network for each set of meta-parameters. (This could be extended to other Machine Learning methods like random forests too.)[53]

## REFERENCES

[1] Robert A Adams and John J F Fournier. *Sobolev spaces; 2nd ed.* Pure and applied mathematics. Academic Press, New York, NY, 2003. URL http://cds.cern.ch/record/1990498. 34, 39, 48

[54]Certain modifications of the network could render the algorithm numerically more stable and adjustments of the regularization could be implemented—e.g. the adapted regression spline can easily be modified to the ordinary regression spline.

[2] Adi Ben-Israel and Thomas NE Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003. URL https://doi.org/10.1007%2Fb97366. 50

[3] M. Bianchini and F. Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*, 25(8):1553–1565, August 2014. ISSN 2162-237X. doi: 10.1109/TNNLS.2013.2293637. URL https://doi.org/10.1109/TNNLS.2013.2293637. 1

[4] Chris M. Bishop. Regularization and complexity control in feed-forward networks. In *Proceedings International Conference on Artificial Neural Networks ICANN'95*, volume 1, pages 141–148. EC2 et Cie, January 1995. URL https://www.microsoft.com/en-us/research/publication/regularization-and-complexity-control-in-feed-forward-networks/. 11, 21, 22, 25, 50

[5] Christopher M Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY, 2006. ISBN 978-0387-31073-2. URL http://cds.cern.ch/record/998831. 5

[6] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046, 2018. URL https://arxiv.org/abs/1805.09545v2 13

[7] Peter Craven and Grace Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403, December 1978. ISSN 0945-3245. doi: 10.1007/BF01404567. URL https://doi.org/10.1007/BF01404567. 5, 10

[8] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, December 1989. ISSN 1435-568X. doi: 10.1007/BF02551274. URL https://doi.org/10.1007/BF02551274. 7, 14

[9] Jerome Friedman and Bogdan E. Popescuy. Gradient directed regularization for linear regression and classification. *Tech rep*, January 2004. URL https://www.researchgate.net/publication/244258820_Gradient_Directed_Regularization_for_Linear_Regression_and_Classification. 11, 21, 22, 25, 50

[10] Carl Friedrich Gauß. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium auctore Carolo Friderico Gauss*. sumtibus Frid. Perthes et IH Besser, 1809. URL https://books.google.at/books?id=VKhu8yPcat8C. 3

[11] Carl-Friedrich Gauß. *Theoria combinationis observationum erroribus minimis obnoxiae.-Gottingae, Henricus Dieterich 1823*. Henricus Dieterich, 1823. URL https://books.google.at/books?id=hrZQAAAAcAAJ. 3

[12] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit Regularization of Discrete Gradient Dynamics in Deep Linear Neural Networks. *arXiv e-prints*, art. arXiv:1904.13262, April 2019. URL https://arxiv.org/abs/1904.13262v1. 6, 9, 11, 21, 25, 50

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 6, 17

[14] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009. ISBN 978-0-387-84858-7. doi: 10.1007/978-0-387-84858-7. URL http://www-stat.stanford.edu/~tibs/ElemStatLearn/. 9

[15] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991. ISSN 0893-6080. doi: 10.1016/0893-6080(91)90009-T. URL https://doi.org/10.1016/0893-6080(91)90009-T. 7, 14

[16] Yoshifusa Ito. Approximation of functions on a compact set by finite sums of a sigmoid function without scaling. *Neural Networks*, 4(6):817 – 826, 1991. ISSN 0893-6080. doi: 10.1016/0893-6080(91)90060-I. URL https://doi.org/10.1016/0893-6080(91)90060-I. 1

[17] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018. URL https://arxiv.org/abs/1806.07572v3. 12, 13

[18] George S. Kimeldorf and Grace Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502, 1970. ISSN 00034851. URL http://www.jstor.org/stable/2239347. 5, 6, 10

[19] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, art. arXiv:1412.6980, December 2014. URL https://arxiv.org/abs/1412.6980. 9

[20] Masayoshi Kubo, Ryotaro Banno, Hidetaka Manabe, and Masataka Minoji. Implicit Regularization in Over-parameterized Neural Networks. *arXiv e-prints*, art. arXiv:1903.01997, March 2019. URL https://arxiv.org/abs/1903.01997. 9, 12

[21] Adrien Marie Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes*. Number 1 in Analyse des triangles tracés sur la surface d'un sphéroide. F. Didot, 1805. URL https://books.google.at/books?id=7C9RAAAAYAAJ. 3

[22] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993. URL https://doi.org/10.1016/S0893-6080(05)80131-5 1, 14, 51

[23] Yuanzhi Li and Yingyu Liang. Learning Overparameterized Neural Networks via Stochastic Gradient Descent on Structured Data. *arXiv e-prints*, art. arXiv:1808.01204, August 2018. URL https://arxiv.org/abs/1808.01204v3. 9, 12

[24] Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient Descent Quantizes ReLU Network Features. *arXiv e-prints*, art. arXiv:1803.08367, March 2018. URL https://arxiv.org/abs/1803.08367v1. 9, 12

[25] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018. URL https://doi.org/10.1073/pnas.1806579115 13

[26] Behnam Neyshabur. Implicit Regularization in Deep Learning. *arXiv e-prints*, art. arXiv:1709.01953, September 2017. URL https://arxiv.org/abs/1709.01953v2. 9

[27] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning. *arXiv e-prints*, art. arXiv:1412.6614, December 2014. URL https://arxiv.org/abs/1412.6614v4. 9

[28] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990. URL https://doi.org/10.1109/5.58326 13

[29] Tomaso Poggio, Qianli Liao, Brando Miranda, Andrzej Banburski, Xavier Boix, and Jack Hidary. Theory IIIb: Generalization in Deep Networks. *arXiv e-prints*, art.

arXiv:1806.11379, June 2018. URL https://arxiv.org/abs/1806.11379v1. 1, 9, 11, 12, 21, 22, 25, 50

[30] Christian H. Reinsch. Smoothing by spline functions. *Numerische Mathematik*, 10(3): 177–183, October 1967. ISSN 0945-3245. doi: 10.1007/BF02162161. URL https://doi.org/10.1007/BF02162161. 5, 10

[31] Uri Shaham, Alexander Cloninger, and Ronald R Coifman. Provable approximation properties for deep neural networks. *Applied and Computational Harmonic Analysis*, 44(3):537–557, 2018. URL https://doi.org/10.1016/j.acha.2016.04.003. 1

[32] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The Implicit Bias of Gradient Descent on Separable Data. *arXiv e-prints*, art. arXiv:1710.10345, October 2017. URL https://arxiv.org/abs/1710.10345v4. 9, 11

[33] Grace Wahba. Improper priors, spline smoothing and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 40(3):364–372, 1978. URL https://doi.org/10.1111/j.2517-6161.1978.tb01050.x 6

## Appendix A. Proofs

In the following, we rigorously prove the results presented within this paper.

A.1. **Proof of Theorem 3.8 ($\mathcal{RN}^{*,\tilde{\lambda}} \to f_{g,\pm}^{*,\lambda}$).** A number of lemmata are required for the proof of Theorem 3.8. These will be presented and proved later in this section. We start by defining the objects that are central to the subsequent derivations.

Throughout this section, we henceforth require Assumptions 1–4 to be in place.

**Definition A.1** (generalized $L$)**.** Let $\forall i \in \{1,\dots,N\} : x_i^{\text{train}} \in \mathbb{R}$ for some $N \in \mathbb{N}$. Let $l_i : \mathbb{R} \to \mathbb{R}_{\geq 0}$, $i = 1,\dots,N$ be convex and continuously differentiable loss functions. Then, the generalized training loss $L$ of function $f : \mathbb{R} \to \mathbb{R}$ is defined as

$$L(f) := \sum_{i=1}^{N} l_i\left(f(x_i^{\text{train}})\right). \tag{29}$$

*Remark* A.2. The training loss $L$ defined in (1) is a special case of (29) with $l_i(\hat{y}) := l(\hat{y}, y_i^{\text{train}})$. This special case is sufficient to prove Theorem 3.8, but the proof is formulated for more general choices of $l_i$ (see Definition A.1) where the shape of the loss $l_i$ can depend on the index $i$.

**Definition A.3.** Let $\forall i \in \{1,\dots,N\} : x_i^{\text{train}}, y_i^{\text{train}} \in \mathbb{R}$ and $\lambda \in \mathbb{R}_{>0}$. Then for a given function $g : \mathbb{R} \to \mathbb{R}_{\geq 0}$ the tuple $(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda})$ is defined[55] as

$$\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right) \overset{55}{:\in} \underset{(f_+,f_-)\in\mathcal{T}}{\arg\min} \underbrace{L(f_+ + f_-) + \lambda P_{+-}^{g}(f_+, f_-)}_{=:F_{+-}^{\lambda,g}(f_+,f_-)}, \tag{30}$$

[55]The tuple $\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right)$ is uniquely defined if $g$ is the probability density function of a distribution with finite first and second moment and $g(0) \neq 0$. Thus, by Remark A.4, the same holds true for the adapted regression spline $f_{g,\pm}^{*,\lambda}$.

with

$$P^g_{+-}(f_+, f_-) := 2g(0)\left(\int_{\mathrm{supp}(g)} \frac{\left(f_+''(x)\right)^2}{g(x)}dx + \int_{\mathrm{supp}(g)} \frac{\left(f_-''(x)\right)^2}{g(x)}dx\right).$$

*Remark* A.4. Note that the adapted regression spline $f^{*,\lambda}_{g,\pm}$ is given by

$$f^{*,\lambda}_{g,\pm} = f^{*,\lambda}_{g,+} + f^{*,\lambda}_{g,-}.$$

**Definition A.5** (estimated kink distance $\bar{h}$ w.r.t. $\mathrm{sgn}(v)$)**.** Let $\mathcal{RN}$ be a randomized shallow neural network with $n$ hidden nodes as introduced in Definition 2.1. The *estimated kink distance w.r.t.* $\mathrm{sgn}(v)$ at the $k^{\text{th}}$ kink position $\xi_k$ corresponding to $\mathcal{RN}$ is defined as[56]

$$\bar{h}_k := \frac{2}{n\, g_\xi(\xi_k)}. \tag{32}$$

**Definition A.6** (spline approximating RSN)**.** Let $\mathcal{RN}$ be a real-valued randomized shallow neural network with $n$ hidden nodes (cp. Definition 2.1) and $f^{*,\lambda}_{g,\pm} = f^{*,\lambda}_{g,+} + f^{*,\lambda}_{g,-} \in \mathcal{C}^2(\mathbb{R})$ be the adapted regression spline as introduced in Definitions 3.5 and A.3. The *spline approximating RSN* $\mathcal{RN}_{\tilde{w}}$ w.r.t. $f^{*,\lambda}_{g,\pm}$ is given by

$$\mathcal{RN}_{\tilde{w}(\omega),\omega}(x) = \sum_{k=1}^{n} \tilde{w}_k(\omega)\, \sigma\left(b_k(\omega) + v_k(\omega)x\right) \quad \forall \omega \in \Omega\ \forall x \in \mathbb{R} \tag{33}$$

with weights $\tilde{w}(\omega)$ defined as[57, 58]

$$\tilde{w}_k(\omega) := w_k^{f^{*,\lambda}_{g,\pm},n}(\omega) := \begin{cases} \frac{\bar{h}_k(\omega)v_k(\omega)}{\mathbb{E}[v^2|\xi=\xi_k(\omega)]} f^{*,\lambda}_{g,+}{}''(\xi_k(\omega)), & v_k(\omega) > 0 \\ \frac{-\bar{h}_k(\omega)v_k(\omega)}{\mathbb{E}[v^2|\xi=\xi_k(\omega)]} f^{*,\lambda}_{g,-}{}''(\xi_k(\omega)), & v_k(\omega) < 0 \end{cases} \qquad \begin{array}{c} \forall k \in \{1,\dots,n\} \\ \forall \omega \in \Omega. \end{array}$$

[56]Without Assumption 3b) one would define:

$$\bar{h}^+{}_k := \frac{1}{n\mathbb{P}\left[v_k > 0\right]\, g^+_\xi(\xi_k)} \tag{31a}$$

$$\bar{h}^-{}_k := \frac{1}{n\mathbb{P}\left[v_k < 0\right]\, g^-_\xi(\xi_k)}. \tag{31b}$$

Under Assumption 3b) we have the equality:

$$\bar{h}_k = \bar{h}^+{}_k = \bar{h}^-{}_k. \tag{31c}$$

[57]Since all $v_k$ are identically distributed and all $\xi_k$ are identically distributed as well, the conditioned expectation $\mathbb{E}\left[v_k^2 \middle| \xi_k = x\right]$ does not depend on the choice of $k \in \{1,\dots,n\}$. Therefore, we will sometimes use the following notation $\mathbb{E}\left[v^2 \middle| \xi = x\right] := \mathbb{E}\left[v_k^2 \middle| \xi_k = x\right]$.

[58]Note that under Assumption 1b), the set $\{v_k = 0\}$ is of zero measure for any $k \in \{1,\dots,n\}$ and hence is not included in the definition of the weights $\tilde{w}(\omega)$. Without Assumption 3b) (and with a weakened form of Assumption 1b)), $\tilde{w}$ would need to be reformulated:

$$\tilde{w}_k(\omega) := w_k^{f^{*,\lambda}_{g+,g-,\pm},n}(\omega) := \begin{cases} \frac{\bar{h}^+{}_k(\omega)v_k(\omega)}{\mathbb{E}\left[v^2|\xi=\xi_k(\omega),v>0\right]} f^{*,\lambda}_{g+,g-,+}{}''(\xi_k(\omega)), & v_k(\omega) > 0 \\ \frac{-\bar{h}^-{}_k(\omega)v_k(\omega)}{\mathbb{E}\left[v^2|\xi=\xi_k(\omega),v<0\right]} f^{*,\lambda}_{g+,g-,-}{}''(\xi_k(\omega)), & v_k(\omega) < 0 \\ \frac{\max(0,b_k(\omega))}{n\mathbb{P}[v=0]\mathbb{E}\left[\max(0,b)^2\right]} \gamma^{*,\lambda}_{g+,g-}, & v_k(\omega) = 0 \end{cases} \qquad \begin{array}{c} \forall k \in \{1,\dots,n\} \\ \forall \omega \in \Omega. \end{array}$$

We further define $\forall \omega \in \Omega$:

$$\mathfrak{K}^+(\omega) := \{\, k \in \{1, \dots, n\} \mid v_k(\omega) > 0 \,\}, \tag{34a}$$

$$\mathfrak{K}^-(\omega) := \{\, k \in \{1, \dots, n\} \mid v_k(\omega) < 0 \,\} \tag{34b}$$

and $\tilde{w}^+ := (\tilde{w}_k)_{k \in \mathfrak{K}^+}$ respectively $\tilde{w}^- := (\tilde{w}_k)_{k \in \mathfrak{K}^-}$. With the above, spline approximating RSNs can be alternatively represented as

$$\mathcal{RN}_{\tilde{w}(\omega),\omega}(x) = \underbrace{\sum_{k \in \mathfrak{K}^+(\omega)} \tilde{w}_k(\omega)\, \sigma\left(b_k(\omega) + v_k(\omega)x\right)}_{=:\mathcal{RN}^+_{\tilde{w}^+(\omega),\omega}} + \underbrace{\sum_{k \in \mathfrak{K}^-(\omega)} \tilde{w}_k(\omega)\, \sigma\left(b_k(\omega) + v_k(\omega)x\right)}_{=:\mathcal{RN}^-_{\tilde{w}^-(\omega),\omega}}. \tag{35}$$

*Remark* A.7. The spline approximating RSN introduced in Definition A.6 is a particular randomized shallow neural network designed to be "close" to the adapted regression spline $f^{*,\tilde{\lambda}}_{g,\pm}$ in the sense that its curvature in between kinks is approximately captured by the size of corresponding weights $\tilde{w}$.

**Definition A.8** (smooth RSN approximation)**.** For $w^{*,\tilde{\lambda}}$ and $\mathcal{RN}^{*,\tilde{\lambda}}$ as in Definition 3.2 with corresponding kink density $g_\xi$ consider for every $x \in \mathbb{R}$ the kernel

$$\kappa_x : \mathbb{R} \to \mathbb{R}, \quad \kappa_x(s) := \mathbb{1}_{B_{\frac{1}{2\sqrt{n} g_\xi(x)}}}(s) \sqrt{n} g_\xi(x) \quad \forall s \in \mathbb{R},$$

where $B_{\frac{1}{2\sqrt{n} g_\xi(x)}} := \{\tau \in \mathbb{R} : |\tau| \leq \frac{1}{2\sqrt{n} g_\xi(x)}\}$. The *smooth RSN approximation* $f^{w^{*,\tilde{\lambda}}}$ is then defined as the convolution[59]

$$f^{w^{*,\tilde{\lambda}}(\omega)}(x) := \left(\mathcal{RN}^{*,\tilde{\lambda}}_\omega * \kappa_x\right)(x) \qquad \forall \omega \in \Omega\ \forall x \in \mathbb{R}. \tag{36}$$

Moreover, with the notation

$$\mathcal{RN}^{*,\tilde{\lambda}}(x) = \underbrace{\sum_{k \in \mathfrak{K}^+} w^{*,\tilde{\lambda}}_k\, \sigma\left(b_k + v_k x\right)}_{=:\mathcal{RN}^{*,\tilde{\lambda}}_+} + \underbrace{\sum_{k \in \mathfrak{K}^-} w^{*,\tilde{\lambda}}_k\, \sigma\left(b_k + v_k x\right)}_{=:\mathcal{RN}^{*,\tilde{\lambda}}_-} \quad \forall x \in \mathbb{R}, \tag{37}$$

with $w^{*+,\tilde{\lambda}} := \left(w^{*,\tilde{\lambda}}_k\right)_{k \in \mathfrak{K}^+}$ and $w^{*-,\tilde{\lambda}}$ analogously defined as $\tilde{w}^+$ and $\tilde{w}^-$, we have

$$f^{w^{*,\tilde{\lambda}}}(x) = \underbrace{\left(\mathcal{RN}^{*,\tilde{\lambda}}_+ * \kappa_x\right)(x)}_{=:f^{w^{*,\tilde{\lambda}}}_+(x)} + \underbrace{\left(\mathcal{RN}^{*,\tilde{\lambda}}_- * \kappa_x\right)(x)}_{=:f^{w^{*,\tilde{\lambda}}}_-(x)} \qquad \forall x \in \mathbb{R}. \tag{38}$$

*Remark* A.9. For any $x \in \mathbb{R}$ the kernel $\kappa_x$ introduced in Definition A.8 satisfies

(1) $\int_{\mathbb{R}} \kappa_x(s)\, ds = 1$ and
(2) $\lim_{n \to \infty} \kappa_x = \delta_0$, where $\delta_0$ denotes the Dirac distribution at zero.

[59] This "convolution" is a bit special, because the kernel $\kappa_x$ changes with $x \in \mathbb{R}$. Therefore, the notation $\mathcal{RN}^{*,\tilde{\lambda}} * \kappa$ would not be properly defined, but we could define $\mathcal{RN}^{*,\tilde{\lambda}} ** \kappa$ as: $\left(\mathcal{RN}^{*,\tilde{\lambda}}_\omega ** \kappa\right)(x) := \left(\mathcal{RN}^{*,\tilde{\lambda}}_\omega * \kappa_x\right)(x) = \int_{\mathbb{R}} \mathcal{RN}^{*,\tilde{\lambda}}_\omega(x - s)\kappa_x(s) ds \quad \forall \omega \in \Omega\ \forall x \in \mathbb{R}$. Hence, $f^{w^{*,\tilde{\lambda}}} := \mathcal{RN}^{*,\tilde{\lambda}} ** \kappa$ would be another correct way to define $f^{w^{*,\tilde{\lambda}}}$.

*Proof of Theorem 3.8.* The two auxiliary functions $\mathcal{RN}_{\tilde{w}}$ and $f^{w^{*,\tilde{\lambda}}}$ defined above in Definitions A.6 and A.8 will play an important role in this proof.[60]

In the end, we want to show the convergence of $\mathcal{RN}^{*,\tilde{\lambda}}$ to $f^{*,\lambda}_{g,\pm}$. Our strategy to achieve this goal is to prove that both these functions $\mathcal{RN}^{*,\tilde{\lambda}}$ and $f^{*,\lambda}_{g,\pm}$ get closer to the same function $f^{w^{*,\tilde{\lambda}}}$ in the limit $n \to \infty$. The first first convergence will be shown in Lemma A.18. The proof of the second convergence $f^{w^{*,\tilde{\lambda}}} \to f^{*,\lambda}_{g,\pm}$ will need more steps—first we will show the convergence $F^{\lambda,g}_{+-}\left(f^{w^{*,\tilde{\lambda}}}_{+}, f^{w^{*,\tilde{\lambda}}}_{-}\right) \to F^{\lambda,g}_{+-}\left(f^{*,\lambda}_{g,+}, f^{*,\lambda}_{g,-}\right)$ (in multiple steps based on Lemmas A.14 and A.19) to further imply with the help of Lemma A.22 the convergence $f^{w^{*,\tilde{\lambda}}} \to f^{*,\lambda}_{g,\pm}$.

Following this strategy, we prove Theorem 3.8 step by step:

step -0.5 Before starting with the proof, we need the auxiliary Lemmas A.10 and A.11

step 0 Lemma A.12 shows

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left\| \mathcal{RN}_{\tilde{w}} - f^{*,\lambda}_{g,\pm} \right\|_{W^{1,\infty}(K)} = 0.$$

step 1 It is directly clear that

$$F^{\tilde{\lambda}}_n\left(\mathcal{RN}^{*,\tilde{\lambda}}\right) \le F^{\tilde{\lambda}}_n\left(\mathcal{RN}_{\tilde{w}}\right),$$

because of the optimality of $\mathcal{RN}^{*,\tilde{\lambda}}$ (see Definition 3.2).

step 1.5 The auxiliary Lemma A.13 will be needed for step 2 and step 4.

step 2 Lemma A.14 shows

$$\mathbb{P}\text{-}\lim_{n\to\infty} F^{\tilde{\lambda}}_n\left(\mathcal{RN}_{\tilde{w}}\right) = F^{\lambda,g}_{+-}\left(f^{*,\lambda}_{g,+}, f^{*,\lambda}_{g,-}\right).$$

step 2.5 The auxiliary Lemmas A.15–A.17 will be needed for step 3 and step 4.

step 3 Lemma A.18 shows

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left\| \mathcal{RN}^{*,\tilde{\lambda}} - f^{w^{*,\tilde{\lambda}}} \right\|_{W^{1,\infty}(K)} = 0.$$

step 4 Lemma A.19 shows

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left| F^{\tilde{\lambda}}_n\left(\mathcal{RN}^{*,\tilde{\lambda}}\right) - F^{\lambda,g}_{+-}\left(f^{w^{*,\tilde{\lambda}}}_{+}, f^{w^{*,\tilde{\lambda}}}_{-}\right) \right| = 0.$$

step 5 After defining $\tilde{\mathcal{T}}$ (see Definition A.20) it follows directly (with help of Remark A.21) that

$$F^{\lambda,g}_{+-}\left(f^{*,\lambda}_{g,+}, f^{*,\lambda}_{g,-}\right) \le F^{\lambda,g}_{+-}\left(f^{w^{*,\tilde{\lambda}}}_{+}, f^{w^{*,\tilde{\lambda}}}_{-}\right)$$

holds, because of the optimality of $\left(f^{*,\lambda}_{g,+}, f^{*,\lambda}_{g,-}\right) \in \tilde{\mathcal{T}}$.

[60] At the end of the proof, we will see that the functions $\mathcal{RN}^{*,\tilde{\lambda}}$, $f^{w^{*,\tilde{\lambda}}}$ and $\mathcal{RN}_{\tilde{w}}$ will converge to the same function $f^{*,\lambda}_{g,\pm}$ in probability with respect to the Sobolev norm [1] $\|\cdot\|_{W^{1,\infty}(K)}$.

step 6 Combining step 1, step 2, step 4 and step 5 we directly get:[61]

$$F_{+-}^{\lambda,g}\left(f_+^{w^*,\tilde{\lambda}}, f_-^{w^*,\tilde{\lambda}}\right) \overset{\text{step 4}}{\approx} F_n^{\tilde{\lambda}}\left(\mathcal{RN}^{*,\tilde{\lambda}}\right) \overset{\mathbb{P}}{\pm} \epsilon_1 \leq$$
$$\overset{\text{step 1}}{\leq} F_n^{\tilde{\lambda}}\left(\mathcal{RN}_{\tilde{w}}\right) \overset{\mathbb{P}}{\pm} \epsilon_1 \approx$$
$$\overset{\text{step 2}}{\approx} F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right) \overset{\mathbb{P}}{\pm} \epsilon_1 \overset{\mathbb{P}}{\pm} \epsilon_2 \overset{\text{step 5}}{\leq} F_{+-}^{\lambda,g}\left(f_+^{w^*,\tilde{\lambda}}, f_-^{w^*,\tilde{\lambda}}\right) \overset{\mathbb{P}}{\pm} \epsilon_1 \overset{\mathbb{P}}{\pm} \epsilon_2,$$

and thus:

$$F_{+-}^{\lambda,g}\left(f_+^{w^*,\tilde{\lambda}}, f_-^{w^*,\tilde{\lambda}}\right) \overset{\substack{\text{step 4}\\ \text{step 2}\\ \text{step 1}}}{\approx} F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right) \overset{\mathbb{P}}{\pm} \epsilon_3 \qquad \overset{\text{step 5}}{\leq} F_{+-}^{\lambda,g}\left(f_+^{w^*,\tilde{\lambda}}, f_-^{w^*,\tilde{\lambda}}\right) \overset{\mathbb{P}}{\pm} \epsilon_3,$$

which directly implies

$$\mathbb{P}\text{-}\lim_{n\to\infty} F_{+-}^{\lambda,g}\left(f_+^{w^*,\tilde{\lambda}}, f_-^{w^*,\tilde{\lambda}}\right) = F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right). \tag{39}$$

step 7 Lemma A.22 shows

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left\| f^{w^*,\tilde{\lambda}} - f_{g,\pm}^{*,\lambda} \right\|_{W^{1,\infty}(K)} = 0,$$

if one applies it on the result (39) of step 6.

step 8 Combining step 4 and step 7 with the triangle inequality directly results in the statement (16) we want show.

□

**Lemma A.10** (Poincaré typed inequality)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be continuously differentiable with $f' : \mathbb{R} \to \mathbb{R}$ Lebesgue integrable. Then, for any interval $K = [a, b] \subset \mathbb{R}$ such that $f(a) = 0$ there exists a $C_K^\infty \in \mathbb{R}_{>0}$ such that*

$$\|f\|_{W^{1,\infty}(K)} \leq C_K^\infty \|f'\|_{L^\infty(K)}. \tag{40}$$

*Additionally, if $f$ is twice differentiable with $f'' : \mathbb{R} \to \mathbb{R}$ Lebesgue integrable, there exists a $C_K^2 \in \mathbb{R}_{>0}$ such that*

$$\|f\|_{W^{1,\infty}(K)} \leq C_K^2 \|f''\|_{L^2(K)}. \tag{41}$$

*Proof.* By the fundamental theorem of calculus, if $\|f'\|_{L^\infty(K)} < \infty$, then

$$\|f\|_{L^\infty(K)} = \sup_{x\in K} \left| \int_a^x f'(y)\, dy \right| \leq |b-a| \sup_{y\in K} |f'(y)|.$$

[61] We are using the following notation:

$$a_n \approx b_n \overset{\mathbb{P}}{\pm} \epsilon_1 :\Leftrightarrow \forall \epsilon_1 \in \mathbb{R}_{>0} : \forall P_1 \in (0,1) : \exists n_0 \in \mathbb{N} : \forall n \in \mathbb{N}_{>n_0} : \mathbb{P}\left[a_n \in b_n + [-\epsilon_1, \epsilon_1]\right] > \rho,$$

but a complete formalization of this notation would be quite long. This notation needs to be interpreted depending on the context—e.g.:

$$b_n \overset{\mathbb{P}}{\pm} \epsilon_1 \approx b_n \overset{\mathbb{P}}{\pm} \epsilon_1 \overset{\mathbb{P}}{\pm} \epsilon_2 :\Leftrightarrow \forall \epsilon_2 \in \mathbb{R}_{>0} : \forall P_2 \in (0,1) : \exists n_0 \in \mathbb{N} : \forall n \in \mathbb{N}_{>n_0} : \mathbb{P}\left[b_n \in c_n + [-\epsilon_2, \epsilon_2]\right] > P_2,$$

or sometimes it makes sense to replace "$\in$" by "$\subseteq$" in a reasonable way. And in the proofs of some later lemmata $\overset{\mathbb{P}}{\pm}\epsilon_2$ can have the meaning of $\overset{\delta,\epsilon_1\to 0}{\overset{\mathbb{P}}{\pm}}\ \epsilon_2$ instead of $\overset{n\to 0}{\overset{\mathbb{P}}{\pm}}\ \epsilon_2$ depending on the context.

Hence it follows that

$$\|f\|_{W^{1,\infty}(K)} = \max\left\{\|f\|_{L^\infty(K)}, \|f'\|_{L^\infty(K)}\right\} \leq \max\{|b-a|,1\}\,\|f'\|_{L^\infty(K)} = C_K^\infty\,\|f'\|_{L^\infty(K)}\,.$$

Similarly, by the Hölder inequality we have

$$\|f'\|_{L^\infty(K)} = \sup_{x\in K}\left|\int_a^b f''(y)\mathbb{1}_{[a,x]}(y)\,dy\right| \leq \sup_{y\in K}\|f''\|_{L^2(K)}\left\|\mathbb{1}_{[a,y]}\right\|_{L^2(K)} = |b-a|\,\|f''\|_{L^2(K)}\,.$$

Thus, (41) follows from

$$\|f\|_{W^{1,\infty}(K)} \leq C_K^\infty\,\|f'\|_{L^\infty(K)} \leq C_K^\infty|b-a|\,\|f''\|_{L^2(K)} = C_K^2\,\|f''\|_{L^2(K)}\,.$$

□

**Lemma A.11.** *Let $\mathcal{RN}$ be a real-valued randomized shallow network. For $\varphi : \mathbb{R}^2 \to \mathbb{R}$ uniformly continuous such that for all $x \in \operatorname{supp}(g_\xi)$, $\mathbb{E}\left[\varphi(\xi,v)\frac{1}{n g_\xi(\xi)}|\xi = x\right] < \infty$, it then holds that*[62]

$$\mathbb{P}\text{-}\lim_{n\to\infty}\sum_{k\in\mathfrak{K}^+:\xi_k<T}\varphi(\xi_k,v_k)\bar{h}_k = \int_{C^\ell_{g_\xi}\wedge T}^{C^u_{g_\xi}\wedge T}\mathbb{E}\left[\varphi(\xi,v)|\xi=x\right]\,dx \tag{42}$$

*uniformly in $T \in K$.*

*Proof.* For $T \leq C^\ell_{g_\xi}$ both sides of (42) are zero, thus we restrict ourselves to $T > C^\ell_{g_\xi}$. By uniform continuity of $\varphi$ and $\frac{1}{g_\xi}$ in $\xi$, for any $\epsilon > 0$ there exists a $\delta(\epsilon)$ such that for every $|\xi' - \xi| < \delta(\epsilon)$ we have $|\varphi(\xi,v)\frac{1}{g_\xi(\xi)} - \varphi(\xi',v)\frac{1}{g_\xi(\xi')}| < \epsilon$ uniformly in $v$. W.l.o.g. assume $\operatorname{supp}(g_\xi)$ is an interval. Thus, by splitting the interval $[C^\ell_{g_\xi}, C^u_{g_\xi}\wedge T]$ into disjoint strips[63] of

[62]The same statement as (42) is analogously true if one replaces $\mathfrak{K}^+$ by $\mathfrak{K}^-$ of course. Also

$$\mathbb{P}\text{-}\lim_{n\to\infty}\sum_{k:\xi_k<T}\varphi(\xi_k,v_k)\frac{\bar{h}_k}{2} = \int_{C^\ell_{g_\xi}\wedge T}^{C^u_{g_\xi}\wedge T}\mathbb{E}\left[\varphi(\xi,v)|\xi=x\right]\,dx$$

holds analogously. Without Assumption 3b) the statement (42) needed to be reformulated as:

$$\mathbb{P}\text{-}\lim_{n\to\infty}\sum_{k\in\mathfrak{K}^+:\xi_k<T}\varphi(\xi_k,v_k)\bar{h^+}_k = \int_{C^\ell_{g^+_\xi}\wedge T}^{C^u_{g^+_\xi}\wedge T}\mathbb{E}\left[\varphi(\xi,v)|\xi=x, v>0\right]\,dx$$

$$\mathbb{P}\text{-}\lim_{n\to\infty}\sum_{k\in\mathfrak{K}^-:\xi_k<T}\varphi(\xi_k,v_k)\bar{h^-}_k = \int_{C^\ell_{g^-_\xi}\wedge T}^{C^u_{g^-_\xi}\wedge T}\mathbb{E}\left[\varphi(\xi,v)|\xi=x, v<0\right]\,dx$$

[63]Assume $\exists \ell_1,\ell_2 \in \mathbb{Z} : C^\ell_{g_\xi} = \delta\ell_1, C^u_{g_\xi} = \delta\ell_2$ to make the notation simpler. For a cleaner proof, one should choose a suitable partition of $\operatorname{supp}(g_\xi)$.

equal length $\delta \leq \delta(\epsilon)$, we have[64]

$$\begin{aligned}
&\sum_{\substack{k\in\mathfrak{K}^+\\ \xi_k<T}} \varphi(\xi_k, v_k)\bar{h}_k = \\
&\overset{63}{=} \sum_{\substack{\ell\in\mathbb{Z}\\ [\delta\ell,\delta(\ell+1))\subseteq[C^\ell_{g_\xi}, C^u_{g_\xi}\wedge T]}} \left( \sum_{\substack{k\in\mathfrak{K}^+\\ \xi_k\in[\delta\ell,\delta(\ell+1))}} \varphi(\xi_k, v_k)\bar{h}_k \right) \\
&\approx \sum_{\substack{\ell\in\mathbb{Z}\\ [\delta\ell,\delta(\ell+1))\subseteq[C^\ell_{g_\xi}, C^u_{g_\xi}\wedge T]}} \left( \sum_{\substack{k\in\mathfrak{K}^+\\ \xi_k\in[\delta\ell,\delta(\ell+1))}} \left( \varphi(\ell\delta, v_k)\frac{2}{n g_\xi(\ell\delta)} \pm \frac{\epsilon}{n} \right) \frac{|\{m\in\mathfrak{K}^+ : \xi_m \in [\delta\ell,\delta(\ell+1))\}|}{|\{m\in\mathfrak{K}^+ : \xi_m \in [\delta\ell,\delta(\ell+1))\}|} \right) \\
&\approx \sum_{\substack{\ell\in\mathbb{Z}\\ [\delta\ell,\delta(\ell+1))\subseteq[C^\ell_{g_\xi}, C^u_{g_\xi}\wedge T]}} \left( \frac{\sum_{\substack{k\in\mathfrak{K}^+\\ \xi_k\in[\delta\ell,\delta(\ell+1))}} \varphi(\ell\delta, v_k)}{|\{m\in\mathfrak{K}^+ : \xi_m \in [\delta\ell,\delta(\ell+1))\}|} \frac{2|\{m\in\mathfrak{K}^+ : \xi_m \in [\delta\ell,\delta(\ell+1))\}|}{n g_\xi(\ell\delta)} \right) \pm \epsilon.
\end{aligned}$$

The number of nodes within a $\delta$-strip follows a binomial distribution with

$$\mathbb{E}\left[|\{m\in\mathfrak{K}^+ : \xi_m \in [\delta\ell,\delta(\ell+1))\}|\right] = \mathbb{P}\left[v_k > 0\right] n \int_{[\delta\ell,\delta(\ell+1))} g_\xi(x)\,dx \approx \frac{1}{2} n(\delta g_\xi(\ell\delta) \pm \delta\tilde{\epsilon}),$$

for any $\delta \leq \delta(\epsilon, \tilde{\epsilon})$, since $g_\xi$ is uniformly continuous on $\operatorname{supp}(g_\xi)$ by Assumption 2b). For $\delta \leq \delta(\epsilon, \tilde{\epsilon})$ small enough, we have $\mathcal{L}(v_k) \approx \mathcal{L}(v|\xi = \ell\delta)\ \forall k \in \mathfrak{K}^+ : \xi_k \in [\delta\ell, \delta(\ell+1))$ and we may apply the law of large numbers to further obtain

$$\begin{aligned}
\sum_{k\in\mathfrak{K}^+:\xi_k<T} \varphi(\xi_k, v_k)\bar{h}_k &\approx \sum_{\substack{\ell\in\mathbb{Z}\\ [\delta\ell,\delta(\ell+1))\subseteq[C^\ell_{g_\xi}, C^u_{g_\xi}\wedge T]}} \left( \mathbb{E}\left[\varphi(\xi, v)|\xi = \ell\delta\right] \overset{\mathbb{P}}{\pm} \tilde{\tilde{\epsilon}} \right) \delta \left( 1 \pm \frac{\tilde{\epsilon}}{g_\xi(\ell\delta)} \right) \pm \epsilon \\
&\approx \left( \sum_{\substack{\ell\in\mathbb{Z}\\ [\delta\ell,\delta(\ell+1))\subseteq[C^\ell_{g_\xi}, C^u_{g_\xi}\wedge T]}} \left( \mathbb{E}\left[\varphi(\xi, v)|\xi = \ell\delta\right] \delta \right) \overset{\mathbb{P}}{\pm} \tilde{\tilde{\epsilon}} |C^u_{g_\xi} - C^\ell_{g_\xi}| \right) \\
&\quad \cdot \left( 1 \pm \frac{\tilde{\epsilon}}{g_\xi(\ell\delta)} \right) \pm \epsilon
\end{aligned}$$

[64]The notation $\pm\epsilon$ from footnote 61 on page 35 and slight adaptions of it will be used in this proof a lot. The relations of all the epsilons will be explicitly described in (43).

Since $1/g_\xi(\cdot)$ and $\mathbb{E}[\varphi(\xi,v)|\xi=\cdot]$ are bounded on $\operatorname{supp}(g_\xi)$, and $\epsilon, \tilde{\epsilon}$ depend on $\delta$ only, we may for some $\epsilon^*, \rho^* \in (0,1)$ define

$$\epsilon := \frac{\epsilon^*}{3}, \tag{43a}$$

$$\tilde{\epsilon} := \frac{\epsilon^* \min_{x\in\operatorname{supp}(g_\xi)} g_\xi(x)}{3|C^u_{g_\xi} - C^\ell_{g_\xi}| \left(\max_{x\in\operatorname{supp}(g_\xi)} \mathbb{E}[\varphi(\xi,v)|\xi=x] + 1\right)}, \tag{43b}$$

$$\tilde{\tilde{\epsilon}} := \frac{\epsilon^*}{3|C^u_{g_\xi} - C^\ell_{g_\xi}|}, \tag{43c}$$

$$\tilde{\tilde{\rho}} := (\rho^*)^{\frac{\delta}{|C^u_{g_\xi} - C^\ell_{g_\xi}|}}, \tag{43d}$$

$$n_0^* := \tilde{\tilde{n}}_0(\tilde{\tilde{\epsilon}}, \tilde{\tilde{\rho}}). \tag{43e}$$

With the above, it follows that for any $\epsilon^*, \rho^* \in (0,1)$ there exists a $n_0^*$ such that $\forall n > n_0^*$ :

$$\mathbb{P}\left[\left|\sum_{k\in\mathfrak{K}^+:\xi_k<T} \varphi(\xi_k, v_k)\bar{h}_k - \sum_{\substack{\ell\in\mathbb{Z} \\ [\delta\ell,\delta(\ell+1))\subseteq[C^\ell_{g_\xi}, C^u_{g_\xi}\wedge T]}} \mathbb{E}[\varphi(\xi,v)|\xi=\ell\delta]\,\delta\right| \le \epsilon^*\right] > \rho^*.$$

For $\delta$ small enough, the above Riemann sum converges uniformly in $T$ to yield the desired result. □

**Lemma A.12** (step 0)**.** *For any choice of the penalty parameter $\lambda > 0$ and $K \subset \mathbb{R}$ compact, the spline approximating RSN $\mathcal{RN}_{\tilde{w}}$ converges to the adapted regression spline $f^{*,\lambda}_{g,\pm}$ in probability w.r.t. $\|\cdot\|_{W^{1,\infty}(K)}$ with increasing number of nodes, i.e. for any $\lambda > 0$ and $K \subset \mathbb{R}$ we have*

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left\|\mathcal{RN}_{\tilde{w}} - f^{*,\lambda}_{g,\pm}\right\|_{W^{1,\infty}(K)} = 0.^{65}$$

*Proof.* Let $\lambda > 0$ and $K \subset \mathbb{R}$ compact with $[C^\ell_g, C^u_g] \subset K$. Directly from the definition (35) of $\mathcal{RN}^+_{\tilde{w}^+}$ and $\mathcal{RN}^+_{\tilde{w}^+}$ and the Definitions 3.5 and A.3 of $f^{*,\lambda}_{g,\pm}$, it follows that it is sufficient to show:

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left\|\mathcal{RN}^+_{\tilde{w}^+} - f^{*,\lambda}_{g,+}\right\|_{W^{1,\infty}(K)} = 0 \quad \text{and} \tag{44a}$$

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left\|\mathcal{RN}^-_{\tilde{w}^-} - f^{*,\lambda}_{g,-}\right\|_{W^{1,\infty}(K)} = 0\,. \tag{44b}$$

W.l.o.g. we restrict ourselves to proving (44a), as the latter limit follows analogously. By Lemma A.10 it suffices to show that

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left\|\mathcal{RN}^{+\,\prime}_{\tilde{w}^+} - f^{*,\lambda\,\prime}_{g,+}\right\|_{L^\infty(K)} = 0. \tag{45}$$

[65] Using the definition of the $\mathbb{P}$-lim, we get:

$$\forall\epsilon\in\mathbb{R}_{>0} : \forall\rho\in(0,1) : \exists n_0\in\mathbb{N} : \forall n \ge n_0 : \mathbb{P}\left[\left\|\mathcal{RN}_{\tilde{w}} - f^{*,\lambda}_{g,\pm}\right\|_{W^{1,\infty}(K)} < \epsilon\right] > \rho.$$

Since for any $x \in K$

$$\mathcal{RN}^{+}_{\tilde{w}^+}{}'(x) = \sum_{k \in \mathfrak{K}^+} \tilde{w}_k v_k = \sum_{k \in \mathfrak{K}^+} f^{*,\lambda}_{g,+}{}''(\xi_k) \frac{v_k^2}{\mathbb{E}\left[v^2 | \xi = \xi_k\right]} \bar{h}_k,$$

we may employ Lemma A.11[66] with $\varphi(z, y) = f^{*,\lambda}_{g,+}{}''(z) \frac{y^2}{\mathbb{E}[v^2|\xi=z]}$ to obtain

$$\mathbb{P}\text{-}\lim_{n\to\infty} \mathcal{RN}^{+}_{\tilde{w}^+}{}'(x) = \int_{C^{\ell}_{g_\xi} \wedge x}^{C^{u}_{g_\xi} \wedge x} \mathbb{E}\left[ f^{*,\lambda}_{g,+}{}''(\xi) \frac{v^2}{\mathbb{E}\left[v^2 | \xi = z\right]} | \xi = z \right] dz = \int_{C^{\ell}_{g_\xi} \wedge x}^{C^{u}_{g_\xi} \wedge x} f^{*,\lambda}_{g,+}{}''(z)\, dz$$

uniformly in $x \in K$. Employing the fundamental theorem of calculus we further obtain

$$\mathbb{P}\text{-}\lim_{n\to\infty} \mathcal{RN}^{+}_{\tilde{w}^+}{}'(x) = f^{*,\lambda}_{g,+}{}'(C^{u}_{g_\xi} \wedge x) - f^{*,\lambda}_{g,+}{}'(C^{\ell}_{g_\xi} \wedge x) \qquad \forall x \in \mathbb{R}.$$

By Remark 3.6, we have that $f^{*,\lambda}_{g,+}{}'(C^{\ell}_{g_\xi} \wedge x) = 0$ for any $x \in \mathbb{R}$. Since by the same remark, $f^{*,\lambda}_{g,+}{}'$ is constant on $[C^{u}_{g_\xi}, \infty)$, we finally obtain

$$\mathbb{P}\text{-}\lim_{n\to\infty} \mathcal{RN}^{+}_{\tilde{w}^+}{}'(x) = f^{*,\lambda}_{g,+}{}'(x) \qquad \text{uniformly in } x \in K.$$

Hence (45) follows.

□

**Lemma A.13** ($L(f_n) \to L(f)$)**.** *For any data $(x_i^{train}, y_i^{train}) \in \mathbb{R}^2$, $i \in \{1, \dots, N\}$, let $(f_n)_{n\in\mathbb{N}}$ be a sequence of functions that converges point-wise[67] in probability to a function $f : \mathbb{R} \to \mathbb{R}$, then the training loss $L$ (c.p. Definition A.1) of $f_n$ converges in probability to $L(f)$ as $n$ tends to infinity, i.e.*

$$\mathbb{P}\text{-}\lim_{n\to\infty} L(f_n) = L(f). \tag{46}$$

*Proof.* By continuity, the result follows directly:

$$\begin{aligned}
\mathbb{P}\text{-}\lim_{n\to\infty} L\left(f_n\right) &= \mathbb{P}\text{-}\lim_{n\to\infty} \sum_{i=1}^{N} l_i\left(f_n(x_i^{\text{train}})\right) \\
&= \sum_{i=1}^{N} l_i\left(\mathbb{P}\text{-}\lim_{n\to\infty} f_n(x_i^{\text{train}})\right) \\
&= \sum_{i=1}^{N} l_i\left(f(x_i^{\text{train}})\right) = L\left(f\right).
\end{aligned}$$

□

[66] Note that $\varphi(x, y)$ is uniformly continuous on $\text{supp}(g_\xi)$ since, by definition, $f^{*,\lambda}_{g,+} \in \mathcal{C}^2(\mathbb{R})$ and $\text{supp}(g_\xi)$ is compact by Assumption 2.

[67] If $\mathbb{P}\text{-}\lim_{n\to\infty} \|f_n - f\|_{W^{1,\infty}(K)} = 0$, then $f_n$ converges point-wise in probability to $f$ (by using Sobolev's embedding theorem [1] or by assuming $f_n$ and $f$ to be continuous). Hence Lemma A.13 can be used together with Lemma A.12 to show $\mathbb{P}\text{-}\lim_{n\to\infty} L(\mathcal{RN}_{\tilde{w}}) = L(f^{*,\lambda}_{g,\pm})$ or together with Lemma A.18 to show $\mathbb{P}\text{-}\lim_{n\to\infty} L\left(\mathcal{RN}^{*,\tilde{\lambda}}\right) = L\left(f^{w^{*,\tilde{\lambda}}}\right)$.

**Lemma A.14** (step 2). *For any $\lambda > 0$ and data $(x_i^{train}, y_i^{train}) \in \mathbb{R}^2$, $i \in \{1, \dots, N\}$, we have*

$$\mathbb{P}\text{-}\lim_{n\to\infty} F_n^{\tilde{\lambda}}\left(\mathcal{RN}_{\tilde{w}}\right) = F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right), \tag{47}$$

*with $\tilde{\lambda}$ and $g$ as defined in Theorem 3.8.*

*Proof.* We start by showing

$$\mathbb{P}\text{-}\lim_{n\to\infty} \tilde{\lambda}\left\|\tilde{w}\right\|_2^2 = \lambda 2 g(0)\left(\int_{\mathrm{supp}(g)} \frac{\left(f_{g,+}^{*,\lambda}{}''(x)\right)^2}{g(x)} dx + \int_{\mathrm{supp}(g)} \frac{\left(f_{g,-}^{*,\lambda}{}''(x)\right)^2}{g(x)} dx\right). \tag{48}$$

Since $\|\tilde{w}\|_2^2 = \|\tilde{w}^+\|_2^2 + \|\tilde{w}^-\|_2^2$, we restrict ourselves to proving

$$\mathbb{P}\text{-}\lim_{n\to\infty} \tilde{\lambda}\left\|\tilde{w}^+\right\|_2^2 = \lambda 2 g(0) \int_{\mathrm{supp}(g_\xi)} \frac{\left(f_{g,+}^{*,\lambda}{}''(x)\right)^2}{g(x)} dx. \tag{49}$$

With the definitions of $\tilde{w}^+$, $\tilde{\lambda}$ and $\bar{h}$ we have

$$\begin{aligned}
\tilde{\lambda}\left\|\tilde{w}^+\right\|_2^2 &= \tilde{\lambda}\sum_{k\in\mathfrak{K}^+}\left(f_{g,+}^{*,\lambda}{}''(\xi_k)\frac{\bar{h}_k v_k}{\mathbb{E}\left[v^2|\xi=\xi_k\right]}\right)^2 \\
&= \tilde{\lambda}\sum_{k\in\mathfrak{K}^+}\left(\left(f_{g,+}^{*,\lambda}{}''\right)^2(\xi_k)\frac{\bar{h}_k v_k^2}{\mathbb{E}\left[v^2|\xi=\xi_k\right]^2}\right)\bar{h}_k \\
&= \lambda 2 g(0)\sum_{k\in\mathfrak{K}^+}\left(\left(f_{g,+}^{*,\lambda}{}''\right)^2(\xi_k)\frac{2v_k^2}{g_\xi(\xi_k)\mathbb{E}\left[v^2|\xi=\xi_k\right]^2}\right)\bar{h}_k.
\end{aligned}$$

An application of Lemma A.11 with $\varphi(x,y) = \left(f_{g,+}^{*,\lambda}{}''\right)^2(x)\frac{2y^2}{g_\xi(x)\mathbb{E}[v^2|\xi=y]^2}$ further yields (49) via

$$\begin{aligned}
\mathbb{P}\text{-}\lim_{n\to\infty} \tilde{\lambda}\left\|\tilde{w}^+\right\|_2^2 &= \lambda 2 g(0)\int_{\mathrm{supp}(g_\xi)} \mathbb{E}\left[\left(f_{g,+}^{*,\lambda}{}''\right)^2(\xi)\frac{2v^2}{g_\xi(\xi)\mathbb{E}\left[v^2|\xi=x\right]^2}\middle|\xi=x\right] dx \\
&= \lambda 2 g(0)\int_{\mathrm{supp}(g_\xi)} \frac{2\left(f_{g,+}^{*,\lambda}{}''\right)^2(x)}{g_\xi(x)\mathbb{E}\left[v^2|\xi=x\right]} dx \\
&= \lambda 2 g(0)\int_{\mathrm{supp}(g_\xi)} \frac{\left(f_{g,+}^{*,\lambda}{}''(x)\right)^2}{g(x)} dx.
\end{aligned}$$

Thus we have proved the convergence of the penalization terms (48). Together with Lemmas A.12 and A.13, (47) follows.

□

Before we prove the very important Lemma A.16, we need an auxiliary Lemma A.15 that would be quite easy to prove in the case of square loss (i.e. $l_i(y) := (y - y_i^{\text{train}})^2$), but gets a bit more envolved in the case of more general forms of training losses $l_i$.

**Lemma A.15** ($l_i'$ -bound)**.** *Using the notation of Definition 3.2, there exists an upper bound:* $\exists C_{l'} \in \mathbb{R}_{>0} : \forall n \in \mathbb{N} : \forall \omega \in \Omega : \forall i \in \{1, \dots, N\} :$

$$\left| l_i' \left( \mathcal{RN}^{*,\tilde{\lambda}}(x_i^{train}) \right) \right| \leq C_{l'} \tag{50}$$

*Proof.* The optimality (11) of $\mathcal{RN}^{*,\tilde{\lambda}}$ implies

$$L\left(\mathcal{RN}^{*,\tilde{\lambda}}\right) + \underbrace{\tilde{\lambda} \left\| w^{*,\tilde{\lambda}} \right\|_2^2}_{\geq 0} \overset{\text{optimality}}{\leq} L\left(\underbrace{0}_{\mathcal{RN}_0}\right) + \| \underbrace{0}_{\in \mathbb{R}^n} \|_2^2, \tag{51}$$

which further implies, that $\forall i \in \{1, \dots, N\} :$

$$l_i\left(\mathcal{RN}^{*,\tilde{\lambda}}(x_i^{\text{train}})\right) \overset{l_\iota \geq 0}{\leq} \sum_{\iota=1}^{N} l_\iota\left(\mathcal{RN}^{*,\tilde{\lambda}}(x_\iota^{\text{train}})\right) \overset{\text{Def. A.1}}{=} L\left(\mathcal{RN}^{*,\tilde{\lambda}}\right) \overset{(51)}{\leq} L(0). \tag{52}$$

In other words, $\forall i \in \{1, \dots, N\}$ the evaluated ridge network

$$\mathcal{RN}^{*,\tilde{\lambda}}(x_i^{\text{train}}) \overset{(52)}{\in} l_i^{-1}\left((-\infty, L(0)]\right) := \{\, y \in \mathbb{R} \mid l_i(y) \leq L(0) \,\} \tag{53}$$

lies in a certain sublevel set of $l_i$.

This implies that $\forall i \in \{1, \dots, N\}$:

$$\left| l_i'\left(\mathcal{RN}^{*,\tilde{\lambda}}(x_i^{\text{train}})\right) \right| \overset{(53)}{\leq} \sup_{y \in l_i^{-1}((-\infty, L(0)])} \left| l_i'(y) \right| =: c_i \in \bar{\mathbb{R}}. \tag{54}$$

So we want to show that the right-hand side $c_i$ of (54) is finite. For this, we need a better understanding of the sublevel set $l_i^{-1}\left((-\infty, L(0)]\right)$.

The sublevel sets of convex functions are convex ($l_i$ is convex by Assumption 4b)). Convex subsets of $\mathbb{R}$ are always intervals. Since $l_i \in \mathcal{C}^1$ is continuous by Assumption 4c), the preimage of the closed set $(-\infty, L(0)]$ is closed.

Hence, $l_i^{-1}\left((-\infty, L(0)]\right)$ is a closed interval. There are only four types of closed intervals: $[\alpha, \beta]$, $[\alpha, \infty)$, $(-\infty, \beta]$ and $(-\infty, \infty)$, where $\alpha, \beta \in \mathbb{R}$. As the domain of $l_i$ is unbounded and as $l_i$ is continuous, we know that $\alpha, \beta \in l_i^{-1}(L(0))$.

Consider these four cases for each $i \in \{1, \dots, N\}$ separately:

case 1: $l_i^{-1}\left((-\infty, L(0)]\right) = [\alpha, \beta]$ is compact:
Since $l_i$ is convex, $l_i'$ is monotonically increasing. Hence, the minimum of $l_i'$ must be attained at the left boundary $\alpha$ and the maximum at the right border $\beta$. So, we can bound

$$c_i \overset{(54)}{:=} \sup_{y \in l_i^{-1}((-\infty, L(0)])} \left| l_i'(y) \right| = \max\left\{ \left| l_i'(\alpha) \right|, \left| l_i'(\beta) \right| \right\} \in \mathbb{R} \tag{55}$$

as a finite number (i.e. the maximum of two finite numbers).

case 2: $l_i^{-1}\left((-\infty, L(0)]\right) = [\alpha, \infty)$ is not compact:
This case allows to imply that

$$l_i'(y) \leq 0 \quad \forall y \in \mathbb{R} \supseteq l_i^{-1}\left((-\infty, L(0)]\right), \tag{56}$$

because of the following contraposition:
Assume $\exists y^+ \in \mathbb{R} : {l_i}'(y^+) > 0$ then, $\forall y \in [y^+, \infty) : {l_i}'(y) \geq {l_i}'(y^+)$, because of monotonicity. Then $\forall y \in [y^+, \infty) : l_i(y) \geq l_i(y^+) + (y - y^+){l_i}'(y^+)$, and further

$$\left( y^+ + \frac{L(0) - l_i(y^+)}{{l_i}'(y^+)}, \infty \right) \cap {l_i}^{-1}\left((-\infty, L(0)]\right) = \emptyset,$$

which would contradict the assumption of case 2. This contraposition has proven ineq. (56).
With the help of ineq. (56) we can bound

$$c_i \overset{(54)}{:=} \sup_{y \in {l_i}^{-1}((-\infty, L(0)])} \left| {l_i}'(y) \right| \overset{(56)}{=} \left| \inf_{y \in {l_i}^{-1}((-\infty, L(0)])} {l_i}'(y) \right| \overset{\text{monotonicity}}{=} \left| {l_i}'(\alpha) \right| \in \mathbb{R}. \tag{57}$$

case 3: ${l_i}^{-1}\left((-\infty, L(0)]\right) = (-\infty, \beta]$ is not compact:
Analogously to (56) we get

$$\forall y \in \mathbb{R} \supseteq {l_i}^{-1}\left((-\infty, L(0)]\right), \quad {l_i}'(y) \geq 0 \tag{56\_}$$

which implies analogously to (57) that we can bound

$$c_i \overset{(54)}{:=} \sup_{y \in {l_i}^{-1}((-\infty, L(0)])} \left| {l_i}'(y) \right| \overset{(56\_)}{=} \left| \sup_{y \in {l_i}^{-1}((-\infty, L(0)])} {l_i}'(y) \right| \overset{\text{monotonicity}}{=} \left| {l_i}'(\beta) \right| \in \mathbb{R}. \tag{57\_}$$

case 4: ${l_i}^{-1}\left((-\infty, L(0)]\right) = (-\infty, \infty)$ is not compact:
Analogously to (56) and (56_) we get

$$\forall y \in \mathbb{R} \supseteq {l_i}^{-1}\left((-\infty, L(0)]\right), \quad {l_i}'(y) = 0 \tag{$56_0$}$$

which directly implies that we can bound

$$c_i \overset{(54)}{:=} \sup_{y \in {l_i}^{-1}((-\infty, L(0)])} \left| {l_i}'(y) \right| \overset{(56_0)}{=} 0 \in \mathbb{R}. \tag{$57_0$}$$

Since this case analysis showed that in each case $c_i \in \mathbb{R}$ is finite, we can use (54) to conclude $\forall n \in \mathbb{N} : \forall \omega \in \Omega : \forall i \in \{1, \dots, N\} :$

$$\left| {l_i}' \left( \mathcal{RN}^{*,\tilde{\lambda}}(x_i^{\text{train}}) \right) \right| \overset{(54)}{\leq} c_i \leq \max_{i \in \{1,\dots,N\}} c_i =: C_{l'} \overset{\text{cases 1–4}}{<} \infty. \tag{58}$$

An equivalent more explicit definition would be:

$$C_{l'} := \max\left( \{0\} \cup \left\{ {l_i}'(y_i) \;\middle|\; i \in \{1, \dots, N\}, y_i \in {l_i}^{-1}(L(0)) \right\} \right). \tag{59}$$

□

**Lemma A.16.** *Using the notation of Definitions 2.6 and 3.2, the following statement holds:*

$$\forall \epsilon \in \mathbb{R}_{>0} : \exists \delta \in \mathbb{R}_{>0} : \forall n \in \mathbb{N} : \forall \omega \in \Omega : \forall \grave{k}, \acute{k} \in \{1, \dots, n\} :$$

$$\left( \left( \left| \underbrace{\xi_{\grave{k}}(\omega) - \xi_{\acute{k}}(\omega)}_{=:\Delta\xi(\omega)} \right| < \delta \;\wedge\; \operatorname{sgn}\left(v_{\grave{k}}(\omega)\right) = \operatorname{sgn}\left(v_{\acute{k}}(\omega)\right) \right) \Rightarrow \left| \frac{w_{\grave{k}}^{*,\tilde{\lambda}}(\omega)}{v_{\grave{k}}(\omega)} - \frac{w_{\acute{k}}^{*,\tilde{\lambda}}(\omega)}{v_{\acute{k}}(\omega)} \right| < \frac{\epsilon}{n} \right),$$

*if we assume that $v_k$ is never zero.*

*Proof.* We will prove the even stronger statement:

$$\left|\frac{w_{\grave{k}}^{*,\tilde{\lambda}}}{v_{\grave{k}}}-\frac{w_{\acute{k}}^{*,\tilde{\lambda}}}{v_{\acute{k}}}\right| \overset{1.}{\underset{\substack{\text{conditioned on}\\ \operatorname{sgn}(v_{\grave{k}})=\operatorname{sgn}(v_{\acute{k}})}}{\leq}} \frac{|\Delta\xi|}{\tilde{\lambda}}\sum_{i=1}^{N}\left|l_i'\left(\mathcal{RN}^{*,\tilde{\lambda}}(x_i^{\text{train}})\right)\right| \overset{2.}{\leq} \tag{60a}$$

$$\overset{2.}{\leq} \frac{|\Delta\xi|}{\tilde{\lambda}} N C_{l'} \tag{60b}$$

because with the help of inequality (60), $\delta := \frac{\epsilon\lambda 2g(0)}{NC_{l'}}$ would be a valid choice of $\delta$ in the statement of Lemma A.16.

1. Proof of (60a): First we define the disturbed weight vector $w^{\Delta s}$ such that

$$w_k^{\Delta s} := w_k^{*,\tilde{\lambda}} + \begin{cases} +\frac{\Delta s}{|v_{\grave{k}}|} & k=\grave{k} \\ -\frac{\Delta s}{|v_{\acute{k}}|} & k=\acute{k} \\ 0 & \text{else-wise} \end{cases}$$

by shifting a little bit of the distributional second derivative $\Delta s$ from the $\acute{k}$th kink to the $\grave{k}$th kink. By a case analysis (or by drawing a sketch) one can easily show that conditioned on $\operatorname{sgn}\left(v_{\grave{k}}\right)=\operatorname{sgn}\left(v_{\acute{k}}\right)$:

$$\forall x\in\mathbb{R}: \left|\mathcal{RN}^{*,\tilde{\lambda}}(x)-\left(\mathcal{RN}_{w^{\Delta s}}(x)\right)\right| \leq \Delta\xi\Delta s. \tag{61}$$

As $\mathcal{RN}^{*,\tilde{\lambda}}$ is optimal the derivative

$$0=\left.\frac{dF_n^{\tilde{\lambda}}\left(\mathcal{RN}_{w^{\Delta s}}\right)}{d\Delta s}\right|_{\Delta s=0} = \tilde{\lambda}2\left(\frac{w_{\grave{k}}^{*,\tilde{\lambda}}}{v_{\grave{k}}}-\frac{w_{\acute{k}}^{*,\tilde{\lambda}}}{v_{\acute{k}}}\right)+\left.\frac{dL\left(\mathcal{RN}_{w^{\Delta s}}\right)}{d\Delta s}\right|_{\Delta s=0} \tag{62}$$

has to be zero. Transforming this equation and taking absolute values on both sides gives:

$$\left|\tilde{\lambda}2\left(\frac{w_{\grave{k}}^{*,\tilde{\lambda}}}{v_{\grave{k}}}-\frac{w_{\acute{k}}^{*,\tilde{\lambda}}}{v_{\acute{k}}}\right)\right| \overset{(62)}{=} \left|\left.\frac{dL\left(\mathcal{RN}_{w^{\Delta s}}\right)}{d\Delta s}\right|_{\Delta s=0}\right| \overset{(61)}{\leq} 2\sum_{i=1}^{N}\left|l_i'\left(\mathcal{RN}^{*,\tilde{\lambda}}(x_i^{\text{train}})\right)\Delta\xi\right|. \tag{63}$$

Dividing both sides by $2\tilde{\lambda}$ results in (60a).

2. (60a)≤(60b) holds because of Lemma A.15

□

**Lemma A.17** ($\frac{w^{*,\tilde{\lambda}}}{v}\approx\mathcal{O}(\frac{1}{n})$)**.** *For any $\lambda>0$ and data $(x_i^{train},y_i^{train})\in\mathbb{R}^2$, $i\in\{1,\dots,N\}$, we have*

$$\max_{k\in\{1,\dots,n\}}\frac{w_k^{*,\tilde{\lambda}}}{v_k} = \mathbb{P}\text{-}\mathcal{O}_{n\to\infty}\left(\frac{1}{n}\right).^{68} \tag{64}$$

[68] Using the definition of $\mathbb{P}$-$\mathcal{O}$, eq. (64) reads as:

$$\forall\rho\in(0,1):\exists C\in\mathbb{R}_{>0}:\exists n_0\in\mathbb{N}:\forall n>n_0:\mathbb{P}\left[\max_{k\in\{1,\dots,n\}}<C\frac{1}{n}\right]>\rho.$$

*Proof.* Let $k^* \in \arg\max_{k\in\{1,\dots,n\}} \frac{w_k^{*,\tilde\lambda}}{v_k}$ and thus $\frac{w_{k^*}^{*,\tilde\lambda}}{v_{k^*}} = \max_{k\in\{1,\dots,n\}} \frac{w_k^{*,\tilde\lambda}}{v_k}$. W.l.o.g. assume $k^* \in \mathfrak{K}^+$.

$$\frac{F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right)}{\tilde\lambda} \overset{\text{Lemma A.14}}{\overset{\mathbb{P}}{\geq}} \frac{1}{2\tilde\lambda} F_n^{\tilde\lambda}\left(\mathcal{RN}^{*,\tilde\lambda}\right) \tag{65a}$$

$$\geq \frac{1}{2} \sum_{k\in\mathfrak{K}^+ : \xi_k \in (\xi_{k^*}, \xi_{k^*}+\delta)} {w_k^{*,\tilde\lambda}}^2 \tag{65b}$$

$$= \frac{1}{2} \sum_{k\in\mathfrak{K}^+ : \xi_k \in (\xi_{k^*}, \xi_{k^*}+\delta)} \frac{{w_k^{*,\tilde\lambda}}^2}{v_k^2} v_k^2 \tag{65c}$$

$$\overset{\text{Lemma A.16}}{\geq} \frac{1}{4} \frac{{w_{k^*}^{*,\tilde\lambda}}^2}{v_{k^*}^2} \sum_{k\in\mathfrak{K}^+ : \xi_k \in (\xi_{k^*}, \xi_{k^*}+\delta)} v_k^2 \tag{65d}$$

$$\overset{\mathbb{P}}{\geq} \frac{1}{8} \frac{{w_{k^*}^{*,\tilde\lambda}}^2}{v_{k^*}^2} \frac{n\delta g_\xi(\xi_{k^*})}{2} \mathbb{E}\left[v_k^2 \middle| \xi_k = \xi_{k^*}\right]. \tag{65e}$$

Transforming inequality (65) and using the definition $\tilde\lambda := \lambda n 2g(0)$ gives:

$$\frac{{w_{k^*}^{*,\tilde\lambda}}^2}{v_{k^*}^2} \overset{\mathbb{P}}{\leq} \frac{16}{n^2} \frac{F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right)}{\delta g_\xi(\xi_{k^*})\lambda 2g(0)}. \tag{66}$$

Taking the square root of both sides and bounding $g_\xi$ with its minimum[69], we get:

$$\frac{w_{k^*}^{*,\tilde\lambda}}{v_{k^*}} \overset{\mathbb{P}}{\leq} \frac{4}{n} \left( \frac{F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right)}{\delta \min_{x\in\mathrm{supp}(g)} g_\xi(x)\lambda 2g(0)} \right)^{\frac{1}{2}}. \tag{67}$$

This proves statement (64) by choosing $C$ from footnote 68 as:

$$C := 4 \left( \frac{F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right)}{\delta \min_{x\in\mathrm{supp}(g)} g_\xi(x)\lambda 2g(0)} \right)^{\frac{1}{2}}. \tag{68}$$

□

**Lemma A.18** (step 3)**.** *For any $\lambda > 0$ and data $(x_i^{train}, y_i^{train}) \in \mathbb{R}^2$, $i \in \{1,\dots,N\}$, we have*

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left\| \mathcal{RN}^{*,\tilde\lambda} - f^{w^{*,\tilde\lambda}} \right\|_{W^{1,\infty}(K)} = 0, \tag{69}$$

*with $\tilde\lambda$ as defined in Theorem 3.8.*

*Proof.* By Lemma A.10 (as $\mathcal{RN}^{*,\tilde\lambda}, f^{w^{*,\tilde\lambda}}$ are zero outside of $\mathrm{supp}(g) + \mathrm{supp}(\kappa_x)$ like described in Remark 3.6), we only need to show that for all $\epsilon > 0$:

$$\lim_{n\to\infty} \mathbb{P}\left[ \left\| \mathcal{RN}^{*,\tilde\lambda'} - f^{w^{*,\tilde\lambda'}} \right\|_{L^\infty(K)} < \epsilon \right] = 1.$$

[69] Assumption 2a) and c) guarantee that $\min_{x\in\mathrm{supp}(g)} g_\xi(x) > 0$.

W.l.o.g. it is sufficient to prove:

$$\lim_{n\to\infty} \mathbb{P}\left[\left\|\mathcal{RN}_+^{*,\tilde{\lambda}'} - f_+^{w^{*,\tilde{\lambda}'}}\right\|_{L^\infty(K)} < \epsilon\right] = 1.$$

For every $x \in K$ and $\omega \in \Omega$, using the Definition A.8 of $f_+^{w^{*,\tilde{\lambda}}}$ we have

$$\begin{aligned}
\mathcal{RN}_+^{*,\tilde{\lambda}'}(x) - f_+^{w^{*,\tilde{\lambda}'}}(x) &= \mathcal{RN}_+^{*,\tilde{\lambda}'}(x) - \left(\mathcal{RN}_+^{*,\tilde{\lambda}'} * \kappa_x\right)(x) \\
&= \int_{\mathbb{R}} \mathcal{RN}_+^{*,\tilde{\lambda}'}(x)\kappa_x(t)\,dt - \int_{\mathbb{R}} \mathcal{RN}_+^{*,\tilde{\lambda}'}(x-t)\kappa_x(t)\,dt \\
&= \int_{\mathbb{R}} \left(\mathcal{RN}_+^{*,\tilde{\lambda}'}(x) - \mathcal{RN}_+^{*,\tilde{\lambda}'}(x-t)\right)\kappa_x(t)\,dt.
\end{aligned}$$

Using the definition of $\mathcal{RN}_+^{*,\tilde{\lambda}}$ we get:

$$\mathcal{RN}_+^{*,\tilde{\lambda}'}(x) = \sum_{k\in\mathfrak{K}^+:\xi_k<x} w_k^{*,\tilde{\lambda}} v_k \tag{70}$$

and hence with $r_n := \frac{1}{2\sqrt{n}g_\xi(x)}$ we can get after some algebraic calculations:

$$\begin{aligned}
\mathcal{RN}_+^{*,\tilde{\lambda}'}(x) - f_+^{w^{*,\tilde{\lambda}'}}(x) = &\sum_{k\in\mathfrak{K}^+:x-r_n<\xi_k<x} w_k^{*,\tilde{\lambda}} v_k \int_{x-r_n}^{\xi_k} \kappa_x(s-x)ds \\
&- \sum_{k\in\mathfrak{K}^+:x<\xi_k<x+r_n} w_k^{*,\tilde{\lambda}} v_k \int_{\xi_k}^{x+r_n} \kappa_x(s-x)ds = \\
= &\sum_{k\in\mathfrak{K}^+:x-r_n<\xi_k<x} \frac{w_k^{*,\tilde{\lambda}}}{v_k} v_k^2 \int_{x-r_n}^{\xi_k} \kappa_x(s-x)ds \\
&- \sum_{k\in\mathfrak{K}^+:x<\xi_k<x+r_n} \frac{w_k^{*,\tilde{\lambda}}}{v_k} v_k^2 \int_{\xi_k}^{x+r_n} \kappa_x(s-x)ds
\end{aligned}$$

Thus, we can use the triangle inequality[70] and the properties of the kernel $\kappa_x$ to get:

$$\left|\mathcal{RN}_+^{*,\tilde{\lambda}'}(x) - f_+^{w^{*,\tilde{\lambda}'}}(x)\right| \leq \frac{1}{2} \sum_{k\in\mathfrak{K}^+:x-r_n<\xi_k<x+r_n} \left|\frac{w_k^{*,\tilde{\lambda}}}{v_k} v_k^2\right| \tag{71a}$$

$$\leq \frac{1}{2} \max_{k\in\mathfrak{K}^+} \left|\frac{w_k^{*,\tilde{\lambda}}}{v_k}\right| \sum_{k\in\mathfrak{K}^+:x-r_n<\xi_k<x+r_n} v_k^2 \tag{71b}$$

$$\overset{\text{Lemma A.17}}{\leq} \mathbb{P}\text{-}\mathcal{O}_{n\to\infty}\left(\frac{1}{n}\right) \mathbb{P}\text{-}\mathcal{O}_{n\to\infty}\left(\sqrt{n}\right) = \mathbb{P}\text{-}\mathcal{O}_{n\to\infty}\left(\frac{1}{\sqrt{n}}\right) \tag{71c}$$

[70] Actually, one could use a much tighter bound the triangle inequality used in inequality (71a), because in asymptotic expectation the positive and negative summands would cancel each other instead of adding up.

uniformly in $x$ on $\operatorname{supp}(g_\xi)$ and thus on $K$ (since outside of $\operatorname{supp}(g_\xi) + (-r_n, r_n)$ both functions and there derivatives are zero). □

**Lemma A.19** (step 4)**.** *For any $\lambda > 0$ and data $(x_i^{train}, y_i^{train}) \in \mathbb{R}^2$, $i \in \{1, \dots, N\}$, we have*

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left| F_n^{\tilde\lambda}\left(\mathcal{RNN}^{*,\tilde\lambda}\right) - F_{+-}^{\lambda,g}\left(f_+^{w^{*,\tilde\lambda}}, f_-^{w^{*,\tilde\lambda}}\right)\right| = 0, \tag{72}$$

*with $\tilde\lambda$ as defined in Theorem 3.8.*

*Proof.* Lemmas A.13 and A.18 combined show that

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left| L\left(\mathcal{RNN}^{*,\tilde\lambda}\right) - L\left(f_+^{w^{*,\tilde\lambda}}, f_-^{w^{*,\tilde\lambda}}\right)\right| = 0.$$

So it is sufficient to show:

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left| \tilde\lambda \left\|w^{*,\tilde\lambda}\right\|_2^2 - \lambda 2 g(0) \left( \int_{\operatorname{supp}(g)} \frac{\left(f_+^{w^{*,\tilde\lambda}}{}''(x)\right)^2}{g(x)} dx + \int_{\operatorname{supp}(g)} \frac{\left(f_-^{w^{*,\tilde\lambda}}{}''(x)\right)^2}{g(x)} dx \right) \right| = 0. \tag{73}$$

Since $\left\|w^{*,\tilde\lambda}\right\|_2^2 = \sum_{k\in\mathfrak{K}^+} {w_k^{*,\tilde\lambda}}^2 + \sum_{k\in\mathfrak{K}^-} {w_k^{*,\tilde\lambda}}^2$, we restrict ourselves to proving

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left| \tilde\lambda \sum_{k\in\mathfrak{K}^+} {w_k^{*,\tilde\lambda}}^2 - \lambda 2 g(0) \int_{\operatorname{supp}(g_\xi)} \frac{\left(f_+^{w^{*,\tilde\lambda}}{}''(x)\right)^2}{g(x)} dx \right| = 0. \tag{74}$$

Using the Definition A.8 of $f_+^{w^{*,\tilde\lambda}}$ we get:

$$f_+^{w^{*,\tilde\lambda}}{}''(x) \overset{\text{Definition A.8}}{=} \sum_{k\in\mathfrak{K}^+ : |\xi_k - x| < \frac{1}{2\sqrt{n} g_\xi(x)}} \sqrt{n} g_\xi(x) w_k^{*,\tilde\lambda} v_k \tag{75a}$$

$$= \sum_{k\in\mathfrak{K}^+ : |\xi_k - x| < \frac{1}{2\sqrt{n} g_\xi(x)}} \sqrt{n} g_\xi(x) \frac{w_k^{*,\tilde\lambda}}{v_k} v_k^2 \tag{75b}$$

$$\overset{\text{Lemma A.16}}{\approx} \left( \frac{w_{l_x}^{*,\tilde\lambda}}{v_{l_x}} \pm \frac{\epsilon}{n} \right) \sum_{k\in\mathfrak{K}^+ : |\xi_k - x| < \frac{1}{2\sqrt{n} g_\xi(x)}} \sqrt{n} g_\xi(x) v_k^2 \tag{75c}$$

$$\approx \left( \frac{w_{l_x}^{*,\tilde\lambda}}{v_{l_x}} \pm \frac{\epsilon}{n} \right) \left(1 \overset{\mathbb{P}}{\pm} \epsilon_1\right) \mathbb{P}\left[v_k > 0\right] n g_\xi(x) \left( \mathbb{E}\left[v_k^2 \middle| \xi_k = x\right] \overset{\mathbb{P}}{\pm} \epsilon_2 \right) \tag{75d}$$

$$\overset{\text{Lemma A.17}}{\approx} \frac{w_{l_x}^{*,\tilde\lambda}}{v_{l_x}} \mathbb{P}\left[v_k > 0\right] n g_\xi(x) \mathbb{E}\left[v_k^2 \middle| \xi_k = x\right] \overset{\mathbb{P}}{\pm} \epsilon_3 \tag{75e}$$

uniformly in $x$ on $K$ for any $l_x$ satisfying $l_x \in \mathfrak{K}^+ : |\xi_l - x| < \frac{1}{2\sqrt{n} g_\xi(x)} \ \forall x \in \text{supp}(g_\xi)$. Therefore we can plug this into the right-hand term of eq. (74):

$$\lambda 2g(0) \int_{\text{supp}(g_\xi)} \frac{\left( {f_+^{w^*,\tilde{\lambda}}}''(x) \right)^2}{g(x)} dx \approx \lambda 2g(0) \int_{\text{supp}(g_\xi)} \frac{\left( \frac{w^{*,\tilde{\lambda}}_{l_x}}{v_{l_x}} \mathbb{P}\left[v_k > 0\right] n g_\xi(x) \mathbb{E}\left[v_k^2 \middle| \xi_k = x\right] \overset{\mathbb{P}}{\pm} \epsilon_3 \right)^2}{g(x)} dx$$

$$\approx \underbrace{\lambda 2g(0) \int_{\text{supp}(g_\xi)} \frac{\left( \frac{w^{*,\tilde{\lambda}}_{l_x}}{v_{l_x}} \mathbb{P}\left[v_k > 0\right] n g_\xi(x) \mathbb{E}\left[v_k^2 \middle| \xi_k = x\right] \right)^2}{g(x)} dx \overset{\mathbb{P}}{\pm} \epsilon_4}$$

$$= \frac{\tilde{\lambda} n}{2} \int_{\text{supp}(g_\xi)} \left( \frac{w^{*,\tilde{\lambda}}_{l_x}}{v_{l_x}} \right)^2 g_\xi(x) \mathbb{E}\left[v_k^2 \middle| \xi_k = x\right] dx$$

by uniformity of approximation (75) and by using the definitions of $\tilde{\lambda} := \lambda n 2g(0)$ and $g(x) := g_\xi(x)\mathbb{E}\left[v_k^2 \middle| \xi_k = x\right] \frac{1}{2}$. In the next steps we show that the left-hand term of eq. (74) converges to the same term as the right-hand side did:[71]

$$\tilde{\lambda} \sum_{k \in \mathfrak{K}^+} {w_k^{*,\tilde{\lambda}}}^2 \overset{71}{=} \tilde{\lambda} \sum_{\substack{\ell \in \mathbb{Z} \\ [\delta\ell, \delta(\ell+1)) \subseteq [C^\ell_{g_\xi}, C^u_{g_\xi}]}} \left( \sum_{\substack{k \in \mathfrak{K}^+ \\ \xi_k \in [\delta\ell, \delta(\ell+1))}} \left( \frac{w_k^{*,\tilde{\lambda}}}{v_k} \right)^2 v_k^2 \right)$$

$$\overset{\text{Lemma A.16}}{\approx} \tilde{\lambda} \sum_{\substack{\ell \in \mathbb{Z} \\ [\delta\ell, \delta(\ell+1)) \subseteq [C^\ell_{g_\xi}, C^u_{g_\xi}]}} \left( \left( \frac{w^{*,\tilde{\lambda}}_{l_{\delta\ell}}}{v_{l_{\delta\ell}}} \pm \frac{\epsilon_5}{n} \right)^2 \underbrace{\sum_{\substack{k \in \mathfrak{K}^+ \\ \xi_k \in [\delta\ell, \delta(\ell+1))}} v_k^2}_{\approx \left(1 \overset{\mathbb{P}}{\pm} \epsilon_6\right) \frac{n}{2} \delta g_\xi(\delta\ell) \left( \mathbb{E}\left[v_k^2 \middle| \xi_k = \delta\ell\right] \overset{\mathbb{P}}{\pm} \epsilon_7 \right)} \right)$$

$$\overset{\text{Lemma A.17}}{\approx} \frac{\tilde{\lambda} n}{2} \sum_{\substack{\ell \in \mathbb{Z} \\ [\delta\ell, \delta(\ell+1)) \subseteq [C^\ell_{g_\xi}, C^u_{g_\xi}]}} \left( \left( \frac{w^{*,\tilde{\lambda}}_{l_{\delta\ell}}}{v_{l_{\delta\ell}}} \right)^2 \delta g_\xi(\delta\ell) \left( \mathbb{E}\left[v_k^2 \middle| \xi_k = \delta\ell\right] \right) \overset{\mathbb{P}}{\pm} \epsilon_8 \right)$$

$$\overset{\text{Riemann}}{\approx} \frac{\tilde{\lambda} n}{2} \int_{\text{supp}(g_\xi)} \left( \frac{w^{*,\tilde{\lambda}}_{l_x}}{v_{l_x}} \right)^2 g_\xi(x) \mathbb{E}\left[v_k^2 \middle| \xi_k = x\right] dx \overset{\mathbb{P}}{\pm} \epsilon_9$$

This proves eq. (72). □

[71] Assume $\exists \ell_1, \ell_2 \in \mathbb{Z} : C^\ell_{g_\xi} = \delta\ell_1, C^u_{g_\xi} = \delta\ell_2$ to make the notation simpler. For a cleaner proof, one should choose a suitable partition of $\text{supp}(g_\xi)$.

**Definition A.20** (extended feasible set $\tilde{\mathcal{T}}$)**.** The *extended feasible set* $\tilde{\mathcal{T}}$ is defined as:

$$\tilde{\mathcal{T}} := \left\{ (f_+, f_-) \in H^2(\mathbb{R}) \times H^2(\mathbb{R}) \middle| \begin{array}{l} \operatorname{supp}(f_+'') \subseteq \operatorname{supp}(g), \operatorname{supp}(f_-'') \subseteq \operatorname{supp}(g), \\ f_+(x) = 0 = f_+'(x) \quad \forall x \leq C_g^\ell, \\ f_-(x) = 0 = f_-'(x) \quad \forall x \geq C_g^u \end{array} \right\}.$$

by replacing $\mathcal{C}^2(\mathbb{R})$ by the Sobolev space [1] $H^2(\mathbb{R}) := W^{2,2}(\mathbb{R}) \supset \mathcal{C}^2(\mathbb{R})$ in $\mathcal{T}$ from Definition 3.5.

*Remark* A.21. If one replaces $\mathcal{C}^2(\mathbb{R})$ by the Sobolev space $H^2(\mathbb{R}) := W^{2,2}(\mathbb{R})$ in Definitions 3.5 and A.3 the minimizer $\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right)$ does not change—i.e.:

$$\operatorname*{arg\,min}_{(f_+,f_-)\in\mathcal{T}} F_{+-}^{\lambda,g}(f_+, f_-) = \operatorname*{arg\,min}_{(f_+,f_-)\in\tilde{\mathcal{T}}} F_{+-}^{\lambda,g}(f_+, f_-).$$

**Lemma A.22** (step 7)**.** *For any $\lambda > 0$ and data $(x_i^{train}, y_i^{train}) \in \mathbb{R}^2$, $i \in \{1, \dots, N\}$, for any sequence of tuples of functions $\left(f_+^n, f_-^n\right) \in H^2(\mathbb{R}) \times H^2(\mathbb{R})$ such that*

$$\mathbb{P}\text{-}\lim_{n\to\infty} F_{+-}^{\lambda,g}\left(f_+^n, f_-^n\right) = F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right), \tag{76}$$

*then it follows that:*

$$\mathbb{P}\text{-}\lim_{n\to\infty} \left\| \left(f_+^n + f_-^n\right) - \underbrace{f_{g,\pm}^{*,\lambda}}_{f_{g,+}^{*,\lambda}+f_{g,-}^{*,\lambda}} \right\|_{W^{1,\infty}(K)} = 0. \tag{77}$$

*Proof.* Define the tuple of $H^2(\mathbb{R})$-functions

$$\left(u_+^n, u_-^n\right) := \left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right) - \left(f_+^n, f_-^n\right) \tag{78}$$

as the difference. The difference $\left(u_+^n, u_-^n\right)$ of elements from $\mathcal{T}$ and $\tilde{\mathcal{T}}$ obviously lies in $\tilde{\mathcal{T}}$.

Recall that the penalty term of $F_{+-}^{\lambda,g}$ is given by

$$P_{+-}^g(f_+, f_-) := 2g(0) \left( \int_{\operatorname{supp}(g)} \frac{\left(f_+''(x)\right)^2}{g(x)} dx + \int_{\operatorname{supp}(g)} \frac{\left(f_-''(x)\right)^2}{g(x)} dx \right). \tag{79}$$

This penalty $P_{+-}^g$ is obviously a quadratic form. Note that $\frac{\left(f_+^n, f_-^n\right) + \left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right)}{2} \in \tilde{\mathcal{T}}$. Since the training loss $L$ is convex, we get the inequality

$$L\left(\frac{f_+^n + f_-^n + f_{g,+}^{*,\lambda} + f_{g,-}^{*,\lambda}}{2}\right) \leq \frac{L\left(f_+^n + f_-^n\right)}{2} + \frac{L\left(f_{g,+}^{*,\lambda} + f_{g,-}^{*,\lambda}\right)}{2}. \tag{80}$$

Since the penalty $P_{+-}^g$ is a quadratic form, we get with the help of some algebraic calculations the inequality

$$P_{+-}^g\left(\frac{\left(f_+^n, f_-^n\right) + \left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right)}{2}\right) \leq \frac{P_{+-}^g\left(f_+^n, f_-^n\right)}{2} + \frac{P_{+-}^g\left(f_{g,+}^{*,\lambda}, f_{g,-}^{*,\lambda}\right)}{2} - \frac{P_{+-}^g\left(u_+^n, u_-^n\right)}{4}. \tag{81}$$

Combining the inequalities (80) and (81) results in

(82)
$$F_{+-}^{\lambda,g}\left(\frac{\left(f_+^n,f_-^n\right)+\left(f_{g,+}^{*,\lambda},f_{g,-}^{*,\lambda}\right)}{2}\right)\leq\underbrace{\frac{F_{+-}^{\lambda,g}\left(f_+^n,f_-^n\right)+F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda},f_{g,-}^{*,\lambda}\right)}{2}}_{\overset{(76)}{\approx}F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda},f_{g,-}^{*,\lambda}\right)\overset{\mathbb{P}}{\pm}\epsilon}-\lambda\frac{P_{+-}^g\left(u_+^n,u_-^n\right)}{4}.$$

Together with the optimality of $\left(f_{g,+}^{*,\lambda},f_{g,-}^{*,\lambda}\right)$ this result leads directly to

(83a)
$$F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda},f_{g,-}^{*,\lambda}\right)\overset{\text{optimality Remark A.21}}{\leq}F_{+-}^{\lambda,g}\left(\frac{\left(f_+^n,f_-^n\right)+\left(f_{g,+}^{*,\lambda},f_{g,-}^{*,\lambda}\right)}{2}\right)$$

(83b)
$$\overset{(82)}{\lessapprox}F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda},f_{g,-}^{*,\lambda}\right)\overset{\mathbb{P}}{\pm}\epsilon-\lambda\frac{P_{+-}^g\left(u_+^n,u_-^n\right)}{4}.$$

By subtracting $\left(F_{+-}^{\lambda,g}\left(f_{g,+}^{*,\lambda},f_{g,-}^{*,\lambda}\right)-\lambda\frac{P_{+-}^g\left(u_+^n,u_-^n\right)}{4}\right)$ from both sides of ineq. (83) and multiplying by 4 we get

$$\lambda P_{+-}^g\left(u_+^n,u_-^n\right)\overset{(83)}{\lessapprox}\overset{\mathbb{P}}{\pm}4\epsilon,$$

which implies that

(84)
$$\mathbb{P}\text{-}\lim_{n\to\infty}P_{+-}^g\left(u_+^n,u_-^n\right)=0.$$

First, we will show that the weak second derivative ${u_+^n}''$ converges to zero. We have

(85)
$$\left\|{u_+^n}''\right\|_{L^2(K)}\leq\frac{max_{x\in\text{supp}(g)}g(x)}{2g(0)}P_{+-}^g\left(u_+^n,u_-^n\right)\quad\forall K\subseteq\mathbb{R},$$

because $\left(u_+^n,u_-^n\right)\in\tilde{\mathcal{T}}$ has zero second derivative outside $\text{supp}(g)$. Thus,

$$\mathbb{P}\text{-}\lim_{n\to\infty}\left\|{u_+^n}''\right\|_{L^2(K)}=0$$

(by combining eqs. (84) and (85)). This can be used to apply two times the Poincaré-typed Lemma A.10 (first on ${u_+^n}''$ then on ${u_+^n}'$) to get for every compact set $K\subset\mathbb{R}$

(86)
$$\mathbb{P}\text{-}\lim_{n\to\infty}\left\|u_+^n\right\|_{W^{1,\infty}(K)}=0,$$

as $\left(u_+^n,u_-^n\right)\in\tilde{\mathcal{T}}$ satisfies the boundary conditions at $C_g^\ell$ (cp. Remark 3.6) because of the compact support of $g$. Analogously, $\mathbb{P}\text{-}\lim_{n\to\infty}\left\|u_-^n\right\|_{W^{1,\infty}(K)}=0$ for every compact set $K\subset\mathbb{R}$ and hence

(87)
$$\mathbb{P}\text{-}\lim_{n\to\infty}\left\|u_+^n+u_-^n\right\|_{W^{1,\infty}(K)}=0.$$

Thus, by the definition (78) of $\left(u_+^n,u_-^n\right)$ we get

$$\mathbb{P}\text{-}\lim_{n\to\infty}\left\|\left(f_+^n+f_-^n\right)-\underbrace{f_{g,\pm}^{*,\lambda}}_{f_{g,+}^{*,\lambda}+f_{g,-}^{*,\lambda}}\right\|_{W^{1,\infty}(K)}\overset{(78)}{=}\mathbb{P}\text{-}\lim_{n\to\infty}\left\|u_+^n+u_-^n\right\|_{W^{1,\infty}(K)}\overset{(87)}{=}0,$$

which shows (77). □

A.2. **Proof of Theorem 3.17 ($\mathcal{RN}_{w^T,\omega} \to \mathcal{RN}_\omega^{*,\frac{1}{T}}$).** In this section we prove all the results (Lemma 3.15, Remark 3.16 and Theorem 3.17) presented in Section 3.2. These results are analogous to the results presented in [4, 9, 29, 12], but we will repeat the proofs briefly in this appendix.

*Proof of Lemma 3.15.* We need to show that for any $\omega \in \Omega$,

("(21)") $$w^T(\omega) = -\exp\left(-2TX^\top(\omega)X(\omega)\right) w^{*,0+}(\omega) + w^{*,0+}(\omega),$$

satisfies (GD). Let $\omega \in \Omega$ be fixed and set $y := (y_1^{\text{train}}, \dots, y_N^{\text{train}})^\top$. Clearly, $w^0 = 0$. Since

$$\nabla_w L(\mathcal{RN}_w) = 2X^\top(Xw - y),$$

(GD) reads as

$$dw^t = -2(X^\top X w^t - X^\top y)\, dt. \tag{88}$$

Differentiating (21) we obtain

$$\frac{d}{dt} w^t = 2X^\top X \exp\left(-2tX^\top X\right) w^{*,0+}. \tag{89}$$

Moreover, since

$$\begin{aligned} -2(X^\top X w^t - X^\top y) &= 2X^\top X \exp\left(-2tX^\top X\right) w^{*,0+} - 2X^\top y w^{*,0+} + 2X^\top y w^{*,0+} \\ &= 2X^\top X \exp\left(-2tX^\top X\right) w^{*,0+} \end{aligned}$$

the result follows (by the Picard—Lindelöf theorem the solution of linear ODEs is unique). □

*Proof of Remark 3.16.* Using basic results on the Moore-Penrose pseudoinverse [2] and singular value decomposition it directly follows that the minimum norm solution $w^{*,0+}$ does not have any singular-value-components in the null-space of the matrix $X$. Combining this with basic knowledge about the matrix exponential of diagonalizable matrices, the result follows. Since the matrix-exponential in eq. (21) only preserves the null-space of $X$, every singular-value-component outside the null-space is scaled down to zero as $T \to \infty$. □

*Proof of Theorem 3.17.* First, we note that obviously

$$\lim_{T\to\infty} w^{*,\frac{1}{T}}(\omega) = w^{*,0+}(\omega) \quad \forall \omega \in \Omega \tag{90}$$

holds by Definition 3.3.

Secondly, the continuity of the map $(\mathbb{R}^n, \|\cdot\|_2) \to W^{1,\infty}(K) : w \mapsto \mathcal{RN}_{w,\omega}$ implies: $\forall \omega \in \Omega$:

$$\lim_{T\to\infty} \left\| \mathcal{RN}_\omega^{*,\frac{1}{T}} - \mathcal{RN}_{w^{*,0+}(\omega),\omega} \right\|_{W^{1,\infty}(K)} = 0, \text{ because of eq. (90)} \tag{91a}$$

$$\lim_{T\to\infty} \left\| \mathcal{RN}_{w^T(\omega),\omega} - \mathcal{RN}_{w^{*,0+}(\omega),\omega} \right\|_{W^{1,\infty}(K)} = 0, \text{ because of Remark 3.16.} \tag{91b}$$

Thirdly, by applying the triangle inequality on eqs. (91) the result (22) follows. □

## A.3. **Proof of Corollary 2.3 and Lemma 2.4.**

**Lemma A.23** (Uniform continuity w.r.t. first-layer weights)**.** *Let $\mathcal{NN}_\theta$ be a shallow neural network as introduced in Definition 1.4 and define $(b, v) \in \mathbb{R}^{n\times(d+1)}$ to be the collection of the network's first layer parameters. Then, for every $\epsilon > 0$ and for any compact $K \subset \mathbb{R}^d$ there exists a $\delta > 0$ such that,*

$$\forall(\tilde{b}, \tilde{v}) \in U_\delta(b, v) : \left\| \sum_{k=1}^{n} w_k \sigma\left(\tilde{b}_k + \sum_{j=1}^{d} \tilde{v}_{k,j} x_j\right) - \mathcal{NN}_\theta \right\|_{L^\infty(K)} < \frac{\epsilon}{2},$$

*with*

$$U_\delta(b, v) := \left\{ (\tilde{b}, \tilde{v}) \in \mathbb{R}^{n\times(d+1)} \middle| \max_{k\in\{1,\dots,n\}} \left\| (b_k, v_k) - (\tilde{b}_k, \tilde{v}_k) \right\|_2 < \delta \right\}.$$

*Proof.* For any $x \in K$, we have

$$\frac{\partial \mathcal{NN}_\theta(x)}{\partial b_k} = w_k \sigma'\left(b_k + \sum_{j=1}^{d} v_{k,j} x_j\right),$$
$$\frac{\partial \mathcal{NN}_\theta(x)}{\partial v_{k,i}} = w_k \sigma'\left(b_k + \sum_{j=1}^{d} v_{k,j} x_j\right) x_i.$$

Both derivatives can be bounded by above by $L := \max_{k\in\{1,\dots,n\}} |w_k| L_\sigma c_K$, with $L_\sigma$ the Lipschitz constant corresponding to $\sigma$ and $c_K > 0$ s.t. $\|x\|_2 \leq c_K \forall x \in K$ as $K$ was assumed to be compact. Since the bound $L$ is independent of $x$ and $w$, the statement follows. □

*Proof of Corollary 2.3.* By uniform approximation in the sense of [22], we have for any $\epsilon > 0$, that there exists an $N^{\epsilon/2} \in \mathbb{N}$, $\mathcal{NN}^{\epsilon/2} : \mathbb{R}^d \to \mathbb{R}$ with

$$\mathcal{NN}^{\epsilon/2}(x) := \sum_{k=1}^{N^{\epsilon/2}} \theta_k \sigma\left(b_k + \sum_{j=1}^{d} v_{k,j} x_j\right)$$

with $\theta_k, b_k, v_{k,j} \in \mathbb{R}$ such that

$$\left\| \mathcal{NN}^{\epsilon/2} - f \right\|_{L^\infty(K)} < \frac{\epsilon}{2}. \tag{92}$$

We now like to consider the probability that a randomly chosen vector of weights $(\tilde{b}_k, \tilde{v}_k)$ corresponding to the $k^{\text{th}}$ neuron in the hidden layer is close to a specific weight vector $(b_i, v_i)$ of $\mathcal{NN}^{\epsilon/2}$. Since $\lambda^{d+1}(U_\delta(b_i, v_i)) > 0$ it follows from $\mu \gg \lambda^{d+1}$ that $\mu(U_\delta(b_i, v_i)) > 0$. Therefore,

$$0 < p := \min_{i\in\{1,\dots,N^{\epsilon/2}\}} \mu(U_\delta(b_i, v_i)) \leq 1.$$

The probability that none of the sampled weights $(\tilde{b}_k, \tilde{v}_k)$, $k = 1, \dots, n$ is in the $\delta$-neighborhood of a specific vector $(b_i, v_i)$ can be bounded as follows:

$$\mathbb{P}^n\left(\left[\forall k \in \{1, \dots, n\} : (\tilde{b}_k, \tilde{v}_k) \notin U_\delta(b_i, v_i)\right]\right) = (1 - \mu(U_\delta(b_i, v_i)))^n \leq (1 - p)^n.$$

This implies

$$\begin{aligned}
&\mathbb{P}^n\left(\underbrace{\left[\exists i \in \{1,\dots,N^{\epsilon/2}\} : \forall k \in \{1,\dots,n\} : (\tilde{b}_k,\tilde{v}_k) \notin U_\delta(b_i,v_i)\right]}_{=:B}\right)\\
&= \mathbb{P}^n\left(\bigcup_{i=1}^{N^{\epsilon/2}}\left[\forall k \in \{1,\dots,n\} : (\tilde{b}_k,\tilde{v}_k) \notin U_\delta(b_i,v_i)\right]\right)\\
&\le \sum_{i=1}^{N^{\epsilon/2}} \mathbb{P}^n\left(\left[\forall k \in \{1,\dots,n\} : (\tilde{b}_k,\tilde{v}_k) \notin U_\delta(b_i,v_i)\right]\right)\\
&\le \sum_{i=1}^{N^{\epsilon/2}} (1-p)^n = (1-p)^n \cdot N^{\epsilon/2} \underset{n\to\infty}{\longrightarrow} 0.
\end{aligned}$$

For every $\omega \in B^c$ define

$$\begin{aligned}
\iota : \left\{1,\dots,N^{\epsilon/2}\right\} &\to \{1,\dots,n\},\\
i &\mapsto \iota(i),
\end{aligned}$$

with $(\tilde{b}_{\iota(i)},\tilde{v}_{\iota(i)})(\omega) \in U_\delta(b_i,v_i)$. Without loss of generality, $\iota$ is injective (choose $\delta$ small enough s.t. $U_\delta(b_i,v_i)$, $i = 1,\dots,N^{\epsilon/2}$ are disjoint). For those $\omega \in B^c$ we further define $\mathcal{RN}_w$ as in the statement of the corollary, with trainable last layer weights

$$w_k := \begin{cases} \theta_k, & \exists i \in \{1,\dots,N^{\epsilon/2}\} : \iota(i) = k,\\ 0, & \nexists i \in \{1,\dots,N^{\epsilon/2}\} : \iota(i) = k. \end{cases}$$

By Lemma A.23, it follows that $\left\|\mathcal{RN}_w - \mathcal{NN}^{\epsilon/2}\right\|_{L^\infty(K)} < \epsilon/2$ on $B^c$. Hence, an application of the triangle inequality, together with (92) yield that

$$\forall \omega \in B^c : \|\mathcal{RN}_w - f\|_{L^\infty(K)} < \epsilon.$$

□

*Proof of Lemma 2.4.* We show that $\mathbb{P}$-almost surely, $\{\psi_{(b,v)}(x_1),\dots,\psi_{(b,v)}(x_N)\}$ are linearly independent, for then the terminal linear regression can be (uniquely in case $N = n$) solved. Consider first the one-dimensional subspace $L_1 := [\psi_{(b,v)}(x_1)] \subseteq \text{range}(\psi_{(b,v)})$, i.e. the linear hull of $\psi_{(b,v)}(x_1)$ restricted to the latent space. By assumption, $\mathbb{P}_\#(\psi_{(b,v)}(x_2))[L_1] = 0$ and hence $\mathbb{P}$-almost surely, $\psi_{(b,v)}(x_2) \notin [\psi_{(b,v)}(x_1)]$. Analogously,

$$L_{N-1} := [\psi_{(b,v)}(x_1),\dots,\psi_{(b,v)}(x_{N-1})] \subseteq \text{range}(\psi_{(b,v)})$$

constitutes a $(N-1)$-dimensional subspace of $\text{range}(\psi_{(b,v)})$ for which

$$\mathbb{P}_\#(\psi_{(b,v)}(x_N))[L_{N-1}] = 0,$$

and thus $\psi_{(b,v)}(x_N) \notin L_{N-1}$ $\mathbb{P}$-almost surely. Thus, almost surely there exists $w \in \mathbb{R}^n$ such that $\sum_{k=1}^n w_k \psi_{(b,v)}(x_i)_k = y_i$ for all $i = 1,\dots,N$. □

## APPENDIX B. INTUITION ABOUT ADAPTED REGRESSION SPLINE $f_{g,\pm}^{*,\lambda}$

Coming soon...[72]

ETH ZÜRICH, D-MATH, RÄMISTRASSE 101, CH-8092 ZÜRICH, SWITZERLAND
*Email address*: jakob.heiss@math.ethz.ch, jteichma@math.ethz.ch, hanna.wutte@math.ethz.ch

[72]The adapted regression spline $f_{g,\pm}^{*,\lambda}$ is a very intuitive, easy to interpret concept, but without the right guidance it can take a few hours instead of a few minutes to acquire this intuition. This is why we will add this section in the next version to make this intuition more accessible, so that it becomes easy to see why the adapted regression spline $f_{g,\pm}^{*,\lambda}$ is very close to the regression spline $f^{*,\lambda}$ under typical circumstances and in which scenarios they differ from each other and how they differ from each other.