On Regularization and Robustness of Deep Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Despite their success, deep neural networks suffer from several drawbacks: they lack robustness to small changes of input data known as "adversarial examples" and training them with small amounts of annotated data is challenging. In this work, we study the connection between regularization and robustness by viewing neural networks as elements of a reproducing kernel Hilbert space (RKHS) of functions and by regularizing them using the RKHS norm. Even though this norm cannot be computed, we consider various approximations based on upper and lower bounds. These approximations lead to new strategies for regularization, but also to existing ones such as spectral norm penalties or constraints, gradient penalties, or adversarial training. Besides, the kernel framework allows us to obtain margin-based bounds on adversarial generalization. We study the obtained algorithms for learning on small datasets, learning adversarially robust models, and discuss implications for learning implicit generative models.

1 Introduction

Learning predictive models for complex tasks often requires large amounts of annotated data. Interestingly, successful models such as convolutional neural networks are also huge-dimensional and typically involve more parameters than training samples. Such a setting is challenging: besides the fact that training with small datasets is difficult, these models also lack robustness to small adversarial perturbations (Szegedy et al., 2013; Biggio & Roli, 2018). In the context of perceptual tasks such as vision (Szegedy et al., 2013) or speech recognition (Carlini & Wagner, 2018), these perturbed examples are often perceived identically to the original ones by a human, but can lead to arbitrarily different model predictions.

In this paper, we present a unified perspective on regularization and robustness, by viewing convolutional neural networks as elements of a particular reproducing kernel Hilbert space (RKHS) following the work of Bietti & Mairal (2018) on deep convolutional kernels. For such kernels, the RKHS contains indeed deep convolutional networks similar to generic ones—up to smooth approximations of rectified linear units. Such a point of view is interesting to adopt for deep networks since it provides a natural regularization function, the RKHS norm, which allows us to control the variations of the predictive model according to the geometry induced by the kernel. Besides, the norm also acts as a Lipschitz constant, which provides a direct control on the stability to adversarial perturbations.

In contrast to traditional kernel methods, the RKHS norm cannot be explicitly computed in our setup. Yet, these norms admit numerous approximations—lower bounds and upper bounds—which lead to new practical strategies for regularization, but also to existing ones. We may indeed recover several approaches such as using penalties or constraints on the spectral norm of the filters (Yoshida & Miyato, 2017), various forms of robust optimization (Madry et al., 2018), variants of gradient penalties (Drucker & Le Cun, 1991), and tangent propagation Simard et al. (1998).

Moreover, regularization and robustness are tightly linked in our kernel framework. In particular, we show a close connection between adversarial training under ℓ_2 perturbations and penalizing with the RKHS norm, in a similar vein to Xu et al. (2009b); from a statistical point of view, we also obtain margin-based bounds based on the RKHS norm in the spirit

of Bartlett et al. (2017); Boucheron et al. (2005) for robust generalization, which highlight the trade-offs between accuracy and robustness (see Schmidt et al., 2018). Following these observations, we evaluate empirically such a trade-off for different regularization strategies, by considering learning tasks with small datasets or with adversarial perturbations. We also provide a new perspective on recent successful approaches to training generative adversarial networks as optimizing a kernel two-sample test (see, *e.g.*, Gretton et al., 2012).

Related work. The construction of hierarchical kernels and the study of neural networks in the corresponding RKHS was studied by Mairal (2016); Zhang et al. (2016; 2017); Bietti & Mairal (2018). Our study of the relationship between robustness and regularization follows from Xu et al. (2009a;b), where the main focus is on linear models with quadratic or hinge losses. Some of the regularization strategies we obtain from our kernel perspective are closely related to previous approaches to adversarial robustness (Cisse et al., 2017; Madry et al., 2018; Simon-Gabriel et al., 2018; Roth et al., 2018), to improving generalization (Drucker & Le Cun, 1991; Miyato et al., 2018b; Sedghi et al., 2018; Simard et al., 1998; Yoshida & Miyato, 2017), and stable training of generative adversarial networks (Roth et al., 2017; Gulrajani et al., 2017; Arbel et al., 2018; Miyato et al., 2018a). The notion of adversarial generalization was considered by Schmidt et al. (2018), who provide lower bounds on a specific example of data distribution. Sinha et al. (2018) provide generalization guarantees from a distributional robustness perspective; however, they consider expected loss instead of classification error, and their bounds do not highlight the dependence on the model complexity, unlike ours.

2 Regularization strategies for deep neural networks

In this section, we recall the kernel perspective on deep networks introduced by Bietti & Mairal (2018), and present upper and lower bounds on the RKHS norm of a given model, leading to various regularization strategies. For simplicity, we first consider real-valued networks and binary classification, before discussing multi-class extensions.

2.1 Relation between deep neural networks and RKHSs

Kernel methods consist of mapping data living in a set $\mathcal{X}$ to a RKHS $\mathcal{H}$ associated to a positive definite kernel K through a mapping function $\Phi : \mathcal{X} \to \mathcal{H}$, and then learning simple machine learning models in $\mathcal{H}$. Specifically, when considering a real-valued regression or binary classification problem, classical kernel methods find a prediction function $f : \mathcal{X} \to \mathbb{R}$ living in the RKHS which can be written in linear form, i.e., such that $f(x) = \langle f, \Phi(x) \rangle_{\mathcal{H}}$ for all x in $\mathcal{X}$. While explicit mapping to a possibly infinite-dimensional space is of course only an abstract mathematical operation, learning f can be done implicitly by computing kernel evaluations and typically using convex programming (Schölkopf & Smola, 2001).

Moreover, the RKHS norm $\|f\|_{\mathcal{H}}$ acts as a natural regularization function, which controls the variations of model predictions according to the geometry induced by Φ:

$$|f(x) - f(x')| \leq \|f\|_{\mathcal{H}} \cdot \|\Phi(x) - \Phi(x')\|_{\mathcal{H}}. \tag{1}$$

Unfortunately, traditional kernel methods become difficult to use when the datasets are large or when evaluating the kernel is intractable. Here, we propose a different approach that considers explicit parameterized representations of functions contained in the RKHS, given by generic convolutional neural networks, and leverage properties of the RKHS and the kernel mapping in order to regularize when learning the network parameters.

Consider indeed a real-valued deep convolutional network $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is simply $\mathbb{R}^d$, with rectified linear unit (ReLU) activations and no bias units. By constructing an appropriate multi-layer hierarchical kernel, Bietti & Mairal (2018) show that the corresponding RKHS $\mathcal{H}$ contains a convolutional network with the same architecture and parameters as f, but with activations that are smooth approximations of ReLU. Although the model predictions might not be strictly equal, we will abuse notation and denote this approximation with smooth ReLU by f as well, with the hope that the regularization procedures derived from the RKHS model will be effective in practice on the original CNN f.

Besides, Bietti & Mairal (2018) show that the mapping $\Phi(\cdot)$ is non-expansive:

$$\|\Phi(x) - \Phi(x')\|_{\mathcal{H}} \leq \|x - x'\|_2, \tag{2}$$

so that controlling $\|f\|_{\mathcal{H}}$ provides some robustness to additive ℓ_2-perturbations, by (1). Additionally, with appropriate pooling operations, Bietti & Mairal (2018) show that the kernel mapping is also stable to deformations, meaning that the RKHS norm also controls robustness to translations and other transformations including scaling and rotations (Engstrom et al., 2017), which can be seen as deformations when they are small.

In contrast to standard kernel methods, where the RKHS norm is typically available in closed form, this norm is difficult to compute in our setup, and requires approximations. The following sections present upper and lower bounds on $\|f\|_{\mathcal{H}}$, with linear convolutional operations denoted by W_k for $k = 1, \dots, L$, where L is the number of layers. Defining $\theta := \{W_k : k = 1, \dots, L\}$, we then leverage these bounds to approximately solve the following penalized or constrained optimization problems on a training set $(x_i, y_i), i = 1, \dots, n$:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) + \lambda \|f_\theta\|_{\mathcal{H}}^2 \quad \text{or} \quad \min_{\theta : \|f_\theta\|_{\mathcal{H}} \leq C} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)). \tag{3}$$

We also note that while the construction of Bietti & Mairal (2018) considers VGG-like networks (Simonyan & Zisserman, 2014), the regularization algorithms we obtain in practice can be easily adapted to different architectures such as residual networks (He et al., 2016).

2.2 Exploiting Lower bounds of the RKHS norm

In this section, we devise regularization algorithms by leveraging lower bounds on $\|f\|_{\mathcal{H}}$, which are obtained by relying on the following variational characterization of Hilbert norms:

$$\|f\|_{\mathcal{H}} = \sup_{\|u\|_{\mathcal{H}} \leq 1} \langle f, u \rangle_{\mathcal{H}}. \tag{4}$$

At first sight, this definition is not useful since the set $U = \{u \in \mathcal{H} : \|u\|_{\mathcal{H}} \leq 1\}$ may be infinite-dimensional and the inner products $\langle f, u \rangle_{\mathcal{H}}$ cannot be computed in general. Thus, we devise tractable lower bound approximations by considering smaller sets $\bar{U} \subset U$.

Adversarial perturbation penalty. The non-expansiveness of Φ allows us to consider the subset $\bar{U} \subset U$ defined as $\bar{U} = \{\Phi(x + \delta) - \Phi(x) : x \in \mathcal{X}, \|\delta\|_2 \leq 1\}$, leading to the bound

$$\|f\|_{\mathcal{H}} \geq \sup_{x \in \mathcal{X}, \|\delta\|_2 \leq 1} f(x + \delta) - f(x), \tag{5}$$

which is reminiscent of adversarial perturbations. Adding a regularization parameter $\epsilon > 0$ in front of the norm then corresponds to different sizes of perturbations:

$$\epsilon \|f\|_{\mathcal{H}} = \sup_{\|u\|_{\mathcal{H}} \leq \epsilon} \langle f, u \rangle_{\mathcal{H}} \geq \sup_{x \in \mathcal{X}, \|\delta\|_2 \leq \epsilon} f(x + \delta) - f(x). \tag{6}$$

Using this lower bound or its square as a penalty in the objective (3) when training a neural network can then provide a way to regularize. In practice, optimizing over all x in $\mathcal{X}$ is difficult, and we can replace the supremum with an expectation over the training set, or with a supremum over a subset of the training set (such as a mini-batch), yielding further lower bounds on the penalty. In the context of a mini-batch stochastic gradient algorithm, one can obtain a subgradient of this penalty by first finding maximizers $\hat{x}, \hat{\delta}$ (where $\hat{x}$ is one of the examples in the mini-batch), and then simply computing gradients of $f_\theta(\hat{x} + \hat{\delta}) - f_\theta(\hat{x})$ (or its square) w.r.t. θ using back-propagation. We compute the perturbations $\hat{\delta}$ for each example x by using a few steps of projected subgradient ascent with constant step-lengths, in a similar fashion to Madry et al. (2018) in the context of robust optimization.

Relationship with robust optimization. This penalized approach turns out to be closely related to solving the robust optimization problem

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \sup_{\|\delta\|_2 \leq \epsilon} \ell(y_i, f_\theta(x_i + \delta)), \tag{7}$$

which is commonly considered for training adversarially robust classifiers (Kolter & Wong, 2017; Madry et al., 2018; Raghunathan et al., 2018). In particular, Xu et al. (2009b) show that the penalized objective and the robust objective are equivalent in the case of the hinge loss with linear predictors, under some additional conditions on the data and perturbations, such as non-separability. They also show the equivalence for kernel methods when considering the (intractable) full perturbation set U around each point in the RKHS $\Phi(x_i)$, that is, predictions $\langle f, \Phi(x_i) + u\rangle_{\mathcal{H}}$ with u in U. Intuitively, when a training example (x_i, y_i) is misclassified, we are in the "linear" part of the hinge loss, so that

$$\sup_{\|u\|_{\mathcal{H}} \leq \epsilon} \ell(y_i, \langle f, \Phi(x_i) + u\rangle_{\mathcal{H}}) = \ell(y_i, \langle f, \Phi(x_i)\rangle_{\mathcal{H}}) + \sup_{\|u\|_{\mathcal{H}} \leq \epsilon} \langle f, u\rangle_{\mathcal{H}} = \ell(y_i, f(x_i)) + \epsilon\|f\|_{\mathcal{H}}.$$

Considering additive perturbations in the input space corresponds to reducing the RKHS perturbations to elements of the form $u = \Phi(x_i + \delta) - \Phi(x_i)$ with $\|\delta\|_2 \leq \epsilon$, giving a lower bound on the above quantity. For other losses such as the the logistic loss, the regularization effect is still present even for correctly classified examples, though it may be smaller since the loss has a reduced slope for such points, leading to a more *adaptive* regularization mechanism which may automatically reduce the amount of regularization on easier datasets. Either way, because both the hinge and logistic loss are 1-Lipschitz, we have that the robust objective (7) is upper bounded by the penalized objective with penalty $\epsilon\|f\|_{\mathcal{H}}$.

Gradient penalties. Taking $\bar{U} = \{\frac{\Phi(x)-\Phi(y)}{\|x-y\|_2} : x, y \in \mathcal{X}\}$, which is a subset of U by Eq. (2)—it turns out that this is the same set as above, since Φ is homogeneous (Bietti & Mairal, 2018) and $\mathcal{X} = \mathbb{R}^p$—we obtain a lower bound based on the Lipschitz constant of f:

$$\|f\|_{\mathcal{H}} \geq \|f\|_L := \sup_{x,y\in\mathcal{X}} \frac{f(x) - f(y)}{\|x - y\|_2}.$$

We can further bound

$$\|f\|_L \geq \sup_{x\in\mathcal{X}} \|\nabla f(x)\|_2,$$

where the supremum is taken over points where f is differentiable, and equality holds when $\mathcal{X}$ is convex. Taking the square and replacing the supremum over x by an expectation over the training data leads to a gradient penalty that has been recently used to stabilize the training of generative adversarial networks (Gulrajani et al., 2017; Roth et al., 2017). Leveraging the connection to robust optimization again, when ϵ is small, we may also consider penalties based on the gradients of the loss, which have been used for regularization (Drucker & Le Cun, 1991) and robustness (Lyu et al., 2015; Roth et al., 2018; Simon-Gabriel et al., 2018). This has the advantage of overcoming the difficulties of the maximization problem over δ by leveraging the closed-form expression of the gradient norm.

Other connections with lower bounds. We may also consider a set $\bar{U} = \{\Phi(\tilde{x}) - \Phi(x) : x \in \mathcal{X},\ \tilde{x} \text{ is a small deformation of } x\}$, where the amount of deformation is dictated by the stability bounds of Bietti & Mairal (2018) in order to ensure that $\bar{U} \subset U$. Then, the obtained penalty (6) and corresponding robust optimization formulation (7) resemble a form of *data augmentation* where transformations are optimized instead of sampled, as in Engstrom et al. (2017). By considering infinitesimal deformations from specific parameterized classes such as translations or rotations, the obtained gradient penalty then resembles the *tangent propagation* regularization strategy of Simard et al. (1998).

One advantage of these approaches based on lower bounds is that the obtained penalties are independent of the model parameterization, making them flexible enough to use with more complex architectures in practice. In addition, the connection with robust optimization can provide a useful mechanism for adaptive regularization. However, these lower bounds do not guarantee a control on the norm $\|f\|_{\mathcal{H}}$, since they might only provide a poor approximation. In particular, a classifier trained with such approximate penalties may simply fit a stable function on ϵ-balls around all training points, while being unstable near the decision boundary, leading to a poor control of the RKHS norm. The next section presents approaches based on upper bounds, which provide guarantees on $\|f\|_{\mathcal{H}}$, overcoming some of the aforementioned drawbacks. Nevertheless, we found that controlling the lower bounds often does help controlling the upper bounds as well in many cases (see Section 4). Moreover, we found that combining upper- and lower-bound strategies may lead to further gains.

2.3 Exploiting upper bounds: optimization with spectral norms

In contrast to lower bounds, upper bounds can provide a guaranteed control on the RKHS norm. Bietti & Mairal (2018) show the following upper bound:

$$\|f\|_{\mathcal{H}} \leq U(\|W_1\|, \ldots, \|W_L\|), \tag{8}$$

where U is increasing in all of its arguments, and $\|W_k\|$ is the spectral norm of the linear convolutional operator W_k. Here, we simply consider the spectral norm on the filters, given by $\|W\| := \sup_{\|x\|_2 \leq 1} \|Wx\|_2$. Other generalization bounds relying on similar quantities have been proposed for controlling complexity (Bartlett et al., 2017; Neyshabur et al., 2018), suggesting that using them for regularization is relevant even beyond our kernel perspective.

Penalizing the spectral norms. One way to control the upper bound (8) when learning a neural network f_θ is to consider a regularization penalty based on spectral norms

$$\min_\theta \frac{1}{n} \sum_{i=1}^n \ell(y_i, f_\theta(x_i)) + \lambda \sum_{l=1}^L \|W_l\|^2, \tag{9}$$

where λ is a regularization parameter. In the context of a stochastic gradient algorithm, one can obtain (sub)gradients of the penalty by computing singular vectors associated to the highest singular value of each W_l. We consider the method of Yoshida & Miyato (2017), which computes such singular vectors approximately using one or two iterations of the power method, as well as a more costly approach using the full SVD.

Constraining the spectral norms. In the constrained setting, we want to optimize:

$$\min_\theta \frac{1}{n} \sum_{i=1}^n \ell(y_i, f_\theta(x_i)) \quad \text{s.t } \|W_l\| \leq \tau \ ; \ l \in 1, \ldots, L \ ,$$

where τ is a user-defined constraint. This objective may be optimized by projecting each W_l in the spectral norm ball of radius τ after each gradient step. Such a projection is achieved by truncating the singular values to be smaller than τ (see Appendix A). We found that the loss was hardly optimized with this approach, but that this issue can be avoided by using an exponentially decaying schedule for τ reaching a constant τ_0 after a few epochs.

2.4 Extension to multiple classes

We now discuss how to extend the regularization strategies to multi-valued networks, in order to deal with multiple classes. First, the upper bound strategies of Section 2.3 are easy to extend, by simply considering spectral norms up to the last layer. This is also justified by the generalization bound of Bartlett et al. (2017), which applies to the multi-class setup.

In the context of lower bounds, we can consider a multi-class penalty $\|f_1\|_{\mathcal{H}}^2 + \ldots + \|f_K\|_{\mathcal{H}}^2$ for an $\mathbb{R}^K$-valued function $f = (f_1, f_2, \ldots, f_K)$. One can then consider lower bounds on this penalty by leveraging lower bounds on each individual f_k. In particular, we use:

$$\|f\|_M^2 := \sum_{k=1}^K \sup_{x \in \mathcal{X}, \|\delta\|_2 \leq \epsilon} (f_k(x+\delta) - f_k(x))^2.$$

Computing this quantity requires finding one perturbation for each class k and each x, which can be costly when K is large. By constraining the perturbations to be the same for different classes, we can obtain another lower bound that is computationally cheaper, yielding a penalty $\|f\|_S^2 := \sup_{x \in \mathcal{X}, \|\delta\|_2 \leq \epsilon} \|f(x+\delta) - f(x)\|_2^2$. For robust optimization formulations (7), the extension is straightforward, given that standard multi-class losses such as the cross-entropy loss can be directly optimized in an adversarial training or gradient penalty setup.

3 Theoretical guarantees and extensions

While various methods have been introduced to empirically gain robustness to perturbations, their robustness on test data, also known as *adversarial generalization* (Schmidt et al., 2018) is poorly understood. In this section, we study how margin bounds for kernel methods can provide theoretical guarantees. We then consider heuristic extensions to different geometries, by analogy with linear models. Finally, we discuss how our kernel approach provides novel interpretations for training generative adversarial networks.

3.1 Guarantees on adversarial generalization

Margin-based bounds have been useful to explain the generalization behavior of learning algorithms that can fit the training data well, such as kernel methods, boosting and neural networks (Koltchinskii et al., 2002; Boucheron et al., 2005; Bartlett et al., 2017). Here, we show how similar bounds can be used to obtain guarantees on adversarial generalization. For a binary classification task with labels in $\mathcal{Y} = \{-1, 1\}$ and data distribution $\mathcal{D}$, we would like to bound the expected adversarial error of a classifier f, given for some $\epsilon > 0$ by

$$P_{(x,y)\sim\mathcal{D}}(\exists \|\delta\|_2 \leq \epsilon : \; yf(x+\delta) < 0). \tag{10}$$

Note that if f is $\|f\|_L$-Lipschitz, we then have

$$P(\exists \|\delta\|_2 \leq \epsilon : \; yf(x+\delta) < 0) \leq P(yf(x) < \epsilon \|f\|_L). \tag{11}$$

We now show how to further bound this quantity using empirical margins, following the usual approach to obtaining margin bounds for kernel methods (*e.g.*, Boucheron et al., 2005). Consider a training dataset $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$. Define

$$L^\gamma(f) := P(yf(x) < \gamma) \qquad \text{and} \qquad L_n^\gamma(f) := \frac{1}{n}\sum_{i=1}^n \mathbb{1}\{y_i f(x_i) < \gamma\}.$$

We then have the following bound, proved in Appendix B:

Proposition 1. *With probability $1-\delta$ over a dataset $\{(x_i, y_i)\}_{i=1,\ldots,n}$, we have, for all choices of $\eta \geq 0$, $\gamma > 0$, and $f \in \mathcal{H}$,*

$$L^\eta(f) \leq L_n^{\eta+\gamma}(f) + \tilde{O}\left(\frac{\|f\|_{\mathcal{H}}}{\gamma\sqrt{n}}\sqrt{\frac{1}{n}\sum_{i=1}^n K(x_i, x_i)} + \sqrt{\frac{\log 1/\delta}{n}}\right). \tag{12}$$

Taking $\eta = 0$ leads to the usual margin bound, while $\eta = \epsilon\|f\|_L$ yields a bound on adversarial error by (11), for some neural network f learned from training data. We note that other complexity measures based on products of spectral norms may be used instead of $\|f\|_{\mathcal{H}}$, as well as multi-class extensions, following Bartlett et al. (2017); Neyshabur et al. (2018).

One can then compare different regularization algorithms by inspecting cumulative distributions of the *normalized margins* $\bar{\gamma}_i = y_i f(x_i)/\|f\|_{\mathcal{H}}$. Given that $\|f\|_L \leq \|f\|_{\mathcal{H}}$, one can then simply look at the part of the plot to the right of $\bar{\gamma} = \epsilon$ in order to assess expected adversarial error with ϵ-bounded perturbations. When n becomes smaller, the best bound is obtained for a larger value of $\bar{\gamma}$, which also suggests that the right side of the plots is indicative of performance on small datasets. An example is given in Section 4.2.

3.2 Beyond Euclidian geometries

The kernel approach from previous sections is well-suited for input spaces $\mathcal{X}$ equipped with the Euclidian distance, thanks to the non-expansiveness property (2) of the kernel mapping. In the case of linear models, this kernel approach corresponds to using ℓ_2-regularization by taking a linear kernel. However, other forms of regularization and geometries can often be useful, for example to encourage sparsity with an ℓ_1 regularizer. Such a regularization approach presents tight links with robustness to ℓ_∞ perturbations on input data, thanks to the duality relation $\|w\|_1 = \sup_{\|u\|_\infty} \langle w, u\rangle$ (see Xu et al., 2009a).

In the context of deep networks, we can leverage such insights to obtain new regularizers, expressed in the same variational form as the lower bounds in Section 2.2, but with different geometries on $\mathcal{X}$. For ℓ_∞ perturbations, we obtain

$$\sup_{x,y\in\mathcal{X}} \frac{f(x)-f(y)}{\|x-y\|_\infty} \quad \geq \quad \sup_{x\in\mathcal{X}} \|\nabla f(x)\|_1. \tag{13}$$

The Lipschitz regularizer (l.h.s.) can also be taken in an adversarial perturbation form, with different strenghts:

$$\sup_{x\in\mathcal{X},\|\delta\|_\infty\leq\epsilon} f(x+\delta)-f(x).$$

As most visible in the gradient ℓ_1-norm in (13), these penalties encourage some sparsity in the gradients of f, which is a reasonable prior for regularization on images, for instance, where we might only want predictions to change based on few salient pixel regions. This can lead to gains in interpretability, as observed by Tsipras et al. (2018). We push this principle one step further in our experiments, by considering other structured penalties on the gradients such as the total variation.

We note that in the case of linear models, our robust margin bound of Section 3.1 can be adapted to ℓ_∞-perturbations, by leveraging Rademacher complexity bounds for ℓ_1-constrained models (Kakade et al., 2009). Obtaining similar bounds for neural networks would be interesting but goes beyond the scope of this paper.

3.3 Regularization of generative adversarial networks

Generative adversarial networks attempt to learn a *generator* neural network $G_\phi : \mathcal{Z} \to \mathcal{X}$, so that the distribution of $G_\phi(z)$ with $z \sim D_z$ a noise vector resembles a data distribution D_x. Various recent approaches have relied on regularization strategies on a *discriminator* network in order to improve the stability of GAN training and the quality of the produced samples. Some of these resemble our approaches from Section 2 such as gradient penalties (Gulrajani et al., 2017; Roth et al., 2017) and spectral norm regularization (Miyato et al., 2018a).

While these regularization methods had different motivations originally, we suggest that viewing them as approximations of an RKHS norm constraint provides some insight into their effectiveness in practice; specifically, we provide an interpretation as optimizing a kernel two-sample test such as MMD with the convolutional kernel introduced in Section 2:

$$\min_\phi \sup_{f\in\mathcal{H}:\|f\|_{\mathcal{H}}\leq 1} \mathbb{E}_{x\sim D_x}[f(x)] - \mathbb{E}_{z\sim D_z}[f(G_\phi(z))]. \tag{14}$$

In contrast to the Wasserstein GAN interpretation of a similar objective where all 1-Lipschitz functions are considered (Arjovsky et al., 2017), the MMD interpretation yields a parametric statistical rate $O(n^{-1/2})$ when learning from an empirical distribution with n samples, which is significantly better than the $O(n^{-1/d})$ rate of the Wasserstein-1 distance for high-dimensional data such as images (Sriperumbudur et al., 2012). While the MMD approach has been used for training generative models, it generally relies on a generic kernel function, such as a Gaussian kernel, that appears explicitly in the objective (Dziugaite et al., 2015; Li et al., 2017; Bińkowski et al., 2018). Although using a learned feature extractor can improve this, the Gaussian kernel might be a poor choice when dealing with natural signals such as images, while the hierarchical kernel of Bietti & Mairal (2018) is better suited for this type of data, by providing useful invariance and stability properties. Leveraging the variational form of the MMD (14) with this kernel suggests using convolutional networks as the discriminator f, with constraints on the spectral norms in order to ensure $\|f\|_{\mathcal{H}} \leq C$ for some C, as done by Miyato et al. (2018a) through normalization.

4 Experiments

We tested the regularization strategies presented in Section 2 in the context of improving generalization on small datasets and training adversarially robust models. Our goal is to use

Table 1: Regularization on CIFAR10 with 5 000 or 1 000 examples for VGG-11 and ResNet-18. Each entry shows the test accuracy with/without data augmentation when all hyper-parameters are optimized on a validation set. TV denotes the total variation norm.

Method	5k VGG-11	5k ResNet-18	1k VGG-11	1k ResNet-18
No weight decay	71.13/57.88	68.14/53.01	50.68/42.82	42.19/38.18
Weight decay	73.13/57.71	71.59/54.71	51.16/44.11	43.93/37.84
SN penalty (PI)	74.23/62.26	73.25/53.19	53.38/45.29	46.93/38.67
SN penalty (SVD)	74.7/60.87	73.49/55.46	52.32/44.52	47.03/40.04
SN projection	74.58/**63.36**	75.87/55.31	53.52/**46.0**	45.11/38.56
$\|f\|_M^2$ penalty	72.95/60.16	70.61/55.47	51.71/45.0	45.23/**44.44**
PGD-ℓ_2	72.7/59.37	72.72/54.99	51.52/43.74	46.83/39.3
PGD-ℓ_∞	73.29/58.04	72.81/54.13	51.92/43.74	46.27/40.97
grad-ℓ_1	**75.13**/58.92	71.11/54.46	53.86/43.65	**48.49**/41.01
grad-ℓ_2	**75.13**/58.63	73.83/55.39	**54.79**/42.73	47.56/42.06
grad-TV	73.67/57.92	71.82/53.13	51.4/44.18	43.47/40.75
PGD-ℓ_2 + SN projection	**75.02**/61.42	**76.06**/**56.99**	53.74/45.93	46.96/40.19
grad-ℓ_2 + SN projection	74.67/**63.12**	**76.17**/**55.53**	**55.98**/**46.68**	**48.94**/**43.26**

common convolutional architectures used for large datasets and improve their performance in different settings through regularization.

For the adversarial training strategies, the inner maximization problems are solved with 5 steps of projected gradient ascent (Madry et al., 2018) with a randomly chosen starting point. In the case of the multi-class lower bound adversarial penalties $\|f\|_M^2$ and $\|f\|_S^2$, which respectively optimize over one perturbation per class and a single perturbation overall (see Section 2.4), we also maximize over examples in the mini-batch, only considering the maximal element when computing gradients w.r.t. parameters. For the robust optimization problem (7), we consider the PGD approach for ℓ_2 and ℓ_∞ perturbations (Madry et al., 2018), as well as the corresponding ℓ_2 (squared) and ℓ_1 gradient norm penalties. For the upper bound approaches with spectral norms (SNs), we consider the SN projection strategy with decaying τ, as well as the SN penalty (9), either using power iteration (PI) or a full SVD for computing gradients.

4.1 Improving generalization on small datasets

In this setting, we use 1 000 and 5 000 examples of the CIFAR10 dataset, with or without data augmentation. We consider a VGG network (Simonyan & Zisserman, 2014) with 11 layers, as well as a residual network (He et al., 2016) with 18 layers, which achieve 91% and 93% test accuracy respectively when trained on the full training set with data augmentation. We do not use any batch normalization layers in order to prevent any interaction with spectral norms. Each regularization strategy derived in Section 2 is trained for 500 epochs using SGD with momentum and batch size 128, halving the step-size every 40 epochs from a fixed initial step-size (0.05 for VGG-11, 0.1 for ResNet-18), a strategy we found to work relatively well for all methods. In order to study the potential effectiveness of each method, we assume that a fairly large validation set is available to select hyper-parameters; thus, we keep 10 000 annotated examples for this purpose.

Table 1 shows the test accuracies we obtain for upper and lower bound approaches, as well as combined approaches and different geometries. Among upper bound strategies, the constrained approach often works best, perhaps thanks to a more explicit control of the SNs compared to the penalized approach. The SN penalty can work well nevertheless, and provides computational benefits when using the PI variant. For lower bound approaches, we found the gradient ℓ_1 and ℓ_2 norm penalties to often work better than the adversarial penalty $\|f\|_M^2$ and adversarial PGD approaches. We note that the best regularization parameters are often quite small on such datasets, making the gradient penalties good approximations of the robust objective (7) used by PGD. Because gradient penalties have closed form gradients while PGD only obtains them by solving (7) approximately, they may

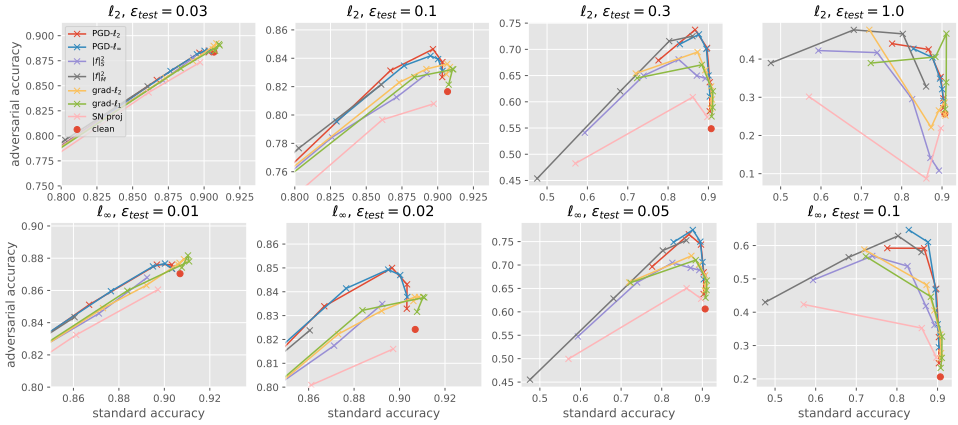

Figure 1: Robustness trade-off curves of different regularization methods for VGG11 on CIFAR10. Each plot shows test accuracy vs adversarial test accuracy for ℓ_2 (top) and ℓ_∞ (bottom) bounded adversaries with a fixed ϵ_{test}. Different points on a curve correspond to training with different regularization strengths.

work better in this setting thanks to optimization benefits. The adaptive nature of the regularization through robust optimization may also be beneficial on these datasets, although the explicit penalization by $\|f\|_M^2$ seems to be more helpful in the case of 1 000 examples with no data augmentation, which is plausibly the hardest setting in our experiments. Finally, we can see that the combined approaches of robust optimization together with SN constraints often yield the best results. Indeed, lower bound approaches do not necessarily control the upper bounds (and this is particularly true for PGD, as discussed in Section 4.2), which might explain why the additional constraints on SNs are helpful.

4.2 Training adversarially robust models

We consider the same VGG architecture as in Section 4.1, trained on CIFAR10 with data augmentation, with different regularization strategies. Each method is trained for 300 epochs using SGD with momentum and batch size 128, dividing the step-size in half every 30 epochs. This strategy was successful in reaching convergence for all methods.

Figure 1 shows the test performance of the different regularization approaches in the presence of ℓ_2 and ℓ_∞-bounded adversaries, plotted against standard accuracy (the leftmost points correspond to stronger regularization). We can see that the robust optimization approaches tend to work better in high-accuracy regimes, perhaps due to a more adaptive form of regularization, while the $\|f\|_M^2$ penalty can be useful in some regimes where robustness to large perturbations is needed. Comparing the different geometries for PGD and gradient penalties, we find that using the right metric can help, but not dramatically, which may suggest that more appropriate optimization algorithms than SGD could be needed to better accommodate the non-smooth case of ℓ_1/ℓ_∞, or perhaps that both algorithms are actually controlling the same notion of complexity on this dataset. The plots also confirm that gradient penalties are preferable for small regularization strengths, possibly due to better optimization, while for stronger regularization, the gradient approximation no longer holds and the adversarial training approaches such as PGD are preferred.

These results suggest that a robust optimization approach such as PGD can work very well on the CIFAR10 dataset, perhaps because the training data is easily separable with a large margin and the method adapts to this "easiness". However, this raises the question of whether the approach is actually controlling $\|f\|_{\mathcal{H}}$, given that it only attempts to minimize a lower bound, as discussed in Section 2.2. Figure 2(left) casts some doubt on this, showing that for PGD in contrast to other methods, the product of spectral norms (representative of an upper bound on $\|f\|_{\mathcal{H}}$) increases when the lower bound $\|f\|_S$ decreases. This suggests that a network learned with PGD with large ϵ may have large RKHS norm, possibly because the approach tries to separate ϵ-balls around the training examples, which can be more

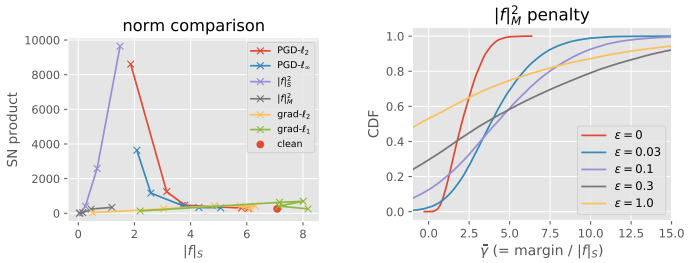

Figure 2: (left) Comparison of lower and upper bound quantities ($\|f\|_S$ vs the product of spectral norms). (right) CDF plot of normalized empirical margins for the $\|f\|_M^2$ penalty with different regularization strengths, normalized by $\|f\|_S$. We consider 1000 fixed training examples when computing $\|f\|_S$.

difficult than simply separating the training examples (see also Madry et al., 2018). For other methods, and in particular the adversarial penalties such as $\|f\|_M^2$, the upper and lower bounds appear more correlated, and thus we may look at the empirical distributions of normalized margins $\bar{\gamma}$ obtained using $\|f\|_S$ for normalization, shown in Figure 2(right). The curves suggest that for small $\bar{\gamma}$, smaller values of ϵ are preferred, while stronger regularization helps when $\bar{\gamma}$ increases, yielding lower test error when an adversary is present according to our bounds in Section 3.1. This qualitative behavior is indeed observed in the results of Figure 1 on test data for the $\|f\|_M^2$ penalty approach.

5 Discussion

Making generic machine learning techniques more data efficient is crucial to reduce the costs related to annotation. While other approaches may also be important to solve this grand challenge, such as incorporating more prior knowledge in the architecture (*e.g.*, Oyallon et al., 2017), semi-supervised learning (Chapelle et al., 2006) or meta-learning (when multiple tasks or datasets are available, see, *e.g.*, Thrun, 1998), basic regularization principles will be needed and those are crucially missing today. Such principles are also essential for obtaining robust models in applications where security is a concern, such as self driving cars. Our paper presents various algorithmic strategies for regularization on generic deep convolutional networks, by leveraging the structure of an appropriate RKHS, leading to many existing approaches to regularization, as well as new ones, in addition to providing theoretical guarantees and insights on different methods.

References

Michael Arbel, Dougal J Sutherland, Mikołaj Bińkowski, and Arthur Gretton. On gradient regularizers for MMD GANs. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Alberto Bietti and Julien Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *arXiv preprint arXiv:1706.03078*, 2018.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 9:323–375, 2005.

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint arXiv:1801.01944*, 2018.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (eds.). *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Harris Drucker and Yann Le Cun. Double backpropagation increasing generalization performance. In *International Joint Conference on Neural Networks (IJCNN)*, 1991.

Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

Logan Engstrom, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *arXiv preprint arXiv:1712.02779*, 2017.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research (JMLR)*, 13 (Mar):723–773, 2012.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

Vladimir Koltchinskii, Dmitry Panchenko, et al. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

J Zico Kolter and Eric Wong. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Chunchuan Lyu, Kaizhu Huang, and Hai-Ning Liang. A unified gradient regularization family for adversarial examples. In *IEEE International Conference on Data Mining (ICDM)*, 2015.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Julien Mairal. End-to-end kernel learning with supervised convolutional kernel networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018a.

Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2018b.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *International Conference on Computer Vision (ICCV)*, 2017.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Adversarially robust training through structured gradient regularization. *arXiv preprint arXiv:1805.08736*, 2018.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Mądry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. 2001.

Hanie Sedghi, Vineet Gupta, and Philip M Long. The singular values of convolutional layers. *arXiv preprint arXiv:1805.10408*, 2018.

Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. Transformation invariance in pattern recognition–tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pp. 239–274. Springer, 1998.

Carl-Johann Simon-Gabriel, Yann Ollivier, Bernhard Schölkopf, Léon Bottou, and David Lopez-Paz. Adversarial vulnerability of neural networks increases with input dimension. *arXiv preprint arXiv:1802.01421*, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, Gert RG Lanckriet, et al. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Sebastian Thrun. Lifelong learning algorithms. In *Learning to learn*, pp. 181–209. Springer, 1998.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. There is no free lunch in adversarial robustness (but there are unexpected benefits). *arXiv preprint arXiv:1805.12152*, 2018.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robust regression and lasso. In *Advances in Neural Information Processing Systems (NIPS)*, 2009a.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research (JMLR)*, 10(Jul):1485–1510, 2009b.

Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

Yuchen Zhang, Jason D Lee, and Michael I Jordan. ℓ_1-regularized neural networks are improperly learnable in polynomial time. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

Yuchen Zhang, Percy Liang, and Martin J. Wainwright. Convexified convolutional neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

A Details on optimization with spectral norms

This section details our optimization approach presented in Section 2.3 for learning with spectral norm constraints. In particular, we rely on a *continuation* approach, decreasing the size of the ball constraints during training, towards a final value τ. The method is presented in Algorithm 1. We use an exponentially decreasing schedule for τ, and found using 2 epochs for κ to work well in practice. In the context of convolutional networks, we simply consider the SVD of a reshaped filter matrix, but we note that alternative approaches based on the singular values of the full convolutional operation may also be used (Sedghi et al., 2018).

Algorithm 1 Stochastic projected gradient with continuation

Input: τ, κ, step-sizes η_t
for $t = 1, \ldots$ **do**
  Sample mini-batch and compute gradients of the loss w.r.t. each W^l, denoted G_t^l
  $\tau_t = \tau(1 + \exp\left(\frac{-t}{\kappa}\right))$
  for $l = 1, \ldots, L$ **do**
    $\tilde{W}_t^l := W_t^l - \eta_t G_t^l$
    Compute SVD: $\tilde{W}_t^l = U \mathrm{diag}(\sigma) V^T$
    $\widehat{\sigma} := \mathrm{proj}_{\|\cdot\|_\infty \leq \tau_t}(\sigma)$
    $W_{t+1}^l := U \mathrm{diag}(\widehat{\sigma}) V^T$
  end for
end for

B Details on generalization guarantees

This section presents the proof of Proposition 1, which relies on standard tools from statistical learning theory (*e.g.*, Boucheron et al., 2005).

B.1 Proof of Proposition 1

Proof. Assume for now that η and γ are fixed in advance, and let $\mathcal{F}_\lambda := \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq \lambda\}$. Consider the function

$$\phi(x) = \begin{cases} 0, & \text{if } x \leq -\gamma - \eta \\ 1, & \text{if } x \geq -\eta \\ 1 + (x+\eta)/\gamma, & \text{otherwise.} \end{cases}$$

Defining $A(f) = \mathbb{E}\,\phi(-yf(x)) \geq L^{\eta}(f)$ and $A_n(f) = \frac{1}{n}\sum_{i=1}^{n}\phi(-y_i f(x_i)) \leq L_n^{\eta+\gamma}(f)$, and noting that ϕ is upper bounded by 1 and $1/\gamma$ Lipschitz, we can apply similar arguments to (Boucheron et al., 2005, Theorem 4.1) to obtain, with probability $1-\delta$,

$$L^{\eta}(f) \leq L_n^{\eta+\gamma}(f) + O\left(\frac{1}{\gamma}R_n(\mathcal{F}_\lambda) + \sqrt{\frac{\log 1/\delta}{n}}\right),$$

where $R_n(\mathcal{F}_\lambda)$ denotes the empirical Rademacher complexity of $\mathcal{F}_\lambda$ on the dataset $\{(x_i, y_i)\}_{i=1,\ldots,n}$. Standard upper bounds on empirical Rademacher complexity of kernel classes with bounded RKHS norm yield the following bound

$$L^{\eta}(f) \leq L_n^{\eta+\gamma}(f) + O\left(\frac{\lambda}{\gamma\sqrt{n}}\sqrt{\frac{1}{n}\sum_{i=1}^{n}K(x_i, x_i)} + \sqrt{\frac{\log 1/\delta}{n}}\right).$$

The final bound follows from this one by relying on a careful union bound on various choices of η, γ and λ, as in (Bartlett et al., 2017, Lemma A.9), which leads to additional logarithmic factors (hidden in the $\tilde{O}$ notation). □