Adversarially Robust Kernel Smoothing

Jia-Jie Zhu
Empirical Inference Department
Max Planck Institute for Intelligent Systems
Tübingen, Germany
and Weierstrass Institute
Berlin, Germany
zhu@wias-berlin.de

Christina Kouridi
Empirical Inference Department
Max Planck Institute for Intelligent Systems
Tübingen, Germany
Currently at InstaDeep Ltd.
London, United Kingdom
christinakouridi@gmail.com

Yassine Nemmour
Empirical Inference Department
Max Planck Institute for Intelligent Systems
Tübingen, Germany
ynemmour@tuebingen.mpg.de

Bernhard Schölkopf
Empirical Inference Department
Max Planck Institute for Intelligent Systems
Tübingen, Germany
bs@tuebingen.mpg.de

Abstract

We propose a scalable robust learning algorithm combining kernel smoothing and robust optimization. Our method is motivated by the convex analysis perspective of distributionally robust optimization based on probability metrics, such as the Wasserstein distance and the maximum mean discrepancy. We adapt the integral operator using supremal convolution in convex analysis to form a novel function majorant used for enforcing robustness. Our method is simple in form and applies to general loss functions and machine learning models. Exploiting a connection with optimal transport, we prove theoretical guarantees for certified robustness under distribution shift. Furthermore, we report experiments with general machine learning models, such as deep neural networks, to demonstrate competitive performance with the state-of-the-art certifiable robust learning algorithms based on the Wasserstein distance.

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

1 Introduction

When learning with finitely many samples, there is an inevitable distribution shift between the training data and the test data, characterized by empirical process theory (van der Vaart and Wellner, 2013). A lack of causal inference can also cause learners to lose robustness under the shifted distribution (Meinshausen, 2018). Furthermore, potential malicious *adversaries* may create large artificial distribution shifts to hamper modern deep learners (Madry et al., 2019). Hence, learning under *distribution shift* presents significant challenges to current machine learning algorithms.

Distributionally robust optimization (DRO) (Delage and Ye, 2010; Scarf, 1958) seeks to robustify against unknown distribution shift explicitly. Given a loss function of interest $l(\theta, \cdot)$, it solves a *robust optimization* (Soyster, 1973; Ben-Tal et al., 2009) problem

$$\min_{\theta} \sup_{P \in \mathcal{C}} \mathbb{E}_{\xi \sim P}\ l(\theta, \xi), \tag{1}$$

where θ is the decision variable, ξ noise or randomness, $\mathcal{C}$ a set of distributions over the uncertain variable ξ that the optimizer wishes to robustify against, often referred to as the *ambiguity set*. DRO is particularly relevant to statistical machine learning as one may construct the ambiguity set $\mathcal{C}$ as a metric ball centering at the empirical distribution $\hat{P}_N$ that also contains the true data-generating distribution P_0. For example, the performance guarantees for Wasserstein DRO have been established by Mohajerin Esfahani and Kuhn (2018); Zhao and Guan (2018). Here, the idea is to use known convergence rate results for empirical es-

timations of the underlying probability metrics, e.g., the Wasserstein metrics (Kantorovich and Rubinshtein, 1958) and the closely related *integral probability metrics* (IPM) (Sriperumbudur et al., 2012; Müller, 1997). Such metrics (or topologies) often correspond to smooth functions as their dual spaces, which characterize the empirical distribution's convergence to the true data-generating distribution (van der Vaart and Wellner, 2013; Billingsley, 1971). Researchers have proposed DRO algorithms with various statistically meaningful ambiguity sets in a large body of literature. While not the focus of previous works, convex analysis tools play important roles in enforcing distributional robustness. They are the primary tools we employ in this paper.

Certain simple DRO problems, such as linear classification with logistic regression losses, admit the tractable reformulation into convex problems, as studied by Ben-Tal et al. (2013); Namkoong and Duchi (2017); Mohajerin Esfahani and Kuhn (2018); Blanchet et al. (2018); Shafieezadeh-Abadeh et al. (2015). However, this only applies to a limited class of convex loss functions and simple models, as also noted by, e.g., Sinha et al. (2017). For general machine learning models, e.g., deep neural networks (DNNs), and common losses l in (1), there exists no tractable reformulation to solve DRO (1). This paper addresses general losses in machine learning tasks, which are less explored in terms of principled distributional robustness, save very few exceptions such as Sinha et al. (2017); Blanchet et al. (2018); Zhu et al. (2020).

Contribution. This paper leverages the critical roles smoothness and function majorants play in enforcing distributional robustness. We summarize our contributions and sketch the main results.

1. We analyze the smooth function majorant perspective of distributional robustness, which generalizes the existing practice of using the Moreau-Yosida regularization in Wasserstein DRO to flexibly chosen general majorants surrogate losses. Specifically, we propose the k-transform (Definition 2) that adapts the convolution in the integral operator to the supremal convolution in convex analysis to form a new function majorant.
2. Using those tools, we propose a novel robust learning algorithm (Section 4), the *adversarially robust kernel smoothing* (ARKS). It solves the minimax program
$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left\{ \sup_{u} \{ l(\theta, u) k(u, \xi_i) \} \right\}. \qquad (2)$$
3. Exploiting a connection between the k-transform and optimal transport (OT), we provide theoretical guarantees in terms of robustness certificate for ARKS under distribution shift. Highlighting the role of kernel bandwidth, our analysis unifies the two perspectives of DRO using OT and kernel methods.
4. While ARKS is derived from kernel methods, it can be easily applied to large-scale machine learning with DNNs. For example, we report an experiment with a ResNet-20 model, where applying exact Wasserstein DRO reformulation techniques (e.g., (Mohajerin Esfahani and Kuhn, 2018; Shafieezadeh-Abadeh et al., 2015)) is out of the question. There, ARKS performs at least competitively with the WRM algorithm (5) proposed by Sinha et al. (2017).
5. Our code is publicly available online at `https://github.com/christinakouridi/arks`.

Notation. In this paper, we refer to the uniform data distribution $\hat{P}_N := \frac{1}{N} \sum_{i=1}^{N} \delta_{\xi_i}$ as the empirical distribution. We use P_0 to denote the (unknown) true data-generating distribution. For the loss functions of interest $l(\theta, \xi)$, we sometimes omit θ when there is no ambiguity. θ denotes the decision variables, such as the weights of neural networks. $\xi \in \mathcal{X}$ denotes the random variable of interest, e.g., input data or features. Its samples are denoted by ξ_i. For conciseness, we limit the discussion to compact $\mathcal{X}$'s. We use $\mathcal{H}$ to denote a function space in the context, e.g., RKHS introduced in the next section. Lip() denotes the Lipschitz seminorm. $\|\|\|_{\mathcal{H}}$ is the RKHS norm in the context. We will assume loss functions l to be bounded continuous functions throughout the paper; extensions to upper semi-continuity in optimization settings is straightforward. See, e.g., (Shapiro et al., 2014). Variables are in their vectorial representation, e.g., $x = [x_1, \ldots, x_N]$, and $f(x) = [f(x_1), \ldots, f(x_N)]$. Throughout the paper, we will refer to DRO using the Wasserstein distance as Wasserstein DRO. We refer to the Moreau-Yosida regularization $\widehat{l_{y,p}}(x) := \sup_u \{ l(u) - y \cdot \|u - x\|^p \}$ as the *supremal convolution* of a function l and the (scaled) norm function $y\|\cdot\|^p$. To avoid ambiguity, we refer to the maximization of a concave function as a convex program. Finally, γ denotes some metric or divergence measure in the probability simplex.

2 Reproducing Kernel Hilbert Spaces

A learning task can be mathematically described as a function approximation problem $\min_{f \in H} \|f - l\|_{\cdot}$, for some criterion $\|\|\|_{\cdot}$, e.g., function norm. The target function l is often only known at certain data points $[x_1, ..., x_N]$. One way to approach the function approximation problem is to consider a function approximator of the form $\sum_{j=1}^{N} a_j k(x_i, x_j) = l(x_i), 1 \leq i \leq N$, where a_j are the coefficients to be determined and

$k(x_i, x_j)$ some bi-variate function. It is in our interest that the matrix $[k(x_i, x_j)]_{i,j}$ should be positive definite. Motivated by this, we now define a symmetric real-valued function k as a positive (semi-)definite kernel if $\sum_{i=1}^n \sum_{j=1}^n a_i a_j k(x_i, x_j) \geq 0$ for any $n \in \mathbb{N}$, $\{x_i\}_{i=1}^n \subset \mathcal{X}$, and $\{a_i\}_{i=1}^n \subset \mathbb{R}$. It is known (e.g. Schölkopf and Smola, 2002, Chapter 2) that there is a one-to-one relationship between every positive semi-definite kernel k and a Hilbert space $\mathcal{H}$, whose feature map $\phi\colon \mathcal{X} \to \mathcal{H}$ satisfies $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$. This Hilbert space is *reproducing*, meaning that $f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}, x \in \mathcal{X}$. We call $\mathcal{H}$ the reproducing kernel Hilbert space (RKHS), also termed the native space of the kernel k. RKHSs are widely used in function approximation based on data due to their attractive properties.

In addition to the functional approximation aspect, the RKHS has also been recently used to manipulate distributions, leveraging its statistical properties as the so-called Glivenko-Cantelli classes; cf. (van der Vaart and Wellner, 2013). Relevant to the robustness aspect, the *maximum mean discrepancy* (MMD, (Gretton et al., 2012)) associated with an RKHS $\mathcal{H}$ is a metric in the probability simplex, $\gamma_{\mathcal{H}}(P, Q) = \sup_{\|f\|_{\mathcal{H}} \leq 1} \int f \, d(P - Q)$, given the associated kernel is characteristic. In particular, the minimax optimal rate for the MMD empirical estimation has been studied by Tolstikhin et al. (2016), which can be used to set the ambiguity set level ϵ in the DRO problem (6) with computable constants, independent of the dimensions. In contrast, measure concentration rates (used for Wasserstein DRO in (Mohajerin Esfahani and Kuhn, 2018)) for the Wasserstein distance are dimension-dependent. The MMD also has a closed-form estimator, while computing Wasserstein distance is hard in general (Peyré and Cuturi, 2019; Santambrogio, 2015). MMD can be generalized to the integral probability metrics (IPM) (Müller, 1997) defined by some function class $\mathcal{F}$, i.e., $\gamma_{\mathcal{F}}(P, \hat{P}) := \sup_{f \in \mathcal{F}} \int f d(P - \hat{P})$. The well-known choices relevant to this paper include: $\mathcal{F} = \{f : \mathrm{Lip}(f) \leq 1\}$ recovers the type-1 Wasserstein metric (Kantorovich metric); the RKHS norm-ball $\mathcal{F} = \{f : \|f\|_{\mathcal{H}} \leq 1\}$ recovers the MMD.

Given a kernel k and probability measure μ, recall that the integral operator $\mathcal{T}\colon L^2_\mu \to \mathcal{H}$ is defined as

$$\mathcal{T} l(x) := \int l(z) k(x, z) d\mu(z). \tag{3}$$

The integral operator maps L^2_μ to a subspace of the RKHS (Wendland, 2004; Conway, 2019) and is used in the celebrated Mercer's theorem to characterize the eigendecomposition of RKHS functions. In the context of this paper, we view the integral operator as a smoothing operation.

3 Distributionally Robust Optimization for Machine Learning

We limit our discussion to DRO using the Wasserstein metrics (Mohajerin Esfahani and Kuhn, 2018; Zhao and Guan, 2018; Gao and Kleywegt, 2016; Blanchet and Murthy, 2017), the MMD (Zhu et al., 2020; Staib and Jegelka, 2019), and general IPMs (Zhu et al., 2020). See (Gao and Kleywegt, 2016; Gretton et al., 2012; Arbel et al., 2021; Peyré and Cuturi, 2019; Weed and Bach, 2017, Section 1.1) for the details of why those probability metrics are more advantageous in many machine learning applications than, e.g., f-divergences. For convenience, we now restate the data-driven DRO primal formulation in (1) with a probability discrepancy constraint.

$$(\textit{DRO-Primal}) : \min_{\theta} \sup_{\gamma(P, \hat{P}_N) \leq \epsilon} \mathbb{E}_P \, l(\theta, \xi), \tag{4}$$

The discrepancy measure γ in (4) can be chosen to be an IPM or OT metric.

In general, solving the minimax DRO problem (4) requires a reformulation via the duality of conic linear optimization, cf. (Shapiro, 2001; Ben-Tal et al., 2015). While the Wasserstein distance has become the most popular choice for the DRO problem, it is important to understand that one *cannot* simply reformulate any Wasserstein DRO problem as a convex program, except for very simple losses such as logistic regression (Shafieezadeh-Abadeh et al., 2015). Unfortunately, for many practical machine learning models, there exists no exact tractable reformulation. Popular Wasserstein DRO approaches such as those proposed in (Mohajerin Esfahani and Kuhn, 2018; Zhao and Guan, 2018) apply to a limited class of loss functions and models, such as logistic regression (linear classification). Moreover, it is also known that estimating the Lipschitz constant for general models is intractable; cf. (Virmaux and Scaman, 2018; Bietti et al., 2019), making Lipschitz regularization in (Shafieezadeh-Abadeh et al., 2019) difficult. This paper does not impose such restrictions on losses or models. For commonly-used machine learning losses, it is well-known that one must resort to general approximate solution methods such as in (Sinha et al., 2017; Blanchet et al., 2018; Zhu et al., 2020).

Most relevant to our work, the authors of (Sinha et al., 2017) proposed to *give up* certifying the exact distributional robustness level ϵ and apply a convexification technique using the Moreau-Yosida regularization, as approximate Wasserstein DRO. They solve the risk minimization problem, which they termed Wasserstein

robust method (WRM),

$$(WRM): \min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\Big\{\widehat{f_\theta^y}(\xi_i) := \sup_u\{l(\theta,u) - y\cdot c(u,\xi_i)\}\Big\}, \quad (5)$$

where c is called the transport cost (Santambrogio, 2015). For example, when c is the squared Euclidean distance, $\widehat{f_\theta^y}(\xi_i)$ is referred to as the Moreau-Yosida regularization or Moreau envelope. In that setting, WRM overcomes the hurdle of the aforementioned hardness of DRO for general machine learning tasks by virtue of a convexification effect. Intuitively, subtracting a strongly convex function makes the inner objective more concave. This technique was also used in robust nonlinear optimization (Houska and Diehl, 2013), trust-regions in numerical optimization (Chapter 4 of (Nocedal and Wright, 2006)), and the S-procedure in robust control (Pólik and Terlaky, 2007; Yakubovich, 1971). We refer interested readers to those works for detailed numerical procedures. Later, we compare our novel kernel smoothing algorithm with WRM (Sinha et al., 2017) in experiments with general machine learning models, e.g., DNNs, to demonstrate our advantages over classical reformulation techniques.

On the other hand, if we choose the metric γ to be an IPM associated with the function class $\mathcal{F} \subseteq \mathcal{H}$, Zhu et al. (2020) proved the IPM-DRO duality. It states that the primal DRO problem (4) is equivalent to solving the variational optimization problem

$$(IPM\text{-}DRO): \min_{\theta, f\in\mathcal{H}} \quad \frac{1}{N}\sum_{i=1}^{N} f(\xi_i) + \epsilon\|f\|_{\mathcal{H}} \quad \text{subject to} \quad l(\theta,\xi) \le f(\xi),\ \forall \xi \in \mathcal{X} a.e. \quad (6)$$

Those authors also proposed approximate solution methods when the IPM is chosen as the MMD (see Section 2 for the advantages of MMD), which generalized the results of Staib and Jegelka (2019) to general loss functions. Through the lens of this paper, (6) explicitly seeks an upper *envelope* f of the loss l as solutions to the variational dual program (6). Instead of the Moreau-Yosida regularization, the smooth majorant role there is played by a more general smooth function $f \in \mathcal{H}$. Note that program (6) is trivial if the loss l is in an $\mathcal{H}$ and has a known RKHS norm. The authors of (Zhu et al., 2020) then proposed Kernel DRO that makes it possible to use the MMD associated with any universal RKHSs for DRO and compute the rate for general losses. To our knowledge, that is the only work aiming to exactly reformulate DRO for general machine learning models. Compared to their method, we provide an approach that produces a function that satisfies the (semi-)infinite constraint in (6), whereas Zhu et al. (2020)'s method can only satisfy that constraint approximately through constraint sampling.

To motivate our method, we make two key observations into (5): (1) the absence of the robustness level ϵ and (2) the fixed dual variable y. That insight is also equivalent to *giving up the exact minimization* w.r.t. $f \in \mathcal{H}$ and $\|f\|_{\mathcal{H}}$ in dual IPM-DRO (6), since fixing y in (5) is to not optimize w.r.t. the Moreau envelope $\widehat{f_\theta^y}$. This is equivalent to Lagrangian relaxation in nonlinear optimization.

4 A Kernel Smoothing Algorithm for Robust Learning

The so-called c-transform $\widehat{f_\theta^y}$ in WRM (5), also known as the Moreau-Yosida regularization, plays a crucial role in the robustness of WRM (5). Importantly, it is a *majorant* function.

Definition 1 (Majorant)**.** *We say that f is a majorant of l if $f(\xi) \ge l(\xi)$ for ξ a.e. in the domain of l.*

Notable examples of majorants relevant to DRO include the Moreau-Yosida regularization as well as the kernel functions in (6). In this paper, we also refer to a majorant as an upper envelope function. To further make clear the roles that majorants play in robustness, we provide a few convex analysis examples in the appendix regarding special cases of majorants relevant to DRO.

Our *key insight* from (6), (5), and Lemma B.1 is that *empirical risk minimization with the loss replaced by a majorant surrogate loss induces distributional robustness.* Motivated by such relationship between robustness and the use of majorants, e.g., c-transform (and Moreau-Yosida regularization) in (5), kernel functions in (6), we now introduce our robust learning algorithm.

4.1 Adversarially Robust Kernel Smoothing

Our starting point is the minimax robust optimization (RO) problem (Ben-Tal et al., 2009; Soyster, 1973)

$$(RO): \min_{\theta}\sup_{u\in\mathcal{X}} l(\theta,u), \quad (7)$$

where the learner assumes the uncertain variable u to take the worst-case value. It is easy to see the pessimism of RO as we do not know the true support of u in machine learning. On the other hand, empirical risk minimization (ERM; also referred to as the sample average approximation) enjoys better performance but is more fragile to shift in distribution and uncertainty. It can be seen as a simple form of smoothing by averaging. Deviating from the typical DRO dual reformulation approaches, our *key idea is to view smoothness as the*

opposite side of robustness by manipulating the integral operator in RO. To that end, let us first establish some tools.

The image of l under the integral operator $\mathcal{T}\, l \in \mathcal{H}$ (3) is a smooth function since it is in an RKHS, but does not directly enable robust learning. To achieve robustness, we notice the following inequality

$$\mathcal{T}\, l(x) = \int l(z)k(x,z)d\mu(z) \leq \sup_u \{l(u)k(u,x)\},\ \forall x \in \mathcal{X}, \quad (8)$$

which is a straightforward inequality between expectation and supremum. Using the right-hand-side expression, we propose the following majorant analogous to c-transform.

Definition 2 (k-transform)**.** *The k-transform of a function l associated with kernel k is defined as*

$$l^k(x) := \sup_u \{l(u)k(u,x)\}.$$

It is helpful to think of concrete examples where $k(u,x)$ is the Gaussian RBF kernel or Laplacian kernel. To make our discussion more general, we now propose the following family of kernels inspired by the transport cost c of OT.

Definition 3 (c-exponential kernel)**.** *Suppose c is the transport cost (as in the Wasserstein distance). The c-exponential kernel with bandwidth $\sigma > 0$ is given by $k(x,x') = e^{-c(x,x')/\sigma}$.*

Note that a relevant kernel on probability metrics was studied in (De Plaen et al., 2020). For conciseness, we focus on the Gaussian RBF kernel and Laplacian kernel in the rest of this section. Other constructions of majorants are possible and discussed in the appendix. It is then straightforward to verify the following:

Proposition 4.1. *The k-transform of l is a majorant of l. Furthermore, we have $l^k \to l$ as $\sigma \to 0$.*

Let us use the tools above to derive our robustification method. We mitigate the conservatism of RO (7) by replacing the the original loss l with a *smoothed* version $\min_\theta \sup_u \mathcal{T}\, l(\theta, u)$. In practice, we can only compute an empirical version of the integral operator based on data ξ_i. We have the following

$$\sup_u \Big\{\hat{\mathcal{T}}\, l(\theta,u) := \frac{1}{N}\sum_{i=1}^N k(\xi_i,u)l(\theta,u)\Big\} \leq \frac{1}{N}\sum_{i=1}^N \sup_u \{l(\theta,u)k(u,\xi_i)\}. \quad (9)$$

The inner objective on the right-hand-side is the k-transform. That objective is indeed less conservative than RO and more robust than ERM since

$$\textit{(ERM)}: \sup_u \frac{1}{N}\sum_{i=1}^N l(\theta,\xi_i) \leq \frac{1}{N}\sum_{i=1}^N \sup_u \{l(\theta,u)k(u,\xi_i)\} \leq \sup_u l(\theta,u)\ : \textit{(RO)} \quad (10)$$

for c-exponential kernels, e.g., Gaussian RBF kernels.

We are now ready to propose the following novel robust learning scheme based on the insight from RO, (6), and (5). The *main idea* here is simple: we minimize the risk using a surrogate loss constructed by the k-transform,

$$(\textit{ARKS}): \min_\theta \frac{1}{N}\sum_{i=1}^N \Big\{ l_\theta^k(\xi_i) := \sup_u \{l(\theta,u)k(u,\xi_i)\}\Big\}. \quad (11)$$

ARKS resembles the risk minimization schemes using majorant surrogate losses in (6) and (5), but with our newly proposed k-transform. Program (11) also bears a clear resemblance to the *Nadaraya-Watson* model, and the *vicinal risk minimization* (Chapelle et al., 2001) in the literature. However, our approach differs in taking supremum to guarantee distributional robustness (see Section 5.2) instead of merely smoothing. Different from the existing robust kernel density estimation methods such as (Kim and Scott, 2012), which was applied by Ning and You (2018) to learn uncertainty sets for robust optimization, ARKS considers specifically the worst-case risk of the loss l, rather than only performing general unsupervised density estimation.

Compared with existing DRO approaches, ARKS (11) does not use *explicit regularization* (e.g., (Shafieezadeh-Abadeh et al., 2019)), but an *implicit* one. To see that, we establish the following.

Proposition 4.2. *Suppose the kernel bandwidth tends to infinity $\sigma \to \infty$, ARKS* (11) *is equivalent to the* worst-case robust optimization *(RO)* (7) *(Ben-Tal et al., 2009; Soyster, 1973).*

If kernel bandwidth tends to zero $\sigma \to 0$, then ARKS (11) *recovers (ERM) $\frac{1}{N}\sum_{i=1}^N \min_\theta l(\theta,\xi_i)$.*

If we choose a bandwidth σ between those cases, the robustness is between RO and ERM, where DRO is.

Remark (Robustness, kernel bandwidth, and size of the function space)**.** *Intuitively, if the kernel bandwidth is large, the function space becomes small. In terms of robustness, our analysis is also consistent with the characterization of dual function space sizes for DRO and RO in (Zhu et al., 2020): large dual function spaces correspond to conservative but more robust optimization. In contrast, smaller ones have better performance*

but are less robust. We see those insights reflected in Proposition 4.2 and further in Section 5.2.

Our risk minimization scheme (11) can be straightforwardly used with stochastic gradient-based optimization for large-scale learning, e.g., with DNNs. We detail the training procedure in Algorithm 1. Note that Step 3

Algorithm 1: Robust Learning with ARKS

1: **input:** data sampler, initial iterate θ_0
2: **for** $k = 0, 1, 2, \ldots, T$ **do**
3: sample $\{\xi_k\}$ and find u_k^* by maximizing $l(\theta_k, u)k(u, \xi_k)$ w.r.t. u
4: update θ by stochastic gradient descent using estimate $\nabla_\theta l(\theta_k, u_k^*)$
5: **output:** approximate solution $\theta^* := \theta^T$

of Algorithm 1 can be seen as a proximal algorithm, discussed further in the next section. ARKS (11) can also be interpreted as a form of *adversarial training* (Kolter and Madry, 2018; Wong and Kolter, 2018; Goodfellow et al., 2015): for each ξ_i, the inner maximization problem of (11) looks for an adversarial example u that hurts the learner the most. In the case of Gaussian RBF kernel, we show that the inner maximization objective in (11) has favorable convexity structures for suitable choices of σ in the next section, as well as in the appendix.

We further illustrate the geometric intuition of ARKS in Figure 1 using a toy problem. For conciseness, we defer detailed experimental setups to the appendix. The idea is to model the distribution shift and adversarial perturbation as a stochastic transition from the original state X (illustrated as the black cross in Figure 1 (right)) to a perturbed uncertain state U (the gray curve). Concretely, the relationship in (8) (restated below for convenience) holds for any distribution $\forall \mu \in \mathcal{P}$,

$$\int l(z)k(x,z)d\mu(z) \leq \sup_u \{l(u)k(u,x)\}, \ \forall x \in \mathcal{X}.$$

The left-hand-side (LHS) above can be interpreted as the *conditional expectation* $\mathbb{E}[l(U)|X = x]$, which is the expected loss under the uncertain state U after a distribution shift from the empirical data distribution. In this context, a smooth kernel k models a conditional density $P(U = z|X = x)$ of this stochastic transition. Unlike f-divergence-based DRO (Ben-Tal et al., 2013; Namkoong and Duchi, 2016), our modeling does not require the shifted distribution to be absolute continuous (i.e., having the same support) w.r.t. the empirical distribution. ARKS uses the robust version of this operation characterized by the k-transform (i.e., right-hand-side (RHS) above) instead of the expectation on the LHS. Therefore, the resulting surrogate worst-case loss (plotted in Figure 1 (left)) upper-bounds the

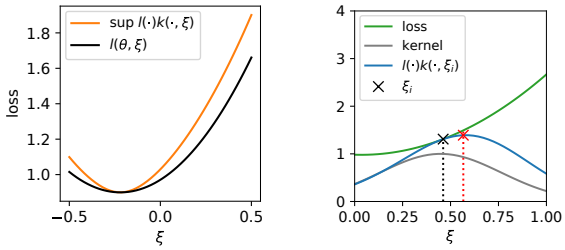

Figure 1: (**left**) Loss landscape of the kernel robust smoothed loss $l^k := \sup_u\{l(u)k_\sigma(u,\cdot)\}$. As analyzed in the text, as the width σ decreases, the ARKS surrogate loss tends towards the original loss, i.e., $l^k \to l$ as $\sigma \to 0$. Note that the kernel-smoothed loss l^k is a majorant of the original loss l. (**right**) Illustration of the inner maximization problem of ARKS. This figure illustrates the mechanism that ARKS finds the adversarial example by kernel smoothing. The figure plots the original loss l in green. The inner objective (using the k-transform in Definition 2) is plotted in blue. The black cross is a sampled data point ξ_i. The red cross is the computed solution to the inner maximization problem of ARKS.

original expected loss. The worst-case adversarial perturbation is also plotted as a red cross (right). See the caption of Figure 1 and the appendix for more details.

5 Certifying distributional robustness

We now detail the theoretical guarantees for our ARKS algorithm. Our analysis is based on an insight on the connection between our robustification scheme using the k-transform (2) and OT. We first show that ARKS can be viewed as a robustification and convexification scheme in the log-transformed space. All proofs are given in the appendix.

5.1 Robustification in log scale

We rewrite the surrogate loss $l_\theta^k(\xi_i)$ of the inner optimization problem in (11) by simplying taking the log transform, obtaining

$$l_\theta^k(\xi_i) = \exp \sup_u \left\{ \ln l(u) - \frac{1}{\sigma} c(u, \xi_i) \right\}. \tag{12}$$

We exchanged the sup and exp above due to the monotonicity and continuity of the exponential function. Using the above relationship, we rewrite ARKS as the

equivalent optimization problem

$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} \exp\left\{ \widehat{\ln l_{1/\sigma}}(\theta, \xi_i) := \sup_{u} \left\{ \ln l(u) - \frac{1}{\sigma} c(u, \xi_i) \right\} \right\}, \quad (13)$$

where $\widehat{\ln l_{1/\sigma}}(\theta, \cdot)$ denotes the (negative) c-transform of the log-loss $\ln l(\theta, \cdot)$. The following lemma states that the inner maximization objective is concave for certain choices of the kernel bandwidth. Hence, standard analysis such as (Lin et al., 2021) applies to our setting.

A function f on $\mathbb{R}^d$ is said to be L-smooth if $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \forall x, y \in \mathcal{X}$, provided that all quantities exist.

Proposition 5.1 (Convexification)**.** *Suppose the function $\ln l(\theta, \cdot)$ is L-smooth, transport cost c is 1-strongly convex, and $\sigma < \frac{1}{L}$. Then, the inner maximization objective of* (13) *is strictly concave.*

In particular, Gaussian kernels with appropriately chosen bandwidths satisfy the assumptions in Proposition 5.1. The intuition is that, by moving the exponentiation outside, we see the convexification mechanism of ARKS more clearly: compared with Wasserstein DRO, ARKS can be viewed as applying the c-transform as robustification in log scale. The log-transform in (13) also gives us an intuition for choosing the kernel bandwidth σ using the practice in proximal algorithms (Moreau-Yosida regularization). Alternatively, (13) can also be seen as optimizing a (differentiable) *softmax* version of the worst-case loss $\sup_{\xi \in \mathcal{X}} \widehat{\ln l_{1/\sigma}}(\theta, \xi)$.

5.2 Certifying robustness under distribution shift using OT

So far, it is not immediately clear how we can produce a certificate for the amount of robustness against distribution shift, or how to measure the distribution shift, e.g., in what metric? This section exploits the connection of our proposed k-transform and the c-transform to provide answers to those questions.

Specifically, we bound the quantity when considering an arbitrary distribution P that differs from the true data-generating distribution P_0. We then show that the ARKS procedure (11) produces a robustness certificate in the setting of distribution shift.

In the following, we assume that the c-transform $\widehat{\ln l_{1/\sigma}}(\theta, \xi)$ (see (13)) is bounded, i.e., $\exists M > 0$ such that $|\widehat{\ln l_{1/\sigma}}(\theta, \xi)| \leq M, \forall \xi \in \mathcal{X}$. As we have already shown that $\widehat{\ln l_{1/\sigma}}$ is a *majorant* of $\ln l$, the assumption also implies $l(\theta, \xi) \leq e^M$. Intuitively, our analysis of ARKS certifies the robustness in log scale. Nonetheless, since the log function is monotone, we can still use OT distances to control the generalization under distribution shift. To ease the notation, put $r_{N,\delta} := \sqrt{\frac{\ln(1/\delta)}{n}}$.

Proposition 5.2 (Certifying robustness against distribution shift)**.** *For any θ pointwise, there exists a constant $C > 0$ such that, $\forall \rho > 0$, any probability measure P, and kernel bandwidth $\sigma > 0$, the following holds except probability δ:*

$$\sup_{\gamma(P, P_0) \leq \rho} \mathbb{E}_P \ \ln l(\theta, \xi) \leq \ln \left\{ \underbrace{\frac{1}{N}\sum_{i=1}^{N} l_\theta^k(\xi_i)}_{ARKS\ objective} \right\} + \frac{\rho}{\sigma} + C \cdot r_{N,\delta}, \quad (14)$$

where γ is the Wasserstein distance associated with transport cost c.

Furthermore, there exits a constant C' that does not depend on θ such that the following holds except probability δ:

$$\sup_{\gamma(P, P_0) \leq \rho} \mathbb{E}_P \ \ln l(\theta, \xi) \leq \ln \left\{ \frac{1}{N}\sum_{i=1}^{N} l_\theta^k(\xi_i) \right\} + \frac{\rho}{\sigma} + C' \cdot r_{N,\delta} + 2 \cdot \mathcal{R}_{\mathrm{N}}(\{\widehat{\ln l_{1/\sigma}}(\theta, \cdot) | \theta \in \Theta\}), \quad (15)$$

where $\mathcal{R}_{\mathrm{N}}$ denotes the Rademacher complexity.

The first two (non-diminishing) terms of the bound in (14) RHS, $\ln \left\{ \frac{1}{N}\sum_{i=1}^{N} l_\theta^k(\xi_i) \right\} + \frac{\rho}{\sigma}$, give us a *computable robustness certificate* under the distribution shift from P_0 to arbitrary P. This is in line with the robustness certificate of (Sinha et al., 2017; Lee and Raginsky, 2018), and different from typical statistical learning theory bounds. Furthermore, the robustness certificate is simply the log-transform of the ARKS objective plus the *regularization* term $\frac{\rho}{\sigma}$.

Remark. *The bound for Rademacher complexity of common function classes is a well-studied topic in statistical learning theory. It also follows the Lipschitz composition rule. For example, for model classes such as RKHS functions with bounded norms $\{f \in \mathcal{H} | \|f\|_{\mathcal{H}} \leq R\}$ for $R > 0$ and bounded kernels, the Rademacher complexity decays at the rate of $\frac{1}{\sqrt{N}}$. However, as mentioned earlier, we are interested in the function spaces that are more general and hence do not further expand on bounding the Rademacher term using specific spaces. We refer to more specialized texts such as (van der Vaart and Wellner, 2013) for more details and (Sinha et al., 2017) for a recent application to robustness certificate.*

Unifying DRO using OT and kernel methods Our analysis above establishes a non-trivial connection between two branches of DRO research using OT (e.g., (Mohajerin Esfahani and Kuhn, 2018; Sinha et al., 2017)) and kernel methods (Zhu et al., 2020; Staib and Jegelka, 2019). In the center stage is the kernel bandwidth parameter σ.

From the OT perspective, we see that larger σ in ARKS corresponds to smaller scaling parameters in the c-transform for OT, which is known to lead to more conservatism in Wasserstein DRO (since we down-weigh the transportation cost, resulting in larger ambiguity region).

On the other hand, and from the kernel perspective, the authors of (Zhu et al., 2020) use functional analysis arguments to characterize that conservative Kernel DRO can be a consequence of using small RKHSs as the dual spaces for DRO, which are associated with kernels with larger bandwidth σ.

In summary, through the bandwidth parameter σ, this paper unifies the DRO performance-robustness trade-off for both OT and kernel methods.

6 Numerical Experiments

In this section, we empirically demonstrate that ARKS can easily work with DNN models, which is a limitation of typical existing DRO reformulation techniques. Our selection of neural architectures is meant to demonstrate the algorithmic robustification effect instead of achieving state-of-the-art benchmarks. To that end, we also ablate factors known to influence robustness, such as dropout if not specified. More details on our experimental setup, hyper-parameter selection, and additional experimental results can be found in the appendix.

6.1 Robust Learning with DNNs

We compare the following algorithms when applicable: *(A)* ARKS Algorithm 1, *(B)* empirical risk minimization (ERM) and *(C)* WRM (Duchi et al., 2018), as well as *(D)* projected gradient descent (PGD) for training (Madry et al., 2019) (reported in the appendix since it is based on RO instead of DRO; We also refer to (Sinha et al., 2017) for extensive comparisons of PGD against WRM) and *(E)* (worst-case) robust optimization (Ben-Tal et al., 2009; Soyster, 1973) (reported in the appendix since it is not applicable for deep learning tasks). We do not test classical Wasserstein DRO algorithms, e.g. (Mohajerin Esfahani and Kuhn, 2018; Zhao and Guan, 2018; Shafieezadeh-Abadeh et al., 2019), since they cannot be applied to our test settings with general losses and DNN models. We further note that the classical type-2 Wasserstein DRO reformulation is equivalent to WRM with the optimal dual variable.

In our evaluation, the test data is perturbed with worst-case disturbances δ within a box $\{\delta : \|\delta\|_\infty \leq \Delta\}$. δ is generated by attacking the model trained with ERM (for each random seed) using the PGD algorithm. This type of attack is referred to as *black-box*. The experiment is also performed with black-box fast-gradient sign method (FGSM) (Goodfellow et al., 2015) attacks with respect to $\|.\|_\infty$, as well as white-box PGD attacks; the results are included in the appendix. We note that our focus is to demonstrate the robustification effect of the algorithm in a known environment instead of benchmarking various attacks exhaustively.

Fashion-MNIST (Xiao et al., 2017) with CNN. The top panel of Figure 2 shows the classification error for increasing perturbation magnitude Δ. We observe that ERM attains good performance when there is no perturbation but quickly underperforms as Δ increases. ARKS and WRM yield improved robustness while also achieving low test error under no perturbation. ARKS

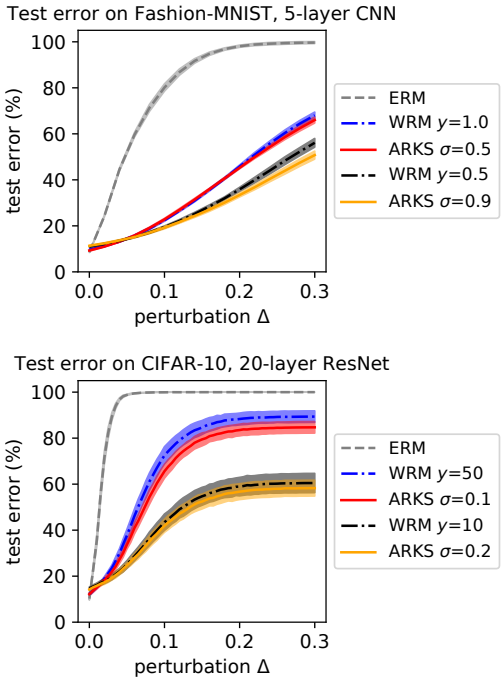

Figure 2: Black-box PGD attack with respect to $\|.\|_\infty$ on the Fashion-MNIST (**top**) and CIFAR-10 (**bottom**) datasets. We show the classification error on perturbed test images versus the allowed magnitude of the adversarial perturbation Δ. ARKS and WRM exhibit similar adversarial performance profiles; ARKS becomes more robust as the kernel width σ increases, while WRM improves with a lower Lagrangian penalty y. For all algorithms, we report the mean and standard deviation across 10 random seeds.

and WRM exhibit similar performance profiles; see the caption of Figure 2. To conclude, ARKS performs at least competitively with WRM.

CIFAR-10 (Krizhevsky et al., 2010) with ResNet-20. The above experiment is repeated for the CIFAR-10 dataset. We choose a deeper architecture, the ResNet-20 (He et al., 2015) with batch normalization (Ioffe and Szegedy, 2015) and ReLU activations. During training, we noticed that WRM might require tuning y to be arbitrarily large for stable performance under highly non-smooth losses, while σ can be easily tuned within a small range. The results are shown on the bottom panel of Figure 2: ARKS exhibits improved robustness under adversarial perturbations with little to no training performance sacrifice and is at least competitive with WRM.

CelebA (Liu et al., 2015) with CNN. Experimental results – similar to other datasets – can be found in the appendix, while Figure 3 illustrates examples of perturbed images generated during training.

Figure 3: **(top)** Perturbed images (u^*) maximizing the inner optimization of ARKS in CelebA binary classification. **(bottom)** Unperturbed counterpart. We observe that ARKS generates worst-case perturbations by creating interference around the eyes, reducing apparent separation between the two classes (with or without eye-wear).

7 Discussion

In this paper, we propose the ARKS algorithm using tools from convex analysis, kernel smoothing, and robust optimization. We have demonstrated state-of-the-art performance in benchmarks of learning under distribution shifts, especially with DNN models that can not be treated with typical Wasserstein DRO convex reformulation techniques. Furthermore, we have provided guarantees that certify robustness under distribution shift.

A future direction is to design specific kernels for robust learning, especially for incorporating rich descriptions of interventions in the real world beyond the typical norm-ball perturbation. For example, we can similarly design the transport cost in our c-exponential kernel (Definition 3) to be the data-dependent Mahalanobis distance to protect against distribution shift for causal inference as in, e.g., (Heinze-Deml and Meinshausen, 2021).

Acknowledgements

We thank the anonymous reviewers for their constructive comments during the review process. We also thank Simon Buchholz for sending us helpful feedback on the initial manuscript. This project received support from the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039B.

References

Arbel, M., Sutherland, D. J., Bińkowski, M., and Gretton, A. (2021). On gradient regularizers for MMD GANs. *arXiv:1805.11565 [cs, stat]*.

Bauschke, H. H. and Combettes, P. L. (2011). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, volume 408. Springer.

Ben-Tal, A., den Hertog, D., De Waegenaere, A., Melenberg, B., and Rennen, G. (2013). Robust Solutions of Optimization Problems Affected by Uncertain Probabilities. *Management Science*, 59(2):341–357.

Ben-Tal, A., den Hertog, D., and Vial, J.-P. (2015). Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming*, 149(1):265–299.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust Optimization*, volume 28. Princeton University Press.

Bietti, A., Mialon, G., Chen, D., and Mairal, J. (2019). A Kernel Perspective for Regularizing Deep Neural Networks. *arXiv:1810.00363 [cs, stat]*.

Billingsley, P. (1971). *Weak Convergence of Measures: Applications in Probability*. SIAM.

Blanchet, J., Murthy, K., and Zhang, F. (2018). Optimal transport based distributionally robust optimization: Structural properties and iterative schemes. *arXiv preprint arXiv:1810.02403*.

Blanchet, J. and Murthy, K. R. A. (2017). Quantifying Distributional Model Risk via Optimal Transport. *arXiv:1604.01446 [math, stat]*.

Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Chapelle, O., Weston, J., Bottou, L., Bottou, L. E., and Vapnik, V. (2001). Vicinal risk minimization. In

Advances in Neural Information Processing Systems, pages 416–422. MIT Press.

Conway, J. B. (2019). *A Course in Functional Analysis*, volume 96. Springer.

De Plaen, H., Fanuel, M., and Suykens, J. A. K. (2020). Wasserstein Exponential Kernels. *arXiv:2002.01878 [cs, stat]*.

Delage, E. and Ye, Y. (2010). Distributionally Robust Optimization Under Moment Uncertainty with Application to Data-Driven Problems. *Operations Research*, 58(3):595–612.

Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5.

Duchi, J., Glynn, P., and Namkoong, H. (2018). Statistics of Robust Optimization: A Generalized Empirical Likelihood Approach. *arXiv:1610.03425 [stat]*.

El Ghaoui, L. and Lebret, H. (1997). Robust Solutions to Least-Squares Problems with Uncertain Data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064.

Gao, R. and Kleywegt, A. J. (2016). Distributionally Robust Stochastic Optimization with Wasserstein Distance. *arXiv:1604.02199 [math]*.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *arXiv:1412.6572 [cs, stat]*.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13:723–773.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Heinze-Deml, C. and Meinshausen, N. (2021). Conditional variance penalties and domain shift robustness. *Machine Learning*, 110(2):303–348.

Houska, B. and Diehl, M. (2013). Nonlinear robust optimization via sequential convex bilevel programming. *Mathematical Programming*, 142(1-2):539–577.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2018). Averaging weights leads to wider optima and better generalization. *CoRR*, abs/1803.05407.

Kantorovich, L. V. and Rubinshtein, S. G. (1958). On a space of totally additive functions. *Vestnik of the St. Petersburg University: Mathematics*, 13(7):52–59.

Kim, J. and Scott, C. D. (2012). Robust kernel density estimation. *The Journal of Machine Learning Research*, 13(1):2529–2565.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kolter, Z. and Madry, A. (2018). Adversarial Robustness - Theory and Practice. http://adversarial-ml-tutorial.org/.

Krizhevsky, A., Nair, V., and Hinton, G. (2010). Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 5(4):1.

Kuhn, D., Esfahani, P. M., Nguyen, V. A., and Shafieezadeh-Abadeh, S. (2019). Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning. In Netessine, S., Shier, D., and Greenberg, H. J., editors, *Operations Research & Management Science in the Age of Analytics*, pages 130–166. INFORMS.

Lee, J. and Raginsky, M. (2018). Minimax Statistical Learning with Wasserstein Distances. *arXiv:1705.07815 [cs, stat]*.

Lin, T., Jin, C., and Jordan, M. I. (2021). On Gradient Descent Ascent for Nonconvex-Concave Minimax Problems. *arXiv:1906.00331 [cs, math, stat]*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2019). Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv:1706.06083 [cs, stat]*.

Meinshausen, N. (2018). Causality from a distributional robustness point of view. In *2018 IEEE Data Science Workshop (DSW)*, pages 6–10.

Mohajerin Esfahani, P. and Kuhn, D. (2018). Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166.

Müller, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443.

Namkoong, H. and Duchi, J. C. (2016). Stochastic Gradient Methods for Distributionally Robust Optimization with f-divergences. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 2208–2216. Curran Associates, Inc.

Namkoong, H. and Duchi, J. C. (2017). Variance-based regularization with convex objectives. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ning, C. and You, F. (2018). Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods. *Computers & Chemical Engineering*, 112:190–210.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer Science & Business Media.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Peyré, G. and Cuturi, M. (2019). Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Pólik, I. and Terlaky, T. (2007). A Survey of the S-Lemma. *SIAM Review*, 49(3):371–418.

Santambrogio, F. (2015). *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*. Progress in Nonlinear Differential Equations and Their Applications. Birkhäuser Basel.

Scarf, H. (1958). A min-max solution of an inventory problem. *Studies in the mathematical theory of inventory and production*.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press, Cambridge, MA, USA.

Shafieezadeh-Abadeh, S., Esfahani, P. M., and Kuhn, D. (2015). Distributionally Robust Logistic Regression. *arXiv:1509.09259 [math, stat]*.

Shafieezadeh-Abadeh, S., Kuhn, D., and Esfahani, P. M. (2019). Regularization via Mass Transportation. *Journal of Machine Learning Research*, 20(103):1–68.

Shapiro, A. (2001). On Duality Theory of Conic Linear Problems. In Pardalos, P., Goberna, M. Á., and López, M. A., editors, *Semi-Infinite Programming*, volume 57, pages 135–165. Springer US, Boston, MA.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. (2014). *Lectures on Stochastic Programming: Modeling and Theory*. SIAM.

Sinha, A., Namkoong, H., Volpi, R., and Duchi, J. (2017). Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*.

Soyster, A. L. (1973). Technical Note—Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming. *Operations Research*, 21(5):1154–1157.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., and Lanckriet, G. R. G. (2012). On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599.

Staib, M. and Jegelka, S. (2019). Distributionally robust optimization and generalization in kernel methods. *arXiv:1905.10943 [cs, stat]*.

Tolstikhin, I. O., Sriperumbudur, B. K., and Schölkopf, B. (2016). Minimax estimation of maximum mean discrepancy with radial kernels. *Advances in Neural Information Processing Systems*, 29:1930–1938.

Tran, P. T. and Phong, L. T. (2019). On the convergence proof of amsgrad and a new version. *IEEE Access*, 7:61706–61716.

van der Vaart, A. and Wellner, J. (2013). *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer Science & Business Media.

Virmaux, A. and Scaman, K. (2018). Lipschitz regularity of deep neural networks: Analysis and efficient estimation. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 3835–3844. Curran Associates, Inc.

Wahba, G. (1990). *Spline Models for Observational Data*. SIAM.

Weed, J. and Bach, F. (2017). Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *arXiv:1707.00087 [math, stat]*.

Wendland, H. (2004). *Scattered Data Approximation*, volume 17. Cambridge university press.

Wong, E. and Kolter, Z. (2018). Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xu, H., Caramanis, C., and Mannor, S. (2009). Robustness and Regularization of Support Vector Machines. *Journal of machine learning research*, 10(7).

Yakubovich, V. A. (1971). S-procedure in nonlinear control theory. *Vestnik Leninggradskogo Universiteta, Ser. Matematika*, pages 62–77.

Zhao, C. and Guan, Y. (2018). Data-driven risk-averse stochastic optimization with Wasserstein metric. *Operations Research Letters*, 46(2):262–267.

Zhu, J.-J., Jitkrittum, W., Diehl, M., and Schölkopf, B. (2020). Kernel distributionally robust optimization. *arXiv preprint arXiv:2006.06981*.

Appendix: Kernel Robust Smoothing

Notation and background. Throughout the appendix, we will consider the cases where the inner suprema of minimax problems are attained. Without further specifications, we consider the default kernel choice to be the Gaussian RBF kernel $k_\sigma(u,x) = e^{-\|u-x\|_2^2/2\sigma}$ (or the Laplacian kernel) in the rest of the appendix. We suppress the kernel bandwidth σ when there is no ambiguity in the context.

A Proofs of theoretical guarantees

A.1 Proof of Proposition 5.1

Proof. The proof is an exercise of calculus, e.g., by using the Taylor expansion of $\ln l(u) - \frac{1}{\sigma}c(u,\xi_i)$ w.r.t. the variable u. □

A.2 Proof of Proposition 5.2

As preparation, we first establish a standard concentration result in this paper's context.

Lemma A.1 (Concentration)**.** *For any θ pointwise, there exists constant $C > 0$ such that, $\forall \rho > 0$, probability measure P, and kernel bandwidth $\sigma > 0$, the following holds except probability δ:*

$$\mathbb{E}_{P_0}\ \widehat{\ln l_{1/\sigma}}(\theta,\xi) \le \frac{1}{N}\sum_{i=1}^{N} \widehat{\ln l_{1/\sigma}}(\theta,\xi_i) + C\cdot r_{N,\delta}.$$

Proof. Since the function of interest $\widehat{\ln l_{1/\sigma}}(\theta,\cdot)$ satisfies the bounded difference condition by assumption, the lemma statement follows directly from the McDiarmid's inequality. □

We now prove Proposition 5.2.

Proof. By the strong duality of DRO using Wasserstein distance (see, e.g., Mohajerin Esfahani and Kuhn (2018); Gao and Kleywegt (2016); Zhao and Guan (2018)) and the c-transform notation, we have, $\forall \sigma > 0, \rho > 0, \theta \in \Theta$,

$$\sup_{\gamma(P,P_0)\le\rho} \mathbb{E}_P\ \ln l(\theta,\xi) = \inf_{\sigma>0}\left\{\mathbb{E}_{P_0}\ \widehat{\ln l_{1/\sigma}}(\theta,\xi) + \frac{\rho}{\sigma}\right\} \le \mathbb{E}_{P_0}\ \widehat{\ln l_{1/\sigma}}(\theta,\xi) + \frac{\rho}{\sigma}. \tag{16}$$

By Lemma A.1, for any fixed θ, the RHS above is bounded by

$$\frac{1}{N}\sum_{i=1}^{N}\widehat{\ln l_{1/\sigma}}(\theta,\xi_i) + \frac{\rho}{\sigma} + C\cdot r_{N,\delta} \le \ln \frac{1}{N}\sum_{i=1}^{N}\exp\left\{\widehat{\ln l_{1/\sigma}}(\theta,\xi_i)\right\} + \frac{\rho}{\sigma} + C\cdot r_{N,\delta},$$

where the inequality is due to the concavity of the log function. Noting the quantity inside the logarithm is simply the ARKS objective (see, e.g., (13)), we obtain the first half of the proposition statement.

We now show the uniform convergence. Put

$$F_N := \sup_{\theta\in\Theta} \mathbb{E}_{P_0}\ \widehat{\ln l_{1/\sigma}}(\theta,\xi) - \frac{1}{N}\sum_{i=1}^{N}\widehat{\ln l_{1/\sigma}}(\theta,\xi_i).$$

Using symmetrization (see, e.g., van der Vaart and Wellner (2013)), we have

$$\mathbb{E}_{P_0}\ F_N \le 2\,\mathcal{R}_{\mathrm{N}}(\{\widehat{\ln l_{1/\sigma}}(\theta,\cdot)|\theta\in\Theta\}).$$

By McDiarmid, with $1-\delta$ probability,

$$F_N \le \mathbb{E}_{P_0}\ F_N + C'\cdot r_{N,\delta},$$

where C' does not depend on θ due to the sup operation. Combining the above relationships with (16) yields the uniform bound. □

B Additional technical details

B.1 Equivalence of type-1 Wasserstein DRO to IPM-DRO

Lemma B.1. *(Motivating example using type-1 Wasserstein DRO) Suppose the loss function $l(\theta, \cdot)$ is $y-$Lipschitz continuous. Let variable f in dual IPM-DRO* (6) *be set to the y-Pasch-Hausdorff envelope $f(\cdot) := \sup_u\{l(\theta, u) - y \cdot \|u - \cdot\|\}$. Then,* (6) *is equivalent to the dual formulation of type-*1 *Wasserstein DRO; cf. (Mohajerin Esfahani and Kuhn, 2018; Zhao and Guan, 2018; Kuhn et al., 2019).*

Unfortunately, estimating Lipschitz constants for general model classes is known to be difficult (Virmaux and Scaman, 2018; Bietti et al., 2019), resulting in the intractability of Wasserstein DRO when used with common machine learning models, e.g., neural networks, which our method in this paper can handle.

B.2 Proof of Lemma B.1

We now prove Lemma B.1. First, it is an exercise to show the following technical lemmas.

B.2.1 Technical lemma and proofs using convex analysis

Lemma B.2. *A function's y-Pasch-Hausdorff envelope dominates itself, i.e.,*

$$l_{y,1}(x) \geq l(x), \forall x \in \mathcal{X}.$$

Furthermore, $l_{y,1}$ is the smallest majorant of l with Lipschitz constant y.

Lemma B.3. *If l is Lipschitz-continuous with constant y, then $l_{y,1}$ coincides with l.*

Similar results concerning the *infimal convolution* (instead of supremal) are well-known (Bauschke and Combettes, 2011, Chapter 12). For completeness, we give self-contained proofs below. We assume the regularity condition that $f(x) := l_{y,1}(x) = \sup_u\{l(\theta, u) - y \cdot \|u - x\|\} < \infty$; we refer to (Bauschke and Combettes, 2011, Proposition 12.14) for the degenerative case when $l_{y,1} = \infty$ and l has no y-Lipschitz majorant.

We now prove Lemma B.2.

Proof. By noting the special choice of $u = x$, the relationship $f(x) \geq l(x)$ is obvious. We now prove the Lipschitz continuity.

For any x, z in the domain,

$$f(x) = \sup_u\{l(u) - y \cdot \|u - x\|\} \geq \sup_u\{l(u) - y \cdot \|u - z\| - y \cdot \|z - x\|\} = \sup_u\{l(u) - y \cdot \|u - z\|\} - y \cdot \|z - x\|$$
$$= f(z) - y \cdot \|z - x\|. \quad (17)$$

Therefore, $f(z) - f(x) \leq y \cdot \|z - x\|$, f is y-Lipschitz.

To show that f is the smallest y-Lipschitz majorant, we let g be any y-Lipschitz majorant of l. Then,

$$g(x) \geq g(z) - y \cdot \|z - x\| \geq l(z) - y \cdot \|z - x\|.$$

Take supremum on both sides,

$$g(x) \geq \sup_z\{l(z) - y \cdot \|z - x\|\} = f(x).$$

Hence, f is the smallest y-Lipschitz majorant. □

Lemma B.3 follows directly from Lemma B.2. See, e.g., Bauschke and Combettes (2011) Chapter 12 for more technical details on the convolution operator.

By plugging in the expression for $l_{y,1}$, we have

$$\frac{1}{N}\sum_{i=1}^{N} \sup_u\{l(u) - y \cdot \|u - \xi_i\|\} + \epsilon y \quad (18)$$

We have thus recovered the type-1 Wasserstein DRO dual as a special case of our analysis.

B.3 Proof of Proposition 4.1

We now verify the relationship $l^k(x) \geq l(x), \forall x \in \mathcal{X},$ and $l^k \to l$ as $\sigma \to 0$. The dominance relationship $l^k(x) \geq l(x)$ can be seen by taking the special case $u = x$ in the supremum. Finally, the convergence of $l^k \to l$ as $\sigma \to 0$ is obvious by examining the expression of the Gaussian RBF kernel and Laplacian kernel.

B.4 Proof of Proposition 4.2 (Robustness-performance trade-off using kernel width σ)

First, we note the continuity of the Gaussian RBF kernel and the loss function l; hence all limits are attained. If we let the kernel width be large $\sigma \to \infty$, then $\lim_{\sigma\to\infty} k(u, x) = 1$. Hence, the robust learning algorithm recovers the worst-case robust optimization (RO)

$$\min_{\theta} \sup_{\xi} l(\theta, \xi).$$

Similarly, if kernel width is small $\sigma \to 0$, then we recover the trivial Dirac function at limit $\lim_{\sigma\to 0} k(u, x) = \delta_x(u)$. Hence ARKS becomes the empirical risk minimization (ERM),

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} l(\theta, \xi_i).$$

B.5 Alternative analysis on convexity properties of inner optimization problem

We now provide an alternative view (to Proposition 5.1) of the convexity properties of the objective function of the inner objective of ARKS, which we denote as $f(u) := l(u)k(u, x)$. For ARKS, our intuition is that, by multiplying the loss $l(u)$ by a function $k(u, x)$ which is strongly concave near its peak, the resulting function is consequently locally concave too. This idea is illustrated in Figure 1. For conciseness, we assume that the loss function l is positive twice-differentiable (cf. Sinha et al. (2017) for why this is not restrictive), and x, u are scalars. We first show that the inner objective $f(u) = l(u)k(u, x)$ is locally concave in a neighborhood of x.

Proof. We compute the curvature $\frac{d^2}{du^2} f(u)$.

$$\begin{aligned}
\frac{d^2}{du^2} f(u) &= \frac{d}{du}\left(\frac{d}{du} l(u)k(u,x) + l(u)\frac{d}{du}k(u,x)\right), \\
&= \frac{d^2}{du^2} l(u)k(u,x) + 2\frac{d}{du} l(u)\frac{d}{du}k(u,x) + l(u)\frac{d^2}{du^2}k(u,x) \\
&= e^{-(u-x)^2/2\sigma}\left[\frac{d^2}{du^2} l(u) + 2\frac{d}{du} l(u)\left(-(u-x)/\sigma\right) + l(u)\left(-1/\sigma + (u-x)^2/\sigma^2\right)\right]. \quad (19)
\end{aligned}$$

Let us choose $\sigma > 0$ small enough such that the following holds.

$$\frac{d^2}{du^2} l(u) - l(u)/\sigma < 0. \quad (20)$$

This can be done trivially if the curvature of the loss l is bounded (similar to the assumptions in Sinha et al. (2017); Houska and Diehl (2013)) and $l(u) > 0$. Then, there exists $\Delta > 0$ such that, for $|u - x| \leq \Delta$, the curvature value (19) is negative. Therefore, the objective $f(u) = l(u)k(u, x)$ is concave in the $\Delta-$neighborhood of x. □

We now show that, for a suitable choice of σ, every stationary point of f is a local maximum, hence explaining the good empirical performance in our experiments. A full convergence analysis is out of the scope of our current paper.

Let

$$\sigma^* = \frac{2(u^* - x)^2}{\sqrt{1 + 4(u^* - x)^2 \cdot \frac{d^2}{du^2} l(u^*)/l(u^*)} - 1},$$

which is a non-negative quantity if $u^* \neq x$ and $\frac{d^2}{du^2} l(u^*) > 0$ by straightforward verification.

Lemma B.4. *Suppose either the loss l is concave or the bandwidth satisfies $\sigma < \sigma^*$. Then, every stationary point of f is a maximum.*

Proof. Suppose u^* is a stationary point of f, which implies

$$\frac{d}{du} f(u) \mid_{u=u^*} = \frac{d}{du} l(u) k(u,x) + l(u) \frac{d}{du} k(u,x) \mid_{u=u^*} = 0.$$

Since $k(u,x) \neq 0$, that further implies

$$\frac{d}{du} l(u) + l(u) \left(-(u-x)/\sigma\right) \mid_{u=u^*} = 0.$$

Plugging the above equality into the last line of (19),

$$\frac{d^2}{du^2} f(u) \mid_{u=u^*} = e^{-(u-x)^2/2\sigma} \left[\frac{d^2}{du^2} l(u) - l(u) \left(1/\sigma + (u-x)^2/\sigma^2\right) \right] \mid_{u=u^*} .$$

Since either $\frac{d^2}{du^2} l(u^*) \leq 0$ or $\sigma < \sigma^*$, we have

$$\frac{d^2}{du^2} l(u^*) - l(u^*) \left(1/\sigma + (u^*-x)^2/\sigma^2\right) < 0.$$

Then,

$$\frac{d^2}{du^2} f(u) \mid_{u=u^*} < 0.$$

Therefore, u^* is a local maximum by the second derivative test of calculus. □

Note that the condition $\sigma < \sigma^*$ can be easily satisfied when l has bounded curvature and is a weaker condition than (20). Therefore, the above lemma implies that a gradient-based algorithm converges to a maximum. Note that the analysis presented in Proposition 5.1 is stronger than the lemma above.

B.6 Additional function approximator and majorant constructions

We now use smooth majorants, as well as interpolants, to construct robustification methods in addition to the ARKS, e.g.,

1. Kernel distance envelope (KE)

$$f_{\sigma,y}(x) := \sup_u \{ l(\theta, u) - y \cdot (1 - k(u,x)) \}. \tag{21}$$

2. Kernel interpolant (KI; l denotes the vector of loss values at some interpolation points $[l(\theta,\xi_1), \ldots, l(\theta,\xi_M)]^\top$)

$$\hat{f}(x) = l^\top k(X,X)^{-1} k(X,x). \tag{22}$$

   While KI is not a strict majorant (since it only interpolates at data sites), we nonetheless show below it can enforce robustness.

Once we take the function approximation perspective, the possibility is by no means limited to those choices. For example, for the inverse multi-quadratic kernels, the approximation $\sup_u \{ l(\theta,u) - y[1/k^2(u,x) - C^2]) \}$ is equivalent to using the Moreau-Yosida regularization in type-2 Wasserstein DRO. The authors of (Zhu et al., 2020) used the RKHS basis expansion $\hat{f}(x) = \sum_{j=1}^M \alpha_j k(\zeta_j, x)$, for some discretization points ζ_j. Compared with their approach, our choices in (21) and ARKS are certified majorants of loss function l. We now examine the specific approximation schemes.

Kernel distance envelope (KE). We adopt a similar insight as Wasserstein DRO in (Sinha et al., 2017) and propose the function approximator, which is a variant of the Moreau-Yosida regularization with the kernel distance (using the Gaussian RBF kernel)

$$f_{y,\sigma}(x) = \sup_u \{l(\theta, u) - y/2 \cdot \|\phi(u) - \phi(x)\|_{\mathcal{H}}^2\} = \sup_u \{l(\theta, u) - y \cdot (1 - k(u, x))\}. \tag{23}$$

Similar to ARKS, one can verify

$$f_{y,\sigma}(x) \geq l(x), \forall x \in \mathcal{X}, \text{ and } f_{y,\sigma} \to l \text{ as } y \to 0,$$

i.e., it is a function majorant of l. Furthermore, $f_{y,\sigma}$ can be viewed as a $y-$Lipschitz continuous mapping from the feature space (i.e., the RKHS $\mathcal{H}$; see the appendix),

$$f_{y,\sigma}(x) - f_{y,\sigma}(z) \leq y\|\phi(x) - \phi(z)\|_{\mathcal{H}}, \ \forall x, z \in \mathcal{X}.$$

Analogous to ARKS, we can simply solve the optimization problem with a fixed y,

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sup_u \{l(\theta, u) - y \cdot (1 - k(u, \xi_i))\}. \tag{24}$$

Similar to (11), for suitable choices of y, σ, the inner function of (24) is locally concave in a neighborhood of ξ_i, facilitating gradient based optimization. Note that KE can again be interpreted as adversarial training like ARKS. The empirical performance of KE is similar to ARKS in our experiments. We thus left it as future work to examine the properties of KE in detail.

Kernel interpolant (KI). We now turn to the *kernel interpolant*, an entirely different scheme from ARKS. Our main idea in this section is to find a map $L^2 \to \mathcal{H}$ such that we can perform robust learning in $\mathcal{H}$. For any given θ, we choose the approximation function f to be the well-known kernel interpolant (Wahba, 1990) of the loss function

$$\hat{f} = l^\top k(X, X)^{-1} k(X, \cdot),$$

where l is defined in (22). This is also referred to as the kernel "ridge-less" regression estimator. Plugging this interpolant back into the IPM-DRO formulation, we arrive at the regularized risk minimization

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} l(\theta, \xi_i) + \epsilon \sqrt{l^\top k(X, X)^{-1} l}. \tag{25}$$

Intuitively, this can be seen as performing the following two steps simultaneously: *1)* interpolating the optimization loss l using kernel regression; and *2)* performing regularized risk minimization w.r.t. θ using the interpolant function's RKHS norm.

Alternatively, using the least-squares loss as an example, $l(\theta, [X, Y]) := (g_\theta(X) - Y)^2$, we may use f to interpolate the model g_θ only, resulting in

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} l(\theta, [x_i, y_i]) + \epsilon \sqrt{[(f(X) - Y)^2]^\top k(X, X)^{-1} [(f(X) - Y)^2]}, \tag{26}$$

where $f = g_\theta(X)^\top k(X, X)^{-1} k(X, \cdot)$. In practice, we may also choose to use a regularizer motivated by kernel ridge regression,

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} (g_\theta(X_i) - Y_i)^2 + \lambda g_\theta(X)^\top k(X, X)^{-1} g_\theta(X).$$

We refer to (Bietti et al., 2019; Xu et al., 2009; Staib and Jegelka, 2019) for more interpretations of RKHS norm regularization.

Remark. *While the above formulations, such as* (26)*, resemble the kernel ridge regression (KRR) estimator, they are not not the same. Our method can learn with either parametric or non-parametric models with loss $l(\theta, \cdot)$, while KRR only works with kernelized models. For example, we have report experiments with DNNs, which cannot be handled by KRR.*

C Experimental set-up and additional results

In this Section, we provide additional information on the numerical experiments presented in Section 6, and report supplementary material. This includes benchmark results with the PGD adversarial learning algorithm (Madry et al., 2019), and a brief discussion on its differences with ARKS. All of our experiments are conducted using the PyTorch (Paszke et al., 2019) and the CVXPY (Diamond and Boyd, 2016) libraries.

C.1 Robust learning under Adversarial Perturbations

Datasets. The numerical experiments in Section 6.1 make use of the following publicly available datasets: Fashion-MNIST[1] (Xiao et al., 2017), CIFAR-10[2] (Krizhevsky et al., 2010), and CelebA[3] (Liu et al., 2015).

The Fashion-MNIST dataset contains greyscale images of garments from 10 categories. Each image x is represented by $x \in [0,1]^{28\times 28}$.

The CIFAR-10 dataset contains colored images of different objects from 10 categories. Each image x is represented by $x \in [0,1]^{32\times 32\times 3}$. As is customary in such settings, we augment the training set with additional samples by randomly cropping and flipping images.

Using the provided attributes, the CelebA dataset is reduced to only contain a balanced number of colored images of celebrities with eye-wear (class 1) or without (class 0). Each image x is represented by $x \in [0,1]^{64\times 48\times 3}$. Our codebase includes a script to modify this dataset.

Model architectures. For Fashion-MNIST, the model architecture consists of two 3×3 convolutional layers with ELU activations and max pooling, followed by two fully connected layers and a softmax layer. For CIFAR-10 we use the ResNet-20 model architecture (He et al., 2015), consisting of 18 convolutional layers with batch normalization (Ioffe and Szegedy, 2015) and ReLU activations, followed by a fully connected and a softmax layer. For CelebA, the model architecture is borrowed from Heinze-Deml and Meinshausen (2021), comprised of four 5×5 convolutional layers with Leaky ReLU activations, followed by a fully connected and a softmax layer. The convolutions produce 16, 32, 64 and 128 channels respectively, using a stride of 2.

Hyper-parameter search. To optimize our algorithms, we performed grid search over the hyper-parameters outlined in Table 1. We first searched for the hyper-parameters of ERM that achieved the lowest classification error on unperturbed images independent of the training set, averaged across seeds $\{0, 10, 20\}$. The same procedure was then repeated for ARKS, WRM and PGD; the hyper-parameters that most closely recovered the lowest possible error given by ERM were selected.

We note that although WRM (5) requires for the Lagrangian relaxation coefficient $2y \geq L$, the Lipschitz constant of the loss gradient, L is hard to compute for our models. In order to provide a fair comparison to ARKS, we searched for the optimal y within a large range. Similarly, the PGD benchmark was tuned to defend against worst-case disturbances δ within a box $\{\delta : \|\delta\|_\infty \leq \Delta\}$. Δ was set to 0.3, the maximum magnitude of the disturbances considered during evaluation. For both the optimization of model weights and the inner optimization of worst-case perturbations, we consider common optimizers such as stochastic gradient descent (SGD), Adam (Kingma and Ba, 2014) and AMSGrad (Tran and Phong, 2019).

In the CIFAR-10 classification task, the learning rate is decayed by a multiplicative factor of 0.1 at steps 25 and 50. Additionally, to prevent ERM from overfitting and ensure a fair comparison across algorithms, they all use a weight decay of 0.0001 as per the authors of the ResNet architecture (He et al., 2015).

Adversarial attacks. In our evaluation, we perturb test images with worst-case disturbances δ within a box $\{\delta : \|\delta\|_\infty \leq \Delta\}$. We consider two types of adversarial attacks using this norm. Firstly, *black-box attacks* for which the disturbances are generated by attacking the model trained with ERM (for each random seed) using PGD and FGSM. Secondly, instead of evaluating each objective on ERM-adversarial loss, we perform *white-box attacks* using PGD on each model individually. In all evaluations, the perturbed images $x + \delta$ are clipped to the valid image range $[0, 1]$. PGD performs 15 iterations of gradient ascent on δ with a learning rate $\alpha = 0.03$ for

[1] available at `https://pytorch.org/vision/stable/datasets.html#fashion-mnist`
[2] available at `https://pytorch.org/vision/stable/datasets.html#cifar`
[3] available at `https://www.kaggle.com/jessicali9530/celeba-dataset`

Algorithm	Hyper-parameters	Fashion-MNIST	CIFAR-10	CelebA
ALL	training epochs	45	65	45
	batch size $\in \{64, 128, 256\}$	256	128	128
	optimizer $\in$ {SGD, AMSGrad, Adam}	AMSGrad	SGD	AMSGrad
	learning rate $\in [0.0001, 0.2]$	0.001	0.1	0.001
	decay learning rate $\in$ {True, False}	False	True	False
ARKS, WRM, PGD	inner optimization epochs	15	15	15
	inner optimizer $\in$ {SGD, AMSGrad, Adam}	AMSGrad	AMSGrad	AMSGrad
	decay inner learning rate $\in$ {True, False}	False	False	False
ARKS	inner learning rate $\in [0.0001, 0.1]$	0.01	0.001	0.002
	kernel bandwidth $\sigma \in [0.01, 10]$	0.5	0.1	0.2
WRM	inner learning rate $\in [0.0001, 0.1]$	0.05	0.001	0.002
	Lagrangian coefficient $y \in [0.01, 1000]$	1.0	50	4.0
PGD	inner learning rate $\in [0.0001, 0.1]$	0.001	0.0001	0.0005

Table 1: Hyper-parameter configuration for classification tasks on the Fashion-MNIST, CIFAR-10 and CelebA datasets. For clarity, we indicate a range for large sets of hyper-parameter values.

Fashion-MNIST, and $\alpha = 0.02$ for CIFAR-10 and CelebA. FGSM performs one iteration of gradient ascent by design.

Supplementary results on black-box attacks. We repeat Figure 2 with the PGD adversarial training algorithm and for FGSM attacks with respect to $\|.\|_\infty$. The results for Fashion-MNIST are shown on Figure 4, and for CIFAR-10 on Figure 5. The same procedure is also applied to binary classifiers trained on CelebA face images, with the results shown on Figure 6.

Across all tests, we see that ERM offers the least robustness. This is expected for an optimistic statistical estimator that underestimates risk and is a well-known fact in stochastic optimization (Shapiro et al., 2014). We emphasize that we included the comparison with the PGD benchmark for completeness. In reality, ARKS is only directly comparable with WRM since they are DRO approaches while PGD is based on RO, as we have discussed in the main text. We do not intend to show PGD to be less robust than ARKS and WRM since the robustness of DRO and RO depends on the choices of uncertainty and ambiguity sets.

ARKS and WRM exhibit similar adversarial profiles, with ARKS offering slightly more robustness as the magnitude of the adversarial perturbations increase. We use the hyper-parameter values outlined in Table 1, but also include ARKS with a higher σ and WRM with a lower y (but otherwise optimal hyper-parameter values), exhibiting improved robustness for a small sacrifice on classifying unperturbed images. Further increasing σ or decreasing y would increase this test-time penalty. However WRM would rapidly become unstable. To use WRM for deep networks such as ResNet, y needs to be tuned to a high value in order to prevent instabilities from the propagation of the input perturbation through the network.

Supplementary results on white-box attacks. In previous experiments, models trained with each learning objective were evaluated on the same set of perturbed images generated by black-box PGD attacks on the models trained with ERM. In this experiment, each model is evaluated on perturbations generated by white-box PGD $\|\|\|_\infty$ attacks on itself. The left panel of Figure 7 shows the evaluation results for Fashion-MNIST, and the right panel for CIFAR-10.

C.2 Comparison with the work of Madry et al. (2019)

In this section, we briefly contrast ARKS against the work of Madry et al. (2019) that introduces PGD, a common adversarial learning algorithm. However, we emphasize that ARKS is best compared to WRM, a state-of-the-art

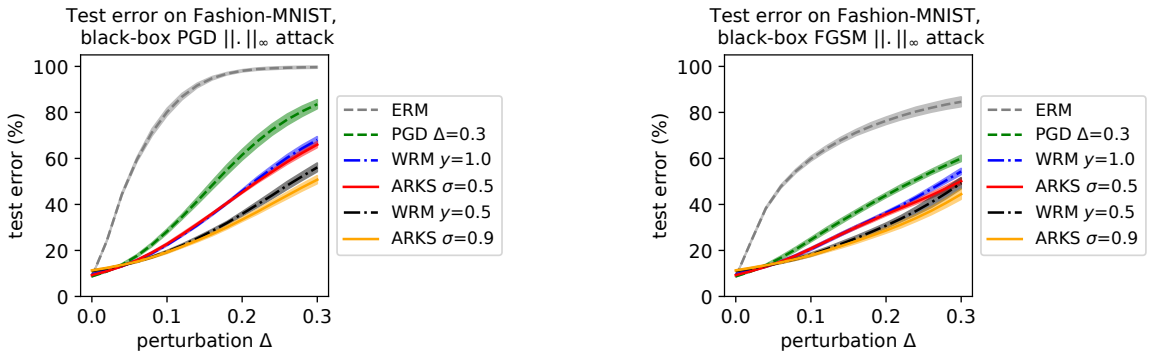

Figure 4: Black-box PGD (**left**) and FGSM (**right**) attacks with respect to $\|.\|_\infty$ on the Fashion-MNIST dataset. We show the classification error on perturbed test images versus the allowed magnitude of the adversarial perturbation Δ. For all algorithms, we report the mean and standard deviation across 10 random seeds.

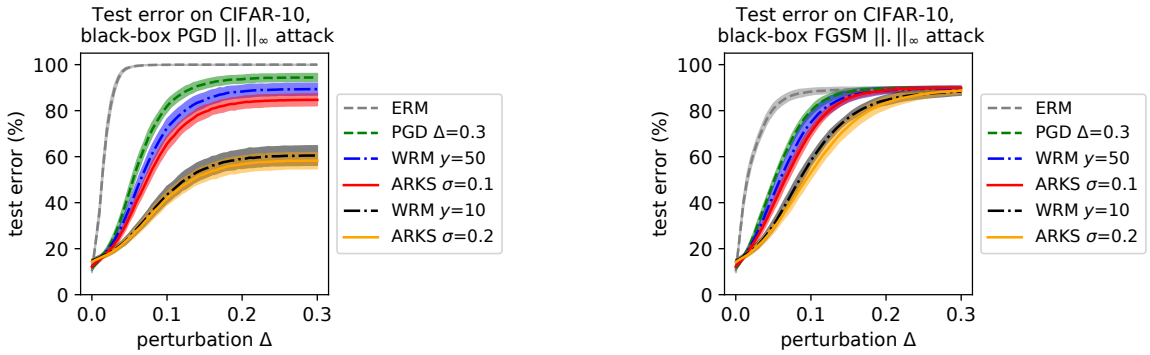

Figure 5: Black-box PGD (**left**) and FGSM (**right**) attacks with respect to $\|.\|_\infty$ on the CIFAR-10 dataset. We show the classification error on perturbed test images versus the allowed magnitude of the adversarial perturbation Δ. For all algorithms, we report the mean and standard deviation across 10 random seeds.

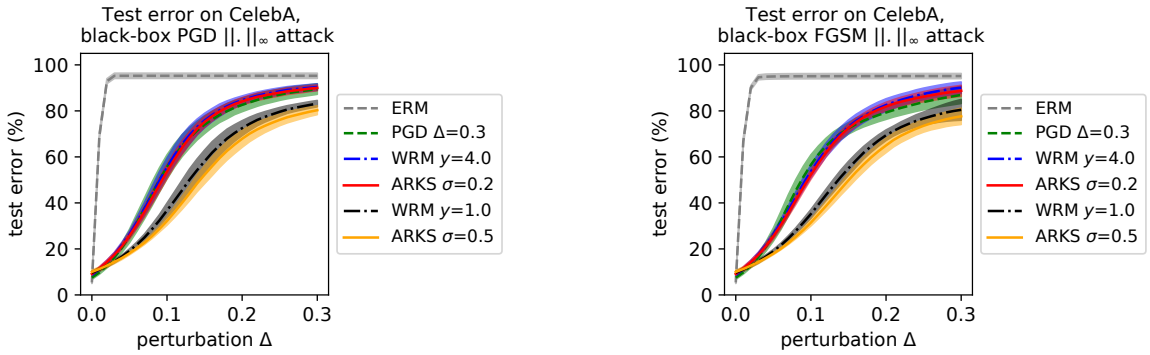

Figure 6: Black-box PGD (**left**) and FGSM (**right**) attacks with respect to $\|.\|_\infty$ on the reduced CelebA dataset. We show the classification error on perturbed test images versus the allowed magnitude of the adversarial perturbation Δ. For all algorithms, we report the mean and standard deviation across 10 random seeds.

method based on DRO, as PGD is based on RO. We refer to Sinha et al. (2017) for extensive comparisons of PGD with WRM.

Theoretically, as ARKS and WRM are derived using the strong duality of DRO, they are less conservative than RO and therefore PGD. This is reflected in robustifying against *imperceptible attacks* in the evaluations of Sinha et al. (2017). Computationally, the inner maximization of PGD is typically difficult in the case of non-convex

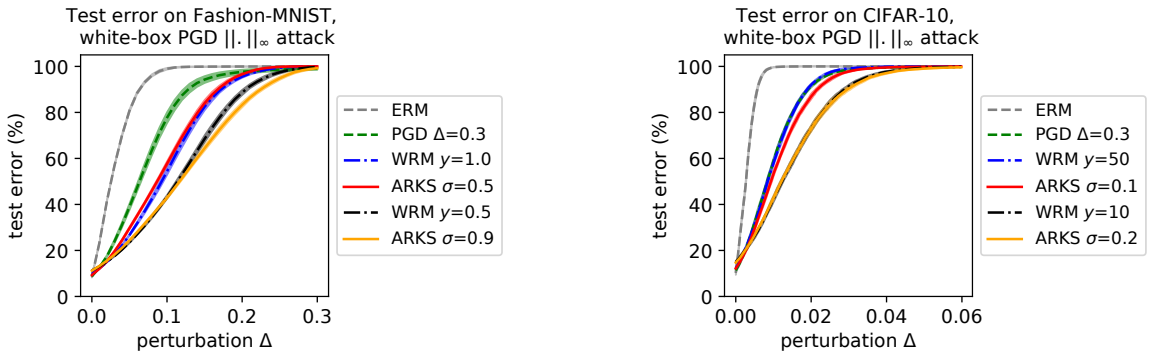

Figure 7: White-box PGD attack with respect to $\|.\|_\infty$ on the Fashion-MNIST (**left**) and CIFAR-10 (**right**) datasets. We show the classification error on perturbed test images versus the allowed magnitude of the adversarial perturbation Δ. For all algorithms, we report the mean and standard deviation across 10 random seeds. On the right plot, the curves for WRM $\gamma = 50$ and PGD almost coincide.

losses, while ARKS and WRM both apply convexification (see our discussion in Section 3) and smoothing. For instance, the projected gradient step is prone to get stuck in local optima.

Furthermore, ARKS makes contributions in areas that substantially differ from the main idea of Madry et al. (2019). Notably, ARKS sheds light on using kernel smoothing theory to robustify deep neural networks. Kernel methods are conventionally not scalable, but our experiments show that ARKS can scale in practical adversarial learning tasks. While previous attempts in robust kernel density estimation exist, the underlying robust kernel methods are not adversarial to specific loss functions nor scalable. ARKS also opens up a new lane of designing kernels for robustness and causal inference. The current work only tested with the default Gaussian kernel, whose empirical performance is already competitive. Finally, our work develops new functional analysis theory for robustness, which does not exist in (Madry et al., 2019).

C.3 Further analysis of ARKS

In the toy problem for Figure 1, we followed the set-up from (Zhu et al., 2020) for the robust least-squares example, which appeared in (El Ghaoui and Lebret, 1997; Boyd et al., 2004). We formulate the optimization problem $\min_\theta \|A(\xi) \cdot \theta - b\|_2^2$, where $A(\xi)$ is assumed to be uncertain and given by $A(\xi) = A_0 + \xi A_1$, where $-1 \leq \xi \leq 1$ is an uncertain variable. We refer to (Zhu et al., 2020) for more details.

The ERM solution to the robust least-squares problem is computed by solving a convex program. The RO solution is obtained by solving the SDP reformulation in (El Ghaoui and Lebret, 1997). Additional visual insights, such as a comparison of our approach with the ERM and RO solution, a comparison of empirical and adversarial distribution, are highlighted in Figure 8. We refer to the caption for more details.

For ARKS, we solved the program (11) using stochastic gradient descent ascent (GDA): in each iteration, we sample a mini-batch $\{\xi_i\}$, then performed gradient ascent to maximize the inner objective of (11) w.r.t. the inner variable u. In practice, we performed 10 steps inner gradient ascent using the L-BFGS routine. The inner maximization problem is illustrated in Figure 1. Following that, we took one outer gradient descent step w.r.t. the decision variable θ, and repeated the loop. See Algorithm 1 for more details. The outer optimization problem is solved using the L-BFGS optimization routine in PyTorch, with a learning rate of 1.

C.4 Results for additional models and data sets

In addition to the linear model in the RLS example and the previously reported benchmarks on CIFAR-10, Fashion-MNIST, and CelebA datasets, we report other results using ARKS with a smaller neural network model. We used a multi-layer perceptron with two fully connected hidden layers, with 32 hidden units for each layer. The multi-layer perceptron (MLP) uses the ELU activation because of its smoothness property. We trained 5 independent models for every setting and use stochastic weight averaging (Izmailov et al., 2018) for all neural network training. We report the results in Figure 9 and the caption therein. For exact hyperparameter

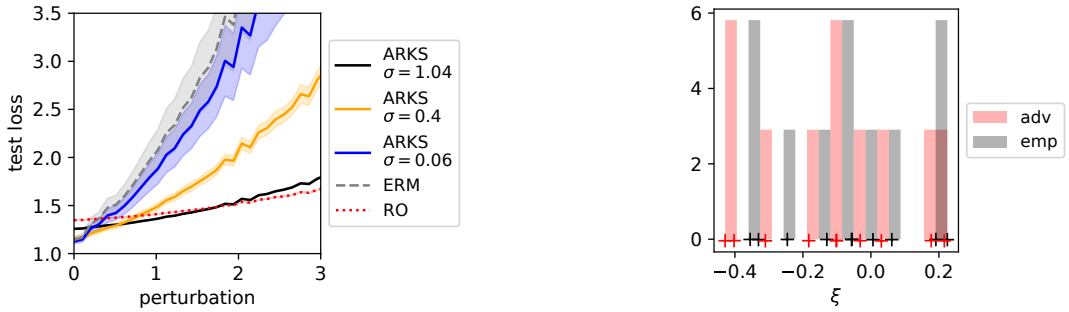

Figure 8: (**left**) We plot the performance-robustness trade-off of ARKS for various width settings (black, yellow, blue). We create settings of perturbed test distribution (different from the training data distribution, with the random variables satisfying $X_{\text{test}} = (1+\delta) \cdot X_{\text{train}}$), with increasing amounts of distribution shift parameter δ. We compare with ERM and the worst-case robust optimization (RO) solution of (El Ghaoui and Lebret, 1997). We see that ARKS with large width σ is more robust and conservative, tending towards RO. When width σ is small, ARKS achieves better performance but less robust under a large distribution shift. Overall, ARKS performs as Proposition 4.2 indicates, achieving a balance of moderate performance and robustness between ERM and RO. For every algorithm, we ran train 10 independent models. The error bars are in standard errors. (**right**) Histogram density estimation with $\sigma = 0.41$ (as used in ARKS) for both the empirical data (black) and the perturbed (adversarial) points (red). The closed-form MMD estimator (Gretton et al., 2012) between the samples and the adversarial samples evaluates to MMD $= 0.167 \pm 0.02$, averaged over 10 independent runs.

configurations of the MLP training, consult Table 2.

Algorithm	Hyper-parameters	Diabetes	Iris
ERM & ARKS	batch size	256	128
	optimizer	Adam	SGD
	learning rate	0.001	0.1
	epochs	2000	2000
ARKS	inner optimizer	L-BFGS	L-BFGS
	inner learning rate	1	1
	inner epochs	10	10

Table 2: Hyperparameter configurations for experiments using a multi-layer perceptron as model

We report results on the diabetes regression dataset [4] and the iris plants classification dataset [4]. To test the robustness property of the methods, we add the perturbation to the test data samples using the following rule

$$X_{\text{perturbed}} = X_{\text{test}} + d \cdot \text{Uniform}(-1, 1).$$

We increase the perturbation magnitude d from 0 to 1. The results are reported in Figure 9.

[4] available at `https://scikit-learn.org/stable/datasets/toy_dataset.html`

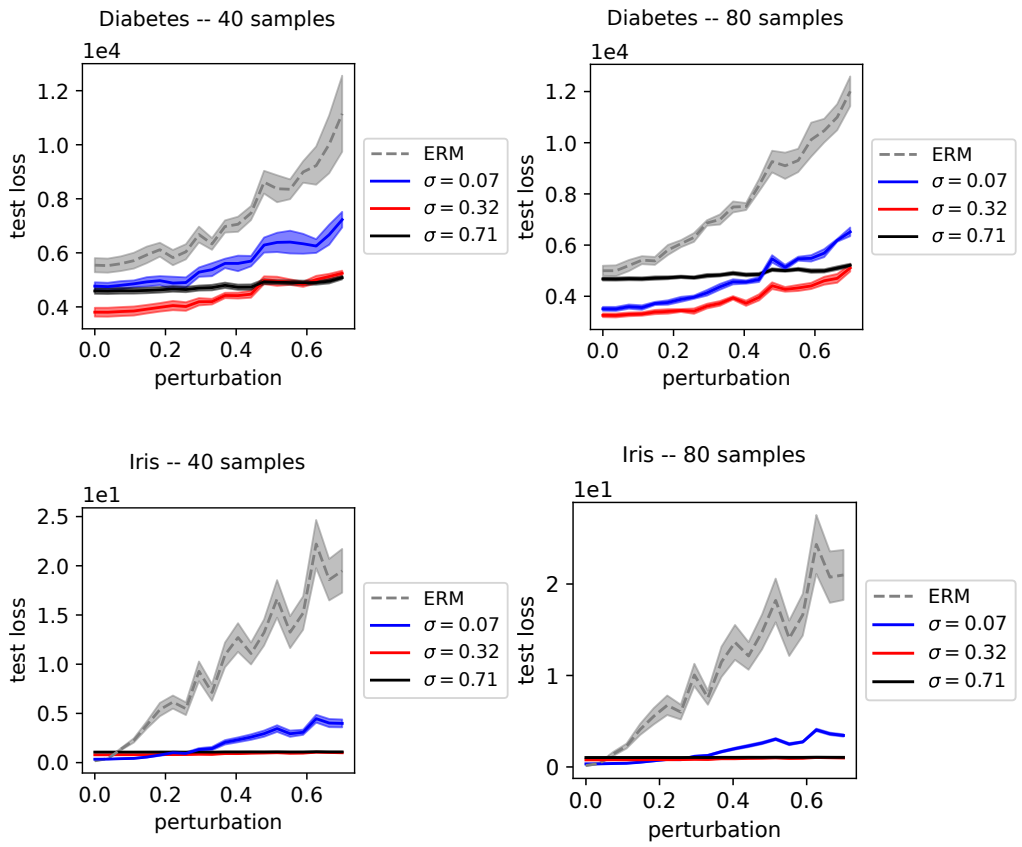

Figure 9: We trained the neural network model with the ARKS algorithm. We compare the results with the ERM solutions. The top-left figure shows model evaluations trained with 40 samples from the diabetes dataset; the top-right figure corresponds to 80 training samples from the diabetes dataset. The bottom-left figure shows model evaluations trained with 40 samples from the iris plant dataset and the bottom-right figure for 80 training samples from the same dataset. Across all figures, we observe that the ERM performance degrades as the perturbation of the test data increases. By contrast, and as expected, ARKS has better robustness against the distribution shift. For smaller kernel width σ, the curve approaches the ERM solution. With increasing kernel widths, the ARKS solution becomes more robust but is also more conservative. Note that the curves are the mean test errors; the error bars denote the standard errors