C. A. G. Trillos, N. G. Trillos *Res Math Sci* (2022) 9:54
https://doi.org/10.1007/s40687-022-00349-9

Research in the Mathematical Sciences

RESEARCH

On the regularized risk of distributionally robust learning over deep neural networks

Camilo Andrés García Trillos[1] and Nicolás García Trillos[2*]

*Correspondence: garciatrillo@wisc.edu
[2]Department of Statistics, University of Wisconsin-Madison, Madison, USA
Full list of author information is available at the end of the article

Abstract

In this paper, we explore the relation between distributionally robust learning and different forms of regularization to enforce robustness of deep neural networks. In particular, starting from a concrete min-max distributionally robust problem, and using tools from optimal transport theory, we derive first-order and second-order approximations to the distributionally robust problem in terms of appropriate regularized risk minimization problems. In the context of deep ResNet models, we identify the structure of the resulting regularization problems as mean-field optimal control problems where the number and dimension of state variables are within a dimension-free factor of the dimension of the original unrobust problem. Using the Pontryagin maximum principles associated with these problems, we motivate a family of scalable algorithms for the training of robust neural networks. Our analysis recovers some results and algorithms known in the literature (in settings explained throughout the paper) and provides many other theoretical and algorithmic insights that to our knowledge are novel. In our analysis, we employ tools that we deem useful for a future analysis of more general adversarial learning problems.

1 Introduction

What is the connection between adversarial learning and regularized risk minimization? This is a question of theoretical and practical relevance that aims at linking two different approaches to enforce robustness in learning models. By an adversarial learning problem, here we mean a *distributionally robust optimization* (DRO) problem of the form:

$$\inf_{\theta\in\Theta} \sup_{\tilde{\mu}:G(\mu_0,\tilde{\mu})\leq\delta} J(\tilde{\mu},\theta), \tag{1.1}$$

where θ denotes the parameters of a statistical learning procedure (for example a neural network, a binary classifier, or the parameters of a linear regression), μ_0 denotes an observed data distribution on $\mathbb{R}^d$, G represents some notion of "distance" between data distributions, $J(\tilde{\mu},\theta)$ is a risk relative to some data distribution $\tilde{\mu}$ and a loss function $j(x,\theta)$, and finally δ is a parameter that describes the "power" of an adversary. On the other hand, by *regularization* we mean an optimization problem of the form

$$\inf_{\theta\in\Theta} J(\mu_0,\theta) + \lambda R(\theta), \tag{1.2}$$

where $\lambda > 0$ is a positive parameter and R is a regularization functional.

The association between these two types problems has been particularly satisfactory in classical statistical parametric learning settings; see [4,5,9] and references within. Roughly speaking, if θ in (1.1) is a finite-dimensional vector representing the parameters of a generalized linear model, $J(\mu,\theta) = \int j(x,\theta)d\mu$, where j is a loss function such as square loss or logistic loss, and the function G is an appropriate Wasserstein distance, then one can show that the family of problems (1.1) coincides with a family of Lasso objectives that includes the popular squared-root Lasso model from [2].

In more general learning settings, and in particular in the setting explored in this paper, there is no direct equivalence between the adversarial problem (1.1) and regularization. There are, however, other settings where one can *exploit* the structure of the given learning model to derive *specific* insights into the type of regularization associated to (1.1). For example, in [18], in the context of nonparametric binary classification, a connection between adversarial learning and regularization is explored by establishing geometric evolution equations that must be satisfied by the ensemble of solutions to the family of adversarial problems (1.1) indexed by δ. In general, an illuminating strategy that can be followed in a variety of settings in order to gain insights into the regularization counterpart of (1.1) is to analyze the max part of the problem for small δ and identify its leading-order terms to use them as regularization terms. This is a strategy that has been followed in many works that study the robust training of neural networks, e.g., [17,25,27,29,30,40]; see more discussion in subsequent sections. However, even with an approximation in hand, the specific structure of the resulting regularization problems will depend on the specific learning models under consideration.

Our goal in this paper is to provide a concrete mathematical connection between a family of distributionally robust learning problems and regularization problems in the context of deep neural network models, and specifically ResNet models. Our analysis provides new theoretical insights into new and existing methods for robust training of neural networks, suggests new algorithms, and revisits older algorithms that can be recast as specific instances of a general unifying family. Our work also suggests new forms of regularization for optimal control problems that are meaningful beyond the applications to machine learning. The main motivation for working with ResNet models is that there is a clear interpretation of truly deep ResNet models (formally, the number of layers is infinity): in the large number of layers limit, the training of a ResNet may be interpreted as a continuous time optimal control problem. This ODE perspective for understanding and training neural networks has received increased attention in the past few years—see [10,20,32]. The specific structure of this setting will then allow us to recognize the resulting regularization problems as mean-field control problems and thus will motivate us to derive their corresponding Pontryagin maximum principles. In turn, these maximum principles can be used to motivate a large class of algorithms for the training of robust networks which includes the double backpropagation algorithm from [12]. The use of Pontryagin maximum principle-based training algorithms has been advocated for in works like [24] given their generality, versatility, and theoretical properties.

In the next two sections, we introduce the specific setup that we will work with throughout the paper. Our main theoretical results are presented in Sect. 1.3 and our algorithms in Sect. 4.

1.1 Network models

As discussed in the introduction, we will focus our discussion on deep ResNet neural networks and use the differential equation in $\mathbb{R}^d$

$$\begin{cases} dX_t = f(X_t, \theta_t)dt, \quad t \in (0, T), \\ X_0 = x \end{cases} \tag{1.3}$$

to model the transformations that an input data point $x \in \mathbb{R}^d$ undergoes along a deep neural network; notice that, with the previous interpretation, X_T is the output of the network when the input is x. The function $\theta : [0, T] \to \Theta_0$ represents the parameters of the network, and Θ is a family of θs. The "time" variable t can be interpreted as index for the layers of the model and the time horizon T as the depth of the network. Θ_0 represents the possible values of parameters at a given layer. We remark once again that a ResNet model found in practice can be seen as a time discretization of (1.3); see [14,32].

Example 1.1 The previous general setting can be used for regression with the following interpretation. We write $x = (v, y) \in \mathbb{R}^{d-1} \times \mathbb{R}$ and interpret v as feature vector and y as label or output. The function $f(\xi, \vartheta)$ can be taken to be

$$f(\xi, \vartheta) = \begin{pmatrix} \sigma(\vartheta \cdot \xi_{1:d-1}) \\ 0 \end{pmatrix},$$

interpreting ϑ as a $(d-1) \times (d-1)$ matrix, and σ as a nonlinear function (e.g., a sigmoid or ReLu) that acts coordinate-wisely on $d-1$-dimensional vectors. Notice that if we write $X_t = (V_t, Y_t)^\top$, then $Y_t = y$ for every t and in particular $X_T = (V_T, y)^\top$.

We introduce two functions $\ell : \mathbb{R}^d \times \Theta_0 \to \mathbb{R}$ and $\Phi : \mathbb{R}^d \times \Theta_0$ which from the control theory perspective can be interpreted as terminal and running costs, respectively. For $x \in \mathbb{R}^d$ and $\theta \in \Theta$ we define:

$$j(x, \theta) := \ell(X_{x,T}, \theta_T) + \int_0^T \Phi(X_{x,t}, \theta_t)dt.$$

In the above, we have used the notation $X_{x,t}$ to represent the solution to (1.3) with the extra subscript highlighting the initial condition x. The value of $j(x, \theta)$ can be interpreted as the loss (including the extra penalization) that the network with parameters θ incurs into when x is the network's input.

Example 1.2 (Continuation of Example 1.1) In the setting considered in Example 1.1 we can take Φ to be either identically equal to zero or be a function penalizing the size of the parameters only. As for the terminal cost, we can take

$$\ell(\xi, \vartheta) = (\vartheta \cdot \xi_{1:d-1} - \xi_d)^2,$$

this time interpreting ϑ as a $d-1$-dimensional vector, i.e., Θ_0 is a subset of $\mathbb{R}^{d-1}$. With this choice we obtain $\ell(X_{x,T}, \theta_T) = (\theta_T \cdot V_{x,T} - y)^2$, i.e., squared loss.

For a given probability distribution μ_0 and for a control $\theta \in \Theta$ we define the risk

$$J(\mu_0, \theta) := \mathbb{E}_{x \sim \mu_0}\left[j(x, \theta)\right],$$

and consider a so-called mean-field control problem (see [14]):

$$\inf_{\theta \in \Theta} J(\mu_0, \theta). \tag{1.4}$$

For us, μ_0 represents the training data distribution which at this stage can be simply assumed to be an empirical measure; problem (1.4) is then a risk minimization problem relative to the training distribution μ_0. We have made the dependence of problem (1.4) on the training data set μ_0 explicit as our goal is precisely to study its sensitivity to changes in the training input as will become more apparent in the next section when we introduce our adversarial learning problem precisely. We remark that in order to rigorously connect the optimization problem characterizing the training of a ResNet model with finitely many layers with the idealized continuous time control problem model considered here one needs to use variational techniques as discussed in [32].

1.1.1 Further notation

- $x, \tilde{x}$ represent vectors in $\mathbb{R}^d$ and will be used to denote inputs of the neural network.
- We use D to denote a matrix of derivatives of a vector valued function whereas we use ∇ to denote the gradient of a scalar valued function. D^2 is reserved to indicate matrices/tensors of second-order derivatives of scalar/vectorial functions. By convention, we identify the derivative of $A : \mathbb{R}^d \to \mathbb{R}^{c_1 \times \cdots \times c_k}; x \mapsto A(x)$ with a tensor of size $(c_1, \ldots, c_k, d)$.
- ξ represents a vector in $\mathbb{R}^d$ and ϑ an element in Θ_0. They will be used as dummy variables for the functions ℓ, Φ, and f. In particular, $\nabla_\xi \Phi$ and $\nabla_\vartheta \Phi$ represent the vector of derivatives of Φ with respect to ξ and ϑ, respectively.
- The matrix $D_\xi f$ has coordinates $[D_\xi f]_{ij} = \frac{\partial}{\partial \xi_j} f_i$, where $(f_1, \ldots, f_d)$ are the coordinate functions of f. We write the tensor $D^2_\xi f$ in coordinates as

  $$[D^2_\xi f]_{ijk} = \frac{\partial^2 f_i}{\partial \xi_j \partial \xi_k}.$$

  Notice that $[D^2_\xi f]_{ijk} = [D^2_\xi f]_{ikj}$. The tensor $(D^2_\xi f)^\top$ is defined as $[(D^2_\xi f)^\top]_{ijk} := [(D^2_\xi f)]_{kij}$. Finally, the tensor $D^3_\xi f$ is defined in coordinates as

  $$[D^3_\xi f]_{ijkl} = \frac{\partial^3 f_i}{\partial \xi_j \partial \xi_k \partial \xi_l},$$

  and we define $[(D^3_\xi f)^\top]_{ijkl} := [(D^3_\xi f)]_{lijk}$.
- θ represents the weights of the "ideal" (continuous time) ResNet neural network.
- $\tilde{\mu}$ will generically represent a probability measure over the variable x and will typically be interpreted as a perturbation of the data distribution μ_0.

1.2 Adversarial learning

In this paper, we restrict our attention to the family of distributionally robust adversarial problems:

$$\inf_{\theta \in \Theta} \sup_{\tilde{\mu}: W_p(\mu_0, \tilde{\mu}) \leq \delta} J(\tilde{\mu}, \theta), \tag{1.5}$$

where W_p stands for the p-Wasserstein distance:

$$W_p(\mu, \tilde{\mu}) := \inf_{\pi \in \Gamma(\mu, \tilde{\mu})} \left\{ \int_{\mathbb{R}^d \times \mathbb{R}^d} c_p(x, \tilde{x}) d\pi(x, \tilde{x}) \right\}^{1/p}$$

defined for two probability measures $\mu, \tilde{\mu}$ over $\mathbb{R}^d$ (or over a compact subset of $\mathbb{R}^d$). The function $c_p : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the p-cost:

$$c_p(x, \tilde{x}) := \|x - \tilde{x}\|^p,$$

where $\|\cdot\|$ denotes an arbitrary norm in $\mathbb{R}^d$. The family of problems (1.5) is indexed by $p \in [0, \infty]$. We interpret the case $p = \infty$ as

$$W_\infty(\mu, \tilde{\mu}) = \min_{\pi \in \Gamma(\mu, \tilde{\mu})} \operatorname{ess\,sup}_\pi \{\|x - \tilde{x}\| \ : \ (x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d\},$$

while the case $p = 0$ as the total variation distance between measures. In the remainder, we will restrict our attention to the case $p \in [2, \infty]$.

Example 1.3 (Continuation of Example 1.2) While problem 1.5 with the cost c_p introduced earlier is meaningful in a regression setting, there are other adversarial settings of interest where for example adversaries are only allowed to perturb feature vectors and not labels. In that case, the cost c_r introduced earlier can be replaced with the closely related cost:

$$\hat{c}_p((v, y), (\tilde{v}, \tilde{y})) = \begin{cases} \|v - \tilde{v}\|^p, & \text{if } y = \tilde{y}, \\ \infty, & \text{else}. \end{cases}$$

The analysis that we present in the remainder of the paper adjusts easily to this cost function. We omit the details.

Remark 1.4 We notice that since the Wasserstein distances satisfy the relation $W_p \leq W_{p'}$ when $p \leq p'$, it is straightforward to see that the adversary in problem (1.5) is stronger than the analogous adversary when choosing p'. In particular, of all the adversaries indexed by p, the weakest one is the one corresponding to $p = \infty$.

Remark 1.5 Problem (1.5) can be equivalently reformulated as

$$\inf_{\theta \in \Theta} \sup_{\pi \in \mathcal{F}_{\mu_0, \delta}} J(\pi, \theta), \tag{1.6}$$

where $\mathcal{F}_{\mu_0, \delta}$ is the set of probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ satisfying $\int_{\mathbb{R}^d \times \mathbb{R}^d} c_p(x, \tilde{x}) d\pi(x, \tilde{x}) \leq \delta^p$ and $P_{1\sharp}\pi = \mu_0$ (where $P_{1\sharp}\pi$ denotes the marginal of π on the first coordinate); we abuse notation slightly and use $J(\pi, \theta)$ to denote $\int_{\mathbb{R}^d \times \mathbb{R}^d} j(\tilde{x}, \theta) d\pi(x, \tilde{x})$. In other words, by replacing the variable $\tilde{\mu}$ with the variable π, the nonlinear constraint in $\tilde{\mu}$ gets replaced by linear constraints in π.

Remark 1.6 In an alternative formulation of (1.5), one can replace the constraint on $\tilde{\mu}$ with an explicit penalization:

$$\inf_{\theta \in \Theta} \sup_{\tilde{\mu}} \left\{ J(\tilde{\mu}, \theta) - \frac{1}{\lambda} W_p^p(\mu_0, \tilde{\mu}) \right\}, \tag{1.7}$$

for some $\lambda > 0$. Just as for the constrained problem, (1.7) admits an equivalent reformulation in terms of couplings:

$$\inf_{\theta \in \Theta} \sup_{P_{1\sharp}\pi = \mu_0} \left\{ J(\pi, \theta) - \frac{1}{\lambda} \int_{\mathbb{R}^d \times \mathbb{R}^d} c_p(x, \tilde{x}) d\pi(x, \tilde{x}) \right\}.$$

1.3 Regularized risk minimization and associated control problems

In order to make our results mathematically precise, we impose some extra conditions on all the terms that determine the min-max problem (1.5) in our setting.

Assumption 1.7

i. f is bounded; f, Φ are continuous in θ; and f, Φ, ℓ are continuously differentiable with respect to x. The first derivatives of f, Φ, ℓ are Lipschitz continuous in space uniformly in θ. Moreover, ℓ is continuously differentiable with respect to θ.
ii. The distribution μ_0 has bounded support in $\mathbb{R}^d$.

Assumption 1.8

i. Assumption 1.7 holds; and
ii. f, Φ, ℓ are twice continuously differentiable with respect to x.

Our first theoretical result is the following.

Theorem 1.9 (Regularization of distributionally robust adversarial learning: first-order case) *Let $p \in [2, \infty]$. Under Assumptions 1.8 the objective function*

$$\sup_{\tilde{\mu}: W_p(\mu_0, \tilde{\mu}) \leq \delta} J(\tilde{\mu}, \theta) \tag{1.8}$$

is equal to

$$J(\mu_0, \theta) + \delta \cdot \left(\mathbb{E}_{x \sim \mu_0}\left[\|\nabla_x j(x, \theta)\|_*^q\right]\right)^{1/q} + O(\delta^2),$$

where the $O(\delta^2)$ term is uniform over all $\theta \in \Theta$, q is p's conjugate, i.e., $\frac{1}{p} + \frac{1}{q} = 1$, and $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. In particular,

$$V_\delta^* - U_\delta^* = O(\delta^2),$$

where V_δ^* is equal to the infimum of (1.8) over all $\theta \in \Theta$ and U_δ^* is equal to

$$\inf_{\theta \in \Theta} \left\{ J(\mu_0, \theta) + \delta \cdot \left(\mathbb{E}_{x \sim \mu_0}\left[\|\nabla_x j(x, \theta)\|_*^q\right]\right)^{1/q} \right\}. \tag{1.9}$$

Remark 1.10 Theorem 1.9 implies that under the given assumptions, for any fixed control in the admissible domain, the value function of the robust problem and the value function of the regularized problem are close with an error of order δ^2. Hence, a minimizer for any of the two problems is a δ^2-minimizer for the other.

Remark 1.11 When $p = \infty$, problem (1.9) can be written more succinctly as

$$\min_{\theta \in \Theta} \left\{ J(\mu_0, \theta) + \delta \cdot \mathbb{E}_{x \sim \mu_0}\left[\|\nabla_x j(x, \theta)\|_*\right] \right\}. \tag{1.10}$$

Notice that the objective is linear in μ_0. This property is useful in connection to the use of stochastic gradient methods for training; see the discussion in Sect. 4. Problem (1.10) and other closely related problems that are linear in μ_0 and that penalize the gradient of the loss function have been considered in several works in the literature including [12,17,25,29,30,40].

It is important to highlight that Theorem 1.9 does not depend on the specific structure of neural network models and indeed continues to be true in other learning settings as long as the following assumption (implied by Assumption 1.8 in our neural network setting) holds:

Assumption 1.12 For every $\theta \in \Theta$ the function $j(\cdot, \theta)$ is twice differentiable with uniformly bounded second derivatives:

$$\|D_x^2 j(\cdot, \theta)\|_{L^\infty(\mathbb{R}^d)} \leq C,$$

where $C > 0$ is independent of $\theta \in \Theta$.

The interpretation of Theorem 1.9 is straightforward: the leading-order regularization effect of the adversarial problem (1.5) with p-cost is given by (1.9). Put in another way, (1.5) for small δ is an approximation to problem (1.9). Moreover, a minimizer of (1.9) is guaranteed to have value of (1.8) within $O(\delta^2)$ of the minimum. Notice that Problem (1.9) is a form of regularized risk minimization of the form (1.2) with regularization term given by

$$\left(\mathbb{E}_{x\sim\mu_0}\left[\|\nabla_x j(x,\theta)\|_*^q\right]\right)^{1/q},$$

only that in this case the regularization does depend on the data distribution μ_0. Intuitively, this new term forces the parameters of the model to be chosen so as to make the loss $j(x, \theta)$ insensitive to perturbations of the data (i.e., the gradient in x of the loss function should be small in the support of μ_0).

In our setting of interest, the regularized problem (1.9) possesses an interesting structure that will be convenient to elaborate on as it provides the basis for the algorithms for training robust neural networks discussed in Sect. 4. First, let us consider the control problem:

$$\begin{aligned} &\inf_{\theta\in\Theta} \quad \mathbb{E}_{x\sim\mu_0}\left[j(x,\theta)\right] \\ &\text{s.t.} \quad \begin{cases} dX_{x,t} = f(X_{x,t},\theta_t)dt, \quad t\in(0,T), \\ X_{x,0} = x, \end{cases} \end{aligned} \tag{1.11}$$

which is nothing but the (unrobust) training problem for the neural network; we have made the constraints in the above problem explicit to facilitate the comparison with the problems introduced below. For each $\theta \in \Theta$ and $x \in \mathbb{R}^d$, there is a corresponding dual variable $P : [0, T] \to \mathbb{R}^d$ associated to the ODE (1.3) which can be written as:

$$\begin{cases} dP_{x,t} = -\nabla_\xi H_0(X_{x,t},\theta_t,P_{x,t})dt, \quad t\in(0,T), \\ P_{x,T} = -\nabla_\xi \ell(X_{x,T},\theta_T), \end{cases} \tag{1.12}$$

where H_0 is the Hamiltonian:

$$H_0(\xi,\vartheta,\varrho) := \varrho\cdot f(\xi,\vartheta) - \Phi(\xi,\vartheta), \quad \xi\in\mathbb{R}^d, \quad \vartheta\in\Theta_0, \varrho\in\mathbb{R}^d.$$

$P_{x,0}$ is known to be equal to the negative gradient (in x) of the function $j(\cdot, \theta)$ when holding θ fixed (see the beginning of section 2.3 for a proof of this fact), that is, $P_{x,0}$ captures the sensitivity of $j(x, \theta)$ to perturbations in the input. This insight has been used in [42] to propose algorithms for the training of robust neural networks.

With this new interpretation, Problem (1.9) can be rewritten as a control problem:

$$\begin{aligned} &\inf_{\theta\in\Theta} \quad \left\{\mathbb{E}_{x\sim\mu_0}\left[j(x,\theta)\right] + \delta\cdot\left(\mathbb{E}_{x\sim\mu_0}\left[\|P_{x,0}\|_*^q\right]\right)^{1/q}\right\} \\ &\text{s.t.} \quad \begin{cases} dX_{x,t} = f(X_{x,t},\theta_t)dt, \quad t\in(0,T) \\ X_{x,0} = x \\ dP_{x,t} = -\nabla_\xi H_0(X_{x,t},\theta_t,P_{x,t})dt, \quad t\in(0,T) \\ P_{x,T} = -\nabla_\xi \ell(X_{x,T},\theta_T). \end{cases} \end{aligned} \tag{1.13}$$

Our second main result identifies the first-order optimality condition for this new control problem.

Theorem 1.13 *Let $p \in [2, \infty]$. Suppose that Assumption 1.8 holds and that $\|\cdot\|$ is the Euclidean norm. Suppose that (θ^*, X^*, P^*) is a minimizer for problem (1.13). Then, there exist absolutely continuous processes α^*, β^* such that for μ_0-a.e. x we have:*

$$\begin{aligned}\alpha^*_{x,t} &= -\nabla_\xi \ell(X^*_{x,T}, \theta^*_T) + D^2_\xi \ell(X^*_{x,T}, \theta^*_T)^\top \beta^*_{x,T} \\ &\quad + \int_t^T \{D_\xi f(X^*_{x,s}, \theta^*_s)^\top \alpha^*_{x,s} - \nabla_\xi \Phi(X^*_{x,s}, \theta_s)\} ds \\ &\quad + \int_t^T \{D^2_\xi \Phi(X^*_{x,s}, \theta^*_s) - D^2_\xi f(X^*_{x,s}, \theta^*_s)^\top P^*_{x,s}\}^\top \beta^*_{x,s} ds \end{aligned} \tag{1.14}$$

$$\beta^*_{x,t} = \delta \left(\mathbb{E}_{x_0 \sim \mu_0}\left[\|P^*_{x,0}\|^q\right]\right)^{-\frac{1}{p}} \|P^*_{x,0}\|^{q-2} P^*_{x,0} + \int_0^t D_\xi f(X^*_{x,s}, \theta^*_s) \beta^*_{x,s} ds, \tag{1.15}$$

and also

$$\theta^*_t \in \arg\max_{\vartheta \in \Theta_0} \left\{\mathbb{E}_{x \sim \mu_0}\left[H(X^*_{x,t}, P^*_{x,t}, \alpha^*_{x,t}, \beta^*_{x,t}, \vartheta)\right]\right\}, \tag{1.16}$$

where

$$H(\xi, \varrho, \alpha, \beta, \vartheta) := \alpha \cdot f(\xi, \vartheta) - \Phi(\xi, \vartheta) - \beta \cdot (D_\xi f(\xi, \vartheta)^\top \varrho - \nabla_\xi \Phi(\xi, \vartheta)) \tag{1.17}$$

is the Hamiltonian of problem (1.13).

Remark 1.14 It is worth highlighting that for the modified problem, the adjoint variable of X^* is *not* P^*, but rather α^*. Indeed, for this problem, P^* becomes part of the state variables, and has its own adjoint variable β^*. The Hamiltonian name for H is then justified since

$$\begin{aligned} dX^*_{x,t} &= \nabla_\alpha H(X^*_{x,t}, P^*_{x,t}, \alpha^*_{x,t}, \beta^*_{x,t}, \theta^*_t), \\ dP^*_{x,t} &= \nabla_\beta H(X^*_{x,t}, P^*_{x,t}, \alpha^*_{x,t}, \beta^*_{x,t}, \theta^*_t), \\ d\alpha^*_{x,t} &= -\nabla_\xi H(X^*_{x,t}, P^*_{x,t}, \alpha^*_{x,t}, \beta^*_{x,t}, \theta^*_t), \\ d\beta^*_{x,t} &= -\nabla_p H(X^*_{x,t}, P^*_{x,t}, \alpha^*_{x,t}, \beta^*_{x,t}, \theta^*_t). \end{aligned}$$

Remark 1.15 A similar result can be derived for more general norms, only that the expressions for the corresponding adjoint variables are more cumbersome.

The derived Pontryagin principle motivates a class of algorithms for training robust neural networks that are discussed in Sect. 4. The double backpropagation algorithm from [12] is a particular instance of this family of algorithms, which, at the moment it was proposed, was used to enhance the generalization properties of a neural network.

1.3.1 Second-order regularization

The results presented in the previous section can be developed further to include higher-order expansions for the function $J(\tilde{\mu}, \theta)$ under the assumption that the function $j(\cdot, \theta)$ is regular enough. However, unlike in the first-order case, an explicit higher-order expansion for (1.8) that does not involve any maximization problems is in general difficult to obtain unless one restricts to specific regimes for the size of the gradient function $\nabla_x j(\cdot, \theta)$ relative

to the size of the parameter δ. Nevertheless, restricting our attention to the case $p \geq 2$ and $\|\cdot\|$ the Euclidean norm, in Sect. 3 we can motivate the following optimization problems:

$$\min_{\theta\in\Theta}\left\{\begin{array}{l} J(\mu_0,\theta)+\delta\left(\mathbb{E}_{x\sim\mu_0}\left[\|\nabla_x j(x,\theta)\|^q\right]\right)^{1/q} \\ \qquad +\dfrac{\delta^2}{2}\left(\mathbb{E}_{x\sim\mu_0}\left[\|\nabla_x j(x,\theta)\|^q\right]\right)^{-2/p}\left(\mathbb{E}_{x\sim\mu_0}\left[\dfrac{\nabla_x j^\top D_x^2 j\nabla_x j}{\|\nabla_x j\|^{2(1-1/(p-1))}}(x,\theta)\right]\right)\end{array}\right\}, \tag{1.18}$$

$$\min_{\theta\in\Theta}\left\{J(\mu_0,\theta)+\frac{\delta^2}{2}(\mathbb{E}_{x\sim\mu_0}\left[(\lambda_{max}(x,\theta))_+^{\tilde{q}}\right])^{1/\tilde{q}}\right\}, \tag{1.19}$$

where $\tilde{q}$ is the conjugate of $p/2$, i.e., $\frac{1}{\tilde{q}}+\frac{2}{p}=1$. Both of these optimization problems can be regarded as second-order regularized risk minimization problems stemming from the adversarial learning problem (1.5), the difference between them being the relative size expected for the gradients of the loss function as compared to δ. As in Remark 1.15, similar second-order problems can be motivated for more general norms $\|\cdot\|$, but we skip the details for concreteness.

It is important to highlight that problems (1.18) and (1.19) are closely connected to other problems in the literature that have been used to train robust neural networks, most prominently the *curvature regularization* problem introduced in [27]. To draw a closer connection between what we do here and what is done in [27], notice that, as for the first-order case, when $p=\infty$ problems (1.18) and (1.19) are linear in μ_0 and read, respectively,

$$\min_{\theta\in\Theta}\left\{J(\mu_0,\theta)+\delta\mathbb{E}_{x\sim\mu_0}\left[\|\nabla_x j(x,\theta)\|\right]+\frac{\delta^2}{2}\mathbb{E}_{x\sim\mu_0}\left[\frac{\nabla_x j^\top D_x^2 j\nabla_x j}{\|\nabla_x j\|^2}(x,\theta)\right]\right\}, \tag{1.20}$$

$$\min_{\theta\in\Theta}\left\{J(\mu_0,\theta)+\frac{\delta^2}{2}\mathbb{E}_{x\sim\mu_0}\left[(\lambda_{max}(x,\theta))_+\right]\right\}. \tag{1.21}$$

Problem (1.21) is closely related to an optimization problem introduced in [27], which effectively uses a regularization term of the form:

$$\mathbb{E}_{x\sim\mu_0}\left[\|D_x^2 j(x,\theta)\|^2\right]$$

(where the matrix norm in the above expectation is the operator norm) instead of the quadratic term in (1.21). Notice that in (1.21) curvature is only penalized when it is positive. This is reasonable as directions with positive curvature are precisely those that an adversary can use to increase the value of the loss function.

Problem (1.20) can also be motivated by considerations discussed in [27]. Indeed, according to [27], in settings of interest involving the use of neural networks (see Remark 3 in [27]) the inner product between the eigenvector corresponding to the maximum eigenvalue of the Hessian matrix $D_x^2 j(x,\theta)$ and the *sign gradient* direction are found to have a large inner product, suggesting that this direction is (almost) parallel to the direction of largest curvature. Note that the quadratic term in (1.20) is precisely the effect of the second derivative along unitary vectors in the direction of the gradient and coincides with the (1-homogeneous) ∞-*Laplacian* of the function $j(\cdot,\theta)$. The works [15,22] also suggest that gradient directions are directions where the loss function $j(\cdot,\theta)$ is highly curved. From these observations, it is thus reasonable to consider the second-order regularization term that appears in (1.20).

As in the first-order case, we can also study the optimal control structure that the objectives in (1.18) and in (1.19) posses. Similarly to the first-order robustness case, we

write the control problem in a different form by expressing the second-order part in terms of dual variables.

Let us consider first the second-order problem (1.18). We can readily rewrite it in terms of the adjoint variables of the first-order expansion.

Proposition 1.16 *Under the same assumptions as Theorem 1.13, Problem (1.18) can be rewritten as*

$$\begin{aligned}
&\inf_{\theta\in\Theta} \left\{ \mathbb{E}_{x\sim\mu_0}\left[\ell(X_{x,T},\theta_T)\right] + \delta\cdot\left(\mathbb{E}_{x\sim\mu_0}\left[\|P_{x,0}\|^q\right]\right)^{1/q} + \frac{\delta}{2}\frac{\mathbb{E}_{x\sim\mu_0}\left[\|P_{x,0}\|^{q-2}\hat{\alpha}_{x,0}\cdot P_{x,0}\right]}{\left(\mathbb{E}_{x\sim\mu_0}\left[\|P_{x,0}\|^q\right]\right)^{1/p}} \right\} \\
&s.t. \quad \begin{cases} X, P \text{ as in} (1.13) \\ \beta \text{ as in Theorem} 1.13 \\ d\hat{\alpha}^*_{x,t} = -D_\xi f(X_{x,s},\theta_s)^\top \hat{\alpha}_{x,s} + \{D^2_\xi f(X_{x,s},\theta_s)^\top P_{x,s}\}^\top \beta_{x,s} \\ \hat{\alpha}_{x,T} = D^2_\xi \ell(X_{x,T},\theta_T)^\top \beta_{x,T}. \end{cases}
\end{aligned} \tag{1.22}$$

The main advantage of writing this case of second-order problem as in (1.22) is that this writing avoids the need to keep track of the whole matrix of second derivatives $D^2_x j(x,\theta)$ which is usually expensive to calculate when the dimension of the problem is large, i.e., $O(d^2)$ as opposed to the cost for keeping track of the gradient only which is $O(d)$).

Remark 1.17 Thanks to the linearity of the adjoint variable α in Theorem 1.13 it is possible to write

$$\alpha_{x,t} = P_{x,t} + \hat{\alpha}_{x,t},$$

so that we could have introduced directly $\hat{\alpha}$ as the adjoint variable of interest.

Remark 1.18 Note that there is no mistake in the powers of δ in the objective function in (1.22): the adjoint variable $\hat{\alpha}$ is already scaled by δ via the initial condition in β.

Another advantage of the formulation based in control variables is that, as in the first-order case, we can use the optimal control tools to deduce a Pontryagin principle: the adjoint variables $\beta, \hat{\alpha}$ are added to the list of primal variables, and new adjoint variables ϕ, π, λ, ψ are found. In this case, the system will be composed by eight variables (four primal and four adjoint), all having the same dimension d. Once more, this shows the complexity of the training problem for this second-order case has increased by 'only' a factor of 2 (i.e., a factor independent of dimension).

As an example, we state, without proof, Pontryagin's principle for the problem (1.16) in the case $p = \infty$.

Theorem 1.19 *Suppose that Assumption 1.8 holds, and further that f, ℓ are three-times continuously differentiable with respect to x. Assume also that $(\theta^*, X^*, P^*, \alpha^*, \beta^*)$ is an optimal minimizer for problem (1.22) (with $p = \infty$). Then, there exist absolutely continuous processes $\phi^*, \pi^*, \lambda^*, \psi^*$ such that for μ_0-a.e. x we have:*

$$\begin{aligned}
d\phi^*_{x,t} &= -D_\xi f(X^*_{x,t},\theta^*_t)^\top \phi^*_{x,t} + \{D^2_\xi f(X^*_{x,t},\theta^*_t)^\top P^*_{x,t}\}^\top \pi^*_{x,t} \\
&\quad + \{D^2_\xi f(X^*_{x,t},\theta^*_t)^\top \alpha^*_{x,t}\}^\top \lambda^*_{x,t} - \{D^2_\xi f(X^*_{x,t},\theta^*_s)^\top \beta^*_{x,t}\}^\top \psi^*_{x,t} \\
&\quad - (\{D^3_\xi f(X^*_{x,t},\theta^*_t)^\top P^*_{x,t}\}^\top \beta^*_{x,t})^\top \lambda^*_{x,t} dt \\
d\pi^*_{x,t} &= D_\xi f(X^*_{x,t},\theta^*_t)\pi^*_{x,t} - \{(\beta^*_{x,t})^\top (D^2_\xi f(X^*_{x,t},\theta^*_t))^\top\}^\top \lambda^*_{x,t} dt,
\end{aligned}$$

$$d\lambda^*_{x,t} = D_\xi f(X^*_{x,t}, \theta^*_t)^\top \lambda^*_{x,t} dt,$$
$$d\psi^*_{x,t} = -\{D^2_\xi f(X^*_{x,t}, \theta^*_t)^\top P^*_{x,t}\}^\top \lambda^*_{x,t} - D_\xi f(X^*_{x,t}, \theta^*_t)\psi^*_{x,t} dt$$

with boundary conditions

$$\phi^*_{x,T} = P^*_{x,T} + D^2_\xi \ell(X^*_{x,T}, \theta^*_T)\pi^*_{x,T} - \{D^3_\xi \ell(X^*_{x,T}, \theta^*_T)\beta^*_{x,T}\}^\top \lambda^*_{x,T}$$
$$\pi^*_{x,0} = \frac{1}{\|P^*_{x,0}\|}\left(P^*_{x,0} + \frac{\delta}{2}\alpha^*_{x,0} - \delta\psi^*_{x,0}\right) - \frac{\delta}{\|P^*_{x,0}\|^3} P^*_{x,0} \cdot \left(\frac{1}{2}\alpha^*_{x,0} - \psi^*_{x,0}\right) P^*_{x,0}$$
$$\lambda^*_{x,0} = \frac{\delta}{2}\frac{P^*_{x,0}}{\|P^*_{x,0}\|}$$
$$\psi^*_{x,T} = -D^2_\xi \ell(X^*_{x,T}, \theta^*_T)^\top \beta^*_{x,T}$$

and also

$$\theta^*_t \in \arg\max_{\vartheta \in \Theta_0} \left\{\mathbb{E}_{x\sim\mu_0}\left[\bar{H}(X^*_{x,t}, P^*_{x,t}, \alpha^*_{x,t}, \beta^*_{x,t}, \phi^*_{x,t}, \pi^*_{x,t}, \lambda^*_{x,t}, \psi^*_{x,t}, \vartheta)\right]\right\},$$

where

$$\begin{aligned}\bar{H}(\xi, \varrho, \alpha, \beta, \varphi, \varpi, \lambda, \Psi, \vartheta) := \varphi^\top f(\xi, \vartheta) - \varrho^\top D_\xi f(\xi, \vartheta)\varpi - \alpha^\top D_\xi f(\xi, \vartheta)\lambda \\ + \lambda^\top (D^2_\xi f(\xi, \vartheta)^\top \varrho)^\top \beta + \beta^\top D_\xi f(\xi, \vartheta)\psi\end{aligned} \tag{1.23}$$

is the Hamiltonian of problem (1.22)

Remark 1.20 Algorithm 2 in Sect. 4 is an optimization algorithm based on the Pontryagin principle presented in Theorem 1.19 for the second-order regularization problem (1.20). Ignoring the δ term in the objective, and considering $j(x, \theta) = \ell(X_{x,T}, \theta_T)$, the resulting problem is related to the curvature regularization algorithm studied in [27]. In what follows, we describe a difference and a similarity between the iterative schemes that we present here and the scheme presented in [27] for the analogous problem.

First, notice that the term $D^2_x j \nabla_x j$, which appears in the objective in (1.20), can be interpreted as the derivative of $\nabla_x j$ in the direction $z = \nabla_x j$. It is clear that z depends on θ and thus it is reasonable (and more accurate) to track this dependence when optimizing over θ, as we do in our schemes. Notice that our adjoint equations precisely contain the information to carry out this "chain rule" computation. In contrast, at each iteration of the algorithm in [27] the value of z gets fixed using the control θ from the previous iteration before proceeding to take a gradient step in the parameters of the network. In fact, in [27] a finite difference approximation $\frac{1}{h}\|\nabla_x j(x + hz, \theta) - \nabla_x j(x, \theta)\|$ for the second derivative is considered, as opposed to the explicit computation $D^2 z$. Our scheme based on Pontryagin's principle does not incur in a higher computational cost than the algorithm in [27].

On the other hand, our schemes based on Pontryagin principles share with that in [27] the fact that data points are never updated during training, in contrast to other works like [19,42] where, at each iteration of their algorithms, data get modified and then used as *the* data for one step of gradient descent in the parameters of the network; see the discussion in Sect. 2.2.

We can now consider the second-order problem (1.19). Unfortunately, it is not simple to express the maximum eigenvalue or its associated eigenvector in terms of a variable

having reasonable dynamics in terms of the problem's data without tracking the full second-order derivative $D_x^2 j(\theta, x)$. As discussed earlier, choosing such a path would be costly for high-dimensional problems.

A common idea to deal with dimensionality issues is to introduce random sampling: for instance, we can consider the action of the second derivative on a sample of directions taken uniformly from the unit ball. Indeed, using the fact that

$$(\lambda_{max}(x,\theta))_+ = \max_{z\in S_1^d}\{(\langle Az, z\rangle)_+\}, \tag{1.24}$$

where S_1^d denotes the surface of the unit ball in dimension d, we could be tempted to consider as candidate

$$\max_{i=1,\dots,m}\{(\langle AZ_i, Z_i\rangle)_+\}; \qquad \{Z_i\}_{i=1,\dots,m} \text{ i.i.d with } Z_1 \sim U(S_1^d).$$

However, one can easily deduce that this technique requires up to $O(\epsilon^{-d})$ samples to guarantee an error of $O(\epsilon)$ in situations where there is an important gap between the maximal eigenvalue and the remaining ones.

A modified version of the problem lends itself to a better reduction complexity through this approach. Noticing that the right-hand side of (1.24) can be understood as taking an $\ell^\infty(S_1^d)$ norm, we suggest replacing $(\lambda_{max}(x,\theta))_+$ with $\left(⨍_{S_1^d}(z^\top D_x^2 j(\theta,x)z)_+^b dz\right)^{1/b}$ for some power $b \geq 1$. Focusing on the case $b = 1$, we substitute (1.19) with

$$\min_{\theta\in\Theta}\left\{J(\mu_0,\theta) + \frac{\delta^2}{2}\left(\mathbb{E}\left[\left\{⨍_{S_1^d}(z^\top D_x^2 j(\theta,x)z)_+ dz\right\}^{\tilde{q}}\right]\right)^{1/\tilde{q}}\right\}. \tag{1.25}$$

Problem (1.25) can be seen as a *different* form of curvature penalization where the stress is put on reducing the overall positive curvature of the second derivative. In this sense, this problem gives less importance to potential worst cases than (1.19). Conveniently, though, it is better suited for an approximation using randomized directions.

Proposition 1.21 *Suppose Assumptions 1.8 and 1.12 hold. Let $\{z_i\}_{i=1,\dots,m}$ be i.i.d. samples with $z_1 \sim U(S_1^d)$ defined on a different probability space $(\Omega', \mathbb{P}', \mathcal{F}')$. Consider the problem*

$$\inf_{\theta\in\Theta}\left\{\mathbb{E}_{x\sim\mu_0}\left[\ell(X_{x,T},\theta_T)\right] + \frac{\delta^2}{2}\left(\mathbb{E}_{x\sim\mu_0}\left[\left\{\frac{1}{m}\sum_{i=1}^m(\rho_{x,0}^{z_i}\cdot z_i)_-\right\}^{\tilde{q}}\right]\right)^{1/\tilde{q}}\right\}$$

$$s.t. \quad \begin{cases} X, P \text{ as in} (1.13); \text{and} \\ d\gamma_{x,t}^i = D_\xi f\left(x_{x,t},\theta_t\right)\gamma_{x,t}^i dt, \\ d\rho_{x,t}^{z_i} = -D_\xi f\left(X_{x,t},\theta_t\right)^\top \rho_{x,t}^{z_i} + \left\{D_\xi^2 f\left(X_{x,t},\theta_t\right)^\top P_{x,t}\right\}^\top \gamma_{x,t}^i dt, \\ \gamma_{x,0}^i = z_i, \qquad \rho_{x,T}^{z_i} = D_\xi^2\ell\left(X_{x,T},\theta_T\right)\gamma_{x,T}^i \qquad \text{for } i = 1,\dots m. \end{cases} \tag{1.26}$$

and let $V^{\delta,m}$ be its optimal value. If V^δ is the optimal value in (1.25), then

$$V^{\delta,m} \to V^\delta \qquad \mathbb{P}' - a.s.$$

and $V^{\delta,m} - V^\delta$ satisfies a central limit theorem (so that the error is of order $m^{-1/2}$).

Problem (1.21) also has an associated Pontryagin maximum principle as the other control problems discussed thus far, only that we do not write it out explicitly for brevity. We remark, however, that the corresponding Pontryagin principle can be used to motivate

algorithms just as with the other problems discussed throughout this section. Notice that the number of state variables in problem (1.26) is $2m + 3$. This control problem is more advantageous than one that directly tracks the Hessian matrix $D^2 j(x, \theta)$ whenever $2m + 3$ is considerably smaller than d.

Remark 1.22 The variance of $V^{\delta,m} - V^{\delta}$ in proposition 1.21 depends on the dimension d. Consider, for example, the case $\tilde{q} = 1$, which corresponds to the case $p = \infty$ in the discussion in Sect. 1.3.1. First, notice that the terms $\rho^{z_i} \cdot z_i$ can be rewritten as $-z_i^\top D_x^2 j(\theta, x) z_i$. Now, given that the matrix $D_x^2 j(\theta, x)$ is real and symmetric, and given that the z_i are uniformly distributed on S_1^d, the spectral decomposition theorem can be used to find a linear growth of this variance with respect to the rank of the matrix, and *a fortiori*, at most linear with respect to d. The linear in d dependence of the variance estimate for fixed x and θ can be upgraded to uniform ones over all x and θ under smoothness assumptions on j. In terms of the sample size, this means that m would need to grow a bit faster than linearly to guarantee convergence from the central limit theorem. Compare with the complexity of using and keeping track of the whole Hessian (as in (1.25)), which is quadratic with respect to the dimension.

Problem (1.25) can be easily extended to other values of $b \geq 1$, with $b \to \infty$ recovering the original problem (1.19). Notice, however, that the randomization strategy for the approximation of the integral degenerates for larger values of b. While problem (1.19) is more closely connected to the original adversarial training problem, formulation (1.25) is superior from a computational perspective.

1.4 Other forms of regularization in the literature

In this section, we review some works in the literature connected to robust training of neural networks and discuss four different ways of introducing regularization penalties.

First, it is always possible to *explicitly penalize the parameters* of a neural network. For example, in [20,24] the loss function $j(x, \theta)$ in (1.11) is replaced with a loss function of the form:

$$\tilde{j}(\theta, x) = \ell(X_{x,T}, \theta_T) + \int_0^T \tilde{\Phi}(X_{x,t}, \theta_t, \dot{\theta}_t) dt,$$

where the running cost Φ explicitly penalizes the derivative of θ in time: this is one way to penalize parameters that is quite reasonable in the deep ResNet setting. A simple choice for $\tilde{\Phi}$ that allows us to draw a direct connection with the regularization problem (1.2) is

$$\tilde{\Phi}(\xi, \vartheta, u) := \Phi(\xi, \vartheta) + \lambda |u|^2,$$

for $\lambda > 0$. From a variational perspective, a regularization problem like the one described above admits optimal controls (parameters) in the classical sense. This contrasts with our problem (1.9) where in general optimal solutions have to be sought in the space of *generalized controls* a.k.a. *Young measures*; see [28].

There are papers that *explicitly penalize the input-to-output mappings* of a neural network. These are works that consider problems of the form:

$$\inf_{\theta \in \Theta} \{J(\mu_0, \theta) + R(g(X_{\cdot,T}))\},$$

where $R(\cdot)$ is some type of regularization functional (typically a seminorm like for example the Lipschitz seminorm) acting explicitly on the input-to-output map $x \mapsto g(X_{x,T})$; here, the function g is a simple function connected to the learning problem in hand (e.g.,

classification or regression) and that in general may depend on trainable parameters. The penalization of parameters via the regularity of their induced input-to-output maps has been considered in papers such as [16,21,36] where it is has also been shown that the resulting function objectives do enforce adversarial robustness (i.e., stability to data perturbations). The paper [33] considers this problem in the setting of graph-based learning.

Another approach to enforce robustness found in the literature is based on *perturbation-based regularization terms*. These are approaches based on the construction of adversarial examples around the observed data that can be used to define a new "perturbed" risk functional that is treated as regularizer. For example, [19] considers the problem

$$\inf_{\theta\in\Theta}\{\mathbb{E}_{x\sim\mu_0}[\tilde{j}(x,\theta)]\},\qquad \text{where } \tilde{j}(x,\theta) := \alpha j(x,\theta) + (1-\alpha) j(x+\delta\cdot\operatorname{sign}(\nabla_x j(x,\theta)),\theta), \tag{1.27}$$

for some $\delta > 0$ and some $\alpha \in [0,1)$. In the above, $x+\delta\cdot\operatorname{sign}(\nabla_x j(x,\theta))$ can be considered as an adversarial example. This specific way of constructing adversarial examples is known in the literature as the Fast Gradient Sign Method (FGSM) from [19]. Although this method has some practical shortcomings, it has been found in [38] that it provides quickly state-of-the-art level robustification results when coupled with a random perturbation of the data μ_0 within the ℓ^∞ ball considered.

Finally, objectives like the ones we presented in Sect. 1.3 have regularization terms that act explicitly on the derivatives of the *loss function* (derivatives with respect to the input data) and thus penalize the parameters implicitly though the regularity that they induce on the loss function. In the literature, the work [12] proposed the use of a regularization term of the form

$$\mathbb{E}_{x\sim\mu_0}[\|\nabla_x j(x,\theta)\|]$$

(i.e., as in problem (1.10)), where it was also noticed that the resulting regularization problem could be implemented easily with a "double backpropagation approach" (see our Remark 4.1). There is a plethora of works that in the past few years have used and analyzed a similar *input gradient regularization* approach (the case $p=\infty$ in our results or similar); see for example [17,25,29,30,40]. More recently, higher-order regularization terms penalizing *curvature* of the loss function have been proposed, e.g., [27].

1.5 Outline

The rest of the paper is organized as follows. In Sect. 2, we present the proof of Theorem 1.9 using two approaches, including a formal one based on the geometric structure of optimal transport. In Sect. 2.2, we discuss connections between regularization problems and perturbation-based training algorithms. In Sect. 2.3, we discuss the Pontryagin principle associated with the optimal control formulation of the first-order regularization problem (1.9) (i.e., Theorem 1.13). Section 3 is devoted to second-order regularization problems and in particular the motivation of problems (1.18) and (1.19), as well as their optimal control reformulations. Using the Pontryagin principles discussed throughout the paper, we motivate a family of algorithms for the training of robust neural networks in Sect. 4, and in Sect. 5 we present a series of numerical experiments to illustrate the performance of the algorithms. We wrap up the paper in Sect. 6 where we present some conclusions.

2 First-order regularization

2.1 Proof of Theorem 1.9

In this section, we present the proof of Theorem 1.9, first rigorously, and then by providing a formal argument that relies on the geometric structure of the space of probability measures $\mathcal{P}_p(\mathbb{R}^d)$ endowed with the W_p distance. For our rigorous proof, we use the fact that for every $\theta \in \Theta$ we have:

$$\max_{\tilde{\mu}:\, W_p(\mu_0,\tilde{\mu})\leq\delta} J(\tilde{\mu},\theta) = \max_{\pi\in\mathcal{F}_{\mu_0,\delta}} \left\{ \int_{\mathbb{R}^d\times\mathbb{R}^d} j(\tilde{x},\theta)d\pi(x,\tilde{x}) \right\}, \tag{2.1}$$

where $\mathcal{F}_{\mu_0,\delta}$ is the set of all Borel probability measures π on $\mathbb{R}^d\times\mathbb{R}^d$ whose first marginal is μ_0 and satisfy

$$\int_{\mathbb{R}^d\times\mathbb{R}^d} \|x-\tilde{x}\|^p d\pi(x,\tilde{x}) \leq \delta^p;$$

see Remark 1.5. Identity (2.1) has been used repeatedly in the literature of distributionally robust optimization to obtain dual representations for generic adversarial problems of the form (1.5) and to in turn propose new frameworks for statistical inference (e.g., [3–5,9,23,37]; see Sect. 2.2 for an explicit formula for the dual problem). In this section, however, we do not focus on the dual representation associated to the adversarial problem, but rather, on the flat geometry of the set of πs that parameterize the right-hand side of (2.1).

Proof of Theorem 1.9 We follow an approach based on lifting the probability distributions π to a space of random variables as used, for example, when defining and studying the L-derivative (see for example sections 5.1-5.2 in [8]). The main idea is to use the fact that, over an atomless probability space $(\Omega,\mathcal{F},\mathbb{P})$, and for any coupling $\pi \in \Gamma(\mu_0,\tilde{\mu})$ where μ_0, $\tilde{\mu}$ are probability distributions on $\mathbb{R}^d$, we can construct $\mathbb{R}^d$-valued random variables $X, \tilde{X}$ on Ω with joint distribution π (for a proof see Proposition 9.1.2 and Theorem 1.13.1 in [13]). We can then operate using these random variables to approximate our optimization problem. Since the development only makes use of our assumptions and the properties of π, the result is independent of the chosen probability space.

With this idea in mind, using Assumptions 1.7, we can write for any coupling $\pi \in \Gamma(\mu_0,\tilde{\mu})$ that

$$\begin{aligned} J(\tilde{\mu},\theta) &= \mathbb{E}[j(\tilde{X},\theta)] \\ &= \mathbb{E}\Big[j(X,\theta) + \nabla_x j(X,\theta)\cdot(\tilde{X}-X) \\ &\quad + \int_0^1 \{\nabla_x j(X+\lambda(\tilde{X}-X),\theta) - \nabla_x j(X,\theta)\}\cdot(\tilde{X}-X)d\lambda \Big], \end{aligned}$$

where $X, \tilde{X}$ are random variables with $(X,\tilde{X}) \sim \pi$. Using now the fact that $\nabla_x j(\cdot,\theta)$ is Lipschitz, it follows that

$$\left| J(\tilde{\mu},\theta) - \mathbb{E}\left[j(X,\theta) + \nabla_x j(X,\theta)\cdot(\tilde{X}-X) \right] \right| \leq \frac{1}{2} Lip(\nabla_x j(\cdot,\theta)) \mathbb{E}[\|\tilde{X}-X\|_e^2],$$

where $\|\cdot\|_e$ is the Euclidean norm.

Note that the constraint $\int_{\mathbb{R}^d\times\mathbb{R}^d} \|x-\tilde{x}\|^p d\pi(x,\tilde{x}) \leq \delta$ means that we only consider couplings where

$$\mathbb{E}[\|\tilde{X}-X\|_e^2] \leq C\mathbb{E}[\|\tilde{X}-X\|^2] \leq C(\mathbb{E}[\|\tilde{X}-X\|^p])^{2/p} \leq C\delta^2$$

where C is a constant appearing due to the equivalence of norms in $\mathbb{R}^d$; in the last line we use the fact that $p \geq 2$ to apply Jensen's inequality. Since the above is a uniform control in the space of feasible solutions π and Θ (thanks to Assumption 1.12), we conclude that

$$\sup_{\tilde{\mu}:\, W_p(\mu_0,\tilde{\mu})\leq\delta} J(\tilde{\mu},\theta) = \sup_{\pi\,:\int_{\mathbb{R}^d\times\mathbb{R}^d}\|x-\tilde{x}\|^p d\pi(x,\tilde{x})\leq\delta^p} \{J(\mu_0,\theta) + \mathbb{E}[\nabla_x j(X,\theta)\cdot(\tilde{X}-X)]\} + O(\delta^2),$$

for $O(\delta^2)$ independent of θ. Clearly, the first term within the supremum on the right-hand side is independent of $\tilde{\mu}$. We then only need to optimize the second term. Now, by a (generalized) Hölder inequality (see for example [39]), it follows that

$$\mathbb{E}[\nabla_x j(X,\theta)\cdot(\tilde{X}-X)] \leq \left(\mathbb{E}[\|\nabla_x j(X,\theta)\|_*^q]\right)^{1/q}\left(\mathbb{E}[\|\tilde{X}-X)\|^p]\right)^{1/p}$$

where equality can be attained whenever $1 \leq p \leq \infty$ for a proper choice of $\tilde{X}-X$ with $\left(\mathbb{E}[\|\tilde{X}-X\|^p]\right)^{1/p} = \delta$. Therefore,

$$\sup_{\tilde{\mu}:\, W_p(\mu_0,\tilde{\mu})\leq\delta} J(\tilde{\mu},\theta) = J(\mu_0,\theta) + \delta\left(\mathbb{E}[\|\nabla_x j(X,\theta)\|_*^q]\right)^{1/q},$$

which in turn deduces our claim since the $O(\delta^2)$ term is uniform over all $\theta \in \Theta$. □

Remark 2.1 Coming back to the literature on L-derivative mentioned in the proof of Theorem 1.9, we remark that in this case $\partial_\mu J(\mu_0,\theta) = \nabla_x j(X,\theta)$ (see section 5.2.2, Example 1 in [8]). As pointed out before, this quantity does not depend on the choice of lifting.

2.1.1 A formal geometric analysis in the space $\mathcal{P}_p(\mathbb{R}^d)$

In this section, we present an alternative formal geometric analysis for Theorem 1.9. We restrict to the case where $p \geq 2$ and for simplicity assume that $\|\cdot\|$ is the Euclidean norm. The idea in this geometric approach is to use the formal differential structure of the space of probability measures $\mathcal{P}_p(\mathbb{R}^d)$ endowed with the W_p distance and carry out a Taylor expansion of the function $\tilde{\mu} \mapsto J(\tilde{\mu},\theta)$ around μ_0. Ultimately, the goal is to replace the function $J(\tilde{\mu},\theta)$ with an approximation written in terms of a natural retraction of $\tilde{\mu}$ to the tangent space of $\mathcal{P}_p(\mathbb{R}^d)$ at the point μ_0. For a discussion on the general differential geometric perspective in the Wasserstein space (i.e., when $p=2$) see Chapter 8.2 in [35], and for more details on the geometry of the space $(\mathcal{P}_p(\mathbb{R}^d), W_p)$ see Chapter 8 in [1].

In more precise terms, let $\tilde{\mu}$ belong to the W_p-ball of radius δ around μ_0. For every such $\tilde{\mu}$ we consider the constant speed geodesic $t \in [0,1] \mapsto (\mu_t, V_t)$ that connects μ_0 with $\tilde{\mu}$. Namely, the continuity equation

$$\partial_t \mu_t + \operatorname{div}(V_t \mu_t) = 0, \quad t \in (0,1).$$

and the relations

$$\int_0^t \int_{\mathbb{R}^d} \|V_s(x)\|^p d\mu_s(x) ds = t(W_p(\mu_0,\tilde{\mu}))^p, \quad t \in [0,1],$$

are satisfied. Here, V_t is a vector field in $\mathbb{R}^d$ and μ_t is a probability measure in $\mathbb{R}^d$ which at time $t=1$ coincides with $\tilde{\mu}$. The pair (μ_t, V_t) can be characterized further. Indeed, assuming that there exists an *optimal* transport map T^* between μ_0 and $\tilde{\mu}$ for the c_p cost (for example assuming that μ_0 has a density with respect to the Lebesgeue measure) we can write:

$$\begin{cases} \mu_t = T_{t\sharp}\mu_0, \quad \forall t \in [0,1], \\ V_t(T_t(x)) = T^*(x) - x, \quad \forall x \in \operatorname{supp}(\mu_0), \forall t \in [0,1], \end{cases} \tag{2.2}$$

where the map T_t is given by

$$T_t(x) := tT^*(x) + (1-t)x.$$

The function $\tilde{\mu} \mapsto V_0 \in L^p(\mathbb{R}^d : \mathbb{R}^d, \mu_0)$ can be understood as a *logarithmic map*, i.e., a map that in particular sends points in the manifold $\mathcal{P}_p(\mathbb{R}^d)$ to tangent vectors at μ_0 and that satisfies:

$$\|V_0\|^p_{L^p(\mathbb{R}^d:\mathbb{R}^d,\mu_0)} = \int_{\mathbb{R}^d} \|V_0(x)\|^p d\mu_0(x) = (W_p(\mu_0, \tilde{\mu}))^p.$$

Let us now find an approximation for the function $J(\tilde{\mu}, \theta)$ in terms of an expression involving V_0. For that purpose, we Taylor-expand the function $j(\cdot, \theta)$ along the geodesic (2.2). First, following equation 8.1.4. in [1] we obtain:

$$\frac{d}{dt}J(\theta, \mu_t) = \frac{d}{dt}\int_{\mathbb{R}^d} j(x, \theta)d\mu_t(x) = \int_{\mathbb{R}^d} \nabla_x j(x, \theta) \cdot V_t(x)d\mu_t(x);$$

notice that, geometrically speaking, when $p = 2$ the above equation is the standard relation between directional derivatives and gradients in the Wasserstein space. We can also compute the second derivative along the geodesic as follows:

$$\begin{aligned}\frac{d^2}{dt^2}J(\theta, \mu_t) &= \frac{d}{dt}\left(\int_{\mathbb{R}^d} \nabla_x j(x, \theta) \cdot V_t(x)d\mu_t(x)\right)\\ &= \frac{d}{dt}\left(\int_{\mathbb{R}^d} \nabla_x j(\theta, T_t(x)) \cdot V_t(T_t(x))d\mu_0(x)\right)\\ &= \int_{\mathbb{R}^d} (D^2_x j(\theta, T_t(x))V_0(x)) \cdot V_0(x)d\mu_0(x).\end{aligned} \tag{2.3}$$

In particular,

$$\left|\frac{d^2}{dt^2}J(\theta, \mu_t)\right| \leq C\int_{\mathbb{R}^d} \|V_0(x)\|^2 d\mu_0(x) \leq C\left(\int_{\mathbb{R}^d} \|V_0(x)\|^p d\mu_0(x)\right)^{2/p} \leq C\delta^2,$$

where the constant C is uniform over all $t \in [0, 1]$, all $\theta \in \Theta$ (thanks to Assumption 1.12) and all $\tilde{\mu}$ with $W_p(\mu_0, \tilde{\mu}) \leq \delta$; notice that in the second to last line we have used Jensen's inequality since $p \geq 2$.

From the above computations we obtain:

$$\begin{aligned}J(\tilde{\mu}, \theta) &= J(\mu_0, \theta) + \frac{d}{dt}J(\theta, \mu_t)\big|_{t=0} + O(\delta^2)\\ &= J(\mu_0, \theta) + \int_{\mathbb{R}^d} \nabla_x j(x, \theta) \cdot V_0(x)d\mu_0(x) + O(\delta^2),\end{aligned}$$

where again we notice that the $O(\delta^2)$ term is uniform over all $\theta \in \Theta$ and all $\tilde{\mu}$ within W_p-distance δ from μ_0. Using the relation between the $\tilde{\mu}$s and their corresponding V_0s, we can thus expect that up to an error of order $O(\delta^2)$ (independent of $\theta \in \Theta$), the expression:

$$\max_{\tilde{\mu}:\, W_p(\mu_0,\tilde{\mu})\leq\delta} J(\theta, \tilde{\mu})$$

(an optimization problem over a curved manifold) is equal to:

$$J(\mu_0, \theta) + \max_{V_0:\, \|V_0\|_{L^p(\mathbb{R}^d:\mathbb{R}^d,\mu_0)}\leq\delta} \left\{\int_{\mathbb{R}^d} \nabla_x j(x, \theta) \cdot V_0(x)d\mu_0(x)\right\}, \tag{2.4}$$

which is an optimization over a flat Banach space. Since (2.4) is simply a dual representation for the $L^q(\mathbb{R}^d : \mathbb{R}^d, \mu_0)$-norm of $\nabla_x j(\cdot, \theta)$, it follows that (2.4) is equal to:

$$J(\mu_0, \theta) + \delta\left(\int_{\mathbb{R}^d} \|\nabla_x j(x, \theta)\|^q d\mu_0(x)\right)^{1/q},$$

which is the objective function in (1.9). Notice that the V_0 achieving the maximum takes the form:

$$V_0(x) := \delta \left[\int_{\mathbb{R}^d} \|\nabla_x j(\tilde{x}, \theta)\|^q d\mu_0(\tilde{x}) \right]^{-1/p} \frac{\nabla_x j(x, \theta)}{\|\nabla_x j(x, \theta)\|^{1-1/(p-1)}}, \tag{2.5}$$

with the convention $\mathbf{0}/0 = \mathbf{0}$.

Remark 2.2 There are a few steps in the above analysis that would need further justification in order to make this analysis into a rigurous proof. Here we offer some comments on this direction.

i) First, we have used the existence of optimal transport maps to write the geodesic (2.2) and to define an associated retraction map $\tilde{\mu} \mapsto V_0$. This can be done, for example, if we assume that μ_0 is absolutely continuous with respect to the Lebesgue measure. The reduction to the absolutely continuous case can be accomplished by an approximation argument since the $O(\delta^2)$ correction terms in the above analysis only depend on the control we have on the Hessian (in x) of the loss function given by Assumption 1.12.

ii) Going from the maximization problem $\max_{\tilde{\mu}:\, W_p(\mu_0,\tilde{\mu})\leq\delta}\{J(\theta, \tilde{\mu})\}$ to (2.4) is motivated by the fact that on a finite-dimensional smooth manifold one can find a one-to-one correspondence between points in a geodesic ball with small enough radius R (in particular smaller than the injectivity radius of the manifold) and tangent vectors at the center of the ball that have norm less than R. In the space $\mathcal{P}_p(\mathbb{R}^d)$, however, this intuition breaks down. To illustrate how one can still recover (2.4) using optimal transport theory let us consider the case $p = 2$ for concreteness and assume that μ_0 is absolutely continuous with respect to the Lebesgue measure. In that case, the V_0s induced by $\tilde{\mu}$s within W_2-distance δ from μ_0 can be written as:

$$\delta(\nabla_x \varphi(x) - \frac{x}{\delta}) = \delta \nabla_x (\varphi(x) - \frac{\|x\|^2}{2\delta})$$

for φ a convex function, as it follows from Brenier's theorem (see Theorem 2.12 in [35]). Now, by Assumption 1.12, the function

$$x \mapsto \frac{j(x, \theta)}{c} + \frac{\|x\|^2}{2\delta}$$

is a convex function for all small enough δ. In the above, $c = \|\nabla_x j(\cdot, \theta)\|_{L^2(\mathbb{R}^d;\mathbb{R}^d,\mu_0)}$. We can thus take $\varphi(x) = \frac{j(x,\theta)}{c} + \frac{\|x\|^2}{2\delta}$ (assuming δ is small enough) and $V_0(x) = \delta(\nabla\varphi(x) - x)$. It is clear that this V_0 maximizes (2.4) when $p = 2$.

iii) It is worth mentioning that the formal analysis presented here does not use specific attributes of the Euclidean norm and in fact can be used for general norms (as done in our rigurous proof of Theorem 1.9) with the difference that the form of the maximizer in problem (2.4) would be in general more cumbersome. One notable exception is the case of the ℓ^∞ norm on $\mathbb{R}^d$-vectors and the induced L^∞ norm on vector fields (i.e., $p = \infty$). Indeed, in that case the maximizer takes the form:

$$V_0(x) = \delta \text{sign}(\nabla_x j(x, \theta)),$$

where the sign function acts coordinatewise on vectors.

iv) Finally, it is worth mentioning that one of the main motivations for presenting this alternate analysis is to introduce additional tools that may come in useful when

considering different adversarial learning problems where for example the energy to maximize is not simply an integral with respect to a measure (here $J(\tilde{\mu}, \theta)$) but rather an energy which in general may include entropic or interaction terms as done recently in [11]. Most modern algorithms used for training robust neural networks include a step where data points are randomly perturbed before considering any drift information induced by the loss function. We believe that making a more concrete connection between said algorithms and a distributionally robust optimization problem would require analyzing this type of entropic term. This is work that is left for the future.

2.2 Connection with perturbation-based training algorithms

The current literature on robust training is dominated by algorithms based on constructing explicit adversarial samples around a given distribution that are then used as *the* training samples for the network. See for example [6,7,26,34], and references therein.

We can relate the results of our study in terms of adversarial samples. Indeed, equation (2.2) suggests considering a transport map of the form:

$$\hat{T}(x) := V_0(x) + x,$$

for the V_0 maximizing (2.4), and in turn consider the associated measure $\hat{\tilde{\mu}} := \hat{T}_\sharp \mu_0$ (the pushforward of μ_0 by $\hat{T}$); notice that $\hat{\tilde{\mu}}$ depends on θ, and for each θ $\hat{\tilde{\mu}}$ is a natural surrogate for the maximizer of problem (1.9). One may then consider an objective function of the form:

$$\theta \in \Theta \mapsto J(\hat{\tilde{\mu}}, \theta)$$

or more generally

$$\theta \in \Theta \mapsto \alpha J(\mu_0, \theta) + (1 - \alpha) J(\hat{\tilde{\mu}}, \theta),$$

for some $\alpha \in [0, 1)$, and the corresponding minimization problem over $\theta \in \Theta$ in order to enforce robustness. It is straightforward to show that, under the assumptions of Theorem 1.9, the minimum value of the resulting problem when $\alpha = 1 - \delta$ is within an error of order δ^2 of the minimum value of the regularization problem (1.9). We remark that if V_0 is taken to be $V_0(x) := \delta \cdot \text{sign}(\nabla_x j(x, \theta))$, i.e., as in iii) in Remark 2.2, then one can recover problem (1.27) introduced in [19].

To conclude our discussion in this section, let us point out that maximizers of problem (1.8) have a clear mathematical characterization. Unfortunately, even with this characterization an optimal $\tilde{\mu}$ is in general difficult to compute explicitly.

Proposition 2.3 *Let $\delta > 0$ and let $\theta \in \Theta$. Then*

$$\max_{\pi \in \mathcal{F}_{\mu_0, \delta}} \left\{ \int_{\mathbb{R}^d \times \mathbb{R}^d} j(\tilde{x}, \theta) d\pi(x, \tilde{x}) \right\} = \min_{\gamma \geq 0} \left\{ \gamma \delta^p + \mathbb{E}_{x \sim \mu_0} \left[j^\gamma(x, \theta) \right] \right\}, \tag{2.6}$$

where

$$j^\gamma(x, \theta) := \sup_{\tilde{x}} \left\{ j(\tilde{x}, \theta) - \gamma \|x - \tilde{x}\|^p \right\}.$$

Moreover, if π^* is a solution to the problem on the left-hand side, then there exists a solution to the problem on the right-hand side γ^* such that for π^*-a.e. (x, x^*):

$$x^* \in \arg\max_{\tilde{x} \in \mathbb{R}^d} \left\{ j(\tilde{x}, \theta) - \gamma^* \|x - \tilde{x}\|^p \right\}.$$

Finally, the second marginal of π^* is a solution to the problem (1.8).

Proof The dual characterization (2.6) is discussed, for example, in any of the following references: [3–5,9,23,37]. The characterization for the minimizers is a straightforward consequence of the zero duality gap. □

2.3 Pontryagin principle for the first-order regularized robust control problem

We start by writing the modified control problem in a slightly different form. Under Assumption 1.7 and fixed but arbitrary control θ, it follows that

$$\nabla_x j(x,\theta) = -P_{x,0}.$$

Indeed, let Δx be an arbitrary unitary vector in $\mathbb{R}^d$. From the flow property of the ordinary differential equation for X, we get that for almost all $t \in [0,T]$

$$\lim_{\epsilon \downarrow 0} \frac{1}{\epsilon}\{X_{x+\epsilon\Delta x,t} - X_{x,t}\} = \zeta_t^{\Delta x}$$

where $\zeta^{\Delta x}$ satisfies

$$\begin{cases} \dot{\zeta}_t^{\Delta x} = D_\xi f(X_{x,t},\theta_t)\zeta_t^{\Delta x} \\ \zeta_0^{\Delta x} = \Delta x. \end{cases} \tag{2.7}$$

Thus, the chain rule implies that

$$\begin{aligned} \nabla_x j(x,\theta)\cdot\Delta x &= \nabla_\xi \ell(X_{x,T},\theta_T)\cdot\zeta_T^{\Delta x} + \int_0^T \nabla_\xi \Phi(X_{x,s},\theta_s)\cdot\zeta_s^{\Delta x} ds \\ &= -P_{x,T}\cdot\zeta_T^{\Delta x} + \int_0^T \nabla_\xi \Phi(X_{x,s},\theta_s)\cdot\zeta_s^{\Delta x} ds \\ &= -P_{x,0}\cdot\zeta_0^{\Delta x} - \int_0^T \dot{P}_{x,s}\cdot\zeta_s^{\Delta x} ds - \int_0^T P_{x,s}\cdot\dot{\zeta}_s^{\Delta x} ds \\ &\quad + \int_0^T \nabla_\xi \Phi(X_{x,s},\theta_s)\cdot\zeta_s^{\Delta x} ds \\ &= -P_{x,0}\cdot\zeta_0^{\Delta x} = -P_{x,0}\cdot\Delta x. \end{aligned}$$

This equivalence between the dual variable P and (minus) the derivative of the loss function with respect to the input allows us to rewrite problem (1.9) in the form (1.13). We are ready to provide a proof of the Pontryagin principle result stated in Theorem 1.13.

Proof of Theorem 1.13 The proof follows the well-known "needle" perturbation approach: we take the optimal control and change it in a small interval; then, we deduce the effect on the overall value function and deduce first-order conditions of optimality from linear expansions and integration by parts.

To simplify the problem, note that we can and will assume, without loss of generality, that there is no running cost (i.e., $\Phi \equiv 0$) by transforming the running cost in a state variable: indeed, set $\hat{x} := (x, x')$ with $\hat{x}_0 = (x_0, 0)$, $\hat{\varrho} := (\varrho, -1)$, $\hat{f} := (f, \Phi)$, and $\hat{\ell}(\bar{x}) := \ell(x) + x'$, and note that the reduced is an equivalent control problem, still satisfies Assumptions 1.8, and does not contain any running cost.

Now, let $\tau \in (0,T)$ be a Lebesgue point for $(f(X_t^*,\theta_t^*), D_\xi f(X_t^*,\theta_t^*))$. By Assumption 1.8, the set of such points is dense in $[0,T]$. For $\epsilon \in (0, T-\tau)$ and $\eta \in \Theta$ let

$$\theta_t^{\epsilon,\tau} = \begin{cases} \eta & \text{if } t \in [\tau-\epsilon,\tau] \\ \theta_t^* & \text{otherwise} \end{cases},$$

and let $X_t^{\epsilon,\tau}, P_t^{\epsilon,\tau}$ be the solutions of

$$X_{x,t}^{\epsilon,\tau} = x + \int_0^t f(X_{x,s}^{\epsilon,\tau}, \theta_s^{\epsilon,\tau})ds$$
$$P_{x,t}^{\epsilon,\tau} = -\nabla_\xi \ell(X_{x,T}^{\epsilon,\tau}, \theta_T^*) + \int_t^T \{D_\xi f(X_{x,s}^{\epsilon,\tau}, \theta_s^{\epsilon,\tau})^\top P_{x,s}^{\epsilon,\tau}\}ds$$

that is, solutions of the forward variable using the control $\theta^{\epsilon,\tau}$ instead of the optimal θ^*.

Let us study the ϵ-order effect of this change of control policy. Let

$$\begin{cases} \dot{u}_{x,t} = D_\xi f(X_{x,t}^*, \theta_t^*)u_{x,t} & \text{for } t > \tau; \\ u_{x,t} = f(X_{x,\tau}^*, \eta) - f(X_{x,\tau}^*, \theta_\tau^*) & \text{for } t = \tau; \\ u_{x,t} = 0 & \text{for } t < \tau. \end{cases}$$

and

$$\begin{cases} \dot{v}_{x,t} = -D_\xi f(X_{x,t}^*, \theta_t^*)^\top v_{x,t} - (D_\xi^2 f(X_{x,t}^*, \theta_t^*)u_{x,t})^\top P_{x,t}^* \\ \qquad\quad + \delta_\tau \{D_\xi f(X_{x,\tau}^*, \theta_\tau^*) - D_\xi f(X_{x,\tau}^*, \eta)\}^\top P_{x,\tau}^* \\ v_{x,T} = -D_\xi^2 \ell(X_{x,T}^*)u_{x,T}; \end{cases}$$

where the term $\delta_\tau \{D_\xi f(X_{x,\tau}^*, \theta_\tau^*) - D_\xi f(X_{x,\tau}^*, \eta)\}^\top P_\tau^*$ denotes the jump arising at time τ due to the change in control. We can show (as in [41]) that

$$\lim_{\epsilon \downarrow 0} \frac{1}{\epsilon}(X_{x,t}^{\epsilon,\tau} - X_{x,t}^*) = u_{x,t};$$
$$\lim_{\epsilon \downarrow 0} \frac{1}{\epsilon}(P_{x,t}^{\epsilon,\tau} - P_{x,t}^*) = v_{x,t}.$$

Since optimality implies

$$\mathbb{E}_{x\sim\mu_0}\left[\ell(X_{x,T}^{\epsilon,\tau})\right] + \delta\left(\mathbb{E}_{x\sim\mu_0}\left[\|P_{x,0}^{\epsilon,\tau}\|^q\right]\right)^{1/q} \geq \mathbb{E}_{x\sim\mu_0}\left[\ell(X_{x,T}^*)\right] + \delta\left(\mathbb{E}_{x\sim\mu_0}\left[\|P_{x,0}^*\|^q\right]\right)^{1/q}$$

we have from Assumption 1.8 and dominated convergence that

$$\begin{aligned} 0 &\leq \lim_{\epsilon\downarrow 0} \frac{1}{\epsilon}\Big\{\mathbb{E}_{x\sim\mu_0}\left[\ell(X_{x,t}^{\epsilon,\tau}) - \ell(X_{x,t}^*)\right] + \delta\big(\big(\mathbb{E}_{x\sim\mu_0}\left[\|P_0^{\epsilon,\tau}(x_0)\|^q\right]\big)^{1/q} \\ &\quad - \big(\mathbb{E}_{x\sim\mu_0}\left[\|P_{x,0}^*\|^q\right]\big)^{1/q}\big)\Big\} \\ &= \mathbb{E}_{x\sim\mu_0}\left[\nabla_\xi \ell(X_{x,T}^*)^\top u_{x,T}\right] + \delta\left(\mathbb{E}_{x\sim\mu_0}\left[\|P_{x,0}^*\|^q\right]\right)^{-\frac{1}{p}} \mathbb{E}_{x\sim\mu_0}\left[\|P_{x,0}^*\|^{q-2}(P_{x,0}^*)^\top v_{x,0}\right] \\ &= \mathbb{E}_{x\sim\mu_0}\left[\nabla_\xi \ell(X_{x,T}^*)^\top u_{x,T} + (\beta_{x,0}^*)^\top v_{x,0}\right] \end{aligned} \tag{2.8}$$

On the other hand, $d((\beta_{x,t}^*)^\top v_{x,t}) = d(\beta_{x,t}^*)^\top v_{x,t} + (\beta_{x,t}^*)^\top dv_{x,t}$, which implies after some cancelations that

$$\begin{aligned} (\beta_{x,0}^*)^\top v_{x,0} &= (\beta_{x,T}^*)^\top v_{x,T} - (\beta_{x,\tau}^*)^\top \{D_\xi f(X_{x,\tau}^*, \theta_\tau^*) - D_\xi f(X_{x,\tau}^*, \eta)\}^\top P_{x,\tau}^* \\ &\quad + \int_0^T (\beta_{x,t}^*)^\top (D_\xi^2 f(X_{x,t}^*, \theta_t^*)u_{x,t})^\top P_{x,t}^* dt. \end{aligned}$$

Therefore, equation (2.8) becomes

$$
\begin{aligned}
0 \leq \mathbb{E}_{x\sim\mu_0}\Big[&\{\nabla_\xi \ell(X^*_{x,T})^\top - (\beta^*_{x,T})^\top D^2_\xi \ell(X^*_{x,T})\}u_{x,T} \\
&+ \int_0^T (\beta^*_{x,t})^\top (D^2_\xi f(X^*_{x,t},\theta^*_t)u_{x,t})^\top P^*_{x,t}dt\Big] \\
&+ \mathbb{E}_{x\sim\mu_0}\Big[\beta^{*\top}_{x,\tau}\{D_\xi f(X^*_{x,\tau},\eta) - D_\xi f(X^*_{x,\tau},\theta^*_\tau)\}^\top P^*_{x,\tau}\Big] \\
= \mathbb{E}_{x\sim\mu_0}\Big[&-\alpha^{*\top}_{x,T}u_{x,T} + \int_0^T \beta^{*\top}_{x,t}(D^2_\xi f(X^*_{x,t},\theta^*_t)u_{x,t})^\top P^*_{x,t}dt + \beta^{*\top}_{x,\tau}\{D_\xi f(X^*_{x,\tau},\eta) \\
&- D_\xi f(X^*_{x,\tau},\theta^*_\tau)\}^\top P^*_{x,\tau}\Big] \\
= \mathbb{E}_{x\sim\mu_0}\Big[&-\alpha^{*\top}_{x,\tau}\{f(X^*_{x,\tau},\eta) - f(X^*_{x,\tau},\theta^*_\tau)\} + \beta^{*\top}_{x,\tau}\{D_\xi f(X^*_{x,\tau},\eta) - D_\xi f(X^*_{x,\tau},\theta^*_\tau)\}^\top P^*_{x,\tau}\Big].
\end{aligned}
$$

This deduces the maximum principle (1.16) in the case without running costs. □

3 Second-order regularization

We motivate now problems (1.18) and (1.19). Continuing our computations from section 2.1.1, we can write for every $\tilde{\mu}$ within W_p-distance δ from $\tilde{\mu}$:

$$
\begin{aligned}
J(\tilde{\mu},\theta) = J(\mu_0,\theta) &+ \int_{\mathbb{R}^d} \nabla_x j(x,\theta)\cdot V_0(x)d\mu_0(x) \\
&+\frac{1}{2}\int_{\mathbb{R}^d}(D^2_x j(x,\theta)V_0(x))\cdot V_0(x)d\mu_0(x) + O(\delta^3),
\end{aligned}
$$

if for example we assume that the function $j(\cdot,\theta)$ has bounded third-order derivatives uniformly over $\theta \in \Theta$. In that case, we could expect that up to an error of order $O(\delta^3)$ (independent of $\theta\in\Theta$), the problem

$$
\max_{\tilde{\mu}:\, W_p(\mu_0,\tilde{\mu})\leq\delta} J(\theta,\tilde{\mu})
$$

(an optimization problem over a curved manifold) is equal to

$$
\begin{aligned}
J(\mu_0,\theta) + \max_{V_0:\, \|V_0\|_{L^p(\mathbb{R}^d;\mathbb{R}^d,\mu_0)}\leq\delta} \Big\{&\int_{\mathbb{R}^d}\nabla_x j(x,\theta)\cdot V_0(x)d\mu_0(x) \\
&+\frac{1}{2}\int_{\mathbb{R}^d}(D^2_x j(x,\theta)V_0(x))\cdot V_0(x)d\mu_0(x)\Big\},
\end{aligned}
\tag{3.1}
$$

which is again an optimization problem over a flat Banach space. However, in contrast with problem (2.4), problem (3.1) does not have an explicit solution. What is more, in general, the correct expansion of (3.1) in δ (up to order two) depends on the size of $\nabla_x j(\cdot,\theta)$ relative to δ as we illustrate with the following analogous finite-dimensional problem.

Remark 3.1 Consider the following maximization problem in $\mathbb{R}^m$:

$$
\max_{v\in\mathbb{R}^m \text{ s.t. } \|v\|\leq\delta} \big\{b\cdot v + (Av)\cdot v\big\}, \tag{3.2}
$$

where A is an arbitrary $m\times m$ symmetric matrix (not necessarily with a sign) and b is an arbitrary vector in $\mathbb{R}^m$. Notice that:

- If δ is small enough and $\frac{\delta}{\|b\|} = o(1)$, then the linear term dominates the problem and we can write:

$$
(3.2) = \delta\|b\| + \delta^2\left(A\frac{b}{\|b\|}\right)\cdot\left(\frac{b}{\|b\|}\right) + o(\delta^2).
$$

This value is obtained by plugging the maximizer of the problem $\max_{v\in\mathbb{R}^m \text{ s.t. } \|v\|\leq\delta} \{b\cdot v\}$ in the objective of (3.2).

- If δ is small enough and $\frac{\|b\|}{\delta} = o(1)$, then the quadratic term dominates the problem and we can actually write:

$$(3.2) = \delta^2(\lambda_{max})_+ + o(\delta^2),$$

where in the above λ_{max} is the largest eigenvalue of A and $(a)_+$ denotes the positive part of $a \in \mathbb{R}$. This value is obtained by plugging the maximizer of the problem $\max_{v\in\mathbb{R}^m \text{ s.t. } \|v\|\leq\delta} \{(Av)\cdot v\}$ in the objective of (3.2).

- When $\|b\| \sim \delta$, an explicit second order expansion for (3.2) is intractable for all practical purposes as can be easily seen by inspection after writing the KKT conditions for this in general non-convex problem.

To connect problems (1.18) and (1.19) with the previous remark, we use the following observations. First, if we plug the V_0 from (2.5) (the maximizer of the problem (2.4)) in the objective function from problem (3.1), we obtain the objective function in problem (1.18). As for the objective in problem (1.19), we notice the following.

Proposition 3.2 *Let $p \geq 2$. Then, for every $\theta \in \Theta$ we have:*

$$\begin{aligned}&\max_{V_0:\, \|V_0\|_{L^p(\mathbb{R}^d:\mathbb{R}^d,\mu_0)}\leq\delta} \left\{\frac{1}{2}\int_{\mathbb{R}^d} (D_x^2 j(x,\theta)V_0(x))\cdot V_0(x) d\mu_0(x)\right\}\\ &= \left(\int_{\mathbb{R}^d} |(\lambda_{max}(x,\theta))_+|^{\tilde{q}} d\mu_0(x)\right)^{1/\tilde{q}},\end{aligned} \tag{3.3}$$

where $\lambda_{max}(\theta,x)$ is the largest eigenvalue of $D_x^2 j(x,\theta)$ and where $\tilde{q}$ is the conjugate of $p/2$, i.e.:

$$\frac{2}{p} + \frac{1}{\tilde{q}} = 1.$$

Proof For each x in the support of μ_0 we select $V_0(x) = g(x)U(x)$ where $U(x)$ is a unit (Euclidean) norm eigenvector of $D_x^2 j(x,\theta)$ with eigenvalue $\lambda_{max}(\theta,x)$ and g is a scalar function that satisfies $g(x) = 0$ if $\lambda_{max}(x,\theta) \leq 0$ and $\int_{\mathbb{R}^d} |g(x)|^p d\mu_0(x)dx \leq \delta^p$. Plugging this V_0 in the objective function of the max problem in (3.3), we obtain:

$$\frac{1}{2}\int_{\mathbb{R}^d} (\lambda_{max}(\theta,x))_+ (g(x))^2 d\mu_0(x).$$

It is then straightforward to show that:

$$\begin{aligned}&\max_{\|V_0\|_{L^p(\mathbb{R}^d:\mathbb{R}^d,\mu_0)}\leq\delta} \left\{\frac{1}{2}\int_{\mathbb{R}^d} (D_x^2 j(x,\theta)V_0(x))\cdot V_0(x) d\mu_0(x)\right\}\\ &\qquad = \max_{\|g^2\|_{L^{p/2}(\mu_0)}\leq\delta^2} \left\{\frac{1}{2}\int_{\mathbb{R}^d} (\lambda_{max}(\theta,x))_+ (g(x))^2 d\mu_0(x)\right\}.\end{aligned}$$

Given that the scalar function $x \mapsto (\lambda_{max}(x,\theta))_+$ is non-negative, we can recognize, by duality, that the right-hand side of the above expression is equal to:

$$\frac{\delta^2}{2}\left(\int_{\mathbb{R}^d} |(\lambda_{max}(x,\theta))_+|^{\tilde{q}} d\mu_0(x)\right)^{1/\tilde{q}},$$

obtaining in this way the desired result. □

In summary, problems (1.18) and (1.19) can be interpreted as second-order expansions for $\inf_{\theta\in\Theta}\max_{\tilde{\mu}\,:W_p(\mu_0,\tilde{\mu})\leq\delta} J(\tilde{\mu},\theta)$ in two distinct regimes: 1) when gradients are *not* small relative to δ, more precisely, when the norms $\|\nabla_x j(x,\theta)\|_{L^q(\mathbb{R}^d:\mathbb{R}^d,\mu_0)}$ are larger than δ, and 2) when gradients are considerably smaller than δ. In the next section, we discuss the structure of both of these regularization problems. Recall that, as discussed in section 1.3.1, both of these problems are closely connected to problems used in the literature to train robust neural networks.

3.1 Adjoint variable formulation of the second-order regularized robust control problems

The main purpose in this section is to examine the transformations on the second-order problems (1.18) and (1.19) in terms of adjoint variables.

3.2 The problem in Proposition (1.16)

We aim to deduce Proposition 1.16. We start by studying how to write the derivative of the regularization term in the first-order expansion problem.

Lemma 3.3 *Under Assumption 1.8, and assuming that $p \geq 2$ and that $\|\cdot\|$ is the Euclidean norm, we have:*

$$\nabla_x \|P_{x,0}\|^q = -q\kappa_{\mu_0,\delta}\hat{\alpha}_{x,0}$$

with $\kappa_{\mu_0,\delta} = \frac{1}{\delta}\left(\mathbb{E}_{x_0\sim\mu_0}[\|P_{x_0,0}\|^q]\right)^{1/p}$.

Remark 3.4 The role of the constant $\kappa_{\mu,\delta}$ is to cancel the terms in the adjoint variables related to the mean-field contribution of the robust problem to allow us to focus on the pointwise result.

Remark 3.5 We had already seen for the original problem with loss function j that $P_{x,0} = -\nabla_x(j(x,\theta))$, i.e., the direction of steepest descent of the loss function with respect to the initial point. Given the decomposition for the adjoint variable α, Lemma 3.3 implies that α plays the analogous role for the first-order robust control problem.

Proof of Lemma 3.3 For an arbitrary Δx of unitary norm in $\mathbb{R}^d$, we have that

$$\nabla_x \|P_{x,0}\|^q \cdot \Delta x = \lim_{\epsilon\downarrow 0}\frac{1}{\epsilon}[\|P_{x+\epsilon\Delta x,0}\|^q - \|P_{x,0}\|^q] = q\|P_{x,0}\|^{q-2}(P_{x,0})^\top \eta_0^{\Delta x},$$

where

$$\eta_t^{\Delta x} = \lim_{\epsilon\downarrow 0}\frac{1}{\epsilon}[P_{x+\epsilon\Delta x,t} - P_{x,t}]. \tag{3.4}$$

From Assumption 1.8 and Leibniz rule, we get that

$$\eta_t^{\Delta x} = -D_\xi^2\ell(X_{x,T},\theta_T)\zeta_T^{\Delta x} + \int_t^T D_\xi f(X_{x,s},\theta_s)^\top \eta_s^{\Delta x} + (D_\xi^2 f(X_{x,s},\theta_s)\zeta_s^{\Delta x})^\top P_{x,s}ds, \tag{3.5}$$

where $\zeta^{\Delta x}$ is defined in (2.7). Using the definition of the process β in (1.15), it follows that

$$\nabla_x \|P_{x,0}\|^q \cdot \Delta x = q\|P_{x,0}\|^{q-2}(P_{x,0})^\top \eta_0^{\Delta x} = q\kappa_{\mu_0,\delta}(\beta_{x,0})^\top \eta_0^{\Delta x}.$$

We can now use the integration rule for products and the dynamics for β and $\eta^{\Delta x}$ to get

$$\begin{aligned}
\nabla_x \|P_{x,0}\|^q \cdot \Delta x &= q\kappa_{\mu_0,\delta}\left((\beta_{x,T})^\top \eta_T^{\Delta x} - \int_0^T (\eta_s^{\Delta x})^\top \dot{\beta}_{x,s} ds - \int_0^T (\beta_{x,s})^\top \dot{\eta}_s^{\Delta x} ds\right) \\
&= q\kappa_{\mu_0,\delta}\left(-(\beta_{x,T})^\top D_\xi^2 \ell(X_{x,T},\theta_T)\zeta_T^{\Delta x} - \int_0^T (\eta_s^{\Delta x})^\top \dot{\beta}_{x,s} ds - \int_0^T (\beta_{x,s})^\top \dot{\eta}_s^{\Delta x} ds\right) \\
&= q\kappa_{\mu_0,\delta}\Big(- (\hat{\alpha}_{x,T})^\top \zeta_T^{\Delta x} - \int_0^T (\eta_s^{\Delta x})^\top D_\xi f(X_{x,s},\theta_s)\beta_{x,s} ds \\
&\quad + \int_0^T (\beta_{x,s})^\top D_\xi f(X_{x,s},\theta_s)^\top \eta_s^{\Delta x} ds \\
&\quad + \int_0^T (\beta_{x,s})^\top (D_\xi^2 f(X_{x,s},\theta_s)\zeta_s^{\Delta x})^\top P_{x,s} ds\Big) \\
&= q\kappa_{\mu_0,\delta}\Big(- (\hat{\alpha}_{x,T})^\top \zeta_T^{\Delta x} + \int_0^T (\beta_{x,s})^\top (D_\xi^2 f(X_{x,s},\theta_s)\zeta_s^{\Delta x})^\top P_{x,s} ds\Big). \qquad (3.6)
\end{aligned}$$

Similarly, from the dynamics of $\hat{\alpha}$ and $\zeta^{\Delta x}$ we get

$$\begin{aligned}
(\hat{\alpha}_{x,T})^\top \zeta_T^{\Delta x} &= (\hat{\alpha}_{x,0})^\top \zeta_0^{\Delta x} + \int_0^T (\hat{\alpha}_{x,s})^\top D_\xi f(X_{x,s},\theta_s)\zeta_s^{\Delta x} ds \\
&\quad - \int_0^T (\zeta_s^{\Delta x})^\top D_\xi f(X_{x,s},\theta_s)^\top \hat{\alpha}_{x,s} ds + \int_0^T (\zeta_s^{\Delta x})^\top \{D_\xi^2 f(X_{x,s},\theta_s)^\top P_{x,s}\}^\top \beta_{x,s} ds \\
&= (\hat{\alpha}_{x,0})^\top \Delta x + \int_0^T (\zeta_s^{\Delta x})^\top \{D_\xi^2 f(X_{x,s},\theta_s)^\top P_{x,s}\}^\top \beta_{x,s} ds.
\end{aligned}$$

Replacing back into (3.6), we conclude that

$$\nabla_x \|P_{x,0}\|^q \cdot \Delta x = -q\kappa_{\mu_0,\delta}\hat{\alpha}_{x,0} \cdot \Delta x$$

from where the claim follows. □

A straightforward consequence of Lemma 3.3 and the analogous result for P is that the original cost function $j(\cdot,\theta)$ is twice differentiable in the direction of the gradient. More precisely we obtain the following result.

Corollary 3.6 *Under Assumption 1.8, $j(\cdot,\theta)$ is twice differentiable in x for any fixed control θ and*

$$\|\nabla_x j(x,\theta)\|^{q-2} D_x^2 j(x,\theta)\nabla_x j(x,\theta) = \kappa_{\mu_0,\delta}\hat{\alpha}_{x,0}.$$

Proof of Proposition 1.16 It follows directly from Corollary 3.6 and the fact that $P_{x,0} = -\nabla_x j(x,\theta)$ in (1.18). □

3.3 The problem in Proposition (1.21)

As in the previous case, we start examining the role of the adjoint variables that we introduce into the problem.

Lemma 3.7 *For a fixed vector $v \in \mathbb{R}^d$, consider the adjoint variables*

$$\gamma_{x,t}^v = v + \int_0^t D_\xi f\,(x_{x,s},\theta_s)\,\gamma_{x,s}^v ds,$$

and

$$\rho^{\nu}_{x,t} = D^2_{\xi}\ell\left(X_{x,T}, \theta_T\right)\gamma^{\nu}_{x,T} + \int_t^T D_{\xi}f\left(X_{x,s}, \theta_s\right)^{\top}\rho^{\nu}_{x,s}ds - \int_t^T \left\{D^2_{\xi}f\left(X_{x,s}, \theta_s\right)^{\top}P_{x,s}\right\}^{\top}\gamma_{x,s}ds.$$

Then

$$\nabla_x(P_{x,0}\cdot\nu) = -\rho^{\nu}_{x,0}.$$

Proof The proof of this result is very similar to that of Lemma 3.3 and thus we skip the details. □

Proof of Proposition 1.21 On the one hand, re-expressing the problem in terms of the adjoint variables is a direct consequence of Lemma 3.7.

On the other hand, under the stated assumptions, the argument inside the expectation in (1.25) has finite variance uniformly in θ and x: therefore, replacing the expectation by the empirical mean using m samples produces an estimator that converges almost surely by the law of large numbers and has errors subject to the central limit theorem. □

4 Training robust neural networks

An approach to robust training is suggested by the results we have presented on the regularized adversarial control problems. The Pontryagin principle in Theorem 1.13 can be used to create optimization algorithms: training the network can be understood as solving a fixed-point problem where the constraints in (1.13), equations (1.14), (1.15), and the maximum principle (1.16) must be simultaneously satisfied. There are many methods to solve numerically such a fixed-point problem, but, undoubtedly, the most popular consists in applying consecutively a step of forward propagation to solve for the primal variables, a step of backward propagation to get the dual variables, and the solution of an optimization algorithm to update the controls (typically this is substituted with a gradient step to solve such optimization problem with an approach like stochastic gradient descent or any of its siblings).

We present in Algorithm 1 an implementation of the first-order regularized control problem applied to ResNets for the case with no running cost. The adjustment to the case with running cost is straightforward. Note that all equations, except for the one of X, are linear in their respective variables. Moreover, they involve only f and its two derivatives. Thus, we can easily implement this algorithm in platforms like TensorFlow or PyTorch.

The algorithm takes an even simpler form when considering a ResNet with ReLu activation functions at each stage: although this activation function is not differentiable, the backpropagation algorithm has been successfully applied using a 'relaxed' gradient. Following the same ideas, Algorithm 1 follows with $D^2_{\xi}f(\xi,\vartheta) = 0$. In this particular case, all linear equations are driven by the same factor $\nabla_{\xi}f$, which makes it simpler to implement.

Remark 4.1 Let us stress (as has been done before, for example in [14,24]) that backpropagation training is by no means the only possible approach to solve the fixed-point problem for training, and in certain problems can have structure favoring alternative algorithms. We notice that Algorithm 1 in the case $p = \infty$ (and $q = 1$) is the *double backpropagation*

Algorithm 1 Backpropagation with SGD for robust control problem - ResNet

Set h, γ small constants
$i = 0$
Initialize $\theta_k^0 \equiv 0$ for all k
while No convergence **do**
  for Every batch **do**
    Set $X_0 = x_0$ for each x_0 in the batch
    Forward propagate using activation function (X):
      $X_{k+1} = X_k + hf(X_k, \theta_k)$
    Backpropagate using derivatives of activation functions (P):
      Set $P_N = -D_\xi \ell(X_N)$
      $P_k = (I + hD_\xi f(X_k, \theta_k)^\top) P_{k+1}$
    Forward propagate using derivatives of activation functions (β):
      Set $\beta_0 = \delta \left(\tilde{\mathbb{E}}\left[\|P_0\|^q\right]\right)^{-\frac{1}{p}} \|P_0\|^{q-2} P_0$; where $\tilde{\mathbb{E}}$ is mean over elements in the batch.
      $\beta_{k+1} = (I + hD_\xi f(X_k, \theta_k))\beta_k$
    Backpropagate using first and second derivatives of the activation function (α):
      $\alpha_T = -D_\xi \ell(X_N) + D_\xi^2 \ell(X_N)\beta_N$
      $\alpha_k = (I + hD_\xi f(X_k, \theta_k)^\top)\alpha_{k+1} - hP_k^\top D_\xi^2 f(X_k, \theta_k)\beta_k$
    Calculate for each k the gradient:
      $\nabla_\vartheta H(X_k, P_k, \alpha_k, \beta_k, \vartheta_k^i) = \alpha_k \cdot D_\vartheta f(X_k, \vartheta_k^i) - \beta_k \cdot (D_{\vartheta,\xi} f(X_k, \vartheta_k^i)^\top P_k)$
    Update the control for each k:
      $\theta_k = \theta_k + \gamma \tilde{\mathbb{E}}[\nabla_\vartheta H(X_k, P_k, \alpha_k, \beta_k, \theta_k)]$
    $i = i + 1$
  end for
end while

algorithm from [12]. This follows from the discussion presented in section 4.2 in [24] on the general relation between the method of successive approximations and gradient descent with backpropagation.

Remark 4.2 In strict terms, we have results that are applicable only to ResNets. However, they can be formally generalized to other types of neural networks. For instance, one can rewrite an instance of a vanilla forward network in terms of a ResNet by setting the diffusion coefficient to be

$$\tilde{f}(t_k, X_k) = \frac{f(t_k, X_k) - X_k}{h},$$

where h is the small coefficient in 1 representing the time discretization. The net effect on Algorithm 1 is that equations are no longer residual.

We present in Algorithm 2 the implementation of second-order Pontryagin principle in (1.22), still in the case of ResNet. Note that there are strong similarities between the forward and backpropagation of two pairs of variables, which is of advantage for any possible implementation. Anologously to the case of Algorithm 1, additional simplifications can be obtained for the case of activation functions like ReLu.

Algorithms 1 and 2 are written in accordance with the most typical way of implementing training algorithms via the use of batched optimization. In this type of implementation, the training sample is subdivided in different batches: We calculate all propagation equations

Algorithm 2 Backpropagation with SGD for robust control problem - second order

1: Set γ small constant (learning rate)
2: $i = 0$
3: Initialize $\theta_k^0 \equiv 0$ for all k
4: **while** No convergence **do**
5: **for** Every batch **do**
6: Set $X_0 = x_0$ for each x_0 in the batch
7: Forward propagate using activation function (X):
8: $X_{k+1} = X_k + hf(X_k, \theta_k)$
9: Backpropagate using derivatives of activation functions (P):
10: Set $P_N = -D_\xi \ell(X_N)$
11: $P_k = (I + hD_\xi f(X_k, \theta_k)^\top) P_{k+1}$
12: Forward propagate using derivatives of activation functions (β, λ):
13: $\beta_0 = \delta \left(\tilde{\mathbb{E}}\left[\|P_0\|^q\right]\right)^{-\frac{1}{p}} \|P_0\|^{q-2} P_0$; where $\tilde{\mathbb{E}}$ is mean of elements in the batch.
14: $\beta_{k+1} = (I + hD_\xi f(X_k, \theta_k))\beta_k$
15: Set $\lambda_0 = \frac{\delta}{2} \frac{P_0}{\|P_0\|}$
16: $\lambda_{k+1} = (I + hD_\xi f(X_k, \theta_k)^\top)\lambda_k$
17: Backpropagate using first and second derivatives of the activation function (α, ψ):
18: Set $\alpha_T = D_\xi^2 \ell(X_N)\beta_N$
19: $\alpha_k = (I + hD_\xi f(X_k, \theta_k)^\top)\alpha_{k+1} - P_k^\top D_\xi^2 f(X_k, \theta_k)\beta_k$
20: Set $\psi_T = -D_\xi^2 \ell(X_N)^\top \beta_N$
21: $\psi_k = (I + hD_\xi f(X_k, \theta_k))\psi_{k+1} + P_k^\top D_\xi^2 f(X_k, \theta_k)\lambda_k$
22: Forward propagate using first- and second-order derivatives (π):
23: Set $\pi_0 = \frac{1}{\|P_0\|}\left(P_0 + \frac{\delta}{2}\alpha_0 - \psi_0\right) - \frac{\delta}{\|P_0\|^3} P_0 \cdot \left(\frac{1}{2}\alpha_0 - \psi_0\right) P_0$
24: $\pi_{k+1} = (I + D_\xi f(X_k, \theta_k))\pi_k - \{(\beta_k)^\top (D_\xi^2 f(X_k, \theta_k))^\top\}^\top \lambda_k$
25: Backpropagate using first-, second-, and third-order derivatives (ψ):
26: $\phi_N = P_N + D_x^2 \ell(X_N)\pi_N - \{D_\xi^3 \ell(X_N, \theta_N)\beta_N\}^\top \lambda_N$

$$\begin{aligned}\phi_k = {} & (I + D_\xi f(X_k, \theta_k)^\top)\phi_{k+1} - \{D_\xi^2 f(X_k, \theta_k)^\top P_k\}^\top \pi_k \\ & - \{D_\xi^2 f(X_k, \theta_k)^\top \alpha_k\}^\top \lambda_k + \{D_\xi^2 f(X_k, \theta_k)^\top \beta_k\}^\top \psi_k \\ & + (\{D_\xi^3 f(X_k, \theta_k)^\top P_k\}^\top \beta_k)^\top \lambda_k\end{aligned}$$

27: Calculate for each k the gradient:

$$\begin{aligned}\nabla_\vartheta H(X_k, P_k, \alpha_k, \beta_k, \theta_k) = {} & \phi_k^\top D_\vartheta f(X_k, \theta) - P_k^\top D_\xi f(X_k, \theta_k)\pi_k - \alpha_k^\top (D_{\vartheta,\xi} f(X_k, \theta_k))\lambda_k \\ & + \lambda_k^\top \{(D_{\vartheta,\xi,\xi} f(X_k, \theta_k)^\top P_k\}^\top \beta_k + \beta_k^\top (D_{\vartheta,\xi} f(X_k, \theta_k))\psi_k\end{aligned}$$

28: Update the control for each k:
29: $\theta_k = \theta_k + \gamma \tilde{\mathbb{E}}[\nabla_\vartheta H(X_k, P_k, \alpha_k, \beta_k, \theta_k)]$
30: $i = i + 1$
31: **end for**
32: **end while**

for initial points in the subsample values and then update the control θ_k using the empirical expectation calculated with the points in the sample.

Frequently, practitioners use rather small values for the batch size (referred in those cases as *mini-batches*). Note, however, that the mean-field effect of the optimization implies that in addition to the control updating term, we also need to calculate an average in the term $\tilde{\mathbb{E}}\left[\|P_0\|^q\right]$ in line 13 to initialize β_0. For stability reasons, we therefore advice

Table 1 Accuracy and relative training time after 5 epochs. Although all training procedures are similarly capable when evaluated with a clean test, the regularization improves the resilience of the network when subject to adversarial attacks at a moderate cost. Note the dependence of results on the chosen vector norm $\|\cdot\|$. Best values in bold

	$r = 2$			$r =$inf		
	Baseline	Order 1	Order 2	Baseline	Order 1	Order 2
Accuracy (clean)	98.41	98.42	98.42	98.41	**98.45**	98.44
Accuracy (adversarial)	0.68	2.09	2.11	0.7	**23.11**	22.94
Training time (factor)	1.0	1.14	1.39	**0.99**	1.13	1.38

against using small batch sizes in this case. A notable exception appears when $p = \infty$ (i.e., $q = 1$), when the initialization of β in line 13 becomes $\beta_0 = \text{sign}(P_0)$. Thus, in this case, the mean-field action disappears and small batches are again perfectly acceptable.

5 Numerical illustration

We illustrate our results numerically in the context of image classification. We train a simple convolutional network[1] to perform the classification task on the MNIST database. We then test the network with a *clean* testing sample, and with an *adversarial* version constructed via modification of the latter using PGD with 20 steps and a step size of 0.04.

We train the network in three different versions: the *baseline* method (i.e., unrobust) which uses the cross-entropy loss function, and the *Order 1* and *Order 2* versions obtained by adding regularization terms as explained in problems 1.13 and 1.22, respectively. As parameters, we fix $\delta = 0.2$, $p = \infty$ (equivalently, $q = 1$), and we take the norm $\|\cdot\|_*$ to be the $r-$ norm with $r \in \{2, \infty\}$.

Table 1 shows the accuracy of the network after training with the three stated procedures. Although all training procedures perform similarly when evaluated with a clean testing sample, the Order 1 regularization significantly improves the robustness of the network when subject to a sample modified by the adversarial attack. The table also shows that the choice of the vector norm to be used plays a significant role. This is not a surprise, since one can understand the PGD attack as directed by taking infinity norms on successive gradients. Different choices of norms might be better suited for other types of adversarial attacks. Importantly, the improved robustness comes with a moderate cost in training time of 14% (39% for Order 2) over the baseline training time.

Table 1 also shows that the Order 2 method does not seem to be contributing to the overall robustness improvement beyond what is already done by Order 1. In order to have a better understanding of the numerical effect of each procedure, we plot in Figure 1 a local view of the loss surface obtained by calculating the cross-entropy for a clean testing image and perturbations around it. The corrupted images are obtained by an additive perturbation of the size marked in each axis: one in an adversarial direction and another in a random direction orthogonal to the adversarial. This way of illustrating the results is suggested in [31].

The robustness effect of the regularized problems is illustrated in Figure 1 by a reduction in values of the level of the surface, which translates in smaller cross-entropy and higher likeliness of obtaining an accurate classification. The plot also suggests that the effect of the

[1] Two layers with a convolutional kernel, ReLu activation functions, and maxpool; and two linear layers at the end

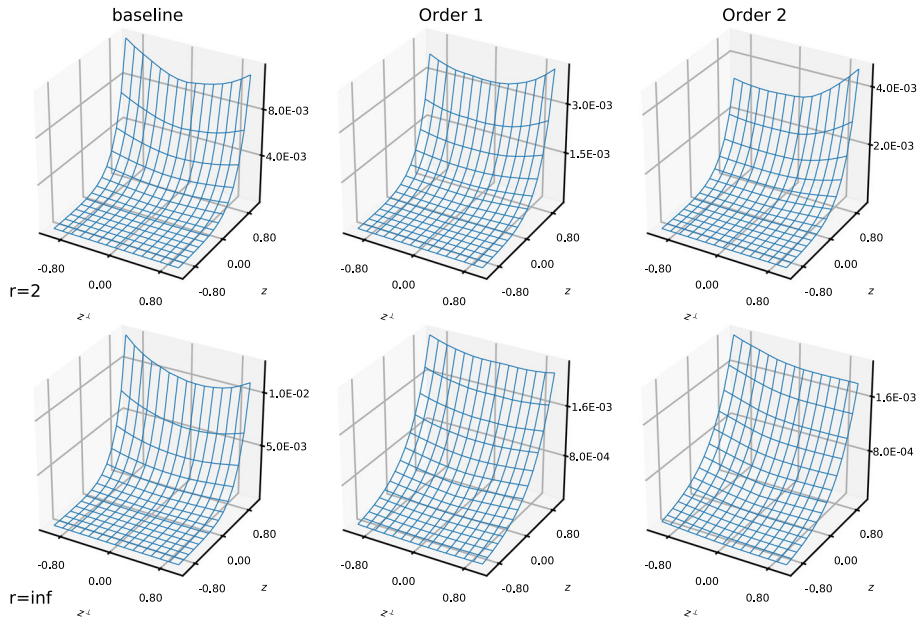

Fig. 1 Cross-entropy loss for one test image perturbed in an adversarial direction z and a random one $z^{\perp}$ that is perpendicular to the adversarial direction. Top $r = 2$, bottom $r =$inf. The first-order regularization effect is manifested in a reduction in the loss function for perturbed images. The second-order regularization effect, if still noticeable, is smaller in comparison and tends to reduce slightly the curvature. Notice the difference of scale from plot to plot

Order 2 method is small compared to the Order 1 procedure and mainly tends to reduce the curvature of the surfaces close to the *clean* image. Hence, one would expect better robustness of Order 2 when considering directions not aligned with the adversarial one. In this sense, a more thorough study of the robustness induced by the Order 2 procedure away from the PGD line of attack would be interesting but outside of the scope of this work.

6 Conclusions

In this paper, we have established a series of connections between distributionally robust learning as modeled by a min-max problem of the form (1.5) and regularized risk minimization problems on the parameters of a deep ResNet neural network. To establish this connection, we study the max part of the min-max problem using tools from optimal transport theory and identify its leading-order terms as a function of δ, i.e., the power of the adversary. We remark that this approach is not restricted to adversarial problems on deep neural networks, and in particular can be used in other learning settings as long as the dependence of the loss function on the input data is regular enough. The specific ResNet deep neural network structure, however, allows us to interpret the resulting regularization problems as mean-field optimal control problems. In turn, these control problems suggest, through their associated Pontryagin maximum principles, a family of algorithms for the training of robust neural networks which can avoid the computation of data perturbations during training. A key property of the resulting control problems is that they are scalable, and in particular the number and dimension of state variables

is within a dimension-free factor of the dimensions of the original (unrobust) learning problem.

Some interesting research directions that stem from this work include: (1) Studying the type of regularity enforced on the input-to-output mappings by the regularization problems discussed in this paper. (2) The analysis of other distributionally robust problems where for example their objective target may contain an entropic term (as motivated in [11]). (3) In general, the use of tools from optimal control theory for the robust training of a wider class of neural networks. (4) The study of adversarial problems in other learning settings of interest where specific structure in the models may be exploited to get novel theoretical or algorithmic insights.

Acknowledgements
The authors would like to thank two anonymous reviewers for their positive and constructive feedback. The authors would like to thank Leon Bungert for enlightening conversations and for providing them with many useful references. NGT was supported by NSF-DMS grant 2005797 and would also like to thank the IFDS at UW-Madison and NSF through TRIPODS grant 2023239 for their support. Part of this work was completed, while NGT was visiting the Simons Institute to participate in the program "Geometric Methods in Optimization and Sampling" during the Fall of 2021. NGT would like to thank the institute for hospitality and support.

Author details
[1]Department of Mathematics, University College London, London, UK, [2]Department of Statistics, University of Wisconsin-Madison, Madison, USA.

Received: 13 September 2021 Accepted: 10 June 2022 Published online: 8 August 2022

References

1. Ambrosio, L., Gigli, N., Savaré, G.: Gradient flows in metric spaces and in the space of probability measures, 2nd edn. Lectures in Mathematics ETH Zürich. Biruser Verlag, Basel (2008)
2. Belloni, A., Chernozhukov, V., Wang, L.: Square-root lasso: pivotal recovery of sparse signals via conic programming. Biometrika **98**(4), 791–806 (2011)
3. Ben-Tal, A., den Hertog, D., Waegenaere, A.D., Melenberg, B., Rennen, G.: Robust solutions of optimization problems affected by uncertain probabilities. Manag. Sci. **59**(2), 341–357 (2013)
4. Blanchet, J., Kang, Y., Murthy, K.: Robust Wasserstein profile inference and applications to machine learning. J. Appl. Probab. **56**(3), 830–857 (2019)
5. Blanchet J, Murthy K, Nguyen VA. Statistical analysis of wasserstein distributionally robust estimators. 2021
6. Carlini N, Athalye A, Papernot N, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On Evaluating Adversarial Robustness arXiv:1902.06705 [cs, math] (2019)
7. Carlini N and Wagner D: Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, San Jose, CA, USA, May 2017. IEEE
8. Carmona R , Delarue F. *Probabilistic Theory of Mean Field Games with Applications II: mean field games with common noise and master equations*, volume 84. Springer, 2018
9. Chen R, Paschalidis IC Distributionally robust learning. *Foundations and Trends®in Optimization*, 4(1-2):1–243, 2020
10. Chen RTQ, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018
11. Dong Y, Deng Z, Pang T, J. Z. 0001, and H. S. 0006. Adversarial distributional training for robust deep learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H.-T. Lin, editors, *Advances in Neural Information Processing Systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020
12. Drucker, H., Le Cun, Y.: Improving generalization performance using double backpropagation. IEEE Trans. Neural Netw. **3**(6), 991–997 (1992)
13. Dudley, R.M.: Real analysis and probability. CRC Press (2018)
14. E W, Han J, Li Q. A Mean-field optimal control formulation of deep learning. arXiv:1807.01083 [cs, math] (2018)
15. Fawzi A, Fawzi H, Fawzi O. Adversarial vulnerability for any classifier. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018
16. Finlay C, Calder J, Abbasi B, Oberman , A: Lipschitz regularized deep neural networks generalize and are adversarially robust. 2018
17. Finlay, C., Oberman, A.M.: Scaleable input gradient regularization for adversarial robustness. Mach. Learn. Appl. **3**, 100017 (2021)
18. García Trillos N, Murray R. Adversarial classification: necessary conditions and geometric flows. arXiv:2011.10797, (2020)

19. Goodfellow I, Shlens J, Szegedy C: Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015
20. Haber, E., Ruthotto, L.: Stable architectures for deep neural networks. Inverse Problems **34**(1), 014004 (2017)
21. Hein M, Andriushchenko M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017
22. Jetley S, Lord N, Torr P. With friends like these, who needs adversaries? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018
23. Kuhn D, Esfahani P, Nguyen V, Shafieezadeh-Abadeh S. *Wasserstein Distributionally robust optimization: theory and applications in machine learning*, pages 130–166. 10 2019
24. Li, Q., Chen, L., Tai, C.W.E.: Maximum principle based algorithms for deep learning. J. Mach. Learn. Res. **18**(165), 1–29 (2018)
25. Lyu C, Huang K, Liang H-N. A unified gradient regularization family for adversarial examples. In *2015 IEEE International Conference on Data Mining*, pages 301–309, 2015
26. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 [cs, stat] (2019)
27. Moosavi-Dezfooli S-M , Fawzi A, Uesato J, Frossard P. Robustness via curvature regularization, and vice versa. In *2019 IEEE/CVF Conference on computer vision and pattern recognition (CVPR)*, pages 9070–9078, 2019
28. Pedregal P. Optimization, relaxation and young measures. *Bull. Amer. Math. Soc. (N.S.)*, 36(1):27–58, 1999
29. Ross AS , Doshi-Velez F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. (2018)
30. Roth K, Lucchi A, Nowozin S, Hofmann T. Adversarially robust training through structured gradient regularization. (2018)
31. Shafahi A, Najibi M, Ghiasi MA, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019
32. Thorpe M, van Gennip Y. Deep limits of residual neural networks. arXiv:1810.11741 [math.CA], 2018
33. Thorpe M, Wang B: Robust certification for laplace learning on geometric graphs. In *Proceedings of Machine Learning Research*, 2021
34. Tramèr, A. Kurakin F, Papernot N, Goodfellow I, Boneh D, McDaniel P. Ensemble adversarial training: attacks and defenses. arXiv:1705.07204 [cs, stat] (2020)
35. Villani C.: Topics in optimal transportation. Graduate Studies in Mathematics, vol. 58. American Mathematical Society, Providence, RI (2003)
36. Weng T-W , Zhang H, Chen P-Y, Yi J, Su D, Gao Y, Hsieh C-J, Daniel L. Evaluating the robustness of neural networks: an extreme value theory approach. In *International Conference on Learning Representations*, 2018
37. Wiesemann, W., Kuhn, D., Sim, M.: Distributionally robust convex optimization. Oper. Res. **62**, 1358–1376 (2014)
38. Wong E, Rice L, Kolter JZ. Fast is better than free: revisiting adversarial training. arXiv:2001.03994 [cs, stat] (2020)
39. Yang WH . On generalized holder inequality. 1991
40. Yeats EC , Chen Y, Li H. Improving gradient regularization using complex-valued neural networks. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on machine learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11953–11963. PMLR, 18–24 Jul 2021
41. Yong J, Zhou XY: *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 1999
42. Zhang D, Zhang T, Lu Y, Zhu Z, Dong B. You only propagate once: Accelerating adversarial training via maximal principle. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in neural information processing systems*, volume 32. Curran Associates, Inc., 2019

Publisher's Note