POLITECNICO DI TORINO

Master's degree course
in Mathematical Engineering

Master's Thesis

Training Kernel Neural ODEs with optimal control and Riemannian optimization

Supervisors
Prof. Fabio Nobile
Prof. Claudio Canuto

Candidate
Matteo Raviola

Accademic Year 2021-2022

To my family

Summary

Nowadays, Machine Learning pipelines permeate the scientific computing world. The flexibility of Neural Networks makes them a formidable tool to perform numerous kinds of tasks, however their training keeps proving to be a computationally challenging optimization problem. This thesis focuses on a specific kind of Neural ODEs, *Kernel Neural ODEs (KerODEs)*, where the usual parametric non-linearities are replaced by elements of a reproducing kernel Hilbert space (RKHS) fixed a priori. Classical training algorithms are based on a variant of stochastic gradient descent, coupled with the celebrated backpropagation algorithm for gradient computations. Though extremely versatile, these approaches potentially suffer from long computational times and/or high cost per iteration. We propose and numerically explore methodologies to overcome both of these issues for the optimization of KerODE parameters in the context of a regression task. In particular, we first exploit the dynamical systems perspective of Deep Learning to link the training problem to an optimal control problem and, in turn, reduce it to the solution of a two-point boundary value problem. Inspired by time-parallel ODE integration techniques, we develop a multi-grid algorithm to speed up optimization and numerically investigate its performance. As an alternative approach, we formally introduce a differential structure on the family of mappings realized by KerODEs, enabling the use of continuous Riemanninan optimization techniques to solve the training problem. This furnishes a new and compelling perspective on Neural ODEs as the realization of a Riemannian gradient/Newton flow which, in practice, leads to layer-by-layer optimization techniques thus alleviating the cost per iteration of classic approaches.

Acknowledgements

First and foremost, I would like to thank my advisors, Prof. Fabio Nobile and Prof. Claudio Canuto, for their precious supervision. I would like to particularly thank Prof. Nobile for giving me plenty of his time and the opportunity to work on my Master' thesis in the CSQI lab at EPFL. This has been an extremely formative experience and has given me a taste of how it feels to do research in a world leading scientific institution. I would also like to thank all the members of the CSQI lab with whom I have had many helpful conversations. Finally, I would like to thank my family and friends for their continuous support, and Alessia for her love and for always believing in me.

Contents

Introduction

The last decade has seen *Machine Learning (ML)* rise to the forefront of scientific research. This phenomenon is in large part due to the success of *Deep Learning (DL)* (Goodfellow et al. [2016]) which has become a primary tool in many modern ML tasks, such as regression, image classification and segmentation. This sub-field furnishes specialized models and algorithms to professionals of all backgrounds in order to satisfy the ever increasing demand from a very heterogeneous group of scientific communities to process data, be it either abundantly available or very scarce. The fundamental idea in DL is to employ regressors/classifiers $v : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output spaces, which feature a parametric and compositional structure. More precisely

$$v(\cdot, \boldsymbol{\theta}) = v_L(\cdot, \theta_L) \circ \ldots \circ v_1(\cdot, \theta_1), \tag{1}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_L)$ are parameters taking values in some parameter set Θ. A popular architecture is given by *Residual Neural Networks (ResNets)* introduced in He et al. [2016]. These essentially enforce the following structure

$$v_1(\cdot, \theta_l) = I + \tilde{v}_l(\cdot, \theta_l), \quad l = 1, \ldots, L, \tag{2}$$

I being the identity mapping. If one introduces step size $h > 0$ and defines (abusing notation) $\tilde{v}_l := \tilde{v}_l/h$, the action of the map in (1) on an input $x \in \mathcal{X}$ is described by the following discrete dynamical system

$$\begin{cases} z_{l+1} &= z_l + h\tilde{v}_l(z_l, \theta_l), \\ z_0 &= x. \end{cases} \tag{3}$$

The key insight is that the above can be seen as an explicit Euler discretization of the continuous-time dynamical system

$$\begin{cases} \dot{z}_t &= \tilde{v}_t(z_t, \theta(t)), \quad t \in (0,1] \\ z_0 &= x. \end{cases} \tag{4}$$

This is the essence of the so-called dynamical systems point of view on DL which was independently put forth by the seminal works of E [2017] and Chen et al. [2018]. Recently, following the same research direction, Owhadi [2020] introduced continuous-time Neural Networks of the structure shown above where, instead of the common parametric nonlinearities populating the vast majority of architectures, the right-hand side $\tilde{v}_t$ is taken

in a reproducing kernel Hilbert space (RKHS). These are the architectures which we focus on in this thesis and refer to them as *Kernel Neural ODEs (KerODEs)*.

Usually, the network parameters are chosen according to some performance metric (commonly referred to as *loss* function or *cost* function) $\mathcal{J} : \Theta \to \mathbb{R}$ which is task-specific. For instance, for a regression task, which is the main focus of the thesis, the most common loss function is the empirical mean square error. Computing the optimal parameters, that is solving

$$\boldsymbol{\theta}^\star \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{J}(\boldsymbol{\theta}), \tag{5}$$

is commonly referred to as the *training* phase of the Neural Network and represents one of the crucial steps for providing an accurate and useful model for the task at hand. From an optimization point of view, training the network is a very hard problem. This is because the loss function $\mathcal{J}$ is usually highly non-convex and Θ is very high dimensional. The most popular strategy in order to tackle this optimization problem consists in stochastic gradient-based optimization starting from random initialization, coupled with the well-known backpropagation algorithm (Rumelhart et al. [1986]). The backpropagation algorithm works by computing the gradient of the loss function with respect to each parameter by the chain rule, computing the gradient one layer at a time, iterating backward from the last layer to avoid redundant calculations of intermediate terms in the chain rule. The gradient-based approach, although extremely successful, is affected by a number of issues, among which the unstable gradient problem (Nielsen [2018]), that essentially refers to the behaviour where different layers *learn* (i.e. are optimized) at vastly different speeds, and the vanishing gradient problem (Pascanu et al. [2013]), which instead refers to the phenomenon where the gradient of the loss is vanishingly small, effectively preventing the parameters from changing their value. These two issues often lead to poor convergence speed to a good set of parameters. Furthermore, each gradient step involves a forward propagation, namely a model evaluation, and a backward one, i.e. the chain rule computation. The larger the model is, the more costly these operations are.

The dynamical system point of view on Deep Learning has also been leveraged for the design of optimization algorithms of DNNs. Indeed, it has been pointed out by Li et al. [2017] that training Deep Neural Networks, bearing in mind that the latter can be seen as discretizations of continuous-time models, shares striking similarities with the solution of classical optimal control problems. In particular, Li et al. [2017] showed that, in this context, one can derive backpropagation from the well-known Pontryagin Maximum Principle (PMP).

The objective of this thesis, is to study the learning problem in the context of Kernel Neural ODEs, shed light on the links with optimal control and exploit it in order to devise training algorithms which speed up the training phase. In particular, we show that training the network is equivalent to solving a two-point boundary value problem (TPBVPs) and we will explore training strategies inspired by time-parallel integration techniques for ODEs.

As a further step, we investigate the link between the learning problem with Kernel Neural ODEs and variational methods for sampling based on Optimal Transport (OT). These techniques tackle the problem of drawing samples from a target distribution by recasting it as an optimization problem. In particular, we consider the *Stein Variational*

Gradient descent (SVGD) introduced by Liu and Wang [2016] which iteratively transports a set of particles to match the target distribution by applying a form of functional gradient descent, with steps taken in a given RKHS, that minimizes the KL Kullback–Leibler (KL) divergence. The follow up works by Duncan et al. [2019] and Nüsken and Renger [2021] then endow the space of probability measures with a Riemannian manifold structure (effectively formally translating the classic works of Otto [2001] in the context of Wasserstein spaces in the RKHS context) in order to recast SVGD as a gradient flow on the KL divergence. In this thesis we highlight some similarities the SVGD shares with KerODEs and the optimal control formulation of the training problem. The fundamental difference lies in the fact that those works take the point of view of the probability measures being transported, while we are interested in the maps which transport the measures. Hence, following Duncan et al. [2019] we introduce a formal Riemannian manifold structure on the set $\mathcal{M}$ of mapping realized by KerODEs. This enables the exploration of Riemannian optimization algorithms, both of first and second order, in order to solve the learning problem. In practice, this leads to layer by layer optimization techniques which reduce the cost per iteration of the optimization procedure.

The thesis is structured as follows. In Chapter 1 we review the necessary background material. In particular, we introduce Optimal Control and the two main approaches based on the Pontryagin Maximum Principle and Hamilton-Jacobi-Bellman equation. Then, we give a formal introduction to learning theory, with a focus on reproducing kernel Hilbert spaces of which we present the main properties. The last part of the background material concerns instead ML, DL and the dynamical systems perspective on DL.

The second Chapter is the theoretical heart of the thesis. First, we introduce KerODEs and the learing problem we want to tackle. Then we show that, in the continuous-time setting, solutions to the problem exist. Furthermore, leveraging the PMP, we recast the learning problem as the solution of a TPBVP. Finally, we introduce a suitable time discretization and show that the solution to the discrete-time problem converges in a suitable sense to the solution of the continuous-time problem. We then make the connection between the infinite-particle limit of the Optimal Control problem and Optimal Transport in the context of reproducing kernel Hilbert spaces as introduced by Duncan et al. [2019]. This prompts the introduction of the manifold $\mathcal{M}$ of diffeomorphisms realized by KerODEs and the statement of the learning problem as an optimization problem on $\mathcal{M}$. We consequently exploit this point of view to identify KerODEs as both first and second order geometric flows on the loss functional.

In Chapter 3 we present concrete training algorithms inspired by the theory developed in Chapter 2. In particular, we first propose a preconditioned version of gradient descent. Then, we introduce a time-parallel training algorithm in both a single and multi-grid context. We next present a Riemannian gradient descent training algorithm along with a stochastic version. The last algorithm we consider is instead a Riemannian damped Newton. Finally, we explore the effectiveness of these training strategies on synthetic data. In particular, we show that the second order methods derived from the OC perspective improve upon simple gradient descent in terms of speed of convergence, however they still lack robustness. The training strategies inspired by the Riemannian optimization perspective, on the other hand, prove to be more robust in terms of problems they manage to find good solutions to. This, together with their simplicity, make Riemannian optimization a

very promising approach.

Chapter 1

Background

1.1 Optimal Control theory

In this section, we will introduce some classical control theory and related discussions, which will be useful later for the study and design of Neural Networks. The main references for this section are Liberzon [2012], Chachuat [2007] and Carlsson et al. [2018].

The fundamental mathematical objects we are interested in are Ordinary Differential Equations (ODEs), which take the following form

$$\begin{cases} \dot{x}_t = f(x_t, t), & t \in (0, T], \\ x_0 = \bar{x}, \end{cases} \tag{1.1}$$

where $T > 0$ and $f : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is commonly denominated *flux*. These objects have been extensively studied throughout the centuries by mathematicians and are ubiquitous both in pure and applied mathematics. In general, we say that a solution to (1.1) is any function $x : [0, T] \to \mathbb{R}^d$ such that

$$x_t = x_0 + \int_0^t f(x_s, s) ds. \tag{1.2}$$

Note that if such function exists then $x \in C([0, T]; \mathbb{R}^d)$ and $\dot{x}_t = f(x_t, t)$, provided f is well behaved.

To guarantee existence and uniqueness of solutions on $[0, T]$ for (1.1) we need to impose some regularity conditions on the right-hand side f. A standard assumption in this regard is that f is continuous on $\mathbb{R}^d \times [0, T]$ and uniformly Lipschitz continuous with respect to x on $\mathbb{R}^d \times [0, T]$, namely there exists a constant $L_f > 0$ such that

$$|f(y, t) - f(x, t)| \leq L_f \|y - x\|_2, \quad \forall\, (y, t), (x, t) \in \mathbb{R}^d \times [0, T]. \tag{1.3}$$

Then, owing to the Picard-Lindelöf theorem, a solution to (1.1) exists and is unique for all $\bar{x} \in \mathbb{R}^d$. Here we are particularly interested in the ability to *control* such solutions. The intuition behind this is perhaps best understood with a practical example.

Example 1.1.1 (Evans [2013])**.** Suppose we own a factory whose output can be controlled. Let us further assume that we can consume some fraction of our output at each time, and likewise can reinvest the remaining fraction which would lead to an increased output. Then, we can construct a mathematical model by setting

- x_t to be the amount produced at time $t \geq 0$,
- β_t to be fraction of output reinvested at time $t \geq 0$.

In this context, x_t is said to be the *state*, while β_t the *control*. The latter is subject to the obvious constraint that $\beta_t \in [0,1]$ for all $t \geq 0$. Given such a control, the corresponding dynamics are provided by the ODE

$$\begin{cases} \dot{x}_t &= k\beta_t x_t, \quad t \in (0,T], \\ x_0 &= \bar{x}, \end{cases} \tag{1.4}$$

the constant $k > 0$ modelling the growth rate of our reinvestment and $\bar{x}$ a given initial state for the system.

This example captures the fundamental elements which characterize a control problem, i.e. the interplay between *state* and *control* variables. The latter are design variables which we assume to be able to modify at will, while the former represent the response of the system as a result of the chosen control. From this point onward, we will write $x = (x^1, \dots, x^d)^\top$ for the (dependent) state function, $\dot{x} = (\dot{x}^1, \dots, \dot{x}^d)^\top$ for their time derivatives, and $\beta = (\beta^1, \dots, \beta^m)^\top$ for the control function. In order to make this precise, let us further define the space of admissible controls on the sub-interval $[t,T]$

$$\mathcal{B}[t,T] = \{\beta : [t,T] \to B \mid \text{regularity condition on } \beta\}. \tag{1.5}$$

This entails that the controls will take values in some control set $B \subset \mathbb{R}^m$, usually a closed subset of $\mathbb{R}^m$ which can be the entire $\mathbb{R}^m$; in principle B can also vary with time, but here we take it to be fixed. Moreover, for the sake of simplicity, if not stated otherwise, we will consider B to be compact and $\mathcal{B}[t,T] = C([t,T];B)$. Finally, we will sometimes use $b = (b^1, \dots, b^m)^\top \in B$ to denote the control variables for some fixed time instant, and, with a little abuse of notation, $x = (x^1, \dots, x^d)^\top \in \mathbb{R}^d$ will denote the state variables for some fixed time instant. With this notation, the control systems that we want to study take the form

$$\begin{cases} \dot{x}_t &= f(x_t, \beta_t, t), \quad t \in [0,T], \\ x_0 &= \bar{x}, \end{cases} \tag{1.6}$$

where now the flux is a function $f : \mathbb{R}^d \times B \times [0,T] \to \mathbb{R}^d$. Sometimes we will refer to the solution of the above system as x^β in order to make its dependence on the specific control function explicit. As the initial condition $x_0 = \bar{x}$ is given while x_T is not, these problems are refereed to in the literature as *fixed-time, free-endpoint problems*. Note that the model in Example 1.1.1 fits into our general framework for $d = m = 1$, once we set

$$B = [0,1] \quad \text{and} \quad f(x,t) = kbx. \tag{1.7}$$

To guarantee local existence and uniqueness of its solutions, we can impose conditions on f and β that let us invoke the previous existence and uniqueness result for the right-hand side

$$\bar{f}(x,t) := f(x,\beta_t,t). \tag{1.8}$$

Here is one such set of assumptions which, although not the weakest possible, is adequate for our purposes: $f(x,b,t)$ is continuous in t, b and C^1 in x and $f_x(x,b,t)$ is bounded uniformly in t and b. Note that this assumptions imply that the following Lipschitz property holds

$$|f(y,b,t) - f(x,b,t)| \leq L_f \|y-x\|_2, \quad \forall\, (y,b,t),(x,b,t) \in \mathbb{R}^d \times B \times [0,T], \tag{1.9}$$

for some $L_f > 0$.

1.1.1 Setting up an Optimal Control problem

To give some intuition on Optimal Control and to introduce the basic concepts let us consider Example 1.1.1 again.

Example 1.1.1 (Continued)**.** Recall that in this imaginary setting we own a factory whose output we can control. With this premise, any sensible owner should then be concerned with controlling the factory in a way that benefits his needs the most. Assuming, for instance, that our satisfaction is directly proportional to the amount consumed, the goal then becomes maximizing the total consumption of the output, our consumption at a given time t being $(1-\beta_t)x_t$, i.e.

$$J(\beta) := \int_0^T (1-\beta_t)x_t dt. \tag{1.10}$$

In addition, we may be concerned with the amount of wealth x_T we leave behind us at time T, which leads to the modified cost functional

$$J(\beta) := \int_0^T (1-\beta_t)x_t dt + \gamma x_T, . \tag{1.11}$$

$\gamma > 0$ being a parameter controlling the trade-off between consumption and heritage.

In general, a *performance criterion* (also called *cost functional*, or simply *cost*) must be specified for evaluating the performance of a system quantitatively, i.e. we can state our problem in Optimal Control terms as the minimization[1] of an objective functional $J : \mathcal{B}[0,T] \to \mathbb{R}$. The latter can be defined in the so-called *Lagrange form*

$$J(\beta) := \int_0^T \ell(x_t^\beta, \beta_t, t)dt, \tag{1.12}$$

[1]Note that in Example 1.1.1 we take the perspective of a maximization problem which, however, can always be turned into a minimization problem as $\max J = -\min(-J)$.

where the running cost $\ell : \mathbb{R}^d \times B \times \mathbb{R} \to \mathbb{R}$ is assumed to be defined and continuous, together with its partial derivatives $\ell_x(x, b, t)$. The objective functional may as well be specified in the *Mayer form*,

$$J(\beta) := h(x_T^\beta, T), \tag{1.13}$$

with $h : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ being a real-valued function which, again, shall be assumed to be defined and continuous along with its partial derivatives $h_x(x, t)$. More generally, we may consider the *Bolza form* which corresponds to the sum of an integral term and a terminal term as

$$J(\beta) := \int_0^T \ell(x_t^\beta, \beta_t, t)dt + h(x_T^\beta, T). \tag{1.14}$$

Interestingly enough, Mayer, Lagrange and Bolza problem formulations can be shown to be theoretically equivalent (see, e.g., Chapter 3 in Liberzon [2012]). Hence, in the following we will stick to the latter as they are often amenable to more natural interpretations.

A mathematical setting for optimally controlling the solution to a deterministic ODE is then to solve

$$\inf_{\beta \in \mathcal{B}[0,T]} J(\beta). \tag{1.15}$$

Similar to problems of the calculus of variations, we shall say that J assumes its minimum value at $\beta^\star$ provided that

$$J(\beta^\star) \leq J(\beta) \quad \forall\, \beta \in \mathcal{B}[0, T], \tag{1.16}$$

in which case the infimum in (1.15) can be replaced by a minimum. This assignment is global in nature and does not require consideration of a norm. The statement (1.16) is global in nature and does not require to introduce a topology on $\mathcal{B}[0, T]$. On the other hand, to describe local minima of J we need to endow $\mathcal{B}[0, T]$ with a topology or even better a structure of normed vector space. Having chosen the class $\mathcal{B}[0, T]$ of admissible controls to be continuous function with values in B, it is natural to choose the supremum norm

$$\|\beta\|_{L^\infty} = \sup_{t \in [0,T]} \|\beta_t\|_2 \,. \tag{1.17}$$

Then, a local minimum $\beta^\star$ of J is such that

$$\exists\, \delta \text{ s.t.} \quad J(\beta^\star) \leq J(\beta) \quad \forall\, \beta \in \mathrm{Ball}_\delta(\beta^\star) \cap \mathcal{B}[0, T], \tag{1.18}$$

where $\mathrm{Ball}_\delta(\beta^\star) = \{\beta \in C([0, T]; \mathbb{R}^m) \mid \|\beta - \beta^\star\|_{L^\infty} < \delta\}$. Finally, we say that a (local) minimum $\beta^\star$ of (1.14) is internal if there exists a $\delta > 0$ such that $\mathrm{Ball}_\delta(\beta^\star) \subset \mathcal{B}[0, T]$.

The question then is how to characterize and compute the Optimal Control. Classically, this problem has been tackled by two approaches, the *Pontryagin maximum principle*, and *dynamic programming*, which we shall introduce in the subsequent sections. Moreover, for simplicity in the following we are going to consider autonomous systems where the the ODE right-hand-side f, the running cost ℓ and the terminal cost h do not depend explicitly on time.

1.1.2 Pontryagin maximum principle

Before reviewing the Pontryagin Maximum Principle, let us introduce the variational approach (or Lagrange principle) which seeks a minimum of the cost with the dynamics as a constraint. Both of them, indeed, lead to the to the solution of a Hamiltonian system of ordinary differential equations which are intimately connected.

Theorem 1.1.1 (Euler-Lagrange equations)**.** *Consider the minimization problem (1.15) subject to (1.6). Suppose f satisfies the assumptions which lead to (1.9) and that it is differentiable in b as well. Assume further $\beta^\star \in \mathcal{B}[0,T]$ is a (local) internal minimizer for the problem, and let $x^\star \in C^1([0,T];\mathbb{R}^d)$ denote the corresponding state trajectory. Then, there is a function $p^\star \in C^1([0,T];\mathbb{R}^d)$ such that the triple $(x^\star, \beta^\star, p^\star)$ satisfies the Euler-Lagrange equations*

$$\dot{x}_t^\star = f(x_t^\star, \beta_t^\star), \qquad t \in [0,T], \qquad x_0^\star = \bar{x}, \tag{1.19}$$

$$\dot{p}_t^\star = \ell_x(x_t^\star, \beta_t^\star) - (p_t^\star)^\top f_x(x_t^\star, \beta_t^\star), \qquad t \in [0,T], \qquad p_T^\star = -h_x(x_T^\star), \tag{1.20}$$

$$0 = f_b(x_t^\star, \beta_t^\star)^\top p_t^\star - \ell_b(x_t^\star, \beta_t^\star), \qquad t \in [0,T]. \tag{1.21}$$

Proof. Define the Lagrangian $\mathcal{L} : C^1([0,T];\mathbb{R}^d) \times \mathcal{B}[0,T] \times C^1([0,T];\mathbb{R}^d) \to \mathbb{R}$

$$\mathcal{L}(x,\beta,p) = h(x_T) + \int_0^T \ell(x_t,\beta_t)dt + \int_0^T p_t^\top \left(\dot{x}_t - f(x_t,\beta_t)\right) dt. \tag{1.22}$$

associated to the minimization problem. Then, the first variations of the Lagrangian provide necessary conditions for optimality:

$$\mathcal{L}_p(x^\star,\beta^\star,p^\star) = 0, \quad \mathcal{L}_x(x^\star,\beta^\star,p^\star) = 0, \quad \mathcal{L}_\beta(x^\star,\beta^\star,p^\star) = 0. \tag{1.23}$$

Consider a one-parameter family of controls $\beta = \beta^\star + \varepsilon v$ for $v \in C([0,T];\mathbb{R}^m)$ and note that the assumption that $\beta^\star$ is an internal minimizer implies that $\beta \in \mathcal{B}[0,T]$ for ε small enough. Then, the function

$$L(\varepsilon) = \mathcal{L}(x^\star, \beta^\star + \varepsilon v, p^\star) \tag{1.24}$$

features a local minimum at $\varepsilon = 0$. Imposing the first order condition $L'(0) = 0$ yields

$$\begin{aligned} 0 &= \int_0^T v_t^\top \ell_b(x_t^\star,\beta_t^\star)dt - \int_0^T v_t^\top f_b(x_t^\star,\beta_t^\star)^\top p_t^\star dt \\ &= \int_0^T v_t^\top \left(f_b(x_t^\star,\beta_t^\star)^\top p_t^\star - \ell_b(x_t^\star,\beta_t^\star)\right) dt. \end{aligned} \tag{1.25}$$

As $f_b(x_t^\star,\beta_t^\star)^\top p_t^\star - \ell_b(x_t^\star,\beta_t^\star)$ is continuous we may conclude that (1.21) holds by the arbitrariness of v. Similarly, set $L(\varepsilon) = \mathcal{L}(x^\star + \varepsilon v, \beta^\star, p^\star)$ where we take $v \in C^1([0,T];\mathbb{R}^d)$ such that $v_0 = 0$ in order to ensure $x_0^\star + \varepsilon v_0 = \bar{x}$. Imposing $L'(0) = 0$ gives

$$\begin{aligned} 0 &= v_T^\top h_x(x_T^\star) + \int_0^T v_t^\top \ell_x(x_t^\star,\beta_t^\star)dt + \int_0^T (p_t^\star)^\top \left(\dot{v}_t - f_x(x_t^\star,\beta_t^\star)v_t\right) dt \\ &= v_T^\top (h_x(x_T^\star) + p_T^\star) + \int_0^T v_t^\top \left(\ell_x(x_t^\star,\beta_t^\star) - f_x(x_t^\star,\beta_t^\star)^\top p_t^\star - \dot{p}_t^\star\right) dt. \end{aligned} \tag{1.26}$$

As the integrand is continuous the above immediately implies (1.20). Finally, taking again $v \in C^1([0,T];\mathbb{R}^d)$, setting $L(\varepsilon) = \mathcal{L}(x^\star, \beta^\star, p^\star + \varepsilon v)$ and imposing $L'(0) = 0$ we obtain

$$\int_0^T v_t^\top \left(\dot{x}_t - f(x_t, \beta_t)\right) \tag{1.27}$$

which, by continuity of $f(x_t, \beta_t) - \dot{x}_t$, is equivalent to (1.19). □

Remarks.

- The optimality conditions consist of d algebraic equations (1.21), together with $2d$ ODEs (1.19) - (1.20) and their respective boundary conditions. Hence, the Euler-Lagrange equations provide a complete set of necessary conditions. However, the boundary conditions for (1.19) and (1.20) are split, i.e., some are given at $t = 0$ and others at $t = T$. Such problems are known as two-point boundary value problems (TPBVPs) and are notably more difficult to solve than initial value problems (IVPs).

- It is convenient to introduce the Hamiltonian function $H : \mathbb{R}^d \times B \times \mathbb{R}^d$ associated with the Optimal Control problem (1.6) - (1.15), by adjoining the right-hand side of the differential equations to the cost integrand as

$$H(x, b, p) = p^\top f(x, b) - \ell(x, b) \tag{1.28}$$

  Thus, the Euler-Lagrange equations (1.19)–(1.21) can be rewritten as

$$\dot{x}_t^\star = H_p(x_t^\star, \beta_t^\star, p_t^\star), \qquad x_0^\star = \bar{x}, \tag{1.29}$$
$$\dot{p}_t^\star = -H_x(x_t^\star, \beta_t^\star, p_t^\star), \qquad p_T^\star = -h_x(x_T^\star), \tag{1.30}$$
$$0 = H_b(x_t^\star, \beta_t^\star, p_t^\star). \tag{1.31}$$

- As expected by classical theory of Hamiltonian systems, H yields a *first integral* to the TPBVP (1.29)-(1.31) assuming $\beta^\star$ is continuously differentiable. Indeed, the variation of the Hamiltonian function along an optimal trajectory is given by

$$\begin{aligned} &\frac{d}{dt} H(x_t^\star, \beta_t^\star, p_t^\star) \\ =&H_x(x_t^\star, \beta_t^\star, p_t^\star)^\top \dot{x}_t^\star + H_b(x_t^\star, \beta_t^\star, p_t^\star)^\top \dot{\beta}_t^\star + H_p(x_t^\star, \beta_t^\star, p_t^\star)^\top \dot{p}_t^\star \\ =&H_x(x_t^\star, \beta_t^\star, p_t^\star)^\top H_p(x_t^\star, \beta_t^\star, p_t^\star) - H_p(x_t^\star, \beta_t^\star, p_t^\star)^\top H_x(x_t^\star, \beta_t^\star, p_t^\star) = 0. \end{aligned} \tag{1.32}$$

As Theorem 1.1.1 states, the conditions given are only necessary for local extrema of J. The question then is if and under which conditions we may characterize global extrema. An answer to this question is provided by the following theorem.

Theorem 1.1.2 (Pontryagin Maximum Principle (PMP))**.** *Let $\beta^\star \in \mathcal{B}[0,T]$ be an Optimal Control and let $x^\star \in C^1([0,T];\mathbb{R}^d)$ be the corresponding optimal state trajectory. Then, there exists a function $p^\star \in C^1([0,T];\mathbb{R}^d)$ having the following properties:*

1. *$x^\star$ and $p^\star$ satisfy the canonical equations*

$$\dot{x}_t^\star = H_p(x_t^\star, \beta_t^\star, p_t^\star) \tag{1.33}$$
$$\dot{p}_t^\star = -H_x(x_t^\star, \beta_t^\star, p_t^\star) \tag{1.34}$$

with the boundary conditions

$$x_0^\star = \bar{x}, \quad p_T^\star = -h_x(x_T^\star). \tag{1.35}$$

2. *For all fixed t, the function $b \mapsto H(x_t^\star, b, p_t^\star)$ has a global maximum at $b = \beta_t^\star$, i.e.*

$$\beta_t^\star \in \arg\max_{b \in B} H(x_t^\star, b, p_t^\star). \tag{1.36}$$

The proof of this result is highly non trivial and is out of the scope of this chapter, hence we refer the reader to Liberzon [2012]. It is noteworthy that a necessary condition for the triple $(x^\star, \beta^\star, p^\star)$ to give a global minimum of J is that $\beta_t^\star$ be a global maximum of the function $H(x_t^\star, b, p_t^\star)$ for all $t \in [0, T]$. In some cases, as we shall see in later chapters, one can express $\beta_t^\star$ as a function of x_t and p_t from (1.36), and then substitute into (1.33)-(1.34) to get a TPBVP in the variables x and p only. Finally, note that (1.31) is implied by (1.36) whenever $\beta^\star$ is an internal minimum and H is b-differentiable, so that the Lagrange principle is implied by the Pontryagin Maximum Principle.

1.1.3 Dynamic programming

The dynamic programming view to solve Optimal Control problems is based on the idea to track the optimal solution backwards. In order to do this, we embed the usual control problem on $[0, T]$ into a larger family of similar problems, by varying the starting times and starting points and defining

$$J(\beta; \bar{x}, t) := \int_t^T \ell(x_s, \beta_s) ds + h(x_T) \tag{1.37}$$

with

$$\begin{cases} \dot{x}_s = f(x_s, \beta_s), & s \in [t, T], \\ x_t = \bar{x}, \end{cases} \tag{1.38}$$

where $\beta \in \mathcal{B}[t, T]$. In this setting, it is convenient to take $\mathcal{B}[t, T]$ to be the set of measurable functions with values in B, so that $\mathcal{B}[0, T] = \cup_{[t,s] \in \Delta} \mathcal{B}[t, s]$ where Δ is any countable partition of $[0, T]$. Note that, in order to be completely explicit, we will sometimes refer to the solution of (1.38) as $x^{\beta, \bar{x}, t}$. The fundamental mathematical object in this context is represented by the *value function*.

Definition 1.1.1 (Value function)**.** Let $\bar{x} \in \mathbb{R}^d$, $t \in [0, T]$, then define the value function as the infimum of the cost functional if the dynamics starts in $\bar{x}$ at time t, namely

$$V(\bar{x}, t) := \inf_{\beta \in \mathcal{B}[t,T]} J(\beta; \bar{x}, t) \tag{1.39}$$

Since at the final time the value function is trivially given by $V(\cdot, T) = h$ we can, recursively for small time steps of size $\Delta t > 0$ backwards, find the Optimal Control to go from each point $(\bar{x}, t)$ on the time level t to the time level $t + \Delta t$ with the value function $V(\cdot, t + \Delta t)$. This procedure leads to the characterization of the value function as the solution to the well known *Hamilton-Jacobi-Bellman* equation, which is a (highly) nonlinear partial differential equation.

Theorem 1.1.3 (Hamilton-Jacobi-Bellman (HJB) equation)**.** *Assume that the value function $V \in C^1(\mathbb{R}^n \times [0, T]; \mathbb{R})$, then it solves the nonlinear partial differential equation*

$$\partial_t V(\bar{x}, t) + \min_{b \in B} \left\{ \partial_x V(\bar{x}, t)^\top f(\bar{x}, b) + \ell(\bar{x}, b) \right\} = 0, \quad \bar{x} \in \mathbb{R}^d, t \in [0, T], \tag{1.40}$$

with the terminal condition

$$V(\cdot, T) = h. \tag{1.41}$$

Proof. We prove this assuming the infimum of the optimization problem is attained. Let $\bar{x} \in \mathbb{R}^d$, $t \in [0, T]$ and let $\Delta t > 0$ be given. Pick any parameter $b \in B$ and use the constant control $\beta \equiv b$ for times $s \in [t, t + \Delta t]$. Then, the definition of the value function implies

$$V(\bar{x}, t) \leq \int_t^{t+\Delta t} \ell(x_s^{\beta, \bar{x}, t}, b) ds + V(x_{t+\Delta t}^{\beta, \bar{x}, t}, t + \Delta t), \tag{1.42}$$

whence

$$\frac{V(x_{t+\Delta t}^{\beta, \bar{x}, t}, t + \Delta t)) - V(\bar{x}, t)}{\Delta t} + \frac{1}{\Delta t} \int_t^{t+\Delta t} \ell(x_s^{\beta, \bar{x}, t}, b) ds \geq 0. \tag{1.43}$$

Letting $\Delta t \to 0$ and recalling V is C^1 yields the differential form of the above inequality, namely[2]

$$\partial_t V(\bar{x}, t) + \partial_x V(\bar{x}, b)^\top f(\bar{x}, b) + \ell(\bar{x}, b) \geq 0, \tag{1.44}$$

where we exploited the fact that $x^{\beta, \bar{x}, t}$ is a solution to (1.38). As the above holds for any $b \in B$, we have that

$$\min_{b \in B} \left\{ \partial_t V(\bar{x}, t) + \partial_x V(\bar{x}, t)^\top f(\bar{x}, b) + \ell(\bar{x}, b) \right\} \geq 0. \tag{1.45}$$

Finally, let us prove that the above is an equality. Assume $\beta^\star$ is optimal starting in $\bar{x}$ at time t with corresponding optimal trajectory $x^\star$, and consider the following decomposition which, again, is a consequence of the definition of value function:

$$V(\bar{x}, t) = \int_t^{t+\Delta t} \ell(x_s^\star, \beta_s^\star) ds + V(x_{t+\Delta t}^\star, t + \Delta t). \tag{1.46}$$

If we let $b^\star = \beta_t^\star$, a similar argument to the one employed earlier yields

$$\partial_t V(\bar{x}, t) + \partial_x V(\bar{x}, t)^\top f(\bar{x}, b^\star) + \ell(\bar{x}, b^\star) = 0, \tag{1.47}$$

which proves the claim. □

[2] In order to avoid confusion, we use the more explicit notation $\partial_t V$ and $\partial_x V$ when referring to the derivatives of the value function, as opposed to the subscript notation adopted throughout the section.

Remark. Note that the HJB equation can be recast as

$$\partial_t V(\bar{x},t) - H^\star(\bar{x}, -\partial_x V(\bar{x},t)) = 0, \quad \bar{x} \in \mathbb{R}^d, t \in [0,T], \tag{1.48}$$

where

$$H^\star(\bar{x},p) := \max_{b\in B} H(\bar{x},b,p). \tag{1.49}$$

1.1.4 Characteristics and the Pontryagin Principle

The last remark hints at the possibility that the Pontryagin principle and the dynamic programming approach are somehow related as the same Hamiltonian maximization is involved in both. On the other hand, one would expect such relation to exist as they are both strategies to solve the same problem. The first step in this direction is to show that the characteristics of the HJB equation solve an Hamiltonian system.

Theorem 1.1.4. *Assume $V \in C^2$, $H \in C^1$ and define*

$$\dot{x}_t := H_p^\star(x_t, p_t), \tag{1.50}$$

with $p_t := -\partial_x V(x_t,t)$. Then, the characteristics of (1.48) satisfy the Hamiltonian system

$$\begin{aligned} \dot{x}_t &= H_p^\star(x_t,p_t), \\ \dot{p}_t &= -H_x^\star(x_t,p_t) \end{aligned} \tag{1.51}$$

Proof. The definition $\dot{x}_t = H_p^\star(x_t,p_t)$ implies by x-differentiation of the HJB equation that along the path (x_t,t) the following holds:

$$\begin{aligned} 0 &= \partial_{xt} V(x_t,t) - H_x^\star(x_t, -\partial_x V(x_t,t)) \\ &= \partial_{tx} V(x_t,t) + \partial_x^2 V(x_t,t) H_p^\star(x_t, -\partial_x V(x_t,t)) - H_x^\star(x_t, -\partial_x V(x_t,t)) \\ &= \frac{d}{dt} V_x(x_t,t) - H_x^\star(x_t, -\partial_x V(x_t,t)) \\ &= -\dot{p}_t - H_x^\star(x_t, -\partial_x V(x_t,t)) \end{aligned} \tag{1.52}$$

as $V \in C^2$, and the claim is proved. □

The next step is to relate the characteristics (x_t,p_t) to the solution of the Pontryagin Maximum Principle. But note first that the Hamiltonian $H^\star$ in general is not differentiable, even if f and ℓ are very regular: for instance given $\dot{x}_t = f(x_t)$ and $\ell(x,b) = xb$ implies for $B = [-1,1]$ that the Hamiltonian becomes $H^\star(x,p) = pf(x) - |x|$ which is only Lipschitz continuous if, for instance, f is differentiable and bounded. In fact, if f and ℓ are bounded differentiable functions the Hamiltonian $H^\star$ will always be Lipschitz continuous satisfying $|H(x,p) - H(y,q)| \le L(\|x-y\|_2 + \|p-q\|_2)$ for some $L > 0$.

Theorem 1.1.5. *Assume that f, h are x-differentiable in $(x,b^\star)$ and a control $b^\star$ is optimal for a point (x,p), i.e.*

$$p^\top f(x,\beta^\star) - \ell(x,\beta^\star) = H^\star(x,p). \tag{1.53}$$

Assume further that $H^\star$ is differentiable in the point or that $\beta^\star$ is unique. Then

$$\begin{aligned} H_p^\star(x,p) &= f(x,\beta^\star) \\ H_x^\star(x,p) &= p^\top f_x(x,\beta^\star) - \ell_x(x,\beta^\star) \end{aligned} \tag{1.54}$$

Proof. See Carlsson et al. [2018]. □

This Theorem shows that the Hamiltonian system (1.51) is the same as the system (1.33)-(1.36), given by the Pontryagin principle using the Optimal Control $\beta^\star$.

If $\beta^\star$ is not unique (i.e not a single point) the proof shows that (1.54) still holds for the Optimal Controls, so that $H_p^\star$ and $H_x^\star$ become set valued. We conclude that non unique local Optimal Controls $\beta^\star$ is the phenomenon that makes the Hamiltonian non differentiable in certain points. In particular a differentiable Hamiltonian gives unique Optimal Control fluxes $H_p^\star$ and $H_x^\star$, even if $\beta^\star$ is not a single point. If the Hamiltonian can be explicitly formulated, it is therefore often practical to use the Hamiltonain system formulation with the variables x and p, avoiding the control variable.

Finally, let us remark that the non linear Hamilton-Jacobi-Bellman partial differential approach has the theoretical advantage of well established theory and that a global minimum is found; its fundamental drawback is that it cannot be used computationally in high dimension $d \gg 1$, since the computational work increases exponentially with the dimension d as is customary when dealing with the numerical solution to PDEs. The Pontryagin principle, on the other hand, has the computational advantage of dealing with ODEs which are much better conditioned with respect to the dimensionality of the ambient space of many practical problems, so that for $d \gg 1$ one can often hope to get a solution, albeit in practice only local minima can be found computationally, often with some additional error introduced by a regularization method (Carlsson et al. [2018]).

1.2 Learning in Reproducing Kernel Hilbert Spaces

Reproducing Kernel Hilbert Spaces are spaces of functions widely used in Machine Learning tasks. Indeed, they are backed by a solid theoretical understanding and prove to be a convenient ambient space for many learning problems. In this work, they represent a fundamental building block for the construction of a large class of Artificial Neural Networks which will be introduced in the next chapter. Hence, in this section we are going to briefly introduce the problem of learning and review Reproducing Kernel Hilbert Spaces and their main properties. The main references for this section are Cucker and Zhou [2007], Álvarez et al. [2012] and Rosasco [2010].

1.2.1 The problem of learning and regularization

We begin by introducing the problem of *learning*. Let $\mathcal{X}$ be a metric space and $\mathcal{Y} = \mathbb{R}^n$. For convenience we will take $n = 1$ for the time being. Let μ be a Borel probability measure on $\mathcal{S} = \mathcal{X} \times \mathcal{Y}$ whose regularity properties will be assumed as required.

A central concept which will accompany us throughout the thesis is the generalization error (or least squares error) of a given function f , for $f : \mathcal{X} \to \mathcal{Y}$, defined by

$$\mathcal{E}_\mu(f) := \int_{\mathcal{S}} (f(x) - y)^2 d\mu(x, y). \tag{1.55}$$

This means that for each input $x \in \mathcal{X}$ and output $y \in \mathcal{Y}$, $(f(x) - y)^2$ is the error incurred through the use of f as a model for the process of producing y from $\mathcal{X}$, which is then averaged over $\mathcal{X} \times \mathcal{Y}$ according to μ. It can be shown that there exists a function f_μ, the so-called regression function, which achieves the minimal error and has the following representation

$$f_\mu(x) = \int_{\mathcal{Y}} y d\mu(y|x), \tag{1.56}$$

where $\mu(\cdot|x)$ denotes the conditional distribution on $\mathcal{Y}$ given $x \in \mathcal{X}$. Then, the generalization error features the following decomposition

$$\mathcal{E}_\mu(f) = \int_{\mathcal{X}} (f(x) - f_\mu(x))^2 d\mu_{\mathcal{X}}(x) + \sigma_\mu^2 \tag{1.57}$$

where

$$\sigma_\mu^2 = \int_{\mathcal{Y}} (y - f_\mu(x))^2 d\mu(x, y) \tag{1.58}$$

and $\mu_{\mathcal{X}}$ denotes the marginal distribution on $\mathcal{X}$, namely

$$\mu_{\mathcal{X}}(A) = \int_{A \times \mathcal{Y}} d\mu(x, y). \tag{1.59}$$

The latter measures the irreducible uncertainty which affects the learning problem and is sometimes called conditioning of μ, in connection to the classical condition number of matrices in linear algebra. The following statement from Cucker and Zhou [2007] encompasses the fundamental objective of learning theory:

> *"The goal is to "learn" (i.e., to find a good approximation of) f_μ from samples on $\mathcal{S}$."*.

A practical example of how the joint measure μ can arise is the following. Consider a function $g : \mathcal{X} \to \mathcal{Y}$ affected by noise represented by some random variable ε on $\mathcal{Y}$, and a random variable x on $\mathcal{X}$. In other words, the underlying process to be learned is $g(x) + \varepsilon$ and μ is the joint probability measure of the random vector $(x, g(x) + \varepsilon)$. Notice that assuming ε to have zero mean implies $g = f_\mu$, so that we achieve the optimal model once we know exactly the function g which generates the process. An interesting particular case is the one of vanishing irreducible uncertainty $\sigma_\mu = 0$, which coincides with the problem of approximating g. Indeed, the generalization error takes the form

$$\mathcal{E}_\mu(f) = \int_{\mathcal{X}} (f(x) - g(x))^2 d\mu_{\mathcal{X}}(x). \tag{1.60}$$

Given this framework, it is clear that dealing with a (in general) infinite-dimensional object μ is computationally impractical on one hand, and outright impossible if we do not have direct access to it. In other words, we can think of two settings:

- the process which gives rise to μ and, consequently, error (1.55), is unknown and accessible only through a finite sample;
- we can draw independent samples from μ at will, albeit arbitrarily computationally expensive.

The reason for this distinction lies in the fact that, in principle, the latter situation allows for a higher number of degrees of freedom which may, and should, be exploited when designing learning algorithms. This is the common situation when μ can be sampled through scientific computing pipelines involving, for instance, the solutions of given parametric PDEs. In both of these cases, however, we are concerned with dealing with a sample in $\mathcal{S}^N$, i.e.

$$\mathbf{s} = ((x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})) \in \mathcal{S}^N, \tag{1.61}$$

where $\mathcal{S}^N$ denotes the N-fold Cartesian product of $\mathcal{S}$. The sample $\mathbf{s}$ should then be thought of as a point-wise discretization of μ. The most natural way this may happen is if $\mathbf{s}$ is a collection of independent and identically distributed (i.i.d.) samples drawn from μ, namely

$$(x^{(i)}, y^{(i)}) \overset{\text{iid}}{\sim} \mu. \tag{1.62}$$

However this is not the only way to discretize a distribution. Indeed, one can employ, for instance, deterministic sets of points in the input space $\mathcal{X}$ in order to represent $\mu_{\mathcal{X}}$. One very popular approach of this kind is Quasi-Monte Carlo (QMC) sampling. QMC is an alternative approach to random sampling in order to generate points in domains of interest (Caflisch [1998]). Let, for example, $\mathcal{X}$ be a compact domain with $\mu_{\mathcal{X}}$ being the uniform distribution. Intuitively, one can think of a Quasi-Monte Carlo sampling as a way to "better cover" the domain than, say, drawing i.i.d. samples from $\mu_{\mathcal{X}}$. This can be made into a rigorous statement and one can prove there exist sequences of points which are optimal in the sense of their ability to cover the domain $\mathcal{X}$. This is indeed the approach we take for our numerical experiments in Chapter 3.

Let us define the *empirical error* of f (w.r.t. $\mathbf{s}$) to be

$$\mathcal{E}_{\mathbf{s}}(f) = \frac{1}{N}\sum_{i=1}^{N}(f(x^{(i)}) - y^{(i)})^2 = \int_{\mathcal{S}}(f(x) - y)^2 d\mu^{(N)}(x, y), \tag{1.63}$$

where

$$\mu^{(N)} := \frac{1}{N}\sum_{i=1}^{N}\delta_{(x^{(i)}, y^{(i)})} \tag{1.64}$$

denotes the empirical measure associated to $\mathbf{s}$. Note that by the law of large numbers we have $\mathcal{E}_{\mathbf{s}} \to \mathcal{E}_{\mu}$ as $N \to \infty$.

Learning does not take place in a vacuum and some structure needs to be present at the beginning of the process. Let $C(\mathcal{X}; \mathcal{Y})$ be the space of continuous functions on $\mathcal{X}$ with values in $\mathcal{Y}$ endowed with the norm

$$\|f\|_{\infty} = \sup_{x \in \mathcal{X}} \|f(x)\|_{\mathcal{Y}}. \tag{1.65}$$

We ambient the learning task in a normed subspace $(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$, the *hypothesis space*, continuously included in $C(\mathcal{X}; \mathcal{Y})$, where algorithms will work to find, as well as is possible, the best approximation for f_μ. Note that $\|f\|_{\mathcal{H}}$ is often taken as a measure of complexity of $f \in \mathcal{H}$.

Definition 1.2.1. Given a measure μ and a hypothesis space $\mathcal{H}$ we define the target function $f_{\mathcal{H}}$, if it exists, as an optimizer of

$$\min_{f \in \mathcal{H}} \mathcal{E}_\mu(f), \tag{1.66}$$

and the empirical target function $f_{\mathbf{s}}$, if it exists, as an optimizer of

$$\min_{f \in \mathcal{H}} \mathcal{E}_{\mathbf{s}}(f). \tag{1.67}$$

The existence of such optimizers is guaranteed under mild conditions on $\mathcal{H}$, provided one also restricts the minimization problems to a ball in $\mathcal{H}$ (see Chapter 1 in Cucker and Zhou [2007]). In practice, then, one looks for $f_{\mathbf{s}}$ in place of the unobtainable $f_{\mathcal{H}}$ and this is called *empirical risk minimization (ERM)* technique.

In the context of ERM, a popular technique is that of regularization whose goal is to restore the well-posedness (specifically, making the result depend smoothly on the data) of the ERM technique by effectively restricting the hypothesis space $\mathcal{H}$. One specific way of doing this is to introduce a penalization term proportional to the complexity of the candidate (empirical) target function f in our minimization as follows:

$$\mathcal{E}_{\mu,\gamma} = \mathcal{E}_\mu + \gamma \|f\|^2_{\mathcal{H}}, \quad \mathcal{E}_{\mathbf{s},\gamma} = \mathcal{E}_{\mathbf{s}} + \gamma \|f\|^2_{\mathcal{H}}, \tag{1.68}$$

where the regularization parameter γ controls the tradeoff between the two terms. This will then cause the minimization to seek out simpler functions, which incur less of a penalty. Regularization, as shown in Figure 1.1, provides one way to strike the appropriate balance in creating our model. It requires a (possibly large) class of models and a method for evaluating the complexity of each model in the class. The concept of kernels will provide us with a flexible, computationally feasible method for implementing this scheme.

1.2.2 Properties of reproducing kernel Hilbert spaces

Definition 1.2.2 (Mercer kernel)**.** Let $\mathcal{X}$ be a metric space. We say that $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is symmetric when $\mathcal{K}(x, x') = \mathcal{K}(x', x)$ for all $x, x' \in \mathcal{X}$ and that it is positive semi-definite when for all finite sets $\mathbf{x} = \{x_1, \ldots, x_N\}$ the $N \times N$ matrix $\mathbf{K}[\mathbf{x}]$, whose (i, j) entry is $\mathbf{K}[\mathbf{x}]_{ij} := \mathcal{K}(x_i, x_j)$, is positive semi-definite. We say that $\mathcal{K}$ is a Mercer kernel if it is continuous, symmetric, and positive semi-definite. The matrix $\mathbf{K}[\mathbf{x}]$ above is called the Gramian (or Gram matrix) of $\mathcal{K}$ at $\mathbf{x}$.

For the remainder of this section we fix a compact metric space $\mathcal{X}$ and a Mercer kernel $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Note that the positive semi-definiteness implies that $\mathcal{K}(x, x) \geq 0$ for each $x \in \mathcal{X}$. We define

$$C_{\mathcal{K}} := \sup_{x \in \mathcal{X}} \sqrt{\mathcal{K}(x, x)} < +\infty. \tag{1.69}$$

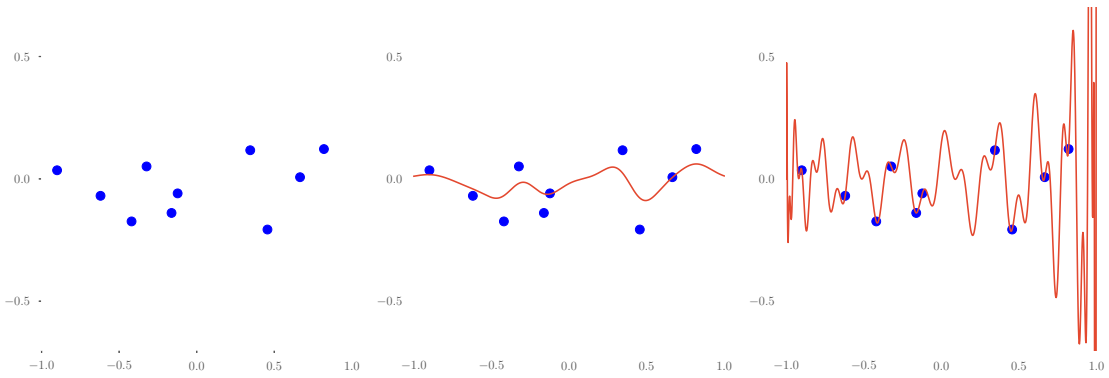

Figure 1.1: The role of regularization. Left: A sample from μ of size $N = 10$. Center: Smooth function that will likely be a good model for the process described by μ. Right: Function that is likely to describe the process described by μ poorly, yet minimizing the empirical error.

Then, $C_{\mathcal{K}} = \sup_{x,x' \in \mathcal{X}} \sqrt{\mathcal{K}(x,x')}$ as the positive semi-definiteness of the matrix $\mathbf{K}[\{x,x'\}]$ implies

$$\mathcal{K}(x,x')^2 \leq \mathcal{K}(x,x)\mathcal{K}(x',x'). \tag{1.70}$$

For $x \in \mathcal{X}$, we denote by $\mathcal{K}_x$ the function $\mathcal{K}(\cdot,x)$. The first main result we are going to state is given in the following theorem.

Theorem 1.2.1. *There exists a unique Hilbert space $(\mathcal{H}_{\mathcal{K}}, \langle\cdot,\cdot\rangle_{\mathcal{H}_{\mathcal{K}}})$ of functions on $\mathcal{X}$ with values in $\mathbb{R}$ satisfying the following conditions:*

1. *$\mathcal{K}_x \in \mathcal{X}$ for all $x \in \mathcal{X}$;*
2. *the span of the set $\{\mathcal{K}_x \mid x \in \mathcal{X}\}$ is dense in $\mathcal{H}_{\mathcal{K}}$;*
3. *$f(x) = \langle f, \mathcal{K}_x\rangle_{\mathcal{H}_{\mathcal{K}}}$ for all $f \in \mathcal{H}_{\mathcal{K}}$ and $x \in \mathcal{X}$.*

Moreover, $\mathcal{H}_{\mathcal{K}}$ consists of continuous functions and the inclusion $I_{\mathcal{K}} : \mathcal{H}_{\mathcal{K}} \to C(\mathcal{X};\mathbb{R})$ is continuous with $\|I_{\mathcal{K}}\| \leq C_{\mathcal{K}}$.

Proof. First, let H_0 be the span of the set $\{\mathcal{K}_x \mid x \in \mathcal{X}\}$. Given $f = \sum_{i=1}^{N} \mathcal{K}(x,x_i)\alpha_i, g = \sum_{i=1}^{M} \mathcal{K}(x,x'_j)\beta_j \in H_0$ define the inner product

$$\langle f,g\rangle_{H_0} = \sum_{i=1}^{N}\sum_{i=1}^{M} \alpha_i \mathcal{K}(x_i,x'_j)\beta_j. \tag{1.71}$$

Note that by the properties of the Gramian matrix this is trivially symmetric, bilinear, associative and positive semi-definite. To prove positive definiteness observe that if $\langle f,f\rangle_{H_0} = 0$ for some $f = \sum_{i=1}^{N} \mathcal{K}(x,x_i)\alpha_i \in H_0$, then the positive semi-definiteness of the Gramian associated to the set $\{x_1,\ldots,x_N\} \cup x'$ implies

$$\sum_{i,j=1}^{N} \alpha_i \mathcal{K}(x_i,x_j)\alpha_j + 2\varepsilon \sum_{i=1}^{N} \mathcal{K}(x',x_i)\alpha_i + \varepsilon^2 \mathcal{K}(x',x') \geq 0 \tag{1.72}$$

for all $\varepsilon \in \mathbb{R}$. The first term vanishes as it coincides with $\langle f, f\rangle_{H_0}$, hence taking ε arbitrarily small implies that $\sum_{i=1}^{N} \mathcal{K}(x', x_i)\alpha_i = 0$, i.e. $f = 0$ by arbitrariness of x'.

Let now $\mathcal{H}_\mathcal{K}$ be the completion of H_0 with the associated norm. It is easy to check that $\mathcal{H}_\mathcal{K}$ satisfies the three conditions in the statement. We now need to prove uniqueness. So, assume H is another Hilbert space of functions on $\mathcal{X}$ satisfying the conditions noted. We want to show that

$$H = \mathcal{H}_\mathcal{K} \quad \text{and} \quad \langle \cdot, \cdot \rangle_H = \langle \cdot, \cdot \rangle_{\mathcal{H}_\mathcal{K}}. \tag{1.73}$$

We first observe that $H_0 \subset H$. Moreover, for any $x, x' \in \mathcal{X}$ it holds $\langle \mathcal{K}_x, \mathcal{K}_{x'} \rangle_H = \mathcal{K}(x, x') = \langle \mathcal{K}_x, \mathcal{K}_{x'} \rangle_{\mathcal{H}_\mathcal{K}}$. Then, as both H and $\mathcal{H}_\mathcal{K}$ are completions of H_0, the claim follows from the uniqueness of the completion.

Finally, to see the remaining assertion, consider $f \in \mathcal{H}_\mathcal{K}$ and $x \in \mathcal{X}$. Then

$$|f(x)| = \left| \langle f, \mathcal{K}_x \rangle_{\mathcal{H}_\mathcal{K}} \right| \leq \|f\|_{\mathcal{H}_\mathcal{K}} \|\mathcal{K}_x\|_{\mathcal{H}_\mathcal{K}} = \|f\|_{\mathcal{H}_\mathcal{K}} \sqrt{\mathcal{K}(x, x)}. \tag{1.74}$$

This implies that $\|f\|_\infty \leq C_\mathcal{K} \|f\|_{\mathcal{H}_\mathcal{K}}$ and, thus, $\|I_\mathcal{K}\| \leq C_\mathcal{K}$. Therefore, convergence in $\|\cdot\|_{\mathcal{H}_\mathcal{K}}$ implies convergence in $\|\cdot\|_\infty$ and this shows that f is continuous since f is the limit of elements in H_0 that are continuous. □

In what follows, to reduce the amount of notation, we will write $\langle \cdot, \cdot \rangle_\mathcal{K}$ instead of $\langle \cdot, \cdot \rangle_{\mathcal{H}_\mathcal{K}}$ and $\|\cdot\|_\mathcal{K}$ instead of $\|\cdot\|_{\mathcal{H}_\mathcal{K}}$.

Definition 1.2.3. The Hilbert space $\mathcal{H}_\mathcal{K}$ in Theorem 1.2.1 is said to be a Reproducing Kernel Hilbert Space (RKHS). Property 3 in Theorem 1.2.1 is referred to as the reproducing property.

Consider now a Borel measure μ on $\mathcal{X}$, the Hilbert space of μ-square-integrable functions $L^2_\mu(\mathcal{X})$ and the linear operator $T_\mu : L^2_\mu(\mathcal{X}) \to C(\mathcal{X})$ associated to $\mathcal{K}$ defined by the following integral transform

$$T_\mu f := \int_\mathcal{X} \mathcal{K}(x, x') f(x') d\mu(x'). \tag{1.75}$$

Note that this is well-defined as $\mathcal{K}$ is continuous and $\mathcal{X}$ is compact. Composition of T_μ with the inclusion $C(\mathcal{X}) \hookrightarrow L^2_\mu(\mathcal{X})$ yields a linear operator from $L^2_\mu(\mathcal{X})$ to $L^2_\mu(\mathcal{X})$ which we again denote by T_μ abusing notation.

Proposition 1.2.2. *If $\mathcal{K}$ is a Mercer kernel, then $T_\mu : L^2_\mu(\mathcal{X}) \to L^2_\mu(\mathcal{X})$ is well-defined, self-adjoint, positive and compact. In addition, $\|T_\mu\| \leq \sqrt{\mu(\mathcal{X})} C^2_\mathcal{K}$.*

The above assertions imply that the Spectral Theorem applies and that the following theorem holds.

Theorem 1.2.3. *Let $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a Mercer kernel. Then, there exists an orthonormal basis $\{\varphi_i\}_i$ of $L^2_\mu(\mathcal{X})$ consisting of eigenfunctions of T_μ. If λ_i is the eigenvalue corresponding to φ_i, then either the set $\{\lambda_i\}_i$ is finite or $\lambda_i \to 0$ as $i \to +\infty$. In addition, if $\lambda_i \neq 0$, then φ_i can be chosen to be continuous on $\mathcal{X}$.*

If $f \in L^2_\mu(\mathcal{X})$ and $\{\varphi_i\}_i$ is an orthonormal basis of $L^2_\mu(\mathcal{X})$ then f can be uniquely written as $f = \sum_{i\geq 1} a_i\varphi_i$ for some $\{a_i\}_i \in \ell^2$. When the basis has infinitely many functions then the partial sums $\sum_{i=1}^N a_i\varphi_i$ converge to f in $L^2_\mu(\mathcal{X})$. If the convergence also holds in $C(\mathcal{X})$ then we say that the convergence is uniform. Moreover, a Borel measure μ on $\mathcal{X}$ is said to be non degenerate if for all nonempty open subsets $U \subseteq \mathcal{X}$ it holds $\mu(U) > 0$. We are now ready to state the second central theorem in RKHS theory.

Theorem 1.2.4 (Mercer's theorem)**.** *Let μ be a non-degenerate Borel measure on $\mathcal{X}$ and $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a Mercer kernel. Let λ_i be the i-th positive eigenvalue of T_μ, and φ_i the corresponding continuous L^2_μ-orthonormal eigenfunction. Then, for all $x, x' \in \mathcal{X}$ it holds*

$$\mathcal{K}(x, x') = \sum_{i\geq 1} \lambda_i\varphi_i(x)\varphi_i(x'), \tag{1.76}$$

where the convergence is absolute (for each $(x, x') \in \mathcal{X} \times \mathcal{X}$) and uniform (on $\mathcal{X} \times \mathcal{X}$).

We now aim at further characterizing the relationship between the RKHS $\mathcal{H}_\mathcal{K}$, the integral operator T_μ and its spectrum.

Theorem 1.2.5. *Let μ be a non-degenerate Borel measure on $\mathcal{X}$ and $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a Mercer kernel. Let λ_i be the i-th positive eigenvalue of T_μ, and φ_k the corresponding continuous orthonormal eigenfunction. Then $\{\sqrt{\lambda_i}\varphi_i \mid \lambda_i > 0\}$ is an orthonormal basis of $\mathcal{H}_\mathcal{K}$.*

Since the RKHS $\mathcal{H}_\mathcal{K}$ is independent of the measure μ, it follows that when μ is nondegenerate and $\dim(\mathcal{H}_\mathcal{K}) = +\infty$, T_μ has infinitely many positive eigenvalues λ_i , $i \geq 1$, and

$$\mathcal{H}_\mathcal{K} = \left\{ f \in L^2_\mu(\mathcal{X}) \;\middle|\; f = \sum_{i=1}^{\infty} a_i\sqrt{\lambda_i}\varphi_i, \quad \{a_i\}_{i=1}^\infty \in \ell^2 \right\}. \tag{1.77}$$

When instead $\dim(\mathcal{H}_\mathcal{K}) = m < \infty$, T_μ has only m positive repeated eigenvalues and, in this case,

$$\mathcal{H}_\mathcal{K} = \left\{ f \in L^2_\mu(\mathcal{X}) \;\middle|\; f = \sum_{i=1}^{m} a_i\sqrt{\lambda_i}\varphi_i, \quad (a_1, \ldots, a_m) \in \mathbb{R}^m \right\}. \tag{1.78}$$

Corollary 1.2.5.1. *Let μ be a non-degenerate Borel measure on $\mathcal{X}$ and $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a positive-definite Mercer kernel. Then, the map*

$$\begin{aligned} T_\mu^{1/2} : L^2_\mu(\mathcal{X}) &\to \mathcal{H}_\mathcal{K} \\ \sum_{i\geq 1} a_i\varphi_i &\mapsto \sum_{i\geq 1} a_i\sqrt{\lambda_i}\varphi_i. \end{aligned} \tag{1.79}$$

defines an isomorphism of Hilbert spaces. In particular, $\mathcal{H}_\mathcal{K} = T_\mu^{1/2}(L^2_\mu(\mathcal{X}))$, namely every function $f \in \mathcal{H}_\mathcal{K}$ can be written as $f = T_\mu^{1/2}g$ for some $g \in L^2_\mu(\mathcal{X})$ with $\|f\|_\mathcal{K} = \|g\|_{L^2_\mu(\mathcal{X})}$. In addition $T_\mu^{1/2}$, considered as an operator on $L^2_\mu(\mathcal{X})$, is the square root of T_μ in the sense that $T_\mu = T_\mu^{1/2} \circ T_\mu^{1/2}$, hence the notation.

We remark that Corollary 1.2.5.1 has the important implication that we can characterize the inner product[3] in $\mathcal{H}_\mathcal{K}$ through the one on $L^2_\mu(\mathcal{X})$ for any non-degenerate Borel measure μ on $\mathcal{X}$ as follows:

$$\langle f, g\rangle_\mathcal{K} = \left\langle T_\mu^{-1/2} f, T_\mu^{-1/2} g\right\rangle_{L^2_\mu(\mathcal{X})} = \left\langle f, T_\mu^{-1} g\right\rangle_{L^2_\mu(\mathcal{X})}. \tag{1.81}$$

Finally, we state and prove a classical theorem which motivates the wide use of reproducing kernel Hilbert spaces in learning theory.

Theorem 1.2.6. *(Representer Theorem) Let $\mathcal{H}_\mathcal{K}$ be an RKHS and $\mathbf{s} = ((x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)}))$ be a sample in $\mathcal{S} = \mathcal{X} \times \mathcal{Y}$. The optimizer $f_{\mathbf{s},\gamma}$ of*

$$\min_{f\in\mathcal{H}_\mathcal{K}} \mathcal{E}_{\mathbf{s},\gamma}(f) = \frac{1}{N}\sum_{i=1}^{N}(f(x^{(i)}) - y^{(i)})^2 + \gamma \|f\|^2_\mathcal{H}, \tag{1.82}$$

enjoys the following representation:

$$f_{\mathbf{s},\gamma} = \sum_{i=1}^{N} \mathcal{K}_{x^{(i)}}\beta^{(i)}, \tag{1.83}$$

for some N-tuple $(\beta^{(1)}, \dots, \beta^{(N)}) \in \mathbb{R}^N$. Hence, the (possibly, depending on $dim(\mathcal{H}_\mathcal{K})$) infinite dimensional minimization problem (1.82) is equivalent to a finite dimensional optimization problem in $\mathbb{R}^N$.

Proof. Define the linear subspace of $\mathcal{H}_\mathcal{K}$

$$H_0 := \operatorname{span}\{\mathcal{K}_{x^{(i)}} \mid i = 1, \dots, N\}, \tag{1.84}$$

namely the space spanned by the representers of the training set. As H_0 is finite dimensional and, hence, closed we have $\mathcal{H}_\mathcal{K} = H_0 \oplus H_0^\perp$. Let now $f = f_0 + f_0^\perp$ with $f_0 \in H_0$ and $f_0^\perp \in H_0^\perp$, then $\|f\|^2_\mathcal{K} = \|f_0\|^2_\mathcal{K} + \|f_0^\perp\|^2_\mathcal{K}$ and by the reproducing property

$$f_0^\perp(x^{(i)}) = \left\langle f_0^\perp, \mathcal{K}_{x^{(i)}}\right\rangle_\mathcal{K} = 0 \tag{1.85}$$

for all $i = 1, \dots, N$. This entails that

$$\mathcal{E}_{\mathbf{s},\gamma}(f) \geq \mathcal{E}_{\mathbf{s},\gamma}(f_0), \tag{1.86}$$

which yields the claim. □

[3]This characterization also holds true with milder assumptions on $\mathcal{X}$. Indeed, the compactness of $\mathcal{X}$ can be relaxed for $\mathcal{X} \subseteq \mathbb{R}^d$, assuming the probability measure μ admits a smooth and strictly positive density with respect to the Lebesgue measure and that $\mathcal{K}$ is continuously differentiable off the diagonal, bounded and integrally strictly positive definite, that is

$$\int_\mathcal{X}\int_\mathcal{X} \mathcal{K}(x, x')d\nu(x)d\nu(x') > 0 \tag{1.80}$$

for all signed nonzero Borel measures ν on $\mathcal{X}$.

Let us conclude this section by introducing *Kernel Ridge Regression (KRR)*.

Corollary 1.2.6.1. *(Kernel Ridge Regression) With the notation of Theorem 1.2.6, assume the kernel $\mathcal{K}$ is positive definite. Then, the unique minimizer of the optimization problem in (1.82), which is commonly referred to as Kernel Ridge Regression, is*

$$f_{\mathbf{s},\gamma} = \sum_{i=1}^{N} \mathcal{K}_{x^{(i)}} \beta^{(i)}, \quad \boldsymbol{\beta} = \mathbf{K}[\mathbf{x}]^{-1}\mathbf{y}. \tag{1.87}$$

Proof. Note that, owing to Theorem 1.2.6 any minimizer admits the representation

$$f = \sum_{i=1}^{N} \mathcal{K}_{x^{(i)}} \beta^{(i)}. \tag{1.88}$$

Then,

$$\mathcal{E}_{\mathbf{s},\gamma}(f) = \|\mathbf{K}[\mathbf{x}]\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \boldsymbol{\beta}^\top \mathbf{K}[\mathbf{x}]\boldsymbol{\beta}. \tag{1.89}$$

The above is a continuous and differentiable function in $\boldsymbol{\beta}$, hence imposing first order optimality conditions and exploiting the fact that $\mathbf{K}[\mathbf{x}]$ is invertible we recover the thesis. □

1.2.3 Some Mercer kernels

Now we have a more concrete concept of what an RKHS is and how we might create such spaces for ourselves. Indeed, if we succeed in writing down a Mercer kernel, we know that there exists an associated RKHS featuring all the properties just listed.

Linear kernel

The first kernel we are going to look at is one of the simplest ones, i.e. the linear kernel with $\mathcal{X} \subset \mathbb{R}^d$

$$\mathcal{K}(x, x') := x^\top x. \tag{1.90}$$

This is obviously symmetric and continuous. To see the positive semi-definiteness note that for any set $\mathbf{x} = \{x_1, \dots, x_N\} \subset \mathcal{X}$ and $\alpha \in \mathbb{R}^N$ we have

$$\alpha^\top \mathbf{K}[\mathbf{x}]\alpha = \sum_{i,j=1}^{N} \alpha_i x_i^\top x_j \alpha_j = \left\| \sum_{i=1}^{N} \alpha_i x_i \right\|^2 \geq 0. \tag{1.91}$$

In this case, the RKHS is that of linear functions $f_w(x) = w^\top x$, and the RKHS norm gives

$$\|f_w\|_{\mathcal{K}}^2 = \langle f_w, f_w \rangle_{\mathcal{K}} = \langle \mathcal{K}_w, \mathcal{K}_w \rangle_{\mathcal{K}} = \mathcal{K}(w, w) = \|w\|^2, \tag{1.92}$$

i.e. the magnitude of the slope of the plane. This example intuitively shows why in learning theory RKHS norms are taken as a measure of function complexity.

Christoffel-Darboux kernel

Let μ be a Borel probability measure on $\mathbb{R}^d$ with compact support, say, $\mathcal{X}$ and denote by $\mathcal{P}_m$ the set of polynomials of degree at most d (of dimension $s(m) := \binom{d+m}{m}$). Then

$$(p, q) \mapsto \langle p, q \rangle_\mu = \int_{\mathcal{X}} p(x)q(x)d\mu(x) \tag{1.93}$$

defines a valid scalar product on $\mathcal{P}_m$, hence $(\mathcal{P}_m, \langle \cdot, \cdot \rangle_\mu)$ is a finite dimensional Hilbert space of functions from $\mathcal{X}$ to $\mathbb{R}$. The interesting question concerns what kernel generates such Hilbert space. Take any basis $\{P_j\}_{j=1}^{s(m)}$ of $\mathcal{P}_m$ and define the map $\mathbf{w}_m : \mathcal{X} \to \mathbb{R}^{s(m)}$ such that

$$\mathbf{w}_m(x) = \Big(P_1(x), \dots, P_{s(m)}(x)\Big). \tag{1.94}$$

Call the matrix of inner products associated to such basis $G_{\mu,m}$, i.e.

$$G_{\mu,m} = \int_{\mathcal{X}} \mathbf{w}_m(x)\mathbf{w}_m(x)^\top d\mu(x). \tag{1.95}$$

Then, it turn out that the kernel generating $(\mathcal{P}_m, \langle \cdot, \cdot \rangle_\mu)$, the so called *Kristoffel-Darboux kernel*, admits the following representation:

$$\mathcal{K}_\mu^m(x, x') = \mathbf{w}_m(x)^\top G_{\mu,m}^{-1}\mathbf{w}_m(x') = \sum_{i,j=1}^{s(m)} P_i(x)(G_{\mu,m}^{-1})_{ij}P_j(x'). \tag{1.96}$$

In the case where $\{P_j\}_{j=1}^{s(m)}$ is an orthonormal basis the above expression takes the simplified form

$$\mathcal{K}_\mu^m(x, x') = \mathbf{w}_m(x)^\top \mathbf{w}_m(x') = \sum_{i=1}^{s(m)} P_i(x)P_i(x'). \tag{1.97}$$

Symmetry and continuity here are again trivial, moreover for any set $\mathbf{x} = \{x_1, \dots, x_N\} \subset \mathcal{X}$ and $\alpha \in \mathbb{R}^N$ we have

$$\begin{aligned} \alpha^\top \mathbf{K}[\mathbf{x}]\alpha &= \sum_{i,j=1}^{N} \sum_{r,l=1}^{s(m)} \alpha_i P_r(x_i)(G_{\mu,m}^{-1})_{rl}P_l(x_j)\alpha_j \\ &= \left\| \sum_{i=1}^{N} \alpha_i G_{\mu,m}^{-1/2}\mathbf{w}_m(x_i) \right\|^2 \geq 0. \end{aligned} \tag{1.98}$$

Translation invariant kernels

Another large class of kernels is represented by the so-called translation invariant kernels. Let $\mathcal{X} = \mathbb{R}^d$ and consider the family of kernels given by

$$\mathcal{K}(x, x') := k(x - x') \tag{1.99}$$

for some even function k on $\mathbb{R}^d$, i.e. $k(x) = k(-x)$ for all $x \in \mathbb{R}^d$.

Proposition 1.2.7. *Let $k \in L^2(\mathbb{R}^d)$ be continuous and even. Assume the Fourier transform $\hat{k}$ of k is nonnegative, that is $\hat{k}(w) \geq 0$ for all $w \in \mathbb{R}^d$. Then, $\mathcal{K}(x, x')$ is a Mercer kernel on $\mathbb{R}^d$ and, hence, a Mercer kernel on any subset of $\mathbb{R}^d$.*

Proof. Symmetry is trivial. We need to show positive semi-definiteness. Consider any $x_1, \ldots, x_N \in \mathbb{R}^d$ and $\alpha_1, \ldots, \alpha_N \in \mathbb{R}^d$ and the inverse Fourier transform

$$k(x) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{k}(w) e^{ix^\top w} dw \tag{1.100}$$

to get

$$\begin{aligned} \sum_{j,l=1}^{N} \alpha_j \mathcal{K}(x_j, x_l) \alpha_l &= \sum_{j,l=1}^{N} \alpha_j \alpha_l \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{k}(w) e^{i(x_j)^\top w} e^{-i(x_l)^\top w} dw \\ &= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{k}(w) \left(\sum_{j=1}^{N} \alpha_j e^{i(x_j)^\top w} \right) \overline{\left(\sum_{j=1}^{N} \alpha_j e^{i(x_j)^\top w} \right)} dw \\ &= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{k}(w) \left| \sum_{j=1}^{N} \alpha_j e^{i(x_j)^\top w} \right|^2 dw \geq 0. \end{aligned} \tag{1.101}$$

Hence, $\mathcal{K}$ is a Mercer kernel in any subset of $\mathbb{R}^d$. □

A prominent example of translation invariant kernel is the Gaussian kernel

$$\mathcal{K}(x, x') = e^{-r\|x - x'\|_2^2}, \quad r > 0. \tag{1.102}$$

Note that for all $r > 0$ the Fourier transform of $e^{-r\|x\|_2^2}$ is given by $(\sqrt{\pi/r})^d e^{-\|w\|_2^2/4r} > 0$ for all $w \in \mathbb{R}^d$, hence this is indeed a Mercer kernel. Moreover, note that $C_{\mathcal{K}} = 1$ in this case. Contrary to the polynomial case, the RKHS generated by the Gaussian kernel is infinite dimensional. This RKHS also enjoys interesting approximating properties, such as density in $C(\mathcal{X})$ if $\mathcal{X}$ is compact (Steinwart [2002]). Rather curiously, however, it can be shown that it does not contain any polynomial on $\mathcal{X}$, including the constant function.

1.2.4 Vector valued Reproducing Kernel Hilbert Spaces

In the previous sections we have introduced the classical $\mathbb{R}$-valued case, however in the following chapters we are mainly going to be interested in spaces of $\mathbb{R}^d$-valued functions for some arbitrary $d \geq 1$. Hence, we now introduce the needed generalization to our relevant case. The construction mirrors the one in the scalar case, with the main difference that the kernel is a continuous and symmetric matrix-valued function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{d \times d}$ such that for all finite sets $\mathbf{x} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^d$ the $dN \times dN$ matrix

$$\mathbf{K}[\mathbf{x}] = \begin{pmatrix} \mathcal{K}(x_1, x_1) & \ldots & \mathcal{K}(x_1, x_N) \\ \vdots & \ddots & \vdots \\ \mathcal{K}(x_N, x_1) & \ldots & \mathcal{K}(x_N, x_N) \end{pmatrix} \tag{1.103}$$

is positive semi-definite. Again, there exists a unique RKHS $\mathcal{H}_{\mathcal{K}}$ (up to isomorphisms) associated with $\mathcal{K}$ defined as the closure of the set of linear combinations

$$f(x) = \sum_{i=1}^{N} \mathcal{K}(x, x_i)\beta_i, \quad \beta \in \mathbb{R}^d \tag{1.104}$$

where we note that in the above equation each term $\mathcal{K}(x, x_i)$ is a matrix acting on a vector β_i. Moreover, $\mathcal{K}$ enjoys the usual reproducing property which, in this setting, reads

$$\langle f, \mathcal{K}(\cdot, x)\beta \rangle_{\mathcal{K}} = f^\top(x)\beta \tag{1.105}$$

for all $\beta \in \mathbb{R}^d$ and $f \in \mathcal{H}_{\mathcal{K}}$. Again, the choice of the kernel corresponds to the choice of the representation (parameterization) for the function of interest. If we then assume the same working hypothesis on $\mathcal{X}$ as in the scalar case, similar properties hold. The reader is referred to, e.g., Álvarez et al. [2012] and Carmeli et al. [2006] for further details.

Example 1.2.1 (Separable Kernels and Sum of Separable Kernels)**.** A simple way one can construct matrix-valued kernels is by taking a scalar-valued Mercer kernel k, a positive semi-definite matrix A and defining

$$\mathcal{K}(x, x') := Ak(x, x'). \tag{1.106}$$

We refer to this type of kernel functions as *separable kernels* as the contribution to the kernel of input and output are decoupled (in particular, A encodes output interactions). Notice that $\mathcal{K}$ inherits symmetry and continuity from k and for any $\mathbf{x} = \{x_1, \dots, x_N\} \subset \mathcal{X}$ the Gram matrix $\mathbf{K}[\mathbf{x}]$ can be realized as the Kronecker product of A and the Gramian $\mathbf{k}[\mathbf{x}]$ of k at $\mathbf{x}$, namely

$$\mathbf{K}[\mathbf{x}] = A \otimes \mathbf{k}[\mathbf{x}], \tag{1.107}$$

the symbol $\otimes$ being the Kronecker product. As the factors A and $\mathbf{k}[\mathbf{x}]$ are positive semi-definite then also $\mathbf{K}[\mathbf{x}]$ is. A straightforward extension of this construction (which yields the so-called *sum of separable kernels*) considers Q scalar-valued Mercer kernels $k_1, \dots, k_Q$ and positive semi-definite matrices $B_1, \dots, B_Q$ and defines

$$\mathcal{K}(x, x') := \sum_{q=1}^{Q} B_q k_q(x, x'). \tag{1.108}$$

It is easy to check that in this case we have for any $\mathbf{x} = \{x_1, \dots, x_N\} \subset \mathcal{X}$

$$\mathbf{K}[\mathbf{x}] = \sum_{q=1}^{Q} B_q \otimes \mathbf{k}_q[\mathbf{x}], \tag{1.109}$$

so that arguing analogously as for separable kernels we prove that this is indeed a valid kernel.

1.3 Machine Learning and Deep Learning

1.3.1 Introduction

In recent years the need for numerical tools to efficiently deal with and interpret all sorts of data types has considerably risen. The demand is very heterogeneous as it arises across different scientific communities, such as biology and neuroscience, as well as industries, such as financial institutions and internet service providers. This is in large part the reason why the field of *Machine Learning (ML)* (James et al. [2013]) has gained immense popularity in the last decades enjoying stable growth showing no signs of slowing down. The problem addressed by ML is indeed the one of processing data, be it either abundantly available or very scarce, relevant for applications. ML can serve as a tool to provide professionals with further understanding of relevant phenomena, forecast the most likely scenarios based on past knowledge, prescribe favourable actions to be taken in highly complex systems, automate and improve processes and much, much more. In particular, the sub-field of *Deep Learning (DL)* (Goodfellow et al. [2016]) has proven to be a flexible framework suited to perform multiple kinds of tasks related to the learning paradigm (e.g. regression, classification, etc...), all without the need for specialized algorithms and hand-crafted features.

The most commonly employed algorithms in Deep Learning are deep versions of *Artificial Neural Networks (ANNs)*. The latter are composite parametric computational models where the fundamental building blocks, the so-called *layers*, are implemented sequentially to produce an output. When the number of layers is large these architectures are referred to as *Deep Neural Networks* (DNNs). To make these concepts more concrete we give a brief mathematical description of ANNs. Let as usual $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output spaces. Consider, further, intermediate spaces $\mathcal{X}_1, \dots, \mathcal{X}_L$ (heuristically, think of these as interpolating between $\mathcal{X}$ and $\mathcal{Y}$) and associated parameter spaces $\Theta_1, \dots, \Theta_L$. Then, given a set of parameters $\boldsymbol{\theta} = (\theta_1, \dots, \theta_L) \subset \prod_i \Theta_i$, an ANN can be described as a (nonlinear) function $v(\cdot, \boldsymbol{\theta}) : \mathcal{X} \to \mathcal{Y}$ realized as the composition of (nonlinear) functions $v_i(\cdot, \theta_i) : \mathcal{X}_{i-1} \to \mathcal{X}_i$, $i = 1, \dots, L$, where we used the notation $\mathcal{X}_0 := \mathcal{X}$, i.e.

$$v(\cdot, \boldsymbol{\theta}) = v_L(\cdot, \theta_L) \circ \dots \circ v_1(\cdot, \theta_1). \tag{1.110}$$

Here each of the $v_i(\cdot, \theta_i)$ represents a layer of the ANN and the total number of layers L is referred to as the *depth* of the network. When $L \gg 1$ such an architecture is referred to as a DNN. Statistical problems with high-dimensional data are frequently plagued by the curse of dimensionality, in which the number of samples required to solve the problem with a given accuracy grows rapidly (often exponentially fast) with the dimensionality of the input. Arguably the merit that makes DNNs so widely used nowadays is their ability to represent arbitrarily complex functions and efficiently learn those from data in high dimension, seemingly beating the curse of dimensionality (although how exactly these methods break such curse remains a fundamental open question) and effectively solving statistical learning problems previously thought intractable. On one hand, indeed, owing to the compositional structure (1.110), one can construct a very rich class of functions by taking the building blocks $v_i(\cdot, \theta_i)$ to be very simple parametric functions. Moreover, these compositional function spaces have shown both experimentally and theoretically to

improve the efficiency of the learning procedure. Usually the layers are realized as a further composition of a linear function and a very simple nonlinear function $\sigma : \mathbb{R} \to \mathbb{R}$. More concretely, let for instance $\mathcal{X}_i = \mathbb{R}^{d_i}$, $\theta_i = (W_i, b_i) \in \Theta_i := \mathbb{R}^{d_i \times d_{i-1}} \times \mathbb{R}^{d_i}$ for $i = 0, \dots, L$, then

$$v_i(x, \theta_i) = \sigma(W_i^\top x + b_i), \tag{1.111}$$

where the action of σ is understood as point-wise. In standard Deep Learning jargon the natural number d_i is said to be the number of *neurons* of the i-th layer, while the matrices W_i and vectors b_i are usually called *weights* and *biases* respectively. This particularly simple structure, which can be summarized as

> *"compose simple nonlinear parametric functions to get a complex nonlinear parametric function"*,

often enables the construction of DNNs that exploit the structure and associated geometric symmetries and invariances which each data type features. These data-specific architectural choices are often referred to *inductive biases* as they embed a-priori knowledge about the target function. A prime example of how inductive biases represent drivers of success for deep neural approaches is image processing, where information locality and grid shape featured by images are effectively exploited by translationally equivariant operators, i.e. convolutional layers.

On the other hand the celebrated *backpropagation* (Rumelhart et al. [1986]) method is an extremely versatile algorithm which, when coupled with a gradient descent type algorithm, allows for efficient *training* (i.e. optimization) of virtually any DNN. Consider for instance a regression setting[4] where a dataset $\mathbf{s} = ((x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})) \in \mathcal{S}^N$ (recall $\mathcal{S} = \mathcal{X} \times \mathcal{Y}$) is given, and consider the error introduced in Section 1.2.1 (or, as is commonly called in Machine Learning literature, the *loss function*) $\mathcal{E}(\mathbf{z})$ which, with a slight abuse of notation, we now denote as a function of the predictions

$$\mathbf{z} := (z^{(1)}; \dots; z^{(N)}) = (v(x^{(1)}, \boldsymbol{\theta}); \dots; v(x^{(N)}, \boldsymbol{\theta})) \tag{1.112}$$

of the model given by the ANN (1.110). Backpropagation works by computing the gradient of the loss function with respect to each weight and bias by the chain rule, computing the gradient one layer at a time, iterating backward from the last layer to avoid redundant calculations of intermediate terms in the chain rule. This is briefly sketched in Algorithm 1.3.1 in the simple case where $N = 1$ for ease of notation, but it trivially extends to the setting $N > 1$ with the appropriate vectorizations.

[4]The backpropagation Algorithm 1.3.1 can be readily implemented with arbitrary loss functions as long as they are differentiable.

Algorithm 1.3.1: Gradient descent algorithm with backpropagation

Input: Data (x, y), maximum steps M, step size ε.
Output: Trained parameters $\boldsymbol{\theta}^\star$.

Initialize $\boldsymbol{\theta}^0 = (\theta_1^0, \dots, \theta_L^0)$;
for $k = 1, \dots, M$ **do**
  Compute

$$w_L = \partial_z \mathcal{E}(v(x, \boldsymbol{\theta}^{k-1})), \quad g_L = \partial_{\theta_L} v_L(\dots, \theta_L^{k-1})^\top w_L; \tag{1.113}$$

  Update $\theta_L^k = \theta_L^{k-1} - \varepsilon g_L$;
  for i = *L-1,. . . ,1* **do**
    Compute

$$w_i = \partial_x v_{i+1}(\dots, \theta_{i+1}^{k-1})^\top w_{i+1}, \quad g_i = \partial_{\theta_i} v_i(\dots, \theta_i^{k-1})^\top w_i; \tag{1.114}$$

    Update $\theta_i^k = \theta_i^{k-1} - \varepsilon g_i$;
  end
end

1.3.2 Neural ODEs

Despite its widespread use and flexibility, there are several challenges associated to the practical use of backpropagation which usually manifest when training very deep networks (hundreds or thousands of layers). The main issue is what is commonly known in DL literature as *exploding* or *vanishing gradients*, namely the well-known phenomenon which characterizes products of a large number of terms (as is the gradient in Algorithm 1.3.1 for layers $l \ll L$) making it vanish or explode depending on the norm of each term in the product. Hence, if one is note careful, training deep networks with backpropagation is an ill-conditioned and very unstable problem. In order to recover stability of the algorithm one should build DNNs such that

$$\|\partial_\theta v_i(x, \theta_i)\| \approx 1. \tag{1.115}$$

Assume for simplicity $\mathcal{X}_i = \mathcal{X}$ for all $i = 1, \dots, L$. The key insight is that (1.115) can be achieved if $v_i(x, \theta_i)$ is represented as a small enough perturbation of the identity mapping $I : \mathcal{X} \to \mathcal{X}$, i.e. instead of layers $v_i(x, \theta_i)$ use

$$\tilde{v}_i(x, \theta_i) = x + h v_i(x, \theta_i), \tag{1.116}$$

the parameter $h > 0$ being an arbitrary scaling factor. The above choice, which yields the well-known neural architecture

$$v(\cdot, \boldsymbol{\theta}) = (I + h v_L(\cdot, \theta_L)) \circ \dots \circ (I + h v_1(\cdot, \theta_1)) \tag{1.117}$$

denoted *Residual Neural Network (ResNet)* (He et al. [2016]), allows for the construction and efficient optimization of very deep networks. Note that composing, e.g., a projection

operation with $v(\cdot, \boldsymbol{\theta})$ permits to build a mapping to an arbitrary output space $\mathcal{Y}$. Given an input $x \in \mathcal{X}$ its evolution according to (1.117) is governed by the following dynamical system:

$$\begin{cases} z_{l+1} &= z_l + h v_l(z_l, \theta_l), \\ z_0 &= x. \end{cases} \tag{1.118}$$

The works of Chen et al. [2018] and E [2017] independently showed the connection of (1.118) with continuous-time dynamical systems which is immediately understood if one sets the scaling factor $h = 1/L$. With this choice, indeed, the propagation of the states through the layers can be seen as an explicit Euler discretization scheme on a uniform partition $\{[(i-1)h, ih)\}_{i=1}^{L}$ of the unitary interval [0,1] for the integration of the following IVP:

$$\begin{cases} \dot{z}_t &= v_t(z_t, \theta(t)), \quad t \in (0,1] \\ z_0 &= x. \end{cases} \tag{1.119}$$

Notice that in order to make sense of (1.119) we should think of the parameters as a function $\theta : [0,1] \to \Theta$.

This connection sparked a lot of research activity focused on both exploiting the flexibility of the continuous-time model and applying classical ideas from the seasoned literature on continuous-time dynamical systems for the construction and optimization of ANNs. Chen et al. [2018] proposed to go beyond the concept of layer which intrinsically imposes uniformity of the time discretization of trajectories. Instead, they integrate the dynamical system exploiting off-the-shelf ODE solvers which provide guarantees about the growth of approximation error, monitor the level of error, and adapt their evaluation strategy on the fly to achieve the requested level of accuracy. This entails choosing the step size for each trajectory in response to the stiffness of the underlying ODE. They further took advantage of this new-found flexibility with respect to time for learning tasks which involve time series data naturally arising irregularly in time. Finally, they provided an application to the computation of transport maps, showing that the continuous-time model allows for the computation of the transported (via the ODE) density with a reduced cost of $\mathcal{O}(d^2)$ operations instead of $\mathcal{O}(d^3)$, d being the dimensionality of the input space $\mathcal{X}$.

Another line of work is centered around employing more stable and accurate discretization schemes for the ODE (1.119). Ruthotto and Haber [2019] proposed to integrate (1.119) with the leapforg method, a second order scheme, to improve convergence of the discretization to the true continuous-time solution. Furthermore, they explored the possibility of imposing Hamiltonian structure to the right-hand side of the ODE which entails preservation of hyper-volumes in phase space by the ODE flow and is argued to make the system more stable. Other attempts to design architectures which correspond to higher order integration schemes can be found in Larsson et al. [2017].

1.3.3 ANNs as Optimal Control problems

The dynamical systems point of view on Deep Learning has also been leveraged for the design of optimization algorithms of DNNs. Indeed, it has been pointed out (Li et al. [2017]) that training Deep Neural Networks, bearing in mind that the latter can

be seen as discretizations of continuous-time models, shares striking similarities with the solution of classical Optimal Control problems (see Section 1.1.1). Given a sample $\mathbf{s} = ((x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)}))$ a supervised learning task can be written as

$$\begin{aligned} \min_{\boldsymbol{\theta}:[0,1]\to\Theta} \quad & \mathcal{E}(z_1^{(1)}, \dots, z_1^{(N)}) + \int_0^1 \ell(\theta_t) dt \\ \text{subject to} \quad & \dot{z}_t^{(i)} = v(z_t^{(i)}, \theta_t), \quad i = 1, \dots, N \\ & z_0^{(i)} = x^{(i)} \end{aligned} \tag{1.120}$$

where $\mathcal{E}$ is the empirical loss function and ℓ is some running cost representing a regularization factor. Note that for simplicity we dropped the explicit time dependence of v meaning that we assume a fixed relation between the parameters and the functional form of v. Clearly, upon defining the state $\mathbf{z}_t := (z_t^{(1)}; \dots; z_t^{(N)}) \in \mathcal{X}^N$ and considering the parameter function $\boldsymbol{\theta}$ as control (1.120) immediately falls into the category of fixed-time, free-endpoint problems introduced in Section 1.1.1.

The review presented in Section 1.1 reveals two fundamental approaches to the solution of OC problems, namely the Pontryagin Maximum Principle and dynamic programming. Recall that the latter is based on the solution of the Hamilton-Jacobi-Bellman PDE and yields control policies in feedback (or closed-loop) form, that is the Optimal Control is expressed as a function of the state $\mathcal{X}^N \ni \mathbf{z} \mapsto \theta^{\text{cl}}(\mathbf{z})$. This property is especially desirable when the forward model is affected by uncertainty, e.g. when the ODE is replaced with an SDE, as it provides the optimal policy whatever state is visited as a result of the noisy dynamics. However, as already mentioned, solving a highly nonlinear and high dimensional PDE poses serious computational hurdles making this strategy unfeasible for Machine Learning applications where usually at least one among $d = \dim(\mathcal{X})$ and N is very large. On the other hand, the PMP entails solving the following two-point boundary value problem

$$\dot{\mathbf{z}}_t = H_{\mathbf{p}}(\mathbf{z}_t, \theta_t, \mathbf{p}_t), \qquad \mathbf{z}_0 = \mathbf{x}, \tag{1.121}$$

$$\dot{\mathbf{p}}_t = -H_{\mathbf{z}}(\mathbf{z}_t, \theta_t, \mathbf{p}_t), \qquad \mathbf{p}_1 = -\partial_{\mathbf{z}}\mathcal{E}(\mathbf{z}_1), \tag{1.122}$$

$$\theta_t \in \arg\max_{\theta' \in B} H(\mathbf{z}_t, \theta', \mathbf{p}_t), \tag{1.123}$$

the function $\mathbf{p} : [0,1] \to \mathcal{X}^N$ being the adjoint state as usual, $\mathbf{x} := (x_t^{(1)}; \dots; x_t^{(N)}) \in \mathcal{X}^N$ and the Hamiltonian

$$\mathcal{X}^N \times \Theta \times \mathcal{X}^N \ni (\mathbf{z}, \theta, \mathbf{p}) \mapsto H(\mathbf{z}, \theta, \mathbf{p}) = \mathbf{p}^\top v(\mathbf{z}, \theta) - \ell(\theta). \tag{1.124}$$

The PMP approach effectively restricts the problem onto a one dimensional manifold embedded in $\mathcal{X}^N$, the trajectory of the forward ODE, and yields an Optimal Control in open-loop form $[0,1] \ni t \mapsto \theta_t^{\text{ol}}$. This property is what makes the PMP the natural approach to the optimization of DNNs.

There are many ways of solving OC problems via the PMP. A very simple algorithm able to solve large scale problems is the Method of Successive Approximation (MSA). This is summarized in Algorithm 1.3.2.

Algorithm 1.3.2: MSA

Input: Dataset $\mathbf{s}$, loss function $\mathcal{E}(\mathbf{s})$, maximum steps M.
Output: Open-loop solution $\boldsymbol{\theta}^{\text{ol}}$ to the PMP.

Initialize $\boldsymbol{\theta}^0$;
for $k = 0, \dots, M$ **do**
  Given $\mathbf{z}_0 = \mathbf{x}$ integrate

$$\dot{\mathbf{z}}_t = H_{\mathbf{p}}(\mathbf{z}_t, \theta_t^k, \mathbf{p}_t); \tag{1.125}$$

  Given $\mathbf{p}_1 = -\partial_{\mathbf{z}}\mathcal{E}(\mathbf{z}_1)$ integrate

$$\dot{\mathbf{p}}_t = -H_{\mathbf{z}}(\mathbf{z}_t, \theta_t^k, \mathbf{p}_t); \tag{1.126}$$

  Set

$$\theta_t^{k+1} = \arg\max_{\theta' \in \Theta} H(\mathbf{z}_t, \theta', \mathbf{p}_t), \tag{1.127}$$

  for all $t \in [0,1]$;
end

A discrete version of the MSA follows from the outlined procedure by discretizing the integration procedure of the state and adjoint ODEs, for instance with a symplectic Euler discretization scheme. Interestingly enough, Li et al. [2017] showed that if one replaces the Hamiltonian maximization step in Algorithm 1.3.2 with a gradient ascent step

$$\theta_t^{k+1} = \theta_t^k + \varepsilon H_\theta(\mathbf{z}_t, \theta_t^k, \mathbf{p}_t), \tag{1.128}$$

and then discretizes the procedure as described, one recovers exactly backpropagation with gradient descent in Algorithm 1.3.1.

Chapter 2

Learning with reproducing kernel Hilbert spaces and Neural ODEs

2.1 Kernel Neural ODEs

This section introduces the neural architecture which will accompany us for the remainder of the thesis. Inspired by the dynamical syestems perspective on Deep Learning, we wish to combine the expressivity and flexibility of reproducing kernel Hilbert spaces and the compositional structure of ANNs for the design of powerful surrogate models amenable to numerical optimization based on data.

Let $\mathcal{X} \subseteq \mathbb{R}^d$, $\mathcal{Y}$ be the input and output spaces respectively, and $\mathcal{H}$ be a given RKHS of functions from $\mathbb{R}^d$ to $\mathbb{R}^d$ along with the associated matrix-valued reproducing kernel $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{d\times d}$. Following Owhadi [2020], we focus in the continuous-time model specified by the following ODE:

$$\begin{cases} \dot{z}_t = v_t(z_t), \quad t \in [0,1], \\ z_0 = x, \end{cases} \tag{2.1}$$

where $v_t \in \mathcal{H}$ for all $t \in [0,1]$. Throughout the thesis we will refer to ANNs of this kind as *Kernel Neural ODEs (KerODEs)*. The main difference with a traditional Neural ODE as described in the previous section is that the model in (2.1) is non-parametric as the right-hand-side v_t is taken in a (possibly infinite-dimensional) RKHS. For $v : [0,1] \to \mathcal{H}$, then, (2.1) defines an (invertible, provided the fields v_t are regular enough) map $\phi_v : \mathcal{X} \to \mathbb{R}^d$ if one assumes existence and uniqueness of solutions for all initial conditions (later in the section we give conditions on $\mathcal{K}$ so that this is always true). Assume further $\mathcal{V}_m \subset C^1(\mathbb{R}^d; \mathcal{Y})$ is some fixed m dimensional space of functions (e.g. polynomials) from $\mathbb{R}^d$ to $\mathcal{Y}$ endowed with a (semi) norm $\|\cdot\|_{\mathcal{V}_m}$, so that $f \circ \phi_v : \mathcal{X} \to \mathcal{Y}$ for all $f \in \mathcal{V}_m$. We are now ready to state the general form of the problem we are interested in.

Problem 1. Consider a target function $g \in L^2_\mu(\mathcal{X};\mathcal{Y})$ where μ is a given Borel probability measure supported on $\mathcal{X}$, $\mathcal{Y} = \mathbb{R}^n$, and whose observation may be affected by noise represented by the random vector ε on $\mathcal{Y}$. Our goal is to approximate g solving the following minimization problem:

$$\begin{aligned} \min_{f\in\mathcal{V}_m,\, v:[0,1]\to\mathcal{H}} \quad & \frac{1}{2}\int_{\mathcal{X}\times\mathcal{Y}} (f(z_1(x)) - y)^2 \, d\nu(x,y) + \frac{\eta}{2}\|f\|^2_{\mathcal{V}_m} + \frac{\gamma}{2}\int_0^1 \|v_t\|^2_{\mathcal{H}}\, dt \\ \text{subject to} \quad & \dot{z}_t(x) = v_t(z_t(x)), \\ & z_0(x) = x. \end{aligned} \tag{2.2}$$

where ν is the joint probability measure of the random vector $(x, g(x)+\varepsilon)$ with $x \sim \mu$, $\gamma > 0$ and $\eta \geq 0$.

Note how this falls into the setting of Section 1.2.1 where the problem of learning and regularization is introduced. In (2.2), indeed,

$$\frac{1}{2}\int_{\mathcal{X}\times\mathcal{Y}} (f(z_1(x)) - y)^2 \, d\nu(x,y) \tag{2.3}$$

measures the misfit with the data associated to the approximation function $f \circ \phi_v$ (notice that with the notation introduced in Problem 1 we have $z_1 = \phi_v$ as a function from $\mathcal{X}$ to $\mathbb{R}^d$). The probability measure μ, which will be sometimes referred to as *reference measure* in the following, should be considered a fixed element of the learning problem as it reflects our needs in terms of approximation, namely it places a larger mass in those regions of the domain $\mathcal{X}$ where the approximation is required to be more accurate and vice-versa for the regions where accuracy is of a smaller concern. On the other hand

$$\frac{\gamma}{2}\int_0^1 \|v_t\|^2_{\mathcal{H}}\, dt \tag{2.4}$$

is a regularization term weighted by the parameter $\gamma > 0$, introduced, in analogy with kernel ridge regression, to promote low-complexity functions in the RKHS. Similarly, term $\frac{\eta}{2}\|f\|^2_{\mathcal{V}_m}$ is weighted by the parameter $\eta \geq 0$ and serves regularization purposes for the last approximation step. Let us now further introduce the notation $\mu_t := z_{t\#}\mu$ for the push-forward of the measure μ through the map z_t, that is

$$\mu_t(B) = \mu(z_t^{-1}(B)) \tag{2.5}$$

for all Borel sets $B \subseteq \mathbb{R}^d$. Assuming ϕ_v is an invertible transformation and that the observation noise term ε vanishes almost surely, the rationale behind the proposed model is revealed by writing

$$\int_{\mathcal{X}} (f(z_1(x)) - g(x))^2 \, d\mu(x) = \int_{\phi_v(\mathcal{X})} \Big(f(z) - g(\phi_v^{-1}(z))\Big)^2 \, d\mu_1(z). \tag{2.6}$$

The above, indeed, shows that the role of the mapping ϕ_v is to apply a change of variables on the input space so that the function being approximated in the mapped space $\phi_v(\mathcal{X})$, namely $g \circ \phi_v^{-1}$, hopefully is more amenable to approximation in the space $\mathcal{V}_m$, albeit with

respect to the push-forward measure μ_1. Clearly, perfect reconstruction can be achieved whenever $g \circ \phi_v^{-1} \in \mathcal{V}_m$.

Problem 1 in not tractable from a computational point of view. Consistently with the typical setting in learning theory, we assume g may only be known through a finite number of possibly noisy observations. Hence, we consider a point-wise space discretization (e.g. with random samples) of (2.2), essentially replacing the misfit in (2.3) with its point-wise approximation.

Problem 2. Given a sample $(x^{(1)}; \dots; x^{(N)})$ and possibly noisy observations $(y^{(1)}; \dots; y^{(N)}) := (g(x^{(1)}) + \varepsilon^{(1)}, \dots, g(x^{(N)}) + \varepsilon^{(N)})$ where $\varepsilon^{(i)} \overset{\text{iid}}{\sim} \varepsilon$, solve

$$\begin{aligned} \min_{f \in \mathcal{V}_m,\, v:[0,1]\to\mathcal{H}} \quad & \frac{1}{2N} \sum_{i=1}^{N} \left(f(z^{(i)}) - y^{(i)} \right)^2 + \frac{\eta}{2} \|f\|_{\mathcal{V}_m}^2 + \frac{\gamma}{2} \int_0^1 \|v_t\|_{\mathcal{H}}^2 \, dt \\ \text{subject to} \quad & \dot{z}_t^{(i)} = v_t(z_t^{(i)}), \quad i = 1, \dots, N, \\ & z_0^{(i)} = x^{(i)}, \quad i = 1, \dots, N. \end{aligned} \tag{2.7}$$

Recall that in the context of empirical risk minimization via kernel ridge regression the role of the penalization term proportional to the RKHS norm of the regressor serves the key purpose, encompassed by the Representer Theorem 1.2.6, of transforming an otherwise intractable infinite-dimensional problem into a convex and finite-dimensional optimization problem. It turns out that a version of the Representer Theorem holds for Problem 2 as well.

Theorem 2.1.1. *Let v be a minimizer of (2.7), then the following representation holds:*

$$v_t = \sum_{i=1}^{N} \mathcal{K}(\cdot, z_t^{(i)}) \beta_t^{(i)}, \tag{2.8}$$

where $\beta^{(i)} : [0,1] \to \mathbb{R}^d$, $i = 1, \dots, N$.

Proof. This result follows as a consequence of a similar Hilbert space decomposition to the one employed in the proof of the Representer Theorem 1.2.6. Assume $v : [0,1] \to \mathcal{H}$ is a minimizer of (2.7). Then, let $z^{(i)} : [0,1] \to \mathbb{R}^d$, $i = 1, \dots, N$ denote the solutions to the N Cauchy problems in (2.7) and define

$$H_t := \left\{ \sum_{i=1}^{N} \mathcal{K}_{z_t^{(i)}} \beta^{(i)} \mid \beta^{(i)} \in \mathbb{R}^d, i = 1, \dots, N \right\}, \quad t \in [0,1], \tag{2.9}$$

where we recall notation $\mathcal{K}_{z_t^{(i)}} = \mathcal{K}(\cdot, z_t^{(i)})$. The subspace H_t is finite dimensional and, hence, closed. This entails that the decomposition

$$v_t = w_t + w_t^{\perp}, \qquad w_t \in H_t, \quad w_t^{\perp} \in H_t^{\perp} \tag{2.10}$$

holds for all $t \in [0,1]$. Then, as usual $\|v_t\|_{\mathcal{H}}^2 = \|w_t\|_{\mathcal{H}}^2 + \|w_t^{\perp}\|_{\mathcal{H}}^2$ and, owing to the reproducing property,

$$\left(w_t^{\perp}(z_t^{(i)}) \right)^{\top} \beta = \left\langle w_t^{\perp}, \mathcal{K}_{z_t^{(i)}} \beta \right\rangle_{\mathcal{H}} = 0 \tag{2.11}$$

for all $i = 1, \dots, N$ and $\beta \in \mathbb{R}^d$, whence

$$w_t^\perp(z_t^{(i)}) = 0 \tag{2.12}$$

for all $i = 1, \dots, N$. As the ODE right-hand-side v_t is evaluated only at points $z_t^{(i)}$, $i = 1, \dots, N$, the term $w_t^\perp$ yields no contribution and

$$\int_0^1 \|v_t\|_{\mathcal{H}} \, dt \geq \int_0^1 \|w_t\|_{\mathcal{H}} \, dt. \tag{2.13}$$

Therefore, as v is a minimizer, it must hold $w_t^\perp = 0$ for all $t \in [0,1]$. □

2.1.1 Existence of minimizers

By virtue of Theorem 2.1.1 the minimization in Problem 2, which is infinite dimensional in both time and space, reduces to a finite-dimensional minimization problem in space (albeit still infinite-dimensional in time). It is interesting to notice how this property follows directly by the space discretization of the misfit (2.3). For ease of presentation we introduce the following vectorized notation:

- $\mathbf{x} := (x^{(1)}; \dots; x^{(N)}) \in \mathbb{R}^{dN}$;
- $\mathbf{y} := (y^{(1)}; \dots; y^{(N)}) \in \mathbb{R}^{dN}$;
- $\mathbf{z}_t := (z^{(1)}; \dots; z^{(N)}) \in \mathbb{R}^{dN}$ and $\mathbf{z} : t \mapsto \mathbf{z}_t$;
- $\boldsymbol{\beta}_t := (\beta^{(1)}; \dots; \beta^{(N)}) \in \mathbb{R}^{dN}$ and $\boldsymbol{\beta} : t \mapsto \mathbf{z}_t$;
- $\mathbf{K}[\cdot] : \mathbb{R}^{dN} \to \mathbb{R}^{dN} \times \mathbb{R}^{dN}$ defined as in Section 1.2.4.

Until now we have not been specific as to the function space it is best to ambient the learning problem in (i.e. concerning the regularity with respect to time of $v : [0,1] \to \mathcal{H}$). Therefore, let us take a further step in the direction of having a well defined problem. Clearly, the first thing that is needed is for the ODE to admit a unique solution on $[0,1]$ for all initial conditions in $\mathcal{X}$. For this purpose, let us introduce the some assumptions on the RKHS $\mathcal{H}$ following Owhadi [2020].

Assumption 1. Assume $\mathcal{K}$ is a uniformly bounded Mercer kernel with first and second order partial derivatives being continuous and uniformly bounded. Assume further that there exists $\delta > 0$ such that $\boldsymbol{\beta}^\top \mathbf{K}[\mathbf{z}]\boldsymbol{\beta} \geq \delta \|\boldsymbol{\beta}\|_2^2$ for all $\boldsymbol{\beta}, \mathbf{z} \in \mathbb{R}^{dN}$.

Assumption 1 has a number of useful consequences which will be thoroughly exploited in the following. For $\mathbf{z} \in \mathbb{R}^{dN}$ denote by $\|\cdot\|_{\mathbf{K}[\mathbf{z}]}$ the norm on $\mathbb{R}^{dN}$ induced by the matrix $\mathbf{K}[\mathbf{z}]$. Then, we have

$$\delta \|\boldsymbol{\beta}\|_2^2 \leq \|\boldsymbol{\beta}\|_{\mathbf{K}[\mathbf{z}]}^2 \leq \Delta \|\boldsymbol{\beta}\|_2^2 \tag{2.14}$$

for constants $\delta, \Delta > 0$ independent of $\mathbf{z}$, that is $\|\cdot\|_2$ and $\|\cdot\|_{\mathbf{K}[\mathbf{z}]}$ are equivalent norms uniformly in $\mathbf{z}$. Note that this also entails $\|\mathbf{K}[\mathbf{z}]\| \leq \Delta$, $\|\cdot\|$ being the spectral norm.

Moreover, for any continuous function $\mathbf{z} \in C([0,1];\mathbb{R}^{dN})$ we have that the bilinear map $\langle \cdot, \cdot \rangle_{\mathbf{K}[\mathbf{z}]}$ defined by

$$\langle \boldsymbol{\beta}, \boldsymbol{\alpha} \rangle_{\mathbf{K}[\mathbf{z}]} := \int_0^1 \boldsymbol{\beta}_t^\top \mathbf{K}[\mathbf{z}_t] \boldsymbol{\alpha}_t dt \tag{2.15}$$

for $\boldsymbol{\alpha}, \boldsymbol{\beta} \in L^2([0,1];\mathbb{R}^{dN})$ defines an equivalent inner product on $L^2([0,1];\mathbb{R}^{dN})$. Finally, owing to the boundedness of first order partial derivatives, we have Lipschitz continuity of $\mathbf{K}[\cdot]$, namely for all $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^{dN}$ it holds

$$\|\mathbf{K}[\mathbf{z}] - \mathbf{K}[\mathbf{z}']\| \leq L_{\mathbf{K}} \|\mathbf{z} - \mathbf{z}'\|_2 \tag{2.16}$$

for some constant $L_{\mathbf{K}} > 0$.

The properties just listed, combined with mild regularity assumptions on $\boldsymbol{\beta} : [0,1] \to \mathbb{R}^{dN}$ are enough to recover existence and uniqueness of solutions of the ODE driving the particles provided we generalize the notion of solution to an ODE introduced at the beginning of Section 1.1.1. For this purpose, consider the Cauchy problem

$$\begin{cases} \dot{\mathbf{z}}_t = \mathbf{K}[\mathbf{z}_t]\boldsymbol{\beta}_t, \quad t \in [0,1] \text{ a.e.}, \\ \mathbf{z}_0 = \mathbf{x}. \end{cases} \tag{2.17}$$

and let a solution be any function $\mathbf{z} : [0,1] \to \mathbb{R}^{dN}$ such that

$$\mathbf{z}_t = \mathbf{z}_0 + \int_0^1 \mathbf{K}[\mathbf{z}_t]\boldsymbol{\beta}_t dt. \tag{2.18}$$

Then, the following result shows that it is sufficient to take

$$\boldsymbol{\beta} \in \mathcal{B} := L^2([0,1];\mathbb{R}^{dN}), \tag{2.19}$$

where the notation $\mathcal{B}$ explicitly recalls the one introduced in Section 1.1.1 for the set of admissible controls. Note further that for simplicity, from now on, we will work under the assumption that $\mathcal{X} = \mathbb{R}^d$.

Lemma 2.1.2. *Assume $\boldsymbol{\beta} \in \mathcal{B}$, and consider the Cauchy Problem*

$$\begin{cases} \dot{\mathbf{z}}_t = \mathbf{K}[\mathbf{z}_t]\boldsymbol{\beta}_t, \quad t \in [0,1] \text{ a.e.}, \\ \mathbf{z}_0 = \mathbf{x}. \end{cases} \tag{2.20}$$

Then, the above admits a unique generalized solution for all $\mathbf{x} \in \mathcal{X}$.

Proof. First, the the map

$$(\mathbf{z}, \boldsymbol{\beta}) \mapsto \mathbf{K}[\mathbf{z}]\boldsymbol{\beta} \tag{2.21}$$

is Lipschitz continuous in both $\mathbf{z}$ and $\boldsymbol{\beta}$. This entails that $\mathbf{z} \mapsto \mathbf{K}[\mathbf{z}]\boldsymbol{\beta}_t$ is continuous for a.e. $t \in [0,1]$, $t \mapsto \mathbf{K}[\mathbf{z}]\boldsymbol{\beta}_t$ is measurable for all $\mathbf{z} \in \mathbb{R}^{dN}$, and, owing to the boundedness of the kernel, we have that

$$\|\mathbf{K}[\mathbf{z}]\boldsymbol{\beta}_t\|_2 \leq \Delta \|\boldsymbol{\beta}_t\|_2, \tag{2.22}$$

with $t \mapsto \|\boldsymbol{\beta}_t\|_2$ being summable. Finally, Lipschitz continuity of the kernel implies

$$\|\mathbf{K}[\mathbf{z}]\boldsymbol{\beta}_t - \mathbf{K}[\mathbf{z}']\boldsymbol{\beta}_t\|_2 \leq L_{\mathbf{K}} \|\mathbf{z} - \mathbf{z}'\|_2 \|\boldsymbol{\beta}_t\|_2 \tag{2.23}$$

for all $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^{dN}$. This is sufficient for a global version of Carathéodory's classical existence and uniqueness theorem (Theorem 1.45 of Chapter 1 in Roubicek [2005]) to hold. □

Remark. Denote $\phi^{\boldsymbol{\beta}}$ the map resulting from the integration of (2.20) for some fixed $\boldsymbol{\beta} \in \mathcal{B}$. Dupuis et al. [1998] show that, under mild regularity assumptions on $\mathcal{H}$ (and, in particular, under Assumption 1), $\phi^{\boldsymbol{\beta}} \in \text{diff}(\mathbb{R}^d)$, where $\text{diff}(\mathbb{R}^d)$ denotes the set of diffeomorphisms from $\mathbb{R}^d$ to $\mathbb{R}^d$. Furthermore, let us remark that for both this property and Lemma 2.1.2 to hold, the assumption $\|\boldsymbol{\beta}\|^2_{\mathbf{K}[\mathbf{z}]} \geq \delta \|\boldsymbol{\beta}\|_2^2$ is not needed.

Remark. Let us now address the fact that the assumption that there exists $\delta > 0$ such that $\|\boldsymbol{\beta}\|^2_{\mathbf{K}[\mathbf{z}]} \geq \delta \|\boldsymbol{\beta}\|_2^2$ may appear too strong. This, however, proves to be crucial for the existence of minimizers result to establish. Thanks to Lemma 2.1.2 we can give a heuristic on how one may consider kernels $\mathcal{K}$ with this property. The idea is to consider the following modified matrix

$$\tilde{\mathbf{K}}[\mathbf{z}] := \mathbf{K}[\mathbf{z}] + \delta I. \tag{2.24}$$

First, we trivially have $\|\boldsymbol{\beta}\|^2_{\tilde{\mathbf{K}}[\mathbf{z}]} \geq \delta \|\boldsymbol{\beta}\|_2^2$ by construction. Then, it is easy to check that the arguments provided in the proof of Lemma 2.1.2 still hold if we replace $\mathbf{K}$ with $\tilde{\mathbf{K}}$. By the uniqueness result we have that the trajectories of two particles $z_t^{(i)}$ and $z_t^{(j)}$ never coincide for a non vanishing Lebesgue measure set in [0,1], unless the initial conditions coincide, which we assume to never be the case. But this means that we can consider the modified kernel

$$\hat{\mathcal{K}}(x, x') := \mathcal{K}(x, x') + \delta\, D(x, x'), \tag{2.25}$$

where $D(x, x')$ denotes the d-dimentional Dirac delta function centered around $x \in \mathbb{R}^d$. Then, denoting $\hat{\mathbf{K}}$ the usual kernel matrix associated to $\hat{\mathcal{K}}$, we have that for all sets of distinct particles $\mathbf{z} = (z^{(1)}; \ldots; z^{(N)})$, that is $z^{(i)} \neq z^{(j)}$ if $i \neq j$, it holds

$$\hat{\mathbf{K}}[\mathbf{z}] = \tilde{\mathbf{K}}[\mathbf{z}]. \tag{2.26}$$

In particular this happens along the solution $\mathbf{z}_t$ to the modified ODE. This entails that $\mathcal{K}$ is a kernel with the desired property as it behaves as a "regularized" version of the original kernel along the ODE solutions.

With the aim of further reducing Problem 2, we notice that the minimization over the finite-dimensional space $\mathcal{V}_m$ can be eliminated. In order to make things more concrete, choose $\mathcal{V}_m$ to be an m-dimensional space of polynomials and assume that μ is absolutely continuous with respect to the Lebesgue measure and is such that $\mathcal{V}_m \subset L^2_\mu(\mathbb{R}^d; \mathcal{Y})$. Then, let us consider $\|\cdot\|_{\mathcal{V}_m} = \|\cdot\|_{L^2_\mu(\mathbb{R}^d;\mathcal{Y})}$, an L^2_μ-orthonormal basis $\{f_1, \ldots, f_m\}$ of $\mathcal{V}_m$, and the associated Vandermonde matrix $\mathbf{V}[\mathbf{z}] \in \mathbb{R}^{N \times m}$, namely $\mathbf{V}[\mathbf{z}]_{ij} = f_j(z^{(i)})$, $i = 1, \ldots, N$, $j = 1, \ldots, m$. Assume further the regularization parameter $\eta > 0$ and $\|f\|_{\mathcal{V}_m} = \|\mathbf{c}\|_2$ for $f = \sum_{i=1}^m c_i f_i$. Then, we have that

$$\mathbf{c}^\star_\eta[\mathbf{z}] := \underset{\mathbf{c} \in \mathbb{R}^m}{\arg\min} \quad \sum_{i=1}^N \left\{ \left(\sum_{j=1}^m c_j f_j(z^{(i)}) - y^{(i)} \right)^2 + \eta c_i^2 \right\} \tag{2.27}$$

is the solution of the well-known normal equations and, therefore, admits the representation

$$\mathbf{c}_\eta^\star[\mathbf{z}] = \Big(\mathbf{V}[\mathbf{z}]^\top \mathbf{V}[\mathbf{z}] + \eta I\Big)^{-1} \mathbf{V}[\mathbf{z}]^\top \mathbf{y}, \tag{2.28}$$

which is continuous in $\mathbf{z}$. Then we can define

$$\mathcal{E}(\mathbf{z},\mathbf{y}) := \min_{\mathbf{c}\in\mathbb{R}^m} \quad \sum_{i=1}^N \left\{ \left(\sum_{j=1}^m c_j f_j(z^{(i)}) - y^{(i)} \right)^2 + \eta c_i^2 \right\} = \left\| \mathbf{V}[\mathbf{z}]\mathbf{c}_\eta^\star[\mathbf{z}] - \mathbf{y} \right\|_2^2 + \eta \left\| \mathbf{c}^\star \right\|_2^2 \tag{2.29}$$

which, owing to the continuity of $\mathbf{c}_\eta^\star[\mathbf{z}]$, is continuous in $\mathbf{z}$. We can now recast Problem 2 in an equivalent yet simplified form.

Problem 3. Given a sample $(x^{(1)};\dots;x^{(N)})$ and possibly noisy observations $(y^{(1)};\dots;y^{(N)}) := (g(x^{(1)})+\varepsilon^{(1)},\dots,g(x^{(N)})+\varepsilon^{(N)})$ where $\varepsilon^{(i)} \overset{\text{iid}}{\sim} \varepsilon$, solve

$$\begin{aligned} \min_{\boldsymbol{\beta}\in\mathcal{B}} \quad & \frac{1}{2N}\mathcal{E}(\mathbf{z}_1,\mathbf{y}) + \frac{\gamma}{2}\int_0^1 \boldsymbol{\beta}_t^\top \mathbf{K}[\mathbf{z}_t]\boldsymbol{\beta}_t dt \\ \text{subject to} \quad & \dot{\mathbf{z}}_t = \mathbf{K}[\mathbf{z}_t]\boldsymbol{\beta}_t, \\ & \mathbf{z}_0 = \mathbf{x}. \end{aligned} \tag{2.30}$$

Thanks to Lemma 2.1.2, for Problem 3 to be well defined under Assumption 1 we only need to show the existence of minimizers, which is the goal of the next two statements.

Lemma 2.1.3. *Let $(\boldsymbol{\beta}^i)_{i\geq 0} \subset \mathcal{B}$ be an L^2-weakly convergent sequence, namely $\boldsymbol{\beta}^i \rightharpoonup \boldsymbol{\beta}$ for some $\boldsymbol{\beta}\in\mathcal{B}$, and denote $\mathbf{z}^{\boldsymbol{\beta}^i} : [0,1] \to \mathbb{R}^{dN}$ the solution of (2.20) for the control $\boldsymbol{\beta}^i$. Then, $\left\|\mathbf{z}^{\boldsymbol{\beta}^i} - \mathbf{z}^{\boldsymbol{\beta}}\right\|_\infty \to 0$ as $i\to\infty$.*

Proof. In this proof we will make use of the constant Δ as defined in the remark following Assumption 1. As $(\boldsymbol{\beta}^i)_{i\geq 0}$ is weakly convergent it is bounded in $\mathcal{B}$ by some constant $C>0$. Moreover, for all $i\geq 0$ it holds

$$\left\|\dot{\mathbf{z}}_t^{\boldsymbol{\beta}^i}\right\|_2 = \left\|\mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}^i}]\boldsymbol{\beta}_t^i\right\|_2 \leq \Delta \left\|\boldsymbol{\beta}_t^i\right\|_2 \tag{2.31}$$

which, by boundedness of $(\boldsymbol{\beta}^i)_{i\geq 0}$, implies that there exists a compact subset $K\subset\mathbb{R}^{dN}$ depending on $\mathbf{x}$ such that $\mathbf{z}_t^{\boldsymbol{\beta}^i} \in K$ for all $i\geq 0$ and $t\in[0,1]$. The bound in (2.31) yields

$$\left\|\dot{\mathbf{z}}^{\boldsymbol{\beta}^i}\right\|_{L^2} \leq \Delta C, \tag{2.32}$$

and we deduce that $(\mathbf{z}^{\boldsymbol{\beta}^i})_{i\geq 0} \subset W^{1,2}([0,1];\mathbb{R}^{dN})$ is bounded. This entails that it is pre-compact[1] and, hence, there exists a $W^{1,2}$-weakly converging subsequence $(\mathbf{z}^{\boldsymbol{\beta}^i})_{i\geq 0}$, the weak limit being denoted by $\mathbf{z}^\infty$. Owing to the compact inclusion $W^{1,2}([0,1];\mathbb{R}^{dN}) \hookrightarrow$

[1] A pre-compact subset, or relatively compact subset, is a subset whose closure is compact.

$C([0,1];\mathbb{R}^{dN})$ it follows that $\mathbf{z}^{\boldsymbol{\beta}^i} \to \mathbf{z}^\infty$ as $i \to \infty$ uniformly. In particular, we have $\mathbf{z}_0^\infty = \mathbf{x}$ and for $\boldsymbol{\alpha} \in \mathcal{B}$

$$\begin{aligned}
&\left| \int_0^1 \boldsymbol{\alpha}_t^\top \left(\mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}^i}]\boldsymbol{\beta}_t^i - \mathbf{K}[\mathbf{z}_t^\infty]\boldsymbol{\beta}_t \right) dt \right| \\
\leq &\left| \int_0^1 \boldsymbol{\alpha}_t^\top \left(\mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}^i}]\boldsymbol{\beta}_t^i - \mathbf{K}[\mathbf{z}_t^\infty]\boldsymbol{\beta}_t^i \right) dt \right| + \left| \int_0^1 \boldsymbol{\alpha}_t^\top \left(\mathbf{K}[\mathbf{z}_t^\infty]\boldsymbol{\beta}_t^i - \mathbf{K}[\mathbf{z}_t^\infty]\boldsymbol{\beta}_t \right) dt \right| \\
\leq &C L_{\mathbf{K}} \|\boldsymbol{\alpha}\|_{L^2} \left\| \mathbf{z}^{\boldsymbol{\beta}^i} - \mathbf{z}^\infty \right\|_\infty + \int_0^1 \left(\mathbf{K}[\mathbf{z}_t^\infty]\boldsymbol{\alpha}_t \right)^\top \left(\boldsymbol{\beta}_t^i - \boldsymbol{\beta}_t \right) dt \xrightarrow[i\to\infty]{} 0
\end{aligned} \tag{2.33}$$

where in the last line we used Lipschitz continuity of $\mathbf{K}[\cdot]$, boundedness of $\mathbf{K}[\cdot]$ (which entails the map $t \mapsto \mathbf{K}[\mathbf{z}_t^\infty]\boldsymbol{\alpha}_t$ is in $\mathcal{B}$) and the fact that $\boldsymbol{\beta}^i \rightharpoonup \boldsymbol{\beta}$. It then follows that

$$\begin{cases} \dot{\mathbf{z}}_t^\infty = \mathbf{K}[\mathbf{z}_t^\infty]\boldsymbol{\beta}_t, \quad t \in [0,1], \text{ a.e.} \\ \mathbf{z}_0^\infty = \mathbf{x}, \end{cases} \tag{2.34}$$

which, in turn, implies $\mathbf{z}^\infty = \mathbf{z}^{\boldsymbol{\beta}}$. Finally, as this holds true for all weak limiting points of subsequences of $(\mathbf{z}^{\boldsymbol{\beta}^i})_{i\geq 0}$, we obtain that the whole sequence is weakly convergent. Repeating the arguments above we then recover the thesis. □

Theorem 2.1.4. *Problem 3 admits a solution.*

Proof. The result follows from the classical Direct Method of Calculus of Variations. Define the functional $\mathcal{J} : \mathcal{B} \to \mathbb{R}$ as

$$\mathcal{J}(\boldsymbol{\beta}) := \frac{1}{2N}\mathcal{E}(\mathbf{z}_1, \mathbf{y}) + \frac{\gamma}{2}\int_0^1 \boldsymbol{\beta}_t^\top \mathbf{K}[\mathbf{z}_t]\boldsymbol{\beta}_t dt \tag{2.35}$$

which trivially satisfies $\mathcal{J}(\boldsymbol{\beta}) \geq 0$ and the relation

$$\mathcal{J}(\boldsymbol{\beta}) \geq \frac{\delta\gamma}{2}\int_0^1 \|\boldsymbol{\beta}_t\|_2^2 \, dt \tag{2.36}$$

owing to Assumption 1. This, in turn, implies that letting $\|\boldsymbol{\beta}\|_{L^2} \to \infty$ we have $\mathcal{J}(\boldsymbol{\beta}) \to \infty$, while assuming $J(\boldsymbol{\beta}) \leq c$ for some $c > 0$ yields $\|\boldsymbol{\beta}\|_{L^2} \leq k$ for some $k > 0$. The former implies that the search for a minimum can be restricted to a sufficiently large bounded subset $K \subset \mathcal{B}$, while the latter means that the functional is coercive on $\mathcal{B}$. Now let $(\boldsymbol{\beta}^i)_{i\geq 0} \subset \mathcal{B}$ be an L^2-weakly convergent sequence to $\boldsymbol{\beta} \in \mathcal{B}$ and consider the following decomposition

$$\begin{aligned}
&\int_0^1 \left(\boldsymbol{\beta}_t^i\right)^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}^i}]\boldsymbol{\beta}_t^i dt \\
= &\int_0^1 \left(\boldsymbol{\beta}_t^i\right)^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}^i}]\boldsymbol{\beta}_t^i dt - \int_0^1 \left(\boldsymbol{\beta}_t^i\right)^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}}]\boldsymbol{\beta}_t^i dt + \int_0^1 \left(\boldsymbol{\beta}_t^i\right)^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}}]\boldsymbol{\beta}_t^i dt.
\end{aligned} \tag{2.37}$$

On one hand we have

$$\begin{aligned}
&\left| \int_0^1 \left(\boldsymbol{\beta}_t^i\right)^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}^i}]\boldsymbol{\beta}_t^i dt - \int_0^1 \left(\boldsymbol{\beta}_t^i\right)^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}}]\boldsymbol{\beta}_t^i dt \right| \\
\leq &\int_0^1 \left\| \left(\boldsymbol{\beta}_t^i\right)^\top \left(\mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}^i}] - \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}}]\right)\boldsymbol{\beta}_t^i \right\|_2 dt \\
\leq &L_{\mathbf{K}} \left\| \boldsymbol{\beta}^i \right\|_{L^2}^2 \left\| \mathbf{z}^{\boldsymbol{\beta}^i} - \mathbf{z}^{\boldsymbol{\beta}} \right\|_\infty \xrightarrow[i\to\infty]{} 0
\end{aligned} \tag{2.38}$$

where we used the fact that $\mathbf{z}^{\boldsymbol{\beta}^i} \to \mathbf{z}^{\boldsymbol{\beta}}$ uniformly as $i \to \infty$ by virtue of Lemma 2.1.3 and $(\boldsymbol{\beta}^i)_{i\geq 0}$ is bounded. On the other hand, exploiting the fact that the map

$$(\boldsymbol{\rho}, \boldsymbol{\alpha}) \mapsto \int_0^1 \boldsymbol{\rho}_t^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}}]\boldsymbol{\alpha}_t dt \tag{2.39}$$

defines an equivalent inner product on $\mathcal{B}$ owing to Assumption 1, we have that the map

$$\boldsymbol{\rho} \mapsto \int_0^1 (\boldsymbol{\rho}_t)^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}}]\boldsymbol{\rho}_t dt \tag{2.40}$$

is weakly lower semi-continuous. Finally, by continuity of the map $\mathbf{z}_1 \mapsto \mathcal{E}(\mathbf{z}_1, \mathbf{y})$, we conclude that $\mathcal{J}$ is weakly lower semi-continuous and the claim follows. □

Remark. As it is remarked in Owhadi [2020], hoping for uniqueness of solutions to problems of the same kind of Problem 3 is in vain as this is violated by the most simple counterexamples.

2.1.2 Characterization of minimizers

We are now ready to leverage the formulation of Problem 3 which, by design, falls exactly into the framework of control problems. This, as we have already seen and discussed in previous sections, allows us to characterize the solutions to the learning problem and, ultimately, will suggest numerical strategies for their explicit computation. Notice finally that, by virtue of the Optimal Control view, in the following we are sometimes going to adopt the relevant jargon introduced in Section 1.1.1 calling, for instance, $\boldsymbol{\beta} \in \mathcal{B}$ a control and the associated ODE solution $\mathbf{z}^{\boldsymbol{\beta}}$ the state trajectory.

Theorem 2.1.5. *Let $\boldsymbol{\beta} \in \mathcal{B}$ be an Optimal Control for Problem 3 and $\mathbf{z} \in C([0,T];\mathbb{R}^{dN})$ be the corresponding optimal state trajectory. Then, there exists a function $\mathbf{p} \in C([0,T];\mathbb{R}^{dN})$, the adjoint state, satisfying the following optimality system:*

$$\begin{cases} \dot{\mathbf{z}}_t = \dfrac{1}{\gamma}\mathbf{K}[\mathbf{z}_t]\mathbf{p}_t, \quad \mathbf{z}_0 = \mathbf{x}, \\ \dot{\mathbf{p}}_t = -\dfrac{1}{2\gamma}\partial_{\mathbf{z}}\left(\mathbf{p}_t^\top \mathbf{K}[\mathbf{z}_t]\mathbf{p}_t\right), \quad \mathbf{p}_1 = -\dfrac{1}{2N}\partial_{\mathbf{z}}\mathcal{E}(\mathbf{z}_1, \mathbf{y}), \\ \boldsymbol{\beta}_t = \dfrac{1}{\gamma}\mathbf{p}_t. \end{cases} \tag{2.41}$$

where $\partial_{\mathbf{z}}\left(\mathbf{p}_t^\top \mathbf{K}[\mathbf{z}_t]\mathbf{p}_t\right)$ and $\partial_{\mathbf{z}}\mathcal{E}(\mathbf{z}_1, \mathbf{y})$ are notation for

$$\partial_{\mathbf{z}}\left(\mathbf{p}_t^\top \mathbf{K}[\mathbf{z}]\mathbf{p}_t\right)\Big|_{\mathbf{z}=\mathbf{z}_t}, \quad \partial_{\mathbf{z}}\mathcal{E}(\mathbf{z}, \mathbf{y})|_{\mathbf{z}=\mathbf{z}_1} \tag{2.42}$$

respectively.

Proof. The proof follows from the Pontryagin Maximum Principle. Indeed, the Hamiltonian $H : \mathbb{R}^{dN} \times \mathbb{R}^{dN} \times \mathbb{R}^{dN} \to \mathbb{R}$ associated to Problem 3 is

$$H(\mathbf{z}, \mathbf{p}, \boldsymbol{\beta}) = \mathbf{p}^\top \mathbf{K}[\mathbf{z}]\boldsymbol{\beta} - \frac{\gamma}{2}\boldsymbol{\beta}^\top \mathbf{K}[\mathbf{z}]\boldsymbol{\beta}. \tag{2.43}$$

The above is easily maximized with respect to the control as Assumption 1 ensures $\mathbf{K}[\mathbf{z}]$ is invertible. Maximization yields

$$\boldsymbol{\beta}_t = \arg\max_{\boldsymbol{\beta}\in\mathcal{B}} H(\mathbf{z}_t, \boldsymbol{\beta}_t, \mathbf{p}_t) = \frac{1}{\gamma}\mathbf{p}_t, \tag{2.44}$$

so that the third equation in (2.41) is proved. The first two then directly follow from the canonical equations

$$\begin{aligned} \dot{\mathbf{z}}_t &= \partial_{\mathbf{p}} H(\mathbf{z}_t, \mathbf{p}_t, \boldsymbol{\beta}_t), \quad \mathbf{z}_0 = \mathbf{x}, \\ \dot{\mathbf{p}}_t &= -\partial_{\mathbf{z}} H(\mathbf{z}_t, \mathbf{p}_t, \boldsymbol{\beta}_t), \quad \mathbf{p}_1 = -\frac{1}{2N}\partial_{\mathbf{z}}\mathcal{E}(\mathbf{z}_1, \mathbf{y}). \end{aligned} \tag{2.45}$$

□

Remark. Owing to Theorem 2.1.5 and the regularity assumptions on $\mathcal{K}$ stated in Assumption 1, we recover regularity of minimizers of Problem 3, indeed we have that any minimizer $\boldsymbol{\beta} \in C^2([0,1]; \mathbb{R}^{dN})$.

There are some interesting consequences stemming from Theorem 2.1.5 as far as the characterization of solutions of the learning problem are concerned. First and foremost, we remark the importance of (2.41) as the problem of minimizing a cost functional is recast as a TPBVP where the control has been effectively eliminated. Moreover, the whole dynamics can be further reduced and be expressed in terms of the state only. Indeed, from (2.41) it follows immediately that the adjoint state is related to the time derivative of the state by the relation

$$\mathbf{p}_t = \gamma \mathbf{K}[\mathbf{z}_t]^{-1}\dot{\mathbf{z}}_t. \tag{2.46}$$

which, in turn, implies that the relevant equations describing the system are

$$\frac{d}{dt}\left(\mathbf{K}[\mathbf{z}_t]^{-1}\dot{\mathbf{z}}_t\right) - \frac{1}{2}\partial_{\mathbf{z}}\left(\dot{\mathbf{z}}_t^\top \mathbf{K}[\mathbf{z}_t]^{-1}\dot{\mathbf{z}}_t\right) = 0 \tag{2.47}$$

along with conditions

$$\mathbf{z}_0 = \mathbf{x}, \quad \dot{\mathbf{z}}_1 = -\frac{1}{2\gamma N}\mathbf{K}[\mathbf{z}_t]\partial_{\mathbf{z}}\mathcal{E}(\mathbf{z}_1, \mathbf{y}). \tag{2.48}$$

Equations of motion such as the ones just derived can be recovered (Owhadi [2020]) as the Euler-Lagrange equations stemming from the least action principle associated to the following action:

$$\mathcal{A}(\mathbf{z}) := \int_0^1 L(\mathbf{z}_t, \dot{\mathbf{z}}_t)dt, \tag{2.49}$$

the Lagrangian being defined by

$$L(\mathbf{z}, \dot{\mathbf{z}}) := \dot{\mathbf{z}}^\top \mathbf{K}[\mathbf{z}]^{-1}\dot{\mathbf{z}} = \|\dot{\mathbf{z}}\|^2_{\mathbf{K}[\mathbf{z}]^{-1}}. \tag{2.50}$$

The action should be minimized over trajectories $\mathbf{z} \in C^1([0,1]; \mathbb{R}^{dN})$ constrained to $\mathbf{z}_0 = \mathbf{x}$ and $\mathbf{z}_1$ as in the solution to the original problem. This formulation suggests that the map

$\mathbb{R}^{dN} \ni \mathbf{z} \mapsto \mathbf{K}[\mathbf{z}]^{-1}$ may be interpreted as a metric tensor and the Euler-Lagrange equations are equivalent to the equations of geodesic motion corresponding to the minimization of the length $\int_0^1 \sqrt{\dot{\mathbf{z}}_t{}^\top \mathbf{K}[\mathbf{z}_t]^{-1}\dot{\mathbf{z}}_t}dt$ of the curve $\mathbf{z} : [0,1] \to \mathbb{R}^{dN}$ connecting $\mathbf{x}$ and $\mathbf{z}_1$. Hence, the role of the kernel is to define a geometry on the space $\mathbb{R}^{dN}$ which, in turn, defines the geodesics along which the dynamics moves. Introducing again the adjoint state $\mathbf{p}$, the Hamiltonian corresponding to the dynamics in (2.41) reads[2]

$$H(\mathbf{z}, \mathbf{p}) := \frac{1}{2\gamma}\mathbf{p}^\top \mathbf{K}[\mathbf{z}]\mathbf{p} = \|\mathbf{p}\|^2_{\mathbf{K}[\mathbf{z}]}\,. \tag{2.51}$$

Owing to its time independence it is well-known that it is conserved along solutions of the ODE. Then, the role of the regularization term is to impose a total budget on $\|\mathbf{p}_t\|_{\mathbf{K}[\mathbf{z}_t]}$ at all times and, as it turns out, the optimal allocation is constant in time. This phenomenon is also reminiscent, from an optimization point of view, of the observation that oftentimes regularization on the norm of model parameters is actually equivalent to a constraint on their maximum norm. Indeed, the control $\boldsymbol{\beta}_t$ may be seen as being restricted to a ball (of radius proportional to γ) in the norm $\|\cdot\|_{\mathbf{K}[\mathbf{z}_t]}$ for all $t \in [0,1]$.

Problem 3 is only concerned with training points and their optimal trajectories. Once the latter are found, we wish to evaluate the model at some $x \in \mathcal{X}$ by integrating the ODE with the appropriate initial condition. Letting as always the mapping $(x,t) \mapsto z_t(x)$ denote such propagation operator, it follows that[3]

$$\dot{z}_t(x) = \mathbf{K}[z_t(x), \mathbf{z}_t]\mathbf{K}[\mathbf{z}_t]^{-1}\dot{\mathbf{z}}_t, \quad z_0 = x. \tag{2.52}$$

As one would expect, the above coincides with the minimum norm interpolation in the RKHS of the time derivative at the "anchor" points $\mathbf{z}_t = (z_t^{(1)}; \dots; z_t^{(N)})$, namely

$$\begin{aligned} \dot{z}_t(\cdot) = \ & \underset{v_t \in \mathcal{H}}{\arg\min} \quad \|v_t\|^2_{\mathcal{H}} \\ & \text{subject to} \quad v_t(x^{(i)}) = \dot{z}_t^{(i)}, \quad i = 1, \dots, N. \end{aligned} \tag{2.53}$$

2.1.3 The role of adjoint variables

We have seen that the adjoint variables can be interpreted as generalized momentum variables as classicaly defined in the context of Hamiltonian dynamics. Furthermore, Theorem 2.1.5 reveals that $p^{(i)}$ is precisely proportional to the Optimal Control and, as such, represents the contribution of data pair $(x^{(i)}, y^{(i)})$ to the regressor[4]. First and

[2] According to the notation introduced in Section 2.1 this corresponds to the reduced Hamiltonian $H^\star$ where the control is eliminated by maximization. As here we mostly work with $H^\star$ we instead denote it by H for ease of notation.

[3] Whenever $\mathbf{x} = (x^{(1)}; \dots; x^{(N_1)}) \in \mathbb{R}^{dN_1}$ and $\mathbf{z} = (z^{(1)}; \dots; z^{(N_2)}) \in \mathbb{R}^{dN_2}$ we extend notation $\mathbf{K}[\mathbf{x}, \mathbf{z}]$ to denote the $dN_1 \times dN_2$ block-matrix where the (i,j)-th block is given by $\mathcal{K}(x^{(i)}, z^{(j)})$.

[4] This is what classically happens in Support Vector Machines (James et al. [2013]). In the context of classification, employing the hinge loss leads to a regressor defined only by the points on the margin (i.e. those closer to the points of the opposite class).

foremost, by the Optimal Control tools introduced in Section 1.1.1 we can adopt the Hamilton-Jacobi-Bellman point of view of Problem 3 which leads to the following PDE:

$$\partial_t V(\mathbf{z},t) + \frac{1}{2\gamma}\left\|\partial_{\mathbf{z}} V(\mathbf{z},t)\right\|^2_{\mathbf{K}[\mathbf{z}]} = 0, \quad V(\cdot,1) = \frac{1}{2N}\mathcal{E}(\cdot,\mathbf{y}), \tag{2.54}$$

where $V : \mathbb{R}^{dN} \times [0,1] \to \mathbb{R}$ denotes the value function. Moreover, owing to the link between PMP and HJB equation through the method of characteristics highlighted in Section 1.1.4, the adjoint variables are related to the value function by

$$\mathbf{p}_t = -\partial_{\mathbf{z}} V(\mathbf{z}_t, t), \tag{2.55}$$

the pair $(\mathbf{z},\mathbf{p}) : [0,1] \to \mathbb{R}^{dN} \times \mathbb{R}^{dN}$ being a solution to the PMP. Hence, for all fixed time instants $t \in [0,1]$ the squared $\|\cdot\|_{\mathbf{K}[\mathbf{z}_t]}$ norm of adjoint variables is interpreted as the instantaneous-in-time decrease of the value function along the optimal path. Also, (2.55) shows that $p_t^{(i)}$ represents the sensitivity of the value function with respect to the position of the i-th particle, therefore supporting the proposed interpretation. Arguably the most important observation in this sense is the one expressed by the following theorem.

Proposition 2.1.6. *Let $(\mathbf{z},\mathbf{p})$ be an optimal pair satisfying the PMP as defined in Theorem 2.1.5. Assume further that $p_1^{(i)} = 0$ for some $i \in \{1,\dots,N\}$, then $p_t^{(i)} = 0$ for all $t \in [0,1]$.*

Proof. Consider the reversed trajectories $(\bar{\mathbf{z}}_t, \bar{\mathbf{p}}_t) = (\mathbf{z}_{1-t}, -\mathbf{p}_{1-t})$, $t \in [0,1]$. It is easy to check that they are still a solution to the Hamiltonian ODE in (2.41) with time-reversed boundary conditions (i.e. the condition on the state is given at time $t = 1$ while the one on the adjoint variable is given at time $t = 0$). Then, $\bar{p}_0^{(i)} = 0$ is sufficient to prove $\dot{\bar{p}}_t^{(i)} = 0$ for all $t \in [0,1]$ and the thesis follows by direct integration. □

As far as interpreting the role of adjoint variables, it is also interesting to consider the instance when the particles are split into two groups, one of them having a fixed control.

Proposition 2.1.7. *Let $\mathbf{x}^1 \in \mathcal{X}^{N_1}$, $\mathbf{x}^2 \in \mathcal{X}^{N_2}$ represent a partition of a sample of total size $N = N_1 + N_2$, and $\mathbf{y}^1 \in \mathcal{Y}^{N_1}$, $\mathbf{y}^2 \in \mathcal{Y}^{N_2}$ the corresponding two sets of observations. Assume further $\boldsymbol{\beta}^1 \in L^2([0,1];\mathbb{R}^{dN_1})$ is given. Write in short $\mathbf{x} = (\mathbf{x}^1;\mathbf{x}^2)$ and similarly for all other vectors concerning the two sets of observations. The solution to*

$$\begin{aligned} \min_{\boldsymbol{\beta}^2 \in L^2([0,1];\mathbb{R}^{dN_2})} \quad & \frac{1}{2N}\mathcal{E}(\mathbf{z}_1,\mathbf{y}) + \frac{\gamma}{2}\int_0^1 \boldsymbol{\beta}_t^\top \mathbf{K}[\mathbf{z}_t]\boldsymbol{\beta}_t dt \\ \textit{subject to} \quad & \dot{\mathbf{z}}_t = \mathbf{K}[\mathbf{z}_t]\boldsymbol{\beta}_t, \\ & \mathbf{z}_0 = \mathbf{x}. \end{aligned} \tag{2.56}$$

satisfies the canonical equations

$$\begin{cases} \dot{\mathbf{z}}_t &= \partial_{\mathbf{p}} H(\mathbf{z}_t,\mathbf{p}_t,t), \quad \mathbf{z}_0 = \mathbf{x}, \\ \dot{\mathbf{p}}_t &= -\partial_{\mathbf{z}} H(\mathbf{z}_t,\mathbf{p}_t,t), \quad \mathbf{p}_1 = -\frac{1}{2N}\partial_{\mathbf{z}}\mathcal{E}(\mathbf{z}_1,\mathbf{y}), \\ \boldsymbol{\beta}_t^2 &= \frac{1}{\gamma}\mathbf{p}_t^2 - \mathbf{K}[\mathbf{z}_t^2,\mathbf{z}_t^2]^{-1}\mathbf{K}[\mathbf{z}_t^2,\mathbf{z}_t^1]\left(\boldsymbol{\beta}_t^1 - \frac{1}{\gamma}\mathbf{p}_t^1\right), \end{cases} \tag{2.57}$$

the Hamiltonian being defined as

$$H(\mathbf{z},\mathbf{p},t) := \frac{1}{2\gamma}\mathbf{p}^\top \mathbf{K}[\mathbf{z}]\mathbf{p} - \frac{\gamma}{2}\left(\boldsymbol{\beta}_t^1 - \frac{1}{\gamma}\mathbf{p}^1\right)^\top \mathbf{K}[\mathbf{z}^1|\mathbf{z}^2]\left(\boldsymbol{\beta}_t^1 - \frac{1}{\gamma}\mathbf{p}^1\right) \tag{2.58}$$

where we introduced the notation

$$\mathbf{K}[\mathbf{z}^1|\mathbf{z}^2] := \mathbf{K}[\mathbf{z}^1,\mathbf{z}^1] - \mathbf{K}[\mathbf{z}^1,\mathbf{z}^2]\mathbf{K}[\mathbf{z}^2,\mathbf{z}^2]^{-1}\mathbf{K}[\mathbf{z}^2,\mathbf{z}^1] \tag{2.59}$$

for the Schur complement of the block $\mathbf{K}[\mathbf{z}^2,\mathbf{z}^2]$ *of the matrix* $\mathbf{K}[\mathbf{z}]$.

Proof. The direct application of the PMP yields

$$\boldsymbol{\beta}_t^2 = \underset{\boldsymbol{\beta}^2\in\mathbb{R}^{dN_2}}{\arg\max}\, H(\mathbf{z},\boldsymbol{\beta}^2,\mathbf{p},t), \tag{2.60}$$

where

$$H(\mathbf{z},\boldsymbol{\beta}^2,\mathbf{p},t) := \mathbf{p}^\top \mathbf{K}[\mathbf{z}]\begin{pmatrix}\boldsymbol{\beta}_t^1\\ \boldsymbol{\beta}^2\end{pmatrix} - \frac{\gamma}{2}\begin{pmatrix}\boldsymbol{\beta}_t^1\\ \boldsymbol{\beta}^2\end{pmatrix}^\top \mathbf{K}[\mathbf{z}]\begin{pmatrix}\boldsymbol{\beta}_t^1\\ \boldsymbol{\beta}^2\end{pmatrix}. \tag{2.61}$$

This immediately implies the third condition in (2.57). Plugging the latter into (2.61) gives the Hamiltonian in (2.58) and the state and adjoint equations in (2.57) follow. □

This proposition reveals how the momentum variables $\mathbf{p}_t^1$ indeed measure the contribution to the regressor of first group of particles, even though the associated control is fixed. The term $\boldsymbol{\beta}_t^1 - \frac{1}{\gamma}\mathbf{p}^1$ represents the "shock" there is at time t between the gradient information relevant to the first group of particles and the employed control (which, upon minimization over $\boldsymbol{\beta}^1 \in L^2([0,1],\mathbb{R}^{dN_1})$, vanishes as we recover the original full solution of Theorem 2.1.5). Then, if the Optimal Control is not used, the contribution of the first group of particles encompassed by momentum variables affects (proportionally to the shock $\boldsymbol{\beta}_t^1 - \frac{1}{\gamma}\mathbf{p}^1$) the Optimal Control of the second group of particles as is shown in the third equation of(2.57).

2.1.4 The discrete-in-time Optimal Control problem

The formulations introduced and studied in the previous sections involve either time or space-time continuity. These, on one hand, provide a simplified framework to study the properties of the learning problem we are concerned with. On the other hand, they provide valuable insight into sensible methodologies for their numerical solution. The fist step in this direction is to write down a discretized version of the problem. In particular, we take Problem 3 as a starting point and introduce a suitable time discretization. In this section we hence consider a partition of the time interval [0,1] defined by a time grid $0 = t_0 < t_1 \ldots < t_L = 1$ where the l-th time step is $h_l := t_{l+1} - t_l$, $l = 0,\ldots,L-1$. Owing to the Deep Learning view highlighted in this chapter, we refer to $l \in \{0,\ldots,L\}$ as the l-th layer and the integer L, which controls the granularity of the grid, as the number of layers. Problem 3 involves both an objective functional as well as the state ODE, therefore it is natural to turn to a Runge-Kutta ODE integration method (Hairer et al. [2006]) characterized by coefficients $\{a_{ij},b_i\}_{i,j=1}^s$ and collocation points $\{c_i\}_{i=1}^s$. Given a

control $\boldsymbol{\beta} \in C([0,1]; \mathbb{R}^{dN})$ and the associated state function $\mathbf{z}$, let us introduce the notation

$$\mathbf{z}_l \approx \mathbf{z}_{t_l}, \quad \boldsymbol{\beta}_{li} = \boldsymbol{\beta}_{t_l + c_i h_l}, \quad i = 1, \dots, s, \ l = 0, \dots, L. \tag{2.62}$$

to denote, respectively, the discrete-in-time state approximations and the evaluations of the control at the collocation points specified by the chosen Runge-Kutta method and time grid. Moreover, with a slight abuse of notation, we denote $\mathbf{z}_{li}$ the i-th Runge-Kutta internal stage of the l-th layer, namely

$$\mathbf{z}_{li} = \mathbf{z}_l + h_l \sum_{i=1}^{L} a_{ij} \mathbf{K}[\mathbf{z}_{li}] \boldsymbol{\beta}_{li}, \quad i = 1, \dots, s, \ l = 0, \dots, L. \tag{2.63}$$

This procedure leads to the statement of the following problem.

Problem 4. Given a sample $(x^{(1)}; \dots; x^{(N)})$ and possibly noisy observations $(y^{(1)}; \dots; y^{(N)}) := (g(x^{(1)}) + \varepsilon^{(1)}, \dots, g(x^{(N)}) + \varepsilon^{(N)})$ where $\varepsilon^{(i)} \overset{\text{iid}}{\sim} \varepsilon$, choose a time grid $0 = t_0 < t_1 \dots < t_L = 1$, a Runge-Kutta method $\{a_{ij}, b_i\}_{i,j=1}^{s}$ and solve

$$\begin{aligned}
\min_{\boldsymbol{\beta}_{li} \in \mathbb{R}^{dN}} \quad & \frac{1}{2N} \mathcal{E}(\mathbf{z}_L, \mathbf{y}) + \frac{\gamma}{2} \sum_{l=0}^{L-1} h_l \sum_{i=1}^{s} b_i \boldsymbol{\beta}_{li}^\top \mathbf{K}[\mathbf{z}_{li}] \boldsymbol{\beta}_{li} \\
\text{subject to} \quad & \mathbf{z}_{l+1} = \mathbf{z}_l + h_l \sum_{i=1}^{s} b_i \mathbf{K}[\mathbf{z}_{li}] \boldsymbol{\beta}_{li}, \quad l = 0, \dots, L-1, \\
& \mathbf{z}_{li} = \mathbf{z}_l + h_l \sum_{j=1}^{s} a_{ij} \mathbf{K}[\mathbf{z}_{lj}] \boldsymbol{\beta}_{lj}, \quad i = 1, \dots, s, \ l = 0, \dots, L-1, \\
& \mathbf{z}_0 = \mathbf{x}.
\end{aligned} \tag{2.64}$$

The above is a fully discrete problem which provides the foundation for the the numerical solution of the learning problem. With standard constrained optimization techniques, first order optimality conditions for Problem 4 can be easily computed. At this point it is natural to ask is how the latter are related to their continuous-in-time counterpart given in Theorem 2.1.5. It turns out (Bonnans and Laurent-Varin [2006]) that the optimality conditions for Problem 4 are equivalent to the discretization of the continuous-time optimality conditions for Problem 3, provided the discretization scheme is carefully chosen. Indeed, the following statement holds.

Proposition 2.1.8. *Let $\mathbf{p}_l, \mathbf{p}_{li} \in \mathbb{R}^{dN}$, $i = 1, \dots, s$, $l = 1, \dots, L$, denote a set of adjoint variables and associated internal stages respectively. Assume the Runge-Kutta method in Problem 4 is such that $b_i \neq 0$ for all $i = 1 \dots, s$ and define*

$$\hat{b}_i := b_i, \quad \hat{a}_{ij} := b_j - \frac{b_j}{b_i} a_{ji}. \tag{2.65}$$

Then, the first order optimality conditions characterizing the solution to Problem 4 are

given by the following system:

$$\begin{cases} \mathbf{z}_{l+1} = \mathbf{z}_l + \frac{h_l}{\gamma}\sum_{i=1}^s b_i \mathbf{K}[\mathbf{z}_{li}]\mathbf{p}_{li}, & \mathbf{z}_{li} = \mathbf{z}_l + \frac{h_l}{\gamma}\sum_{i=1}^s a_{ij}\mathbf{K}[\mathbf{z}_{lj}]\mathbf{p}_{lj}, \\ \mathbf{p}_{l+1} = \mathbf{p}_l - \frac{h_l}{2\gamma}\sum_{i=1}^s \hat{b}_i \partial_{\mathbf{z}}\left(\mathbf{p}_{li}^\top \mathbf{K}[\mathbf{z}_{li}]\mathbf{p}_{li}\right), & \mathbf{p}_{li} = \mathbf{p}_l - \frac{h_l}{2\gamma}\sum_{i=1}^s \hat{a}_{ij}\partial_{\mathbf{z}}\left(\mathbf{p}_{lj}^\top \mathbf{K}[\mathbf{z}_{lj}]\mathbf{p}_{lj}\right), \\ \boldsymbol{\beta}_{li} = \frac{1}{\gamma}\mathbf{p}_{li}. \end{cases} \tag{2.66}$$

for $i = 1, \dots, s$, $l = 0, \dots, L-1$ *and*

$$\mathbf{z}_0 = \mathbf{x}, \quad \mathbf{p}_L = -\frac{1}{2N}\partial_{\mathbf{z}}\mathcal{E}(\mathbf{z}_L, \mathbf{y}). \tag{2.67}$$

Furthermore, the above coincides with the discretization of the optimality system in Theorem 2.1.5 through the partitioned Runge-Kutta method defined by the pair of coefficients $(\{a_{ij}, b_i\}_{i,j=1}^s, \{\hat{a}_{ij}, \hat{b}_i\}_{i,j=1}^s)$. *Finally, the latter is a symplectic integrator for the Hamiltonian system 2.41 and* $\|\mathbf{p}_l\|^2_{\mathbf{K}[\mathbf{z}_l]}$ *fluctuates like* $\mathcal{O}(h^k)$ *with respect to* l, *where* $h := \max_{i=1}^L h_i$ *and* k *denotes the order of the partitioned Runge-Kutta method.*

Proof. Introduce Lagrange multipliers $\mathbf{p}_l, \boldsymbol{\alpha}_{li} \in \mathbb{R}^{dN}$, $i = 1, \dots, s$, $l = 1, \dots, L$, and define

$$S_{li} := \mathbf{K}[\mathbf{z}_{li}]\boldsymbol{\beta}_{li} \tag{2.68}$$

so that

$$\mathbf{z}_{l+1} = \mathbf{z}_l + h_l \sum_{i=1}^s b_i S_{li}, \quad S_{li} = \mathbf{K}[\mathbf{z}_l + h_l \sum_{j=1}^L a_{ij} S_{lj}]\boldsymbol{\beta}_{li}. \tag{2.69}$$

Then, using variables S_{li} instead of $\mathbf{z}_{li}$ for convenience, the Lagrangian associated to the minimization problem reads

$$\begin{aligned} \mathcal{L}(\mathbf{z}_l, S_{li}, \boldsymbol{\beta}_{li}, \mathbf{p}_l, \boldsymbol{\alpha}_{li}) =& \frac{1}{2N}\mathcal{E}(\mathbf{z}_L, \mathbf{y}) + \frac{\gamma}{2}\sum_{l=0}^{L-1} h_l \sum_{i=1}^s b_i \boldsymbol{\beta}_{li}^\top \mathbf{K}[\mathbf{z}_{li}]\boldsymbol{\beta}_{li} \\ &+ \sum_{l=0}^{L-1} \mathbf{p}_{l+1}^\top \left(\mathbf{z}_{l+1} - \mathbf{z}_l - h_l \sum_{i=1}^s b_i S_{li}\right) + \mathbf{p}_0^\top(\mathbf{z}_0 - \mathbf{x}) \\ &+ \sum_{l=0}^{L-1}\sum_{i=1}^s \boldsymbol{\alpha}_{li}^\top \left(S_{li} - \mathbf{K}[\mathbf{z}_{li}]\boldsymbol{\beta}_{li}\right). \end{aligned} \tag{2.70}$$

Setting derivatives with respect to $\boldsymbol{\beta}_{li}$ to zero yields

$$\boldsymbol{\alpha}_{li} = \gamma h_l b_i \boldsymbol{\beta}_{li} \quad \Longrightarrow \quad \boldsymbol{\beta}_{li} = \frac{1}{\gamma}\mathbf{p}_{li} \tag{2.71}$$

if we define $\mathbf{p}_{li} := \boldsymbol{\alpha}_{li}/(h_l b_i)$, effectively exploiting the hypothesis that $b_i \neq 0$. Then, differentiation with respect to variables $\mathbf{z}_l$ gives

$$\begin{aligned} &\frac{\gamma h_l}{2}\sum_{j=1}^s b_j \partial_{\mathbf{z}}\left(\boldsymbol{\beta}_{lj}^\top \mathbf{K}[\mathbf{z}_{lj}]\boldsymbol{\beta}_{lj}\right) - \mathbf{p}_{l+1} + \mathbf{p}_l - \sum_{j=1}^s \partial_{\mathbf{z}}\left(\boldsymbol{\alpha}_{lj}^\top \mathbf{K}[\mathbf{z}_{lj}]\boldsymbol{\beta}_{lj}\right) = 0 \\ \Longleftrightarrow\ &\mathbf{p}_{l+1} = \mathbf{p}_l - \frac{h_l}{2\gamma}\sum_{j=1}^s \hat{b}_j \partial_{\mathbf{z}}\left(\mathbf{p}_{lj}^\top \mathbf{K}[\mathbf{z}_{lj}]\mathbf{p}_{lj}\right) \end{aligned} \tag{2.72}$$

for $l = 0, \dots, L-1$ and, doing the same for variables S_{li},

$$\begin{aligned} &\frac{\gamma h_l^2}{2} \sum_{j=1}^{s} b_j a_{ji} \partial_{\mathbf{z}} \left(\boldsymbol{\beta}_{lj}^\top \mathbf{K}[\mathbf{z}_{lj}] \boldsymbol{\beta}_{lj} \right) - h_l b_i p_{l+1} + \boldsymbol{\alpha}_{li} - h_l \sum_{j=1}^{s} a_{ji} \partial_{\mathbf{z}} \left(\boldsymbol{\alpha}_{lj}^\top \mathbf{K}[\mathbf{z}_{lj}] \boldsymbol{\beta}_{lj} \right) = 0 \\ \iff & p_{li} = p_l - \frac{h_l}{2\gamma} \sum_{j=1}^{s} \hat{b}_j \partial_{\mathbf{z}} \left(\mathbf{p}_{lj}^\top \mathbf{K}[\mathbf{z}_{lj}] \mathbf{p}_{lj} \right), \end{aligned} \tag{2.73}$$

for $l = 0, \dots, L-1$, $i = 1, \dots, s$. The rest of the conditions are easily derived taking the remaining derivatives. It is also easy to verify that (2.66) can be equivalently derived by direct discretization of the optimality system in Theorem 2.1.5 through the partitioned Runge-Kutta method defined by the pair of coefficients $(\{a_{ij}, b_i\}_{i,j=1}^s, \{\hat{a}_{ij}, \hat{b}_i\}_{i,j=1}^s)$. Furthermore, the relation in (2.65) immediately implies that the partitioned Runge-Kutta method satisfies Sun's condition and, hence, is symplectic. Finally, if k is the order of the method, owing to its symplecticity we have that the Hamiltonian is conserved with an error $\mathcal{O}(h^k)$ with respect to l. □

Remark. Though we discretized the learning problem with respect to time starting from the more convenient parametric formulation (obtained originally in Theorem 2.1.1 as a consequence of the space-discretization), we remark that an equivalent course of action would have been the time-discretization of the nonparametric Problem 2 directly through a Runge-Kutta scheme characterized by coefficients $\{a_{ij}, b_i\}_{i,j=1}^s$ and collocation points $\{c_i\}_{i=1}^s$, namely

$$\begin{aligned} \min_{v_{li} \in \mathcal{H}} \quad & \frac{1}{2N} \mathcal{E}(\mathbf{z}_L, \mathbf{y}) + \frac{\gamma}{2} \sum_{l=0}^{L-1} h_l \sum_{i=1}^{s} b_i \left\| v_{li} \right\|_{\mathcal{H}}^2 \\ \text{subject to} \quad & z_{l+1}^{(k)} = z_l^{(k)} + h_l \sum_{i=1}^{s} b_i v_{li}(z_{li}^{(k)}), \quad l = 0, \dots, L-1, \ k = 1, \dots, N, \\ & z_{li}^{(k)} = z_l^{(k)} + h_l \sum_{j=1}^{s} a_{ij} v_{lj}(z_{lj}^{(k)}), \quad i = 1, \dots, s, \ l = 0, \dots, L-1, \ k = 1, \dots, N, \\ & z_0^{(k)} = x^{(i)}, \quad k = 1, \dots, N. \end{aligned} \tag{2.74}$$

This formulation, indeed, may be easily recast as Problem 4 trough the usual argument exploiting the orthogonal decomposition of the functions v_{li}.

Though it is of interest to state the formulation of the discrete problem for a generic Runge-Kutta method as in Problem 4, in what follows the main discretization scheme employed is the explicit Euler method due to its simplicity and efficiency. Therefore, we wish to study the properties of Problem 4 with $a_{11} = 0$, $b_1 = 1$ and $c_1 = 0$ in further detail. In particular, we are concerned with the existence of solutions and how they are related to those of the continuous-in-time Problem 3. In this context, the natural technique to produce such results is provided by Γ-convergence (Dal Maso [1993]). The fist step we take is to ambient both the discrete functionals and the continuous one on the same function space $\mathcal{B} := L^2([0,1], \mathbb{R}^{dN})$. Hence, we prove the following lemma.

Lemma 2.1.9. *Under Assumption 1 and with the notation introduced in Problem 4, define the discrete functional* $\mathcal{J}^L : \mathcal{B} \to \mathbb{R}$ *as*

$$\mathcal{J}^L(\boldsymbol{\beta}) := \frac{1}{2N}\mathcal{E}(\mathbf{z}_1^L, \mathbf{y}) + \frac{\gamma}{2}\int_0^1 \boldsymbol{\beta}_t^\top \mathbf{K}[\mathbf{z}_t^L]\boldsymbol{\beta}_t dt, \tag{2.75}$$

where the discretized state $\mathbf{z}_t^L$ *follows the dynamics*

$$\begin{aligned} \mathbf{z}_0^L &= \mathbf{x}, \\ \mathbf{z}_{t_{l+1}}^L &= \mathbf{z}_{t_l}^L + h_l \int_{t_l}^{t_{l+1}} \mathbf{K}[\mathbf{z}_{t_l}^L]\boldsymbol{\beta}_t dt, \quad l = 0, \dots, L-1, \\ \mathbf{z}_t^L &= \mathbf{z}_{t_l}^L, \quad t \in (t_{l-1}, t_l], \quad l = 1, \dots, L. \end{aligned} \tag{2.76}$$

Then,

$$\min_{\boldsymbol{\beta}\in\mathcal{B}} \mathcal{J}^L(\boldsymbol{\beta}) \tag{2.77}$$

admits a solution and is equivalent to Problem 4 with the explicit Euler method.

Proof. First, define the set $PC_L \subset \mathcal{B}$ of piecewise-constant functions on the grid $\{t_l\}_{l=0}^L$, which is finite dimensional. Then, consider the decomposition

$$\boldsymbol{\beta} = \boldsymbol{\alpha} + \boldsymbol{\rho}, \qquad \boldsymbol{\alpha} \in PC_L, \quad \boldsymbol{\rho} \in PC_L^\perp, \tag{2.78}$$

and notice that

$$\begin{aligned} &\int_0^1 \boldsymbol{\beta}_t^\top \mathbf{K}[\mathbf{z}_t^L]\boldsymbol{\beta}_t dt \\ &= \int_0^1 \boldsymbol{\alpha}_t^\top \mathbf{K}[\mathbf{z}_t^L]\boldsymbol{\alpha}_t dt + 2\left\langle \boldsymbol{\rho}, \mathbf{K}[\mathbf{z}^L]\boldsymbol{\alpha} \right\rangle_{\mathcal{B}} + \int_0^1 \boldsymbol{\rho}_t^\top \mathbf{K}[\mathbf{z}_t^L]\boldsymbol{\rho}_t dt \\ &= \int_0^1 \boldsymbol{\alpha}_t^\top \mathbf{K}[\mathbf{z}_t^L]\boldsymbol{\alpha}_t dt + \int_0^1 \boldsymbol{\rho}_t^\top \mathbf{K}[\mathbf{z}_t^L]\boldsymbol{\rho}_t dt \\ &\geq \int_0^1 \boldsymbol{\alpha}_t^\top \mathbf{K}[\mathbf{z}_t^L]\boldsymbol{\alpha}_t dt, \end{aligned} \tag{2.79}$$

where in the second equality we used the fact that $\boldsymbol{\rho} \in PC_L^\perp$ and $\mathbf{K}[\mathbf{z}^L]\boldsymbol{\alpha} \in PC_L$ since $\mathbf{z}^L \in PC_L$. Moreover, the fact that $\boldsymbol{\rho} \in PC_L^\perp$ also implies

$$\int_{t_l}^{t_{l+1}} \mathbf{K}[\mathbf{z}_{t_l}^L]\boldsymbol{\beta}_t dt = \int_{t_l}^{t_{l+1}} \mathbf{K}[\mathbf{z}_{t_l}^L]\boldsymbol{\alpha}_t dt, \tag{2.80}$$

for all $l = 0, \dots, L-1$, that is $\boldsymbol{\rho}$ does not contribute to the evolution of the state. Then, owing to the inequality in (2.79), we immediately have that any minimizer $\boldsymbol{\beta}$ of $\mathcal{J}^L$ must be an element of PC_L. This observation entails that we can equivalently ambient the problem on the finite dimensional space PC_L of piecewise constant functions which, upon defining $\boldsymbol{\beta}_l := 1/h_l \int_{t_l}^{t_{l+1}} \boldsymbol{\beta}_t dt \in \mathbb{R}^{dN}$, $l = 0, \dots, L-1$, is enough to prove the equivalence with Problem 4 with the explicit Euler method, namely

$$\begin{aligned} \min_{\boldsymbol{\beta}_l \in \mathbb{R}^{dN}} \quad & \frac{1}{2N}\mathcal{E}(\mathbf{z}_L, \mathbf{y}) + \frac{\gamma}{2}\sum_{l=0}^{L-1} h_l \boldsymbol{\beta}_l^\top \mathbf{K}[\mathbf{z}_l^L]\boldsymbol{\beta}_l \\ \text{subject to} \quad & \mathbf{z}_{l+1} = \mathbf{z}_l^L + h_l \mathbf{K}[\mathbf{z}_l]\boldsymbol{\beta}_l, \quad l = 0, \dots, L-1, \\ & \mathbf{z}_0 = \mathbf{x}. \end{aligned} \tag{2.81}$$

Denote $\hat{\mathcal{J}}^L(\boldsymbol{\beta}_0, \dots, \boldsymbol{\beta}_{L-1})$ the functional in the above minimization problem defined on $\mathbb{R}^{dN} \times \cdots \times \mathbb{R}^{dN}$. It follows from the continuity of $\mathbf{K}[\cdot]$ that $\hat{\mathcal{J}}^L$ is continuous in $\boldsymbol{\beta}_l$, $l = 0, \dots, L-1$. As it further holds

$$\hat{\mathcal{J}}^L(\boldsymbol{\beta}) \geq \frac{\delta\gamma}{2} \sum_{l=0}^{L-1} h_l \left\|\boldsymbol{\beta}_l\right\|_2^2, \tag{2.82}$$

the constant δ being the one defined in Assumption 1, then $\hat{\mathcal{J}}^L \to \infty$ as $\left\|\boldsymbol{\beta}_l\right\|_2 \to \infty$ for any $l = 1, \dots, L-1$. This is enough to prove the existence of a minimizer of (2.81) and, in turn, of $\mathcal{J}^L$. □

The tools of Γ-convergence require the domain where the functionals are defined to be equipped with a metrizable topology. Recalling that the weak topology of L^2 is metrizable only on bounded sets, we need to properly restrict the functionals. For every $r > 0$ we set

$$\mathcal{B}_r := \{\boldsymbol{\beta} \in \mathcal{B} \mid \left\|\boldsymbol{\beta}\right\|_{L^2} \leq r\}, \tag{2.83}$$

and define the corresponding restricted functionals $\mathcal{J}_r^L := \mathcal{J}^L|_{\mathcal{B}_r}$, $L = 1, 2, \dots, \infty$, with the convention that $\mathcal{J}^\infty := \mathcal{J}$ represents the continuous functional defined as in Theorem 2.1.4. Then, take $\boldsymbol{\beta} \equiv 0$ and note that

$$\mathcal{J}^L(0) = \frac{1}{2N}\mathcal{E}(\mathbf{x}, \mathbf{y}) =: C, \tag{2.84}$$

the constant $C > 0$ being independent from $L = 1, 2, \dots, \infty$. Denote $\boldsymbol{\beta}^{L_\star} \in \mathcal{B}$, $L = 1, 2, \dots, \infty$, a minimizer of $\mathcal{J}^L$, then it holds

$$\frac{\delta\gamma}{2} \left\|\boldsymbol{\beta}^{L_\star}\right\|_{L^2}^2 \leq \mathcal{J}^L(\boldsymbol{\beta}^{L_\star}) \leq \mathcal{J}^L(0) = C, \tag{2.85}$$

where we used both Assumption 1 and the definition of $\boldsymbol{\beta}^{L_\star}$. This entails that taking $r = \sqrt{\frac{2C}{\delta\gamma}}$ is enough to state

$$\arg\min_{\boldsymbol{\beta} \in \mathcal{B}_r} \mathcal{J}_r^L = \arg\min_{\boldsymbol{\beta} \in \mathcal{B}} \mathcal{J}^L, \tag{2.86}$$

for all $L = 1, 2, \dots, \infty$. With this choice we restrict the minimization problem to a bounded subset of $\mathcal{B}$, without losing any minimizer. Let us now recall the definition of Γ-convergence.

Definition 2.1.1. The family of functionals $(\mathcal{J}_r^L)_{L \geq 1}$ is said to Γ-converge to a functional $\mathcal{J}_r : \mathcal{B}_r \to \mathbb{R} \cup \{+\infty\}$ with respect to the weak topology of $\mathcal{B}$ as $L \to \infty$ if the following conditions hold:

- for all $(\boldsymbol{\beta}^L)_{L \geq 1} \subset \mathcal{B}_r$ such that $\boldsymbol{\beta}^L \rightharpoonup \boldsymbol{\beta} \in \mathcal{B}_r$ as $L \to \infty$ it holds

$$\liminf_{L \to \infty} \mathcal{J}_r^L(\boldsymbol{\beta}^L) \geq \mathcal{J}_r(\boldsymbol{\beta}); \tag{2.87}$$

- for all $\boldsymbol{\beta} \in \mathcal{B}_r$ there exists a sequence $(\boldsymbol{\beta}^L)_{L\geq 1} \subset \mathcal{B}_r$ such that $\boldsymbol{\beta}^L \rightharpoonup \boldsymbol{\beta} \in \mathcal{B}_r$ as $L \to \infty$ and

$$\limsup_{L\to\infty} \mathcal{J}_r^L(\boldsymbol{\beta}^L) \leq \mathcal{J}_r(\boldsymbol{\beta}). \tag{2.88}$$

If (2.87) and (2.88) are satisfied, then we write $\mathcal{J}_r^L \xrightarrow{\Gamma} \mathcal{J}_r$ as $L \to \infty$.

Lemma 2.1.10. *Under Assumption 1, let $\boldsymbol{\beta} \in \mathcal{B}$ and $\mathbf{z}$, $\mathbf{z}^L$ denote the solutions of the continuous time dynamics (2.20) and the discrete time dynamics (2.76) respectively. Then,*

$$\left\|\mathbf{z}^L - \mathbf{z}\right\|_\infty \leq Ch, \tag{2.89}$$

where $C = C(\|\boldsymbol{\beta}\|_{L^2})$ denotes a constant monotonically increasing in $\|\boldsymbol{\beta}\|_{L^2}$ and $h := \max_l h_l$.

Proof. The structure of this proof follows closely the classical arguments to prove estimates on the global error of Runge-Kutta methods (see, e.g., Hairer et al. [1993]). The fist step is to prove local convergence, i.e. convergence of the discretized solution to the true solution on any time interval $(t_l, t_{l+1}]$, $l = 0, \dots, L-1$, when the initial condition at time t_l is the same. For notational simplicity we concentrate on the first time interval $(0, t_1]$, bearing in mind the following statements hold for any other time interval. Define the local error $e_1 := \left\|\mathbf{z}_{t_1}^L - \mathbf{z}_{t_1}\right\|_2$ and note that

$$e_1 = \left\| \int_0^{h_0} \left(\mathbf{K}[\mathbf{x}] - \mathbf{K}[\mathbf{z}_s]\right) \boldsymbol{\beta}_s ds \right\|_2 \leq L_{\mathbf{K}} \|\boldsymbol{\beta}\|_{L^2(0,h_0)} \left(\int_0^{h_0} \|\mathbf{z}_s - \mathbf{x}\|_2^2 \right)^{1/2}. \tag{2.90}$$

Furthermore

$$\|\mathbf{z}_s - \mathbf{x}\|_2 = \left\| \int_0^s \mathbf{K}[\mathbf{z}_r] \boldsymbol{\beta}_r dr \right\|_2 \leq \sqrt{\Delta} \int_0^{h_0} \|\boldsymbol{\beta}_r\|_2 \, dr \leq \sqrt{\Delta} h_0^{1/2} \|\boldsymbol{\beta}\|_{L^2(0,h_0)}, \tag{2.91}$$

which implies

$$e_1 \leq L_{\mathbf{K}} \Delta \|\boldsymbol{\beta}\|_{L^2(0,h_0)}^2 h. \tag{2.92}$$

Clearly, we can extend the definition of the local error e_l to any other time interval $(t_{l-1}, t_l]$, $l = 1, \dots, L$, and the same bound will hold with h_{l-1} in place of h_0. Fix now $\bar{L} \leq L-1$ and define the global error in the time interval $(t_{\bar{L}}, t_{\bar{L}+1}]$ as

$$E := \sup_{t\in(t_{\bar{L}}, t_{\bar{L}+1}]} \left\|\mathbf{z}_t^L - \mathbf{z}_t\right\|_2. \tag{2.93}$$

Define further

$$E_l := \sup_{t\in(t_{\bar{L}}, t_{\bar{L}+1}]} \left\|\mathbf{z}_t(\mathbf{z}_{t_l}^L, t_l) - \mathbf{z}_t(\mathbf{z}_{t_{l-1}}^L, t_{l-1})\right\|_2, \quad l = 1, \dots, \bar{L}, \tag{2.94}$$

where $\mathbf{z}_t(\mathbf{x}, s)$ denotes the true ODE solution with initial condition $\mathbf{x}$ at time s. Let $t \in (t_{\bar{L}}, t_{\bar{L}+1}]$, then

$$\begin{aligned}
&\left\|\mathbf{z}_t(\mathbf{z}^L_{t_l}, t_l) - \mathbf{z}_t(\mathbf{z}^L_{t_{l-1}}, t_{l-1})\right\|_2 \\
\leq &\left\|\mathbf{z}^L_{t_l} - \mathbf{z}_{t_l}(\mathbf{z}^L_{t_{l-1}}, t_{l-1})\right\|_2 + \int_{t_l}^{t} \left\|\left(\mathbf{K}[\mathbf{z}_s(\mathbf{z}^L_{t_l}, t_l)] - \mathbf{K}[\mathbf{z}_s(\mathbf{z}^L_{t_{l-1}}, t_{l-1})]\right)\boldsymbol{\beta}_s\right\|_2 ds \\
\leq &\left\|\mathbf{z}^L_{t_l} - \mathbf{z}_{t_l}(\mathbf{z}^L_{t_{l-1}}, t_{l-1})\right\|_2 + L_{\mathbf{K}} \int_{t_l}^{t} \left\|\mathbf{z}_s(\mathbf{z}^L_{t_l}, t_l) - \mathbf{z}_s(\mathbf{z}^L_{t_{l-1}}, t_{l-1})\right\|_2 \|\boldsymbol{\beta}_s\|_2 \, ds
\end{aligned} \tag{2.95}$$

and, owing to Gronwall's inequality,

$$\begin{aligned}
\left\|\mathbf{z}_t(\mathbf{z}^L_{t_l}, t_l) - \mathbf{z}_t(\mathbf{z}^L_{t_{l-1}}, t_{l-1})\right\|_2 &\leq (1 + L_{\mathbf{K}} \|\boldsymbol{\beta}\|_{L^2} e^{\|\boldsymbol{\beta}\|_{L^2}}) \left\|\mathbf{z}^L_{t_l} - \mathbf{z}_{t_l}(\mathbf{z}^L_{t_{l-1}}, t_{l-1})\right\|_2 \\
&\leq (1 + L_{\mathbf{K}} \|\boldsymbol{\beta}\|_{L^2} e^{\|\boldsymbol{\beta}\|_{L^2}}) e_l,
\end{aligned} \tag{2.96}$$

which, in turn, yields

$$E_l \leq (1 + L_{\mathbf{K}} \|\boldsymbol{\beta}\|_{L^2} e^{\|\boldsymbol{\beta}\|_{L^2}}) e_l, \quad l = 1, \ldots, \bar{L}. \tag{2.97}$$

Finally

$$\begin{aligned}
E \leq &\sum_{l=1}^{\bar{L}} E_l \\
\leq &\sum_{l=1}^{\bar{L}} (1 + L_{\mathbf{K}} \|\boldsymbol{\beta}\|_{L^2} e^{\|\boldsymbol{\beta}\|_{L^2}}) e_l \\
\leq &L_{\mathbf{K}} \Delta (1 + L_{\mathbf{K}} \|\boldsymbol{\beta}\|_{L^2} e^{\|\boldsymbol{\beta}\|_{L^2}}) h \sum_{l=1}^{\bar{L}} \|\boldsymbol{\beta}\|^2_{L^2(0,h_l)} \\
\leq &L_{\mathbf{K}} \Delta \|\boldsymbol{\beta}\|^2_{L^2} (1 + L_{\mathbf{K}} \|\boldsymbol{\beta}\|_{L^2} e^{\|\boldsymbol{\beta}\|_{L^2}}) h,
\end{aligned} \tag{2.98}$$

which proves the claim as the above holds uniformly for all $\bar{L}$. □

This result gives that, as expected, the explicit Euler method is globally convergent in [0,1]. In particular, it proves that the convergence to the true solution is uniform with order $\mathcal{O}(h)$ and a constant of convergence dependent only on the norm of the control $\boldsymbol{\beta}$. We shall now see that this is enough to prove the Γ-convergence of the discrete functionals to the continuous one.

Theorem 2.1.11. *Under Assumption 1, the family of functionals $(\mathcal{J}^L_r)_{L\geq 1}$ Γ-converges to $\mathcal{J}_r$ with respect to the weak topology of $\mathcal{B}$ as $L \to \infty$.*

Proof. Let us prove condition (2.88) first. Take $\boldsymbol{\beta} \in \mathcal{B}_r$ and consider the constant sequence $\boldsymbol{\beta}^L := \boldsymbol{\beta}$, $L \geq 1$. Then, owing to Lemma 2.1.10, the discretized solution $\mathbf{z}^L$ uniformly converges to the true ODE solution $\mathbf{z}$ and, by the Lipschitz continuity of $\mathbf{K}[\cdot]$, we have

$$\mathcal{J}^L(\boldsymbol{\beta}) \xrightarrow[L\to\infty]{} \mathcal{J}(\boldsymbol{\beta}) \tag{2.99}$$

as desired. In order to prove condition (2.87), let $\boldsymbol{\beta} \in \mathcal{B}_r$ and consider $(\boldsymbol{\beta}^L)_{L\geq 1} \subset \mathcal{B}_r$ such that $\boldsymbol{\beta}^L \rightharpoonup \boldsymbol{\beta} \in \mathcal{B}_r$ as $L \to \infty$. Call $\mathbf{z}^{\boldsymbol{\beta}^L}$ and $\mathbf{z}^{L,\boldsymbol{\beta}^L}$ the solutions to the continuous time dynamics (2.20) and discrete time dynamics (2.76) with control $\boldsymbol{\beta}^L$ respectively, $L \geq 1$, with the usual convention $\boldsymbol{\beta}^\infty := \boldsymbol{\beta}$. Then

$$\left\|\mathbf{z}^{L,\boldsymbol{\beta}^L} - \mathbf{z}^{\boldsymbol{\beta}^\infty}\right\|_\infty \leq \left\|\mathbf{z}^{L,\boldsymbol{\beta}^L} - \mathbf{z}^{\boldsymbol{\beta}^L}\right\|_\infty + \left\|\mathbf{z}^{\boldsymbol{\beta}^L} - \mathbf{z}^{\boldsymbol{\beta}^\infty}\right\|_\infty. \tag{2.100}$$

As $(\boldsymbol{\beta}^L)_{L\geq 1}$ is weakly convergent it is bounded by a constant, say, $K > 0$. This and Lemma 2.1.10 entail that the first term in the right-hand side of (2.100) can be uniformly bounded with respect to L, namely

$$\left\|\mathbf{z}^{L,\boldsymbol{\beta}^L} - \mathbf{z}^{\boldsymbol{\beta}^L}\right\|_\infty \leq Ch, \tag{2.101}$$

the constant C being independent of L. Furthermore, Lemma 2.1.3 implies that the second term in the right-hand side of (2.100) vanishes as $L \to \infty$, effectively proving uniform convergence of $\mathbf{z}^{L,\boldsymbol{\beta}^L}$ to $\mathbf{z}^{\boldsymbol{\beta}^\infty}$ as $L \to \infty$. This is enough to prove that

$$\liminf_{L\to\infty} \mathcal{J}_r^L(\boldsymbol{\beta}^L) \geq \mathcal{J}_r(\boldsymbol{\beta}) \tag{2.102}$$

arguing as in the proof of Theorem 2.1.4. □

Establishing Γ-convergence allows us to immediately characterize the behaviour of the minimizers of the discrete functionals in the limit $L \to \infty$.

Corollary 2.1.11.1. *With the notation and assumptions of Theorem 2.1.11, it holds that*

$$\lim_{L\to\infty} \min_{\mathcal{B}_r} \mathcal{J}^L = \min_{\mathcal{B}_r} \mathcal{J}, \tag{2.103}$$

and any cluster point $\boldsymbol{\beta}$ of a sequence of minimizers $(\boldsymbol{\beta}^L)_{L\geq 1}$ is a minimizer of $\mathcal{J}_r$. Furthermore, any sequence of minimizers $(\boldsymbol{\beta}^L)_{L\geq 1}$ is is pre-compact with respect to the strong topology of L^2.

Proof. The inequality

$$\mathcal{J}^L(\boldsymbol{\beta}) \geq \frac{\delta\gamma}{2} \|\boldsymbol{\beta}\|_{L^2} \tag{2.104}$$

entails that the sequence $(\mathcal{J}^L)_{L\geq 1}$ of functionals is equi-coercive. This, together with [Dal Maso, 1993, Corollary 7.20], implies (2.103) and that any cluster point $\boldsymbol{\beta}$ of a sequence of minimizers $(\boldsymbol{\beta}^L)_{L\geq 1}$ is a minimizer of $\mathcal{J}_r$. Assume the sequence of minimizers is weakly converging to $\boldsymbol{\beta}$ (which we stress must be a minimizer of $\mathcal{J}_r$), then, arguing as in Theorem 2.1.4, we have that

$$\frac{1}{2N}\mathcal{E}(\mathbf{z}_1^{L,\boldsymbol{\beta}^L}) + \int_0^1 \left(\boldsymbol{\beta}_t^L\right)^\top \mathbf{K}[\mathbf{z}_t^{L,\boldsymbol{\beta}^L}]\boldsymbol{\beta}_t^L dt - \int_0^1 \left(\boldsymbol{\beta}_t^L\right)^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}}]\boldsymbol{\beta}_t^L dt \xrightarrow[L\to\infty]{} \frac{1}{2N}\mathcal{E}(\mathbf{z}_1^{\boldsymbol{\beta}}). \tag{2.105}$$

This, together with (2.103), implies

$$\int_0^1 \left(\boldsymbol{\beta}_t^L\right)^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}}]\boldsymbol{\beta}_t^L dt \xrightarrow[L\to\infty]{} \int_0^1 (\boldsymbol{\beta}_t)^\top \mathbf{K}[\mathbf{z}_t^{\boldsymbol{\beta}}]\boldsymbol{\beta}_t dt. \tag{2.106}$$

Owing to the equivalence of the L^2 norm and the $\mathbf{K}[\mathbf{z}^{\boldsymbol{\beta}}]-$weighed L^2 norm observed in Theorem 2.1.4, this is enough to prove that the sequence $(\boldsymbol{\beta}^L)_{L\geq 1}$ converges in L^2 with respect to the strong topology as well, and the claim follows. □

2.2 A Riemannian Optimization perspective on the learning problem

2.2.1 Remarks on the infinite data points limit for the Optimal Control problem

Before we go any further, it is insightful to take a step back and consider again the limiting case where an infinite number of data pairs are available. Ultimately, we wish to show, though only formally, that the extension of the results obtained in Section 2.1.2 hold when Problem 1 is directly tackled. In this setting we directly deal with absolutely continuous probability measures (with respect to the Lebesgue measure) instead of particles in space. Hence, we first give a brief review of the learning Problem at heart of the thesis highlighting the connection with the field of Optimal Transport (OT) (Figalli and Glaudo [2021]) which is exactly concerned with optimization over the space of probability measures. Hopefully, this will make it clearer to the reader how the need for an RKHS arises in the context of learning and transporting measures in space according to an optimality criterion.

Given complete information on the target function g, the most general way we can (formally) state the problem we seek to solve is the following:

$$\inf_{f\in\mathcal{V}_m,\ \pi\in\Pi(\mu)} \int_{\mathcal{X}\times\mathbb{R}^d} \left(f(z)-g(x)\right)^2 d\pi(x,z), \tag{2.107}$$

where $\Pi(\mu) := \{\pi \mid \pi(\cdot,\mathbb{R}^d)=\mu\}$ denotes the set of joint probability measures on $\mathcal{X}\times\mathbb{R}^d$ with marginal μ on $\mathcal{X}$. As always, the measure μ is imposed by accuracy requirements on the reconstruction of the target g, while the marginal $\pi(\mathcal{X},\cdot)$ can be chosen arbitrarily. This formulation is reminiscent of Kantorovich's formulation of the OT problem[5] and, in some way, represents a non-parametric (with respect to the involved measures) version of the learning problem introduced at the beginning of Section 2.1. In OT literature, Kantorovich's formulation was introduced as an improvement (with respect to existence of minima) over Monge's formulation which, in our setting, is instead analogous to the problem

$$\inf_{f\in\mathcal{V}_m,\ \mu_1,\ \phi\in\Phi(\mu,\mu_1)} \int_{\mathcal{X}\times\mathbb{R}^d} \left(f(\phi(x))-g(x)\right)^2 d\mu(x) \tag{2.109}$$

with $\Phi(\mu,\mu_1) := \{\phi : \mathcal{X}\to\mathbb{R}^d \mid \phi_*\mu=\mu_1\}$ being the set of transport maps from μ to μ_1. Within either Kantorovich's or Monge's framework, one very common approach to the solution of the OT problem is through point-wise discretizations of space. Typically, a number of particles are dispatched in space and some numerical scheme is employed in order to make such particles approximate the sought-after joint distribution, or otherwise

[5] Kantorovich's and Monge's formulations of the OT problem read

$$\inf_{\pi\in\Pi(\mu,\mu_1)} \int_{\mathcal{X}\times\mathbb{R}^d} c(x,z)d\pi(x,z), \qquad \inf_{\phi\in\Phi(\mu,\mu_1)} \int_{\mathcal{X}\times\mathbb{R}^d} c(x,\phi(x))d\mu(x) \tag{2.108}$$

respectively, where $c:\mathcal{X}\times\mathbb{R}^d\to[0,\infty]$ denotes a cost function.

the optimal transport map. This clearly reminds of the space discretization introduced in Section 2.1 which led to the statement of Problem 2. However, once the space discretization is introduced, we actually have a stronger requirement in that we wish to extrapolate what was learned with the "training" particles to the whole ambient space, for in OT there is no need to "interpolate" between particles. In other words, we must in some capacity turn to a parametric model where not only particles are moved around in space, but their trajectories should inform about the movement of any other chosen particle. Our own way to do so, as described in Section 2.1, is to approximate the transport map in (2.109) as the flow of an ODE whose flux is taken in a particular function space, i.e. an RKHS. At first glance this RKHS may seem like an untameable object as it possibly is an infinite dimensional space. However, owing to the representer-like Theorem 2.1.1, after the space discretization introduced for Problem 2 (which is ultimately the only one we can hope to solve in practice) reduces to a finite dimensional parametric problem in space. Let us introduce some assumptions and notation. For simplicity let $\mathcal{X} = \mathbb{R}^d$ and denote $\mathcal{P}(\mathbb{R}^d)$ the space of Borel probability measures on $\mathbb{R}^d$. Abusing notation, we will use the same letter for their Lebesgue densities in case they exist. Furthermore, in this section we consider an RKHS associated to a kernel with the following properties.

Assumption 2. Assume $\mathcal{K}$ is a C^∞ diagonal kernel on $\mathbb{R}^d$, i.e. $\mathcal{K}(x, x') = Ik(x, x')$, $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ being a scalar-valued kernel function, satisfying Assumption 1. Assume further that k is integrally strictly positive definite, that is

$$\int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \mathcal{K}(x, x')d\nu(x)d\nu(x') > 0 \tag{2.110}$$

for all signed Borel nonzero measures ν on $\mathbb{R}^d$, and that $\sup_{x\in\mathbb{R}^d} k(x, x) < \infty$.

Remark. Note that the regularity assumptions on the kernel imply $\mathcal{H} \subseteq C^\infty(\mathbb{R}^d)$ and are motivated by the wish to tackle the problem in its simplest form, however some of what follows holds under milder regularity conditions as well. Furthermore, throughout this and the following sections we will refer to the RKHS of functions from $\mathbb{R}^d$ to $\mathbb{R}$ associated to k as $\mathcal{G}$. Note that with this notation $\mathcal{H} = \mathcal{G}^d$.

Given a kernel $\mathcal{K}$ satisfying Assumption 2 let us define the following subset of $\mathcal{P}(\mathbb{R}^d)$,

$$\mathcal{P}_\mathcal{K}(\mathbb{R}^d) = \left\{\mu \in \mathcal{P}(\mathbb{R}^d) \;\middle|\; \frac{d\mu}{d\lambda} \text{ is } C^\infty,\ \operatorname{supp}(\mu) = \mathbb{R}^d\right\}. \tag{2.111}$$

where λ denotes the Lebesgue measure on $\mathbb{R}^d$. We remark that for any fixed ODE flow $v : [0,1] \to \mathcal{H}$ and measure $\mu \in \mathcal{P}_\mathcal{K}(\mathbb{R}^d)$ the evolution of the density of the push-forward measure μ_t through the ODE flow associated to v at time t is governed by the well-known continuity equation which, formally, reads

$$\partial_t\mu_t + \operatorname{div}(\mu_t v_t) = 0. \tag{2.112}$$

A number of works on variational methods for sampling (see, e.g., Duncan et al. [2019] and references therein) have recently taken the above as the starting point to construct a mathematically sound kernel approach to the transport of measures, as opposed to

the usual one ambiented in Wasserstein spaces[6]. Indeed, they define the so-called Stein distance on $\mathcal{P}_{\mathcal{K}}(\mathbb{R}^d)$ as

$$d_{\mathcal{K}}^2(\rho,\nu) := \inf_{v\in\Gamma(\rho,\nu)} \int_0^1 \|v_t\|_{\mathcal{H}}^2 \, dt \tag{2.114}$$

where

$$\begin{aligned}\Gamma(\rho,\nu) := \{v : [0,1] \to \mathcal{H} \mid \mu_0 = \rho, \mu_1 = \nu,\\ \text{(2.112) holds in the sense of distributions}\}\,.\end{aligned} \tag{2.115}$$

Then, they show that the Stein distance defines an extended metric[7] on $\mathcal{P}_{\mathcal{K}}(\mathbb{R}^d)$ and that there exists a constant $C > 0$ such that

$$\mathcal{W}_2(\rho,\nu) \le C d_{\mathcal{K}}(\rho,\nu), \tag{2.116}$$

namely $d_{\mathcal{K}}$ induces a stronger topology on $\mathcal{P}_{\mathcal{K}}(\mathbb{R}^d)$ than the topology induced by the Wasserstein distance. We refer the reader to Duncan et al. [2019] and references therein for further details. The connection with our methodology as presented in Problem 1 is then clear. In (2.114) the concern is to join two given measures through a flow in the RKHS $\mathcal{H}$, minimizing the length of the resulting curve in the space of measures $\mathcal{P}_{\mathcal{K}}(\mathbb{R}^d)$. Our goal is very similar but, instead of having a pre-specified target measure to connect the original measure with, we choose the target measure through a performance criterion specified by the misfit with the target function and have a regularization parameter γ tuning the trade-off between the minimization of the curve length and the misfit.

We now want to show through a formal argument that in this infinite-dimensional setting the natural extension to the infinite data points limit $N \to \infty$ (i.e. formally replacing sums with integrals) of the discrete optimality conditions shown in Theorem 2.1.5 holds. Recall from Section 1.2.2 that, under Assumption 2, for all $\mu \in \mathcal{P}_{\mathcal{H}}(\mathbb{R}^d)$ the operator $T_\mu : L^2_\mu(\mathbb{R}^d) \to \mathcal{H}$ defined by

$$T_\mu w = \int_{\mathbb{R}^d} \mathcal{K}(x,x')w(x')d\mu(x') \tag{2.117}$$

is compact, self-adjoint and positive semi-definite. Furthermore, $T_\mu^{1/2}$ defines an isometry between $L^2_\mu(\mathbb{R}^d)$ and $\mathcal{H}$, so that the RKHS inner product features the characterization

$$\langle v, w\rangle_{\mathcal{H}} = \left\langle v, T_\mu^{-1} w\right\rangle_{L^2_\mu} \tag{2.118}$$

for all $v, w \in \mathcal{H}$.

---

[6] In $\mathbb{R}^d$ and for $p \ge 1$, the p-Wasserstein space is defined as the set of Borel probaility measures $\mathcal{P}(\mathbb{R}^d)$ endowed with the p-Wasserstein distance

$$\mathcal{W}_p(\rho,\nu) = \left(\inf_{\pi\in\Pi(\rho,\nu)} \int_{\mathbb{R}^d} (z-x)^p d\pi(x,z)\right)^{1/p}, \quad \rho,\nu \in \mathcal{P}_{\mathcal{K}}(\mathbb{R}^d). \tag{2.113}$$

[7] An extended metric satisfies the usual metric axioms but it allows for infinite distance between elements.

Lemma 2.2.1. *Under Assumption 2 the RKHS $\mathcal{H}$ consists of L^2_μ-functions for any $\mu \in \mathcal{P}_\mathcal{H}(\mathbb{R}^d)$. Furthermore, let $\mu \in \mathcal{P}_\mathcal{H}(\mathbb{R}^d)$ be a given reference measure, $v \in L^2([0,1];\mathcal{H})$ a flow in the RKHS and denote ϕ and μ_ϕ respectively the map obtained by integrating the usual ODE $\dot{z} = v_t(z)$ up to time $t = 1$ and the corresponding pushforward measure $\phi_\# \mu$. Then, $\mu_\phi \in \mathcal{P}_\mathcal{H}(\mathbb{R}^d)$.*

Proof. The proof of the first statement, which we recall for completeness, is canonical in RKHS literature and can be found in, e.g., Steinwart and Christmann [2008]. Let $\mu \in \mathcal{P}_\mathcal{H}(\mathbb{R}^d)$ and $w \in \mathcal{H}$, then

$$\begin{aligned} \|w_i\|^2_{L^2_\mu} &= \int_{\mathbb{R}^d} |w_i(x)|^2 \, d\mu(x) \\ &= \int_{\mathbb{R}^d} |\langle w_i, k(\cdot, x)\rangle_\mathcal{H}|^2 \, d\mu(x) \\ &\leq \int_{\mathbb{R}^d} \|w_i\|^2_\mathcal{H} \, \|k(\cdot, x)\|^2_\mathcal{H} \, d\mu(x) \\ &= \|w_i\|^2_\mathcal{H} \int_{\mathbb{R}^d} k(x,x) d\mu(x) < \infty. \end{aligned} \tag{2.119}$$

Consider now the ODE $\dot{z} = v_t(z)$, then, owing to the regularity of the RKHS we have that ϕ is a C^∞ diffeomorphism (Dupuis et al. [1998]). This concludes the proof. □

Theorem 2.2.2 (Formal)**.** *Under Assumption 2, assume Problem 1, which we recall is*

$$\begin{aligned} \min_{f \in \mathcal{V}_m,\, v:[0,1]\to\mathcal{H}} \quad & \frac{1}{2}\int_\mathcal{X} (f(z_1(x)) - g(x))^2 \, d\mu(x) + \frac{\eta}{2}\|f\|^2_{\mathcal{V}_m} + \frac{\gamma}{2}\int_0^1 \|v_t\|^2_\mathcal{H} \, dt \\ \textit{subject to} \quad & \dot{z}_t(x) = v_t(z_t(x)), \\ & z_0(x) = x, \end{aligned} \tag{2.120}$$

admits a solution $v \in C([0,1];\mathcal{H})$. Then, there exist functions $z \in C^1([0,1] \times \mathbb{R}^d)$ with $\dot{z}_t \in L^2_\mu(\mathbb{R}^d)$ for all $t \in [0,1]$ and $p \in C^1([0,1]; L^2_\mu(\mathbb{R}^d))$ such that

$$\begin{cases} \dot{z}_t(x) = \dfrac{1}{\gamma}\displaystyle\int_{\mathbb{R}^d} \mathcal{K}(z_t(x), z_t(x'))p_t(x')d\mu(x'), \quad z_0(x) = x, \\ \dot{p}_t(x) = -\dfrac{1}{2\gamma}\displaystyle\int_{\mathbb{R}^d} \partial_z \left(p_t(x)^\top \mathcal{K}(z_t(x), z_t(x'))p_t(x')\right) d\mu(x'), \quad p_1 \circ z_1^{-1} = -\dfrac{\delta}{\delta z_1}\mathcal{E}(z_1, g) \\ v_t = \dfrac{1}{\gamma} T_{\mu_t}(p_t \circ z_t^{-1}) \end{cases} \tag{2.121}$$

where we defined

$$\mathcal{E}(z_1, g) := \min_{f \in \mathcal{V}_m} \left\{ \frac{1}{2}\int_{\mathbb{R}^d} (f(z_1(x)) - g(x))^2 \, d\mu(x) + \frac{\eta}{2}\|f\|^2_{\mathcal{V}_m} \right\} \tag{2.122}$$

Proof. First, owing to Lemma 2.2.1, taking $v \in C([0,1];\mathcal{H})$ implies that $\mu_t \in \mathcal{P}_\mathcal{K}(\mathbb{R}^d)$ for all $t \in [0,1]$. Introduce the adjoint state $p \in C^1([0,1]; L^2_\mu(\mathbb{R}^d))$ and consider the Lagrangian associated to the minimization problem

$$\mathcal{L}(z, p, v) := \mathcal{E}(z_1, g) + \frac{\gamma}{2}\int_0^1 \|v_t\|^2_\mathcal{H} \, dt + \int_0^1 \langle p_t, \dot{z}_t - v_t \circ z_t\rangle_{L^2_\mu} \, dt. \tag{2.123}$$

Note that the above is well defined as $v_t \circ z_t \in L^2_\mu$ if and only if $v_t \in L^2_{\mu_t}$ which is true by Lemma 2.2.1. Let us now compute the first variation of $\mathcal{L}$ in v: let $\bar{v} \in C([0,1]; \mathcal{H})$, then

$$\begin{aligned} \frac{d}{d\varepsilon}\mathcal{L}(z,p,v+\varepsilon\bar{v})\Big|_{\varepsilon=0} &= \gamma \int_0^1 \langle \bar{v}_t, v_t \rangle_{\mathcal{H}}\, dt - \int_0^1 \left\langle \bar{v}_t, p_t \circ z_t^{-1} \right\rangle_{L^2_{\mu_t}} dt \\ &= \gamma \int_0^1 \langle \bar{v}_t, v_t \rangle_{\mathcal{H}}\, dt - \int_0^1 \left\langle \bar{v}_t, T_{\mu_t}(p_t \circ z_t^{-1}) \right\rangle_{\mathcal{H}} dt \end{aligned} \tag{2.124}$$

where in the last line we used the characterization of the RKHS inner product through the one in $L^2_{\mu_t}(\mathbb{R}^d)$. Setting the above to be equal to zero then implies

$$\begin{aligned} v_t &= \frac{1}{\gamma} T_{\mu_t}(p_t \circ z_t^{-1}) \\ &= \frac{1}{\gamma} \int_{\mathbb{R}^d} \mathcal{K}(\cdot, \tilde{z}) p_t(z_t^{-1}(\tilde{z})) d\mu_t(\tilde{z}) \\ &= \frac{1}{\gamma} \int_{\mathbb{R}^d} \mathcal{K}(\cdot, z_t(x)) p_t(x) d\mu(x). \end{aligned} \tag{2.125}$$

and the third equation in (2.121) is proved. Plugging this result into the Lagrangian yields a reduced Lagrangian

$$\begin{aligned} \tilde{\mathcal{L}}(z,p) := \mathcal{E}(z_1, g) &+ \int_0^1 \left(\int_{\mathbb{R}^d} p_t(x)^\top \dot{z}_t(x) d\mu(x) \right) dt \\ &- \frac{1}{2\gamma} \int_0^1 \left(\int_{\mathbb{R}^d \times \mathbb{R}^d} p_t(x)^\top \mathcal{K}(z_t(x), z_t(x')) p_t(x') d\mu(x) d\mu(x') \right) dt \end{aligned} \tag{2.126}$$

Computing variations with respect to z and p gives the first two equations in (2.121) and the claim is proved.

□

Write $\tilde{p}_t := p_t \circ z_t^{-1}$. Then, the continuity equation describing the evolution of the optimal measure μ_t as defined by the optimality system (2.121) reads

$$\partial_t \mu_t + \operatorname{div}(\mu_t T_{\mu_t} \tilde{p}_t) = 0. \tag{2.127}$$

Owing to the results obtained by Duncan et al. [2019], the above describes the motion in measure space $\mathcal{P}_{\mathcal{K}}(\mathbb{R}^d)$ along the geodesics[8] defined by the Stein distance. The role of the field $T_{\mu_t}\tilde{p}_t$, then, is to guide the flow towards a favourable target measure according to the misfit with the target function g.

2.2.2 The manifold of diffeomorphisms

The remarks of the previous section highlight the connection between the learning problem we have introduced and studied in this chapter so far and some of the works in the field of

[8] Duncan et al. [2019] show that the space $\mathcal{P}_{\mathcal{K}}(\mathbb{R}^d)$ can be formally equipped with the structure of a Riemannian manifold.

variational inference, especially Duncan et al. [2019] and the follow-up paper Nüsken and Renger [2021], both centered on the analysis of the so-called Stein variational gradient descent (SVGD) algorithm. In this section we take a further step inspired by SVGD and devise alternative learning algorithms for the regression task we are interested in. First, let us briefly report the idea of SVGD. With the notation of the previous section, consider a target measure $\pi \in \mathcal{P}_{\mathcal{H}}$ to be approximated. SVGD achieves this goal through the minimization of the Kullback–Leibler (KL) divergence[9]. In particular, this is done through the following gradient flow

$$\partial_t \mu_t = -\text{gradKL}(\mu_t|\pi), \tag{2.129}$$

where

$$\text{gradKL}(\rho_t|\pi) = -\nabla \cdot \left(\mu_t T_{\mu_t} \nabla \frac{\delta \text{KL}(\mu_t|\pi)}{\delta \mu_t} \right) \tag{2.130}$$

and $\frac{\delta \text{KL}(\mu_t|\pi)}{\delta \mu_t}$ represents the functional (Fréchet) derivative. If an appropriate differential structure on $\mathcal{P}_{\mathcal{H}}(\mathbb{R}^d)$ is defined, Duncan et al. [2019] show, $\text{gradKL}(\rho_t|\pi)$ is the Riemannian gradient of the KL divergence. Let us underline the fact that the driving vector field of the continuity equation (2.128) is an element of the RKHS, i.e.

$$T_{\mu_t} \nabla \frac{\delta \text{KL}(\mu_t|\pi)}{\delta \mu_t} \in \mathcal{H}. \tag{2.131}$$

In the framework of a regression task the main interest lies in the diffeomorphism instead of the related pushforward measure. In order to correctly ambient the problem we hence introduce the set of diffeomorphisms realized as the ODE flow at time $t = 1$ of the usual form

$$\begin{cases} \dot{z}_t = v_t(z_t), \quad t \in [0,1], \\ z_0 = x. \end{cases} \tag{2.132}$$

with $v \in L^2([0,1]; \mathcal{H})$. More precisely, let us define

$$\begin{aligned} \mathcal{M} := \Big\{ \phi : \mathbb{R}^d \to \mathbb{R}^d \ : \ &\text{there exists } v \in L^2([0,1]; \mathcal{H}) \text{ such that } \phi(x) = z_1, \\ &\text{where } z_t \text{ satisfies an ODE of the form (2.132)} \Big\}. \end{aligned} \tag{2.133}$$

Note that, under Assumption 2, $\mathcal{M}$ is well defined and consists of C^∞ diffeomorphisms (see the proof of Theorem 2.2.2).

In order to get acquainted with this object we show that it has a group structure.

[9] The KL divergence between a measure $\rho \in \mathcal{P}_{\mathcal{H}}$ a $\pi \in \mathcal{P}_{\mathcal{H}}$ is defined as follows:

$$\text{KL}(\rho|\pi) := \int_{\mathbb{R}^d} \log \left(\frac{\rho(x)}{\pi(x)} \right) d\rho(x). \tag{2.128}$$

Proposition 2.2.3. *Under Assumption 2, the set $\mathcal{M}$ can be endowed with a group structure with operation given by function composition.*

Proof. Clearly, the neutral element is the identity map, corresponding to the zero flow $0 \in L^2([0,1];\mathcal{H})$. Let now $w, v \in L^2([0,1];\mathcal{H})$ and $\phi, \psi \in \mathcal{M}$ be the associated maps, respectively. Then, $\phi \circ \psi \in \mathcal{M}$ as it is realized through the flow $s \in L^2([0,1];\mathcal{H})$ defined as

$$\begin{cases} s_t = 2w_{2t}, & \text{if } t \leq \frac{1}{2}, \\ s_t = 2v_{2t}, & \text{if } t > \frac{1}{2}. \end{cases} \tag{2.134}$$

Finally, $\phi^{-1} \in \mathcal{M}$ as it is sufficient to consider the time inverted ODE, namely the one associated to the vector field $t \mapsto v_{1-t}$. □

Remark. Proposition 2.2.3, though trivial, suggests that the best course of action for a rigorous treatment of $\mathcal{M}$ is to model it as an infinite-dimensional Lie group (Brooks and Trauber [1978]). Such an object, indeed, can be shown (Rudolf [2010]) to be locally diffeomorphic to an infinite-dimensional vector space. Here we sidestep this crucial step and instead proceed to set up a formal Riemannian calculus on $\mathcal{M}$, acting as though $\mathcal{M}$ were a smooth manifold.

Let us introduce a notion of tangent space equipped with positive-definite quadratic forms, playing the role of Riemannian metrics.

Definition 2.2.1. Let $\phi \in \mathcal{M}$, then define the tangent space of $\mathcal{M}$ at ϕ as

$$\mathcal{T}_\phi \mathcal{M} := \mathcal{H} \tag{2.135}$$

and the Riemannian metric $g_\phi : \mathcal{T}_\phi\mathcal{M} \times \mathcal{T}_\phi\mathcal{M} \to \mathbb{R}$ as

$$g_\phi(v, w) = \langle v, w \rangle_{\mathcal{H}}. \tag{2.136}$$

The rationale behind the above definition is clear if one thinks about the structure of $\mathcal{M}$. The maps in $\mathcal{M}$ are realized as ODE flows where the vector field is taken in the RKHS, hence it is reasonable to think of infinitesimal variations of such maps as elements of $\mathcal{H}$ themselves. Then, the choice of Riemannian metric is natural given the nature of the tangent space. The latter, in turn, induces a Riemannian distance on $\mathcal{M}$ as follows.

Definition 2.2.2. Let us define the distance

$$d^2_{\mathcal{K}}(\phi, \psi) := \inf_{v \in \Gamma(\phi,\psi)} \int_0^1 \|v_t\|^2_{\mathcal{H}} \, dt \tag{2.137}$$

where $\Gamma(\phi, \psi)$ denotes the set of connecting vector fields, namely

$$\Gamma(\phi, \psi) := \left\{ v \in L^2([0,1];\mathcal{H}) \mid v \text{ defines a map } \xi \text{ such that } \xi \circ \phi = \psi \right\}. \tag{2.138}$$

Note that the only difference between (2.114) and (2.137) lies in the fact that our perspective is on transport maps instead of the the probability measures being transported. The two viewpoints, however, are very much intertwined.

Proposition 2.2.4. *Under Assumption 2, the distance $d_{\mathcal{K}}$ defined in (2.137) is a metric on $\mathcal{M}$.*

Proof. First, $d_{\mathcal{K}} : \mathcal{M} \times \mathcal{M} \to [0, \infty)$ and is clearly symmetric. Assume $\phi, \psi \in M$ such that $d_{\mathcal{K}}(\phi, \psi) = 0$, then, by definition, there exists $\xi \in \mathcal{M}$ such that $\xi \circ \phi = \psi$. Moreover, ξ is realized as the flow of the zero vector field $0 \in L^2([0,1]; \mathcal{H})$, namely ξ is the identity mapping and, hence, $\phi = \psi$. On the other hand, if $\phi = \psi$ then $d_{\mathcal{K}}(\phi, \psi) = 0$ trivially. Finally, let $\phi, \psi, \xi \in M$, then $d_{\mathcal{K}}(\phi, \xi) \leq d_{\mathcal{K}}(\phi, \psi) + d_{\mathcal{K}}(\psi, \xi)$. This is true since

$$\begin{aligned} d^2_{\mathcal{K}}(\phi, \xi) &= \inf_{u \in \Gamma(\phi, \xi)} \int_0^1 \|u_t\|^2_{\mathcal{H}} \, dt \\ &\leq \frac{1}{\alpha} \inf_{v \in \Gamma(\phi, \psi)} \int_0^\alpha \frac{1}{\alpha} \left\| v_{\frac{t}{\alpha}} \right\|^2_{\mathcal{H}} dt + \frac{1}{1-\alpha} \inf_{w \in \Gamma(\psi, \xi)} \int_\alpha^1 \frac{1}{1-\alpha} \left\| w_{\frac{t}{1-\alpha} - \frac{\alpha}{1-\alpha}} \right\|^2_{\mathcal{H}} dt \\ &\leq \frac{1}{\alpha} d^2_{\mathcal{K}}(\phi, \psi) + \frac{1}{1-\alpha} d^2_{\mathcal{K}}(\psi, \xi) \end{aligned} \tag{2.139}$$

as any path from ϕ to ξ through ψ is in $\Gamma(\phi, \xi)$. Then, choosing

$$\alpha = \frac{d_{\mathcal{K}}(\phi, \psi)}{d_{\mathcal{K}}(\phi, \psi) + d_{\mathcal{K}}(\psi, \xi)} \tag{2.140}$$

in the above yields the claim. □

Remark. The distance $d_{\mathcal{K}}$ is constructed in a way that, formally,

$$d^2_{\mathcal{K}}(\phi, \psi) = \inf_{\xi} \left\{ \int_0^1 g_{\xi_t}(\partial_t \xi_t, \partial_t \xi_t) dt \;\middle|\; \xi_0 = \phi, \; \xi_1 = \psi \right\}, \tag{2.141}$$

however sidestepping the issue of defining the appropriate notion of differentiation for $\partial_t \xi_t$.

2.2.3 First order Riemannian optimization

Recall now that the $L^2_\mu(\mathbb{R}^d)$ functional derivative of a given functional $\mathcal{J} : \mathcal{M} \to \mathbb{R}$ is defined via

$$\int_{\mathbb{R}^d} \frac{\delta \mathcal{J}}{\delta \phi}(\phi) \psi d\mu = \frac{d}{d\varepsilon} \mathcal{J}(\phi + \varepsilon \psi) \Big|_{\varepsilon = 0} \tag{2.142}$$

where $\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \in L^2_\mu(\mathbb{R}^d)$ and $\psi \in C^\infty_c(\mathbb{R}^d; \mathbb{R}^d)$ is a test function.

Proposition 2.2.5 (Riemannian gradient)**.** *Let $\mathcal{J} : \mathcal{M} \to \mathbb{R}$ be a L^2_μ-differentiable functional. Then, under Assumption 2, the Riemannian gradient of $\mathcal{J}$ at $\phi \in \mathcal{M}$ is*

$$T_{\mu_\phi} \left(\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \circ \phi^{-1} \right) = \int_{\mathbb{R}^d} \mathcal{K}(\cdot, \phi(x)) \frac{\delta \mathcal{J}}{\delta \phi}(\phi)(x) d\mu(x) \in \mathcal{H}. \tag{2.143}$$

Proof. By definition, the Riemannian gradient $\operatorname{grad}(\mathcal{J})(\phi) \in \mathcal{T}_\phi \mathcal{M}$ of $\mathcal{J}$ at ϕ is the element such that

$$\frac{d}{dt} \mathcal{J}(\psi_t) \Big|_{t=0} = g_\phi \left(\operatorname{grad}(\mathcal{J})(\phi), \, \partial_t \psi_t|_{t=0} \right), \tag{2.144}$$

for all sufficiently regular curves $(\psi_t)_{t\in(-\varepsilon,\varepsilon)} \subset \mathcal{M}$ with $\psi_0 = \phi$ and $\partial_t \psi_t|_{t=0} \in \mathcal{T}_\phi \mathcal{M}$. Note that for any such curve with corresponding vector fields $(v_t)_{t\in(-\varepsilon,\varepsilon)}$ formally it holds

$$\partial_t \psi_t|_{t=s} = v_s. \tag{2.145}$$

Then, let us compute the left-hand side of (2.144)

$$\begin{aligned} \frac{d}{dt}\mathcal{J}(\psi_t)\Big|_{t=0} &= \frac{d}{d\varepsilon}\mathcal{J}\left(\phi + \int_0^\varepsilon v_s \circ \psi_s ds\right)\Big|_{\varepsilon=0} \\ &= \int_{\mathbb{R}^d} \frac{\delta \mathcal{J}}{\delta \phi}(\phi)(v_0 \circ \phi) d\mu \\ &= \int_{\mathbb{R}^d} \left(\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \circ \phi^{-1}\right) v_0 d\mu_\phi \\ &= \left\langle \left(\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \circ \phi^{-1}\right), v_0 \right\rangle_{L^2_{\mu_\phi}} \end{aligned} \tag{2.146}$$

where $\mu_\phi = \phi_\# \mu$. Note that this is well defined as $\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \in L^2_\mu(\mathbb{R}^d)$ by assumption and $v_0 \in L^2_{\mu_\phi}(\mathbb{R}^d)$ as a consequence of Assumption 2. Then, we can exploit the operator T_{μ_ϕ} in order to state that

$$\begin{aligned} \frac{d}{dt}\mathcal{J}(\psi_t)\Big|_{t=0} &= \left\langle T_{\mu_\phi}\left(\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \circ \phi^{-1}\right), v_0 \right\rangle_{\mathcal{H}}, \\ &= g_\phi\left(T_{\mu_\phi}\left(\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \circ \phi^{-1}\right), \partial_t \psi_t|_{t=0}\right), \end{aligned} \tag{2.147}$$

which proves the claim. □

Let us now rigorously restate in this context the regression problem of interest. Note that, for simplicity of exposition we assume the observations of g are not affected by noise, but all of the following can be readily stated in the context of noisy observations as well, as in Section 2.1.

Problem 5. Consider a target function $g \in L^2_\mu(\mathbb{R}^d; \mathcal{Y})$ where μ is a given Borel probability measure supported on $\mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}^n$. Our goal is to approximate g through the composition of maps in $\mathcal{M}$ and elements of the space $\mathcal{V}_m$, i.e. solving the following problem:

$$\min_{\phi \in \mathcal{M}} \quad \mathcal{J}(\phi) \tag{2.148}$$

where

$$\mathcal{J}(\phi) := \min_{f \in \mathcal{V}_m} \left\{ \int_{\mathbb{R}^d} (f(\phi(x)) - g(x))^2 d\mu(x) + \eta \|f\|^2_{\mathcal{V}_m} \right\} \tag{2.149}$$

and η is a parameter controlling the amount of regression regularization.

Now that the formal Riemannian structure is in place, we can leverage it in order to solve Problem 5. We propose to do so through a Riemannian geometric flow, of which

the simplest is a Riemannian gradient flow. In other words, we consider the following geometric evolution equation

$$\partial_t \phi_t = -\mathrm{grad}(\mathcal{J})(\phi_t). \tag{2.150}$$

Assuming $\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \in L^2_\mu(\mathbb{R}^d)$, Proposition 2.2.5 implies that (2.150) is equivalent to

$$\partial_t \phi_t = -T_{\mu_\phi}\left(\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \circ \phi^{-1}\right). \tag{2.151}$$

Lemma 2.2.6. *Let $(\phi_t)_{t\in[0,\infty)} \subset \mathcal{M}$ be the curve defined by the gradient flow (2.150), then*

$$\frac{d}{dt}\mathcal{J}(\phi_t) = -\int_{\mathbb{R}^d\times\mathbb{R}^d} \frac{\delta \mathcal{J}}{\delta \phi}(\phi_t)(x)\mathcal{K}(\phi_t(x),\phi_t(x'))\frac{\delta \mathcal{J}}{\delta \phi}(\phi_t)(x')d\mu(x)d\mu(x'). \tag{2.152}$$

Furthermore

$$\lim_{t\to\infty} \|\mathrm{grad}(\mathcal{J})(\phi_t)\|_{\mathcal{H}} = 0 \tag{2.153}$$

Proof. The first claim follows from the fact that

$$\frac{d}{dt}\mathcal{J}(\phi_t) = -\|\mathrm{grad}(\mathcal{J})(\phi_t)\|^2_{\mathcal{H}}. \tag{2.154}$$

Moreover for all $t > 0$

$$\mathcal{J}(\phi_0) \geq \mathcal{J}(\phi_0) - \mathcal{J}(\phi_t) = \int_0^t \|\mathrm{grad}(\mathcal{J})(\phi_s)\|^2_{\mathcal{H}}\, ds, \tag{2.155}$$

where we exploited the fact that $\mathcal{J} \geq 0$. This entails

$$\int_0^\infty \|\mathrm{grad}(\mathcal{J})(\phi_s)\|^2_{\mathcal{H}}\, ds \leq \mathcal{J}(\phi_0), \tag{2.156}$$

which then yields the claim. □

Problem 5 stands as an ideal guiding principle for the design of learning methods, however we must introduce a point-wise space discretization in order to state a problem which resembles what in practice we can hope to have access to and solve. We perform this step analogously to Section 2.1 (again, taking noiseless observations for simplicity).

Problem 6. Given a sample $(x^{(1)};\dots;x^{(N)})$ and observations $(y^{(1)};\dots;y^{(N)}) = (g(x^{(1)}),\dots,g(x^{(N)}))$, solve

$$\min_{\phi\in\mathcal{M}} \quad \mathcal{J}^N(\phi) \tag{2.157}$$

where

$$\mathcal{J}^N(\phi) := \min_{f\in\mathcal{V}_m}\left\{\frac{1}{2N}\sum_{i=1}^N (f(\phi(x^{(i)})) - y^{(i)})^2 + \eta\,\|f\|^2_{\mathcal{V}_m}\right\} \tag{2.158}$$

and η is a parameter controlling the amount of regression regularization.

The next step is to define a discrete gradient flow, namely compute the gradient of the discrete functional $\mathcal{J}^N$.

Proposition 2.2.7. *The gradient of the discrete functional $\mathcal{J}^N$ in (2.158) reads*

$$\operatorname{grad}(\mathcal{J}^N)(\phi) = \sum_{i=1}^{N} \mathcal{K}(\cdot, z^{(i)}) p^{(i)} \tag{2.159}$$

where we defined

$$z^{(i)} = \phi(x^{(i)}), \quad p^{(i)} = \partial_{z^{(i)}} \hat{\mathcal{J}}^N(z^{(1)}, \dots, z^{(N)}), \tag{2.160}$$

and $\hat{\mathcal{J}}^N(\phi(x^{(1)}), \dots, \phi(x^{(N)})) = \mathcal{J}^N(\phi)$.

Proof. The proof closely follows the one presented for the continuous case. Let $(\psi_t)_{t\in(-\varepsilon,\varepsilon)} \subset \mathcal{M}$ be a sufficiently regular curve with $\psi_0 = \phi$ and $\partial_t \psi_t|_{t=0} \in \mathcal{T}_\phi \mathcal{M}$. Then, let us compute

$$\begin{aligned}
\frac{d}{dt}\mathcal{J}^N(\psi_t)\Big|_{t=0} &= \frac{d}{d\varepsilon}\mathcal{J}^N\left(\phi + \int_0^\varepsilon v_s \circ \psi_s ds\right)\Big|_{\varepsilon=0} \\
&= \frac{d}{d\varepsilon}\hat{\mathcal{J}}^N\left(\phi(x^{(1)}) + \int_0^\varepsilon v_s(\psi_s(x^{(1)}))ds, \dots, \phi(x^{(N)}) + \int_0^\varepsilon v_s(\psi_s(x^{(N)}))ds\right)\Big|_{\varepsilon=0} \\
&= \sum_{i=1}^{N} (p^{(i)})^\top v_0(z^{(i)}) \\
&= \left\langle \sum_{i=1}^{N} \mathcal{K}(\cdot, z^{(i)}) p^{(i)}, v_0 \right\rangle_{\mathcal{H}}.
\end{aligned} \tag{2.161}$$

which, owing to the definition of Riemannian gradient, proves the claim. □

In view of this result, we can state the space-discretized version of the Riemannian gradient flow, namely

$$\partial_t \phi_t = -\sum_{i=1}^{N} \mathcal{K}(\cdot, z_t^{(i)}) p_t^{(i)}, \tag{2.162}$$

where $z_t^{(i)}$ and $p_t^{(i)}$, $i = 1, \dots, N$, refer to the map ϕ_t and are defined in the usual way.

Remark. As it is the case in Section 2.1, the introduction of a space discretization of the objective functional naturally induces a discretization on the Riemannian gradient. A priori, it is not at all obvious that such a property should hold, however, once again, the properties of the RKHS inner product lead to a (space-wise) finite dimensional problem. Indeed, the gradient flow in (2.162) is completely determined by the trajectories $z_t^{(i)}$ of the data points and the discrete functional Euclidean gradients $p_t^{(i)}$, $i = 1, \dots, N$.

Exploiting the vectorized notation introduced in the previous sections, we can hence equivalently consider the finite-dimensional ODE

$$\dot{\mathbf{z}}_t = -\mathbf{K}[\mathbf{z}_t]\mathbf{p}_t. \tag{2.163}$$

Notice the similarity of this with respect to the optimality conditions in (2.41). There, the gradient at time $t = 1$ is propagated backwards through the adjoint equation in order to define the adjoint variables at times $t < 1$. In some sense, the gradient flow discards the sensitivity equation and greedily attempts to minimize the objective functional as much as possible at all times.

Remark. Notice that Problem 6 is theoretically not well posed as it does not feature a uniquely determined solution. This may seem as a deal-breaker both from the analytical point of view and the algorithmic one. However, even though regularization in not explicitly present in the objective functional, recent works in Deep Learning literature (see, e.g., Belabbas [2020]) suggest that it implicitly lies in the chosen optimization algorithm (Riemannian gradient flow in this instance). This means that the optimization algorithm selects the solution, among the many available ones, according to some optimality criterion. Then, the latter can be exploited in order to make statements about the generalization error of the solution provided and, hence, justify the algorithm as a whole. Even though we do not attempt to characterize this optimality criterion here, it is important to keep implicit regularization in mind for explaining how this algorithm may potentially be successful.

The appeal of this approach is clear in that the it effectively decouples the control across the layers (here, still thought as a continuum) as it is simply determined by the Riemannian gradient of the loss functional. This makes the computation of the control at each time a lot cheaper.

2.2.4 Second order Riemannian optimization

The Riemannian gradient flow presented in the previous section may feature slow convergence speed. Given its simplicity, in this section we aim at enriching the dynamics with second order information, effectively setting up a Riemannian Newton-type flow. As it is always the case for second order optimization algorithms, our wish is to trade the cost to compute each descent direction (i.e. the control in our setting) for convergence speed. The first step in this direction, then, is to compute the Riemannian Hessian of a given functional $\mathcal{J} : \mathcal{M} \to \mathbb{R}$. We hence mimic the steps taken to calculate the gradient in the previous section.

Recall that the functional L^2_μ-Hessian of $\mathcal{J}$ evaluated at ϕ is a linear operator $\frac{\delta^2 \mathcal{J}}{\delta \phi^2}(\phi)$ from $L^2_\mu(\mathbb{R}^d)$ to itself defined via

$$\left\langle \frac{\delta^2 \mathcal{J}}{\delta \phi^2}(\phi)(\psi), \psi \right\rangle_{L^2_\mu} = \frac{d^2}{d\varepsilon^2} \mathcal{J}(\phi + \varepsilon\psi)\Big|_{\varepsilon=0}, \tag{2.164}$$

where $\psi \in C^\infty_c(\mathbb{R}^d; \mathbb{R}^d)$ is a test function.

Proposition 2.2.8 (Riemannian Hessian)**.** *Let $\mathcal{J} : \mathcal{M} \to \mathbb{R}$ be a doubly L^2_μ-differentiable functional. Then, under Assumption 2, the Riemannian Hessian of $\mathcal{J}$ at $\phi \in \mathcal{M}$ is a linear operator from $\mathcal{T}_\phi \mathcal{M}$ to itself acting in the following way*

$$\mathrm{Hess}(\mathcal{J})(\phi)(v) = T_{\mu_\phi}\left(\frac{\delta^2 \mathcal{J}}{\delta \phi^2}(\phi)\,(v \circ \phi) \circ \phi^{-1}\right) + T'_{\mu_\phi}\left(\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \circ \phi^{-1}, v\right), \tag{2.165}$$

where we defined the operator $T'_{\mu_\phi} : L^2_{\mu_\phi} \times \mathcal{H} \to \mathcal{H}$ *as*

$$T'_{\mu_\phi}(f, w) = \int_{\mathbb{R}^d} (f(x)w(x)^\top) k_2(\cdot, x) d\mu_\phi, \tag{2.166}$$

and k_2 *denotes the gradient of the kernel* k *in the second component, namely* $k_2(x, x')_i := \partial_{x'_i} k(x, x')$.

Proof. The Riemannian Hessian $\mathrm{Hess}(\mathcal{J})(\phi)$ of $\mathcal{J}$ at ϕ is the linear operator such that

$$\left.\frac{d^2}{dt^2}\mathcal{J}(\psi_t)\right|_{t=0} = g_\phi\left(\mathrm{Hess}(\mathcal{J})(\phi)(\left.\partial_t\psi_t\right|_{t=0}), \left.\partial_t\psi_t\right|_{t=0}\right), \tag{2.167}$$

for all geodesics $(\psi_t)_{t\in(-\varepsilon,\varepsilon)} \subset \mathcal{M}$ with $\psi_0 = \phi$ and $v_t := \partial_t\psi_t \in \mathcal{T}_\phi\mathcal{M}$. Note that for any geodesic it formally holds

$$g_\phi\left(\left.\partial_t^2\psi_t\right|_{t=0}, w\right), \quad \forall w \in \mathcal{T}_\phi\mathcal{M}, \tag{2.168}$$

where $\partial_t^2\psi_t$ formally denotes the covariant acceleration. Let us now compute the left-hand side of (2.167)

$$\begin{aligned}
\left.\frac{d^2}{dt^2}\mathcal{J}(\psi_t)\right|_{t=0} &= \left.\frac{d^2}{dt^2}\mathcal{J}\left(\phi + \int_0^t v_s \circ \psi_s ds\right)\right|_{t=0} \\
&= \left.\frac{d}{dt}\int_{\mathbb{R}^d} \frac{\delta\mathcal{J}}{\delta\phi}\left(\phi + \int_0^t v_s \circ \psi_s ds\right)^\top (v_t \circ \psi_t) d\mu\right|_{t=0} \\
&= \left.\int_{\mathbb{R}^d} (v_t \circ \psi_t)^\top \frac{d}{dt}\frac{\delta\mathcal{J}}{\delta\phi}\left(\phi + \int_0^t v_s \circ \psi_s ds\right) d\mu\right|_{t=0} \\
&+ \left.\int_{\mathbb{R}^d} \frac{\delta\mathcal{J}}{\delta\phi}\left(\phi + \int_0^t v_s \circ \psi_s ds\right)^\top \frac{d}{dt}(v_t \circ \psi_t) d\mu\right|_{t=0}.
\end{aligned} \tag{2.169}$$

The first term of the last expression in (2.169) reads

$$\begin{aligned}
&\int_{\mathbb{R}^d} \frac{\delta^2\mathcal{J}}{\delta\phi^2}(\phi)\,(v_0 \circ \phi)^\top (v_0 \circ \phi) d\mu \\
&= \int_{\mathbb{R}^d} \left(\frac{\delta^2\mathcal{J}}{\delta\phi^2}(\phi)\,(v_0 \circ \phi) \circ \phi^{-1}\right)^\top v_0 d\mu_\phi \\
&= \left\langle T_{\mu_\phi}\left(\frac{\delta^2\mathcal{J}}{\delta\phi^2}(\phi)\,(v_0 \circ \phi) \circ \phi^{-1}\right), v_0\right\rangle_{\mathcal{H}}.
\end{aligned} \tag{2.170}$$

Note that this is well defined as $\frac{\delta^2\mathcal{J}}{\delta\phi^2}(\phi)\,(v_0 \circ \phi) \in L^2_\mu(\mathbb{R}^d)$ by assumption and $v_0 \in L^2_{\mu_\phi}(\mathbb{R}^d)$ as a consequence of Assumption 2. Before advancing in the computations, we remark that Zhou [2008] shows how, owing to the regularity of $\mathcal{K}$, $k_2(\cdot, x)^\top\beta \in \mathcal{G}$ for all $\beta \in \mathbb{R}^d$ and the following characterization of derivative evaluation holds

$$(\partial_x w(x)\beta)_i = \left\langle k_2(\cdot, x)^\top\beta, w_i\right\rangle_{\mathcal{G}}, \tag{2.171}$$

for all $\beta \in \mathbb{R}^d$ and $w \in \mathcal{H}$. Then, the second term in (2.169) can be recast as

$$\begin{aligned}
&\int_{\mathbb{R}^d} \frac{\delta \mathcal{J}}{\delta \phi}(\phi)^\top (((\partial_x v_0) \circ \phi) v_0 \circ \phi)\, d\mu \\
&= \int_{\mathbb{R}^d} \frac{\delta \mathcal{J}}{\delta \phi}(\phi)(x)^\top \left\langle k_2(\cdot, \phi(x))^\top v_0(\phi(x)), v_0 \right\rangle_{\mathcal{H}} d\mu(x) \\
&= \left\langle \int_{\mathbb{R}^d} \frac{\delta \mathcal{J}}{\delta \phi}(\phi)(x) k_2(\cdot, \phi(x))^\top v_0(\phi(x)) d\mu(x), v_0 \right\rangle_{\mathcal{H}} \\
&= \left\langle T'_{\mu_\phi}\left(\frac{\delta \mathcal{J}}{\delta \phi}(\phi) \circ \phi^{-1}, v_0\right), v_0 \right\rangle_{\mathcal{H}},
\end{aligned} \tag{2.172}$$

where we exploited (2.168) in order to discard the term involving $\partial_t^2 \psi_t$. This completes the proof. □

Let us now formally state the Riemannian Newton flow we are interested in. Sidestepping the important issue of Hessian invertibility, the relevant geometric evolution equation is

$$\partial_t \phi_t = -\mathrm{Hess}(\mathcal{J})(\phi_t)^{-1} (\mathrm{grad}(\mathcal{J})(\phi_t)). \tag{2.173}$$

We leave a more thorough analysis of this evolution equation as future work. Mirroring the gradient flow case, the next step is to more precisely specify the Newton flow for the the discrete functional $\mathcal{J}^N$ in (2.158).

Proposition 2.2.9. *With the same notation as in the previous section, the Hessian of the discrete functional $\mathcal{J}^N$ in (2.158) acts as*

$$\mathrm{Hess}(\mathcal{J}^N)(\phi)(v) = \sum_{i=1}^N \left((p^{(i)} v(z^{(i)})^\top) k_2(\cdot, z^{(i)}) + \mathcal{K}(\cdot, z^{(i)}) \sum_{j=1}^N h^{(i,j)} v(z^{(j)}) \right) \tag{2.174}$$

where we defined

$$h^{(i,j)} := \partial_{z^{(j)}} \partial_{z^{(i)}} \hat{\mathcal{J}}^N(z^{(1)}, \dots, z^{(N)}). \tag{2.175}$$

Proof. Here we take a slightly different approach with respect to the infinite dimensional case and exploit the fact that

$$\mathrm{Hess}(\mathcal{J}^N)(\phi)(\partial_t \psi_t|_{t=0}) = \frac{d}{dt} \mathrm{grad}(\mathcal{J}^N)(\psi_t)\Big|_{t=0}, \tag{2.176}$$

$(\psi_t)_{t \in (-\varepsilon, \varepsilon)} \subset \mathcal{M}$ being a sufficiently regular curve with $\psi_0 = \phi$ and $v_s := \partial_t \psi_t|_{t=s} \in \mathcal{T}_\phi \mathcal{M}$.

Then, let us compute

$$
\begin{aligned}
&\frac{d}{dt}\mathrm{grad}(\mathcal{J}^N)(\psi_t)\Big|_{t=0} \\
=&\frac{d}{dt}\sum_{i=1}^{N}\mathcal{K}\left(\cdot,\phi(x^{(i)})+\int_0^t v_s(\psi_s(x^{(i)}))ds\right) \\
\times&\partial_{z^{(i)}}\hat{\mathcal{J}}^N\left(\phi(x^{(1)})+\int_0^t v_s(\psi_s(x^{(1)}))ds,\dots,\phi(x^{(N)})+\int_0^t v_s(\psi_s(x^{(N)}))ds\right)\Big|_{t=0} \\
=&\sum_{i=1}^{N}\frac{d}{dt}\mathcal{K}\left(\cdot,\phi(x^{(i)})+\int_0^t v_s(\psi_s(x^{(i)}))ds\right)\Big|_{t=0}p^{(i)} \\
+&\mathcal{K}\left(\cdot,z^{(i)}\right)\frac{d}{dt}\partial_{z^{(i)}}\hat{\mathcal{J}}^N\left(\phi(x^{(1)})+\int_0^t v_s(\psi_s(x^{(1)}))ds,\dots,\phi(x^{(N)})+\int_0^t v_s(\psi_s(x^{(N)}))ds\right)\Big|_{t=0} \\
=&\sum_{i=1}^{N}\left((p^{(i)}v_0(z^{(i)})^\top)k_2(\cdot,z^{(i)})+\mathcal{K}(\cdot,z^{(i)})\sum_{j=1}^{N}h^{(i,j)}v_0(z^{(j)})\right),
\end{aligned}
\tag{2.177}
$$

which concludes the proof. □

Propagating the Newton flow dynamics entails computing the Newton descent direction at each time t, i.e. one must solve

$$
\mathrm{Hess}(\mathcal{J}^N)(\phi)(v) = -\mathrm{grad}(\mathcal{J}^N)(\phi), \tag{2.178}
$$

for $v \in \mathcal{H}$. The fist challenge which arises in this setting, then, is that there is no obvious finite dimensional subspace of $\mathcal{H}$ where to look for v in. Furthermore, it is not clear if no solutions, one solution or multiple solutions exist. In order to gain further insight, let us first define the subspace $\mathcal{S} := \mathcal{S}_1 \cup \mathcal{S}_2$ where

$$
\begin{aligned}
\mathcal{S}_1 &:= \left\{\mathcal{K}(\cdot,z^{(i)})\beta \;\middle|\; \beta\in\mathbb{R}^d, i=1,\dots,N\right\}, \\
\mathcal{S}_2 &:= \left\{p^{(i)}(k_2(\cdot,z^{(i)})^\top\beta) \;\middle|\; \beta\in\mathbb{R}^d, i=1,\dots,N\right\}.
\end{aligned}
\tag{2.179}
$$

Proposition 2.2.10. *Under assumption 2, for all $\phi \in \mathcal{M}$ the Riemannian Hessian of $\mathcal{J}^N$ at ϕ is a linear, continuous and compact operator. Furthermore, $\mathcal{S}_1^\perp$ is in the kernel of* $\mathrm{Hess}(\mathcal{J}^N)(\phi)$.

Proof. The linearity is trivial, while the continuity follows from the boundedness of the kernel $\mathcal{K}$ and its first derivatives. Furthermore, note that the image of $\mathrm{Hess}(\mathcal{J}^N)(\phi)(v)$ is in $\mathcal{S}$ which is finite dimensional, hence the compactness. Finally, taking $v \in \mathcal{S}_1^\perp$ yields $v(z^{(i)}) = 0$ for all $i = 1,\dots,N$, which trivially implies $\mathrm{Hess}(\mathcal{J}^N)(\phi)(v) = 0$. □

Proposition 2.2.10 immediately implies that $\mathrm{Hess}(\mathcal{J}^N)(\phi)(v)$ is never invertible. This makes the Newton system in (2.178) ambiguous. Among the strategies to overcome this

issue one may pursue, we propose to consider, instead of $\mathrm{Hess}(\mathcal{J}^N)(\phi)$, a damped version

$$\mathrm{Hess}(\mathcal{J}^N)(\phi) + \omega I, \tag{2.180}$$

$\omega > 0$ being the damping parameter. In other words, we modify (2.178) and consider the following equation

$$\Big(\mathrm{Hess}(\mathcal{J}^N)(\phi) + \omega I\Big)(v) = -\mathrm{grad}(\mathcal{J}^N)(\phi), \quad v \in \mathcal{H}, \tag{2.181}$$

defining the descent direction at ϕ. Then, the resulting Riemannian damped Newton flow formally reads

$$\begin{aligned} \partial_t \phi_t &= v_t, \\ \Big(\mathrm{Hess}(\mathcal{J}^N)(\phi_t) + \omega_t I\Big)(v_t) &= -\mathrm{grad}(\mathcal{J}^N)(\phi_t), \end{aligned} \tag{2.182}$$

where we introduced a possible time-dependency in the damping parameter ω_t. Let us now prove that the linear equation in $\mathcal{H}$ in (2.182) can be reduced to a finite dimensional problem. First, let us introduce some vectorized notation:

$$\mathbf{K}'[\mathbf{z},\mathbf{p}] := \begin{pmatrix} p^{(1)} k_2(z^{(1)}, z^{(1)})^\top & \dots & p^{(N)} k_2(z^{(1)}, z^{(N)})^\top \\ \vdots & \ddots & \vdots \\ p^{(1)} k_2(z^{(N)}, z^{(1)})^\top & \dots & p^{(N)} k_2(z^{(N)}, z^{(N)})^\top \end{pmatrix} \in \mathbb{R}^{dN \times dN} \tag{2.183}$$

and

$$\mathbf{K}''[\mathbf{z},\mathbf{p}] := \begin{pmatrix} (p^{(1)})^\top p^{(1)} k_{12}(z^{(1)}, z^{(1)}) & \dots & (p^{(1)})^\top p^{(N)} k_{12}(z^{(1)}, z^{(N)}) \\ \vdots & \ddots & \vdots \\ (p^{(N)})^\top p^{(1)} k_{12}(z^{(N)}, z^{(1)}) & \dots & (p^{(N)})^\top p^{(N)} k_{12}(z^{(N)}, z^{(N)}) \end{pmatrix} \in \mathbb{R}^{dN \times dN} \tag{2.184}$$

where

$$k_{12}(x, x')_{ij} := \partial_{x_i} \partial_{x'_j} k(x, x'), \tag{2.185}$$

are the kernel matrix derivatives. Note the dependence on $\mathbf{z}$ and $\mathbf{p}$, hence on ϕ and $\mathcal{J}^N$ respectively. Moreover, let

$$\begin{aligned} \mathbf{v}[\mathbf{z}] &:= (v(z^{(1)}); \dots; v(z^{(N)})) \in \mathbb{R}^{dN}, \\ \mathbf{v}'[\mathbf{z},\mathbf{p}] &:= (\partial_z v(z^{(1)})^\top p^{(1)}; \dots; \partial_z v(z^{(N)})^\top p^{(N)}) \in \mathbb{R}^{dN}, \\ \mathbf{H}[\mathbf{z}] &:= \left(h^{(i,j)}\right)_{ij} \in \mathbb{R}^{dN \times dN}, \end{aligned} \tag{2.186}$$

be the vector of evaluations of the descent direction v in the points $z^{(i)}$, the vector of derivatives of the descent direction in $z^{(i)}$ along $p^{(i)}$ and the Euclidean Hessian of $\hat{\mathcal{J}}^N$.

Proposition 2.2.11. *The damped Newton linear system in (2.181) in $\mathcal{H}$ is equivalent to the following linear systems*

$$(\mathbf{K}'[\mathbf{z},\mathbf{p}] + \mathbf{K}[\mathbf{z}]\mathbf{H}[\mathbf{z}] + \omega I)\mathbf{v}[\mathbf{z}] = -\mathbf{K}\mathbf{p}, \tag{2.187}$$

$$(\mathbf{K}''[\mathbf{z},\mathbf{p}] + \mathbf{K}'[\mathbf{z},\mathbf{p}]^\top \mathbf{H}[\mathbf{z}])\mathbf{v}[\mathbf{z}] + \omega \mathbf{v}'[\mathbf{z},\mathbf{p}] = -(\mathbf{K}')^\top \mathbf{p}, \tag{2.188}$$

$$\begin{pmatrix} \mathbf{K}[\mathbf{z}] & \mathbf{K}'[\mathbf{z},\mathbf{p}] \\ \mathbf{K}'[\mathbf{z},\mathbf{p}]^\top & \mathbf{K}''[\mathbf{z},\mathbf{p}] \end{pmatrix} \boldsymbol{\beta} = \begin{pmatrix} \mathbf{v}[\mathbf{z}] \\ \mathbf{v}'[\mathbf{z},\mathbf{p}] \end{pmatrix}, \tag{2.189}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{2dN}$ *and*

$$v = \sum_{i=1}^{N} \left(\mathcal{K}(\cdot, z^{(i)})\beta^{(i)} + p^{(i)}(k_2(\cdot, z^{(i)})^\top \beta^{(i+N)}) \right). \tag{2.190}$$

Proof. Denote $P_{\mathcal{S}^\perp}$ the projection operator from $\mathcal{H}$ onto the subspace $\mathcal{S}$. Applying $P_{\mathcal{S}^\perp}$ to both sides of (2.181) yields

$$P_{\mathcal{S}^\perp} v = 0, \tag{2.191}$$

namely $v \in \mathcal{S}$. Hence, we can represent v as in (2.190). Then, projecting $\mathrm{Hess}(\mathcal{J}^N)(\phi)(v)$ onto $\mathcal{K}(\cdot, z^{(r)})\beta$ gives

$$\begin{aligned} &\left\langle \mathcal{K}(\cdot, z^{(i)})\beta, \mathrm{Hess}(\mathcal{J}^N)(\phi)(v) \right\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{N} \left(\beta^\top p^{(i)} v(z^{(i)})^\top k_2(z^{(r)}, z^{(i)}) + \beta^\top \mathcal{K}(z^{(r)}, z^{(i)}) \sum_{j=1}^{N} h^{(i,j)} v(z^{(j)}) \right), \end{aligned} \tag{2.192}$$

while the result of doing the same for v and $\mathrm{grad}(\mathcal{J}^N)(\phi)$ is

$$\beta^\top v(z^{(r)}), \qquad \beta^\top \sum_{i=1}^{N} \mathcal{K}(z^{(r)}, z^{(i)}) p^{(i)}, \tag{2.193}$$

respectively. As this is true for all $\beta \in \mathbb{R}^d$ and $r = 1, \dots, N$ equation (2.187) follows. We then must perform the same projections onto $p^{(r)} k_2(\cdot, z^{(r)})^\top \beta$ which yield

$$\begin{aligned} &\left\langle p^{(r)} k_2(\cdot, z^{(r)})^\top \beta, \mathrm{Hess}(\mathcal{J}^N)(\phi)(v) \right\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{N} \left(\beta^\top (p^{(r)})^\top p^{(i)} k_{12}(z^{(r)}, z^{(i)}) v(z^{(i)}) + \beta^\top k_2(z^{(i)}, z^{(r)}) (p^{(r)})^\top \sum_{j=1}^{N} h^{(i,j)} v(z^{(j)}) \right), \end{aligned} \tag{2.194}$$

and

$$\beta^\top \partial_z v(z^{(r)})^\top p^{(r)}, \qquad \beta^\top \sum_{i=1}^{N} k_2(z^{(i)}, z^{(r)}) (p^{(r)})^\top p^{(i)}, \tag{2.195}$$

respectively. This holds for all $\beta \in \mathbb{R}^d$ and $r = 1, \dots, N$ and equation (2.188) is proven. Finally, equation (2.189) follows from the fact that $v \in \mathcal{S}$. □

The above proposition effectively reduces (2.182) to a finite-dimensional problem. The damped Newton evolution equation, indeed, is equivalent to

$$\dot{\mathbf{z}}_t = \mathbf{v}_t[\mathbf{z}_t], \tag{2.196}$$

where computing $\mathbf{v}_t[\mathbf{z}_t]$ involves the solution of two linear systems of sizes dN and $2dN$ respectively (note that $\mathbf{v}'_t[\mathbf{z}_t, \mathbf{p}_t]$ can be deduced from (2.188) by rearranging the terms). Therefore, we achieved the goal of enriching the gradient flow with second order information, yet keeping the problem computationally tractable and, more importantly, decoupled

across time. Clearly, the cost to compute each descent direction is larger, however we experimentally show in Chapter 3 that this comes with a significant gain in convergence speed.

Note that the difference between the damped Newton flow (2.182) and the gradient flow (2.162) lies not only in the optimization strategy, but in principle they may converge (if they converge at all) to a different solution. In other words, not only the controls are different as a consequence of the optimization algorithm, but the maps $\lim_{t\to\infty} \phi_t^{\mathrm{GF}}$ and $\lim_{t\to\infty} \phi_t^{\mathrm{DNF}}$ may be different as well since $\mathcal{J}^N$ does not feature a unique minimum in general. This is also evident from the fact that, for the damped Newton flow, the subspace where the control lies is different than any other we have seen so far as it involves the kernel derivatives. If this is indeed the case, the interpretation from the implicit regularization (assuming there is any) point of view would be that the two optimization algorithms select solutions according to different optimality criteria.

Chapter 3

Training algorithms and numerical experiments

In this Section we detail some training algorithms for Kernel ODEs as introduced in Chapter 2. The underlying motivating factors are essentially two: either the speed-up of training with respect to classic approaches (however at the cost of increased computational complexity), or the wish to reduce the cost per iteration. A batch of algorithms fulfill the former goal and is based on the formulation of the problem given in Section 2.1, while the other proposed algorithms stem from the latter motivating factor and follow from the ideas introduced in Section 2.2. Finally, we perform numerical tests on synthetic data in order to explore their effectiveness.

3.1 Algorithms

Throughout this section we will make use of the following vectorized notation with respect the layer index l, namely

$$\begin{aligned} \vec{\mathbf{z}} &:= (\mathbf{z}_0; \dots; \mathbf{z}_L) \in \mathbb{R}^{dN(L+1)}, \\ \vec{\boldsymbol{\beta}} &:= (\boldsymbol{\beta}_0; \dots; \boldsymbol{\beta}_L) \in \mathbb{R}^{dN(L+1)}, \\ \vec{\mathbf{p}} &:= (\mathbf{p}_0; \dots; \mathbf{p}_L) \in \mathbb{R}^{dN(L+1)}. \end{aligned} \tag{3.1}$$

Furthermore, we consider a regression space $\mathcal{V}_m$ which is the linear span of basis functions $f_1, \dots, f_m$. Then, given a vector of N data points $\mathbf{z} \in \mathbb{R}^{dN}$ we define the associated Vandermonde matrix $\mathbf{V}[\mathbf{z}] \in \mathbb{R}^{N \times m}$ as $\mathbf{V}[\mathbf{z}] := (f_j(z^{(i)}))_{ij}$.

3.1.1 Preconditioned gradient descent

The first algorithm we propose is a preconditioned version of gradient descent. With the notation introduced in Proposition 2.1.8, consider a uniform time grid with step size h, $s = 1$, and $a_{11} = 0$, $b_1 = 1$, namely the Explicit Euler method. Recall that the discrete

Hamiltonian reads

$$H(\mathbf{z}, \boldsymbol{\beta}, \mathbf{p}) = \mathbf{p}^\top \mathbf{K}[\mathbf{z}]\boldsymbol{\beta} - \frac{\gamma}{2}\boldsymbol{\beta}^\top \mathbf{K}[\mathbf{z}]\boldsymbol{\beta}. \tag{3.2}$$

Fix some layer l, then, taking the gradient with respect to the control of the above gives

$$\partial_{\boldsymbol{\beta}} H(\mathbf{z}, \boldsymbol{\beta}, \mathbf{p}) = \mathbf{K}[\mathbf{z}](\mathbf{p} - \gamma\boldsymbol{\beta}), \tag{3.3}$$

where, for notational simplicity, we dropped the layer index. Standard gradient descent prescribes to update the control as

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \omega \mathbf{K}[\mathbf{z}^{(k)}](\mathbf{p}^{(k)} - \gamma\boldsymbol{\beta}^{(k)}), \tag{3.4}$$

$\omega > 0$ being a damping parameter and $k = 1,2,\dots$ the iteration index (note the "+" sign in the gradient update since we take sign conversions such that the Hamiltonian must be maximized). On the other hand, if the kernel satisfies Assumption 1 and, hence, $\mathbf{K}[\mathbf{z}]$ is invertible, the MSA Algorithm 1.3.2 suggests to maximize the Hamiltonian and set

$$\boldsymbol{\beta}^{(k+1)} = \frac{1}{\gamma}\mathbf{p}^{(k)}. \tag{3.5}$$

Here we choose to interpolate between these two methodologies and propose a soft Hamiltonian maximization update by solving the system

$$(I + \omega\gamma \mathbf{K}[\mathbf{z}^{(k)}])\Delta = \omega \mathbf{K}[\mathbf{z}^{(k)}](\mathbf{p}^{(k)} - \boldsymbol{\beta}^{(k)}) \tag{3.6}$$

and setting

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \Delta. \tag{3.7}$$

Note that for $\omega \to \infty$ we recover the MSA update, while for $\omega \to 0$ the gradient descent step. The choice of ω is then delegated to a line-search-type procedure, which checks that the time-discretized functional, which we recall being

$$\mathcal{J}(\vec{\mathbf{z}}, \vec{\boldsymbol{\beta}}) := \frac{1}{2N}\mathcal{E}(\mathbf{z}_L, \mathbf{y}) + \frac{h\gamma}{2}\sum_{l=0}^{L-1} \boldsymbol{\beta}_l^\top \mathbf{K}[\mathbf{z}_l]\boldsymbol{\beta}_l, \tag{3.8}$$

decreases with the new set of controls. The full procedure is described in Algorithm 3.1.1.

Algorithm 3.1.1: Preconditioned gradient descent (PGD) algorithm

Input: A sample $\mathbf{x}$, observations $\mathbf{y}$, initial controls $\boldsymbol{\beta}_0^{(0)}, \dots, \boldsymbol{\beta}_{L-1}^{(0)}$, kernel $\mathcal{K}$, maximum iterations M, tolerance tol, line search parameter ω_0, regression basis functions $f_1, \dots, f_m$.

Output: Approximation function $\hat{g}$.

Set iteration counter $k \leftarrow 0$;
Set $\mathbf{z}_0^{(0)} = \mathbf{x}$
for $l = 0, \dots, L-1$ **do**
  Set $\mathbf{z}_{l+1}^{(0)} = \mathbf{z}_l^{(0)} + h\mathbf{K}[\mathbf{z}_l^{(0)}]\boldsymbol{\beta}_l^{(0)}$
end
Compute regression parameters $c^{(0)} = \arg\min_{c \in \mathbb{R}^m} \left\| \mathbf{V}[\mathbf{z}_L^{(0)}]c - \mathbf{y} \right\|_2^2$
Set iteration counter $k \leftarrow 0$
while $k < M$ **do**
  Set $\mathbf{p}_L^{(k)} = -\frac{1}{2N}\partial_{\mathbf{z}}\mathcal{E}(\mathbf{z}_L^{(k)}, \mathbf{y})$
  for *$l = 0, \dots, L$-1* **do**
    Set $\mathbf{p}_l^{(k)} = \mathbf{p}_{l+1}^{(k)} + h\partial_{\mathbf{z}} \left((\mathbf{p}_{l+1}^{(k)})^\top \mathbf{K}[\mathbf{z}_l^{(k)}]\boldsymbol{\beta}_l^{(k)} - \frac{\gamma}{2}(\boldsymbol{\beta}_l^{(k)})^\top \mathbf{K}[\mathbf{z}_l^{(k)}]\boldsymbol{\beta}_l^{(k)} \right)$
  end
  Set $\omega = \omega_0$
  do
    Set $\mathbf{z}_l^{\text{new}} = \mathbf{z}_l^{(k)}$, $\boldsymbol{\beta}_l^{\text{new}} = \boldsymbol{\beta}_l^{(k)}$, $l = 0, \dots, L$
    for $l = 0, \dots, L-1$ **do**
      Compute update Δ_l solving

$$(I + \omega\gamma\mathbf{K}[\mathbf{z}_l^{(k)}])\Delta_l = \omega\mathbf{K}[\mathbf{z}_l^{(k)}](\mathbf{p}_{l+1}^{(k)} - \boldsymbol{\beta}_l^{(k)}) \tag{3.9}$$

      Update $\boldsymbol{\beta}_l^{\text{new}} \leftarrow \boldsymbol{\beta}_l^{\text{new}} + \Delta_l$
      Set $\mathbf{z}_{l+1}^{\text{new}} = \mathbf{z}_l^{\text{new}} + h\mathbf{K}[\mathbf{z}_l^{\text{new}}]\boldsymbol{\beta}_l^{\text{new}}$
    end
    Update $\omega \leftarrow \alpha\omega$
  while $\mathcal{J}(\vec{\mathbf{z}}^{new}, \vec{\boldsymbol{\beta}}^{new}) \geq \mathcal{J}(\vec{\mathbf{z}}^{(k)}, \vec{\boldsymbol{\beta}}^{(k)})$;
  Set $\mathbf{z}_l^{(k+1)} = \mathbf{z}_l^{\text{new}}$, $\boldsymbol{\beta}_l^{(k+1)} = \boldsymbol{\beta}_l^{\text{new}}$, $l = 0, \dots, L$
  Compute regression parameters $c^{(k+1)} = \arg\min_{c \in \mathbb{R}^m} \left\| \mathbf{V}[\mathbf{z}_L^{(k+1)}]c - \mathbf{y} \right\|_2^2$
end
Compute the surrogate function $\hat{g}$ from $\vec{\mathbf{z}}^{(M)}$, $\vec{\boldsymbol{\beta}}^{(M)}$ and $c^{(M)}$.

3.1.2 Time-parallel training algorithm

Consider the usual time grid time grid $0 = t_0 < t_1 \dots < t_L = 1$ on the time interval [0,1] and introduce two sets of parameters $\{\mathbf{a}_l\}_{l=0}^L$ and $\{\mathbf{b}_l\}_{l=0}^L$ corresponding to approximations of the state $\mathbf{z}$ and the adjoint state $\mathbf{p}$ at times $t_0, \dots, t_L$. Define the nonlinear solution

operators $\mathbf{d}$ and $\mathbf{q}$ for the two point boundary value problem (2.41) on the subinterval $[t_l, t_{l+1}]$ with initial condition $\mathbf{z}_{t_l} = \mathbf{a}_l$ and final condition $\mathbf{p}_{t_{l+1}} = \mathbf{b}_{l+1}$, defined so that $\mathbf{d}$ propagates the state $\mathbf{z}$ forward to t_{l+1} and $\mathbf{q}$ propagates the adjoint backward to t_l:

$$\begin{pmatrix} \mathbf{z}_{t_{l+1}} \\ \mathbf{p}_{t_l} \end{pmatrix} = \begin{pmatrix} \mathbf{d}(\mathbf{a}_l, \mathbf{b}_{l+1}) \\ \mathbf{q}(\mathbf{a}_l, \mathbf{b}_{l+1}) \end{pmatrix}. \tag{3.10}$$

Using these solution operators, we can write the two-point boundary value problem as a system of subproblems, which have to satisfy the matching conditions

$$\begin{aligned} \mathbf{a}_0 - \mathbf{x} &= 0, & \mathbf{b}_0 - \mathbf{q}(\mathbf{a}_0, \mathbf{b}_1) &= 0, \\ \mathbf{a}_1 - \mathbf{d}(\mathbf{a}_0, \mathbf{b}_1) &= 0, & \mathbf{b}_1 - \mathbf{q}(\mathbf{a}_1, \mathbf{b}_2) &= 0, \\ &\vdots & &\vdots \\ \mathbf{a}_{L-1} - \mathbf{d}(\mathbf{a}_{L-2}, \mathbf{b}_{L-1}) &= 0, & \mathbf{b}_{L-1} - \mathbf{q}(\mathbf{a}_{L-1}, \mathbf{b}_L) &= 0, \\ \mathbf{a}_L - \mathbf{d}(\mathbf{a}_{L-1}, \mathbf{b}_L) &= 0, & \mathbf{b}_L + \frac{1}{2N}\partial_{\mathbf{z}}\mathcal{E}(\mathbf{a}_L, \mathbf{y}) &= 0. \end{aligned} \tag{3.11}$$

This nonlinear system of equations can be solved using Newton's method. Collecting the unknowns in vectors $\vec{\mathbf{a}} = (\mathbf{a}_0; \ldots; \mathbf{a}_L)$ and $\vec{\mathbf{b}} = (\mathbf{b}_0; \ldots; \mathbf{b}_L)$, we obtain the nonlinear system

$$\mathbf{f}(\vec{\mathbf{a}}, \vec{\mathbf{b}}) = \begin{pmatrix} \mathbf{a}_0 - \mathbf{x} \\ \mathbf{a}_1 - \mathbf{d}(\mathbf{a}_0, \mathbf{b}_1) \\ \vdots \\ \mathbf{a}_L - \mathbf{d}(\mathbf{a}_{L-1}, \mathbf{b}_L) \\ \mathbf{b}_0 - \mathbf{q}(\mathbf{a}_0, \mathbf{b}_1) \\ \vdots \\ \mathbf{b}_{L-1} - \mathbf{q}(\mathbf{a}_{L-1}, \mathbf{b}_L) \\ \mathbf{b}_L + \frac{1}{2N}\partial_{\mathbf{z}}\mathcal{E}(\mathbf{a}_L, \mathbf{y}) \end{pmatrix} \tag{3.12}$$

Using Newton's method to solve $\mathbf{f} = 0$ gives the iteration

$$\mathbf{J}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}) \begin{pmatrix} \Delta\vec{\mathbf{a}}^{(k)} \\ \Delta\vec{\mathbf{b}}^{(k)} \end{pmatrix} = -\mathbf{f}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}), \quad \begin{pmatrix} \vec{\mathbf{a}}^{(k+1)} \\ \vec{\mathbf{b}}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \vec{\mathbf{a}}^{(k)} \\ \vec{\mathbf{b}}^{(k)} \end{pmatrix} + \omega \begin{pmatrix} \Delta\vec{\mathbf{a}}^{(k)} \\ \Delta\vec{\mathbf{b}}^{(k)} \end{pmatrix} \tag{3.13}$$

where, again, $\omega > 0$ denotes a damping parameter and $k = 1,2,\ldots$ denotes the iteration index. The Jacobian matrix here is given by

$$\mathbf{J}(\vec{\mathbf{a}}, \vec{\mathbf{b}}) = \left(\begin{array}{cccc|ccc} \mathbf{I} & & & & 0 & & \\ -\partial_{\mathbf{a}}\mathbf{d}(\mathbf{a}_0, \mathbf{b}_1) & \mathbf{I} & & & -\partial_{\mathbf{b}}\mathbf{d}(\mathbf{a}_0, \mathbf{b}_1) & & \\ & \ddots & \ddots & & & \ddots & \\ & & -\partial_{\mathbf{a}}\mathbf{d}(\mathbf{a}_{L-1}, \mathbf{b}_L) & \mathbf{I} & & & -\partial_{\mathbf{b}}\mathbf{d}(\mathbf{a}_{L-1}, \mathbf{b}_L) \\ \hline -\partial_{\mathbf{a}}\mathbf{q}(\mathbf{a}_0, \mathbf{b}_1) & & & & \mathbf{I} & -\partial_{\mathbf{b}}\mathbf{q}(\mathbf{a}_0, \mathbf{b}_1) & \\ & \ddots & & & \ddots & \ddots & \\ & & -\partial_{\mathbf{a}}\mathbf{q}(\mathbf{a}_{L-1}, \mathbf{b}_L) & & & \mathbf{I} & -\partial_{\mathbf{b}}\mathbf{q}(\mathbf{a}_{L-1}, \mathbf{b}_L) \\ & & & \frac{1}{2N}\partial_{\mathbf{z}}^2\mathcal{E}(\mathbf{a}_L, \mathbf{y}) & & & \mathbf{I} \end{array}\right). \tag{3.14}$$

This is already enough to have a working algoritm to solve the problem. Indeed, we can introduce the discretized versions $\hat{\mathbf{d}}_{\mathrm{RK}}$ and $\hat{\mathbf{q}}_{\mathrm{RK}}$ of the solution operators corresponding to a specific Runge-Kutta method, and, in turn, the discrete function $\hat{\mathbf{f}}_{\mathrm{RK}}$ and Jacobian $\hat{\mathbf{J}}_{\mathrm{RK}}$. Then, we can set up a Newton iteration with the discretized system. A particularly favourable choice in this sense is given by the symplectic Euler method, namely

$$\begin{aligned}\hat{\mathbf{q}}_{\mathrm{SE}}(\mathbf{a}_l, \mathbf{b}_{l+1}) &= \mathbf{a}_l + \frac{h_l}{\gamma}\mathbf{K}[\mathbf{a}_l]\mathbf{b}_{l+1},\\ \hat{\mathbf{q}}_{\mathrm{SE}}(\mathbf{a}_l, \mathbf{b}_{l+1}) &= \mathbf{b}_{l+1} + \frac{h_l}{2\gamma}\partial_{\mathbf{z}}\left(\mathbf{b}_{l+1}^{\top}\mathbf{K}[\mathbf{a}_l]\mathbf{b}_{l+1}\right).\end{aligned} \tag{3.15}$$

The desirable property here is that symplectic Euler effectively decouples the solution operators so that computing

$$\begin{pmatrix}\mathbf{z}_{t_{l+1}}\\ \mathbf{p}_{t_l}\end{pmatrix} = \begin{pmatrix}\hat{\mathbf{d}}_{\mathrm{SE}}(\mathbf{a}_l, \mathbf{b}_{l+1})\\ \hat{\mathbf{q}}_{\mathrm{SE}}(\mathbf{a}_l, \mathbf{b}_{l+1})\end{pmatrix} \tag{3.16}$$

is a matter of a single forward and backward propagation. This, in turn, decouples the Jacobian computation, namely $\partial_{\mathbf{a}}\hat{\mathbf{d}}_{\mathrm{SE}}(\mathbf{a}_0, \mathbf{b}_1)$, $\partial_{\mathbf{b}}\hat{\mathbf{d}}_{\mathrm{SE}}(\mathbf{a}_0, \mathbf{b}_1)$, $\partial_{\mathbf{a}}\hat{\mathbf{q}}_{\mathrm{SE}}(\mathbf{a}_0, \mathbf{b}_1)$ and $\partial_{\mathbf{b}}\hat{\mathbf{q}}_{\mathrm{SE}}(\mathbf{a}_0, \mathbf{b}_1)$ can be separately computed. Recall now from the proof of Proposition 2.1.8 that, once the symplectic Euler discretization scheme is employed the reduced (with respect to the controls) Lagrangain reads

$$\mathcal{L}(\vec{\mathbf{z}}, \vec{\mathbf{p}}) := \frac{1}{2N}\mathcal{E}(\mathbf{z}_L, \mathbf{y}) + \frac{h}{2\gamma}\sum_{l=0}^{L-1}\mathbf{p}_{l+1}^{\top}\mathbf{K}[\mathbf{z}_l]\mathbf{p}_{l+1} + \sum_{l=0}^{L-1}\mathbf{p}_{l+1}^{\top}(\mathbf{z}_{l+1} - \mathbf{z}_l) + \mathbf{p}_0^{\top}(\mathbf{z}_0 - \mathbf{x}). \tag{3.17}$$

Then, it is easy to see that $\hat{\mathbf{f}}_{\mathrm{SE}}(\vec{\mathbf{a}}, \vec{\mathbf{b}})$ is (up to a reordering of the components) the gradient of $\mathcal{L}(\vec{\mathbf{a}}, \vec{\mathbf{b}})$, namely

$$\partial_{\vec{\mathbf{z}}}\mathcal{L}(\vec{\mathbf{a}}, \vec{\mathbf{b}}) = \hat{\mathbf{f}}_{\mathrm{SE}}(\vec{\mathbf{a}}, \vec{\mathbf{b}})_{dN(L+1)+1:2dN(L+1)}, \qquad \partial_{\vec{\mathbf{p}}}\mathcal{L}(\vec{\mathbf{a}}, \vec{\mathbf{b}}) = \hat{\mathbf{f}}_{\mathrm{SE}}(\vec{\mathbf{a}}, \vec{\mathbf{b}})_{1:dN(L+1)}. \tag{3.18}$$

Then, instead of the standard Newton iteration, we consider the damped Newton system

$$\left(\hat{\mathbf{J}}_{\mathrm{SE}}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}) + \rho^{(k)}\begin{pmatrix}0 & -I\\ I & 0\end{pmatrix}\right)\begin{pmatrix}\Delta\vec{\mathbf{a}}^{(k)}\\ \Delta\vec{\mathbf{b}}^{(k)}\end{pmatrix} = -\hat{\mathbf{f}}_{\mathrm{SE}}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}), \tag{3.19}$$

$\rho^{(k)} > 0$ being the relevant damping parameter. Note that the minus sign is a consequence of the fact that, ideally for $\rho \to \infty$, we want to perform a gradient descent step in the state variables, while a gradient ascent step for the adjoint variables. Experimentally, it turns out the correct scaling (with respect to the iteration k) for $\rho^{(k)}$ is to make it proportional to the norm of the residual, namely

$$\rho^{(k)} = \rho^{(0)}\left\|\hat{\mathbf{f}}_{\mathrm{SE}}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)})\right\|_2. \tag{3.20}$$

Instead, in order to adaptively choose the correct amount of damping in ω we implement a a backtracking line-search based on the Armijo–Goldstein (Armijo [1966]) condition. The

very simple idea behind this scheme is to make sure at each iteration that the updated solution achieves a sufficiently large amount of decrease in a performance metric, which here we take to be $\left\|\hat{\mathbf{f}}_{\mathrm{SE}}(\vec{\mathbf{a}},\vec{\mathbf{b}})\right\|_2^2$, namely we require

$$\begin{aligned}&\left\|\hat{\mathbf{f}}_{\mathrm{SE}}(\vec{\mathbf{a}}^{(k)}+\omega\Delta\vec{\mathbf{a}}^{(k)},\vec{\mathbf{b}}^{(k)}+\omega\Delta\vec{\mathbf{b}}^{(k)})\right\|_2^2\\ <&\left\|\hat{\mathbf{f}}_{\mathrm{SE}}(\vec{\mathbf{a}}^{(k)},\vec{\mathbf{b}}^{(k)})\right\|_2^2+2\omega\tau\hat{\mathbf{f}}_{\mathrm{SE}}(\vec{\mathbf{a}}^{(k)},\vec{\mathbf{b}}^{(k)})^\top\hat{\mathbf{J}}_{\mathrm{SE}}(\vec{\mathbf{a}}^{(k)},\vec{\mathbf{b}}^{(k)})\begin{pmatrix}\Delta\vec{\mathbf{a}}^{(k)}\\ \Delta\vec{\mathbf{b}}^{(k)}\end{pmatrix}\end{aligned}\tag{3.21}$$

for the update to be accepted. Here

$$\hat{\mathbf{f}}_{\mathrm{SE}}(\vec{\mathbf{a}}^{(k)},\vec{\mathbf{b}}^{(k)})^\top\hat{\mathbf{J}}_{\mathrm{SE}}(\vec{\mathbf{a}}^{(k)},\vec{\mathbf{b}}^{(k)})\begin{pmatrix}\Delta\vec{\mathbf{a}}^{(k)}\\ \Delta\vec{\mathbf{b}}^{(k)}\end{pmatrix}\tag{3.22}$$

measures the expected amount of decrease of $\hat{\mathbf{f}}_{\mathrm{SE}}$ moving along descent direction $\left(\Delta\vec{\mathbf{a}}^{(k)};\Delta\vec{\mathbf{b}}^{(k)}\right)$, while τ represents a scaling parameter, usually set to $\tau=10^{-4}$. Each time the Armijo–Goldstein condition is not satisfied the damping amount is shrunken by a factor of $\alpha\in(0,1)$ (hence the term "backtracking"). The resulting algorithm is summarized[1] in Algorithm 3.1.2.

The interesting property of this optimization scheme lies in the fact that the training is parallelized across the layers. The drawback, however, is in the computational price for each iteration which involves the solution of a linear system of size $\mathcal{O}(dNL)$. This must be compared with the linear systems for the preconditioned gradient method which are instead of size $\mathcal{O}(dN)$. The wish to control the computational complexity with respect to the number of layers motivates the next proposed training algorithm.

[1]Note that we use $\mathbf{f}$ and $\mathbf{J}$ instead of $\hat{\mathbf{f}}_{\mathrm{SE}}$ and $\hat{\mathbf{J}}_{\mathrm{SE}}$ for ease of notation.

Algorithm 3.1.2: Time-parallel (TP) algorithm

Input: A sample $\mathbf{x}$, observations $\mathbf{y}$, initial solutions $\vec{\mathbf{a}}^{(0)}, \vec{\mathbf{b}}^{(0)}$, kernel $\mathcal{K}$, maximum iterations M, tolerance tol, damping $\rho^{(0)}$, line search parameters ω, τ and α, regression basis functions $f_1, \ldots, f_m$.

Output: Approximation function $\hat{g}$.

Set iteration counter $k \leftarrow 0$ and initialize residual $r \leftarrow \left\| \mathbf{f}(\vec{\mathbf{a}}^{(0)}, \vec{\mathbf{b}}^{(0)}) \right\|_2$;

while $r > tol$ *and* $k \leq M$ **do**

  Compute $\mathbf{J}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)})$, $\mathbf{f}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)})$ and $\rho^{(k)} = \rho^{(0)} \left\| \mathbf{f}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}) \right\|_2$;

  Compute regression parameters $c^{(k)} = \arg\min_{c \in \mathbb{R}^m} \left\| \mathbf{V}[\mathbf{z}_L^{(k)}]c - \mathbf{y} \right\|_2^2$;

  Compute descent direction by solving

$$\left(\mathbf{J}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}) + \rho^{(k)} \begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix} \right) \begin{pmatrix} \Delta\vec{\mathbf{a}}^{(k)} \\ \Delta\vec{\mathbf{b}}^{(k)} \end{pmatrix} = -\mathbf{f}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}); \tag{3.23}$$

  Compute expected decrease

$$m \leftarrow 2\mathbf{f}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)})^\top \mathbf{J}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}) \begin{pmatrix} \Delta\vec{\mathbf{a}}^{(k)} \\ \Delta\vec{\mathbf{b}}^{(k)} \end{pmatrix} \tag{3.24}$$

  of squared residual ;

  Set $t \leftarrow \tau m$, line search counter $s \leftarrow 0$ and initialize $\omega^{(0)} \leftarrow \omega$;

  while

$$\left\| \mathbf{f}(\vec{\mathbf{a}}^{(k)} + \omega^{(j)}\Delta\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)} + \omega^{(j)}\Delta\vec{\mathbf{b}}^{(k)}) \right\|_2^2 > \left\| \mathbf{f}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}) \right\|_2^2 + \omega^{(j)} t \tag{3.25}$$

  do

    Set $\omega^{(j+1)} \leftarrow \alpha\omega^{(j)}$;

    Update counter $j \leftarrow j + 1$;

  end

  Update solution

$$\begin{pmatrix} \vec{\mathbf{a}}^{(k+1)} \\ \vec{\mathbf{b}}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \vec{\mathbf{a}}^{(k)} \\ \vec{\mathbf{b}}^{(k)} \end{pmatrix} + \omega^{(j)} \begin{pmatrix} \Delta\vec{\mathbf{a}}^{(k)} \\ \Delta\vec{\mathbf{b}}^{(k)} \end{pmatrix}; \tag{3.26}$$

  Update residual $r \leftarrow \left\| \mathbf{f}(\vec{\mathbf{a}}^{(k+1)}, \vec{\mathbf{b}}^{(k+1)}) \right\|_2$ and counter $k \leftarrow k + 1$;

end

Compute the surrogate function $\hat{g}$ from $\vec{\mathbf{a}}^{(k)}$, $\vec{\mathbf{b}}^{(k)}$ and $c^{(k)}$.

3.1.3 Time-parallel multi-grid training algorithm

Here we introduce a training method which exploits two levels of time discretization in order to improve the scalability of Algorithm 3.1.2 with respect to the number of layers

L. To this end, introduce a coarse uniform grid $0 = T_0 < T_1 \ldots < T_{L_{\text{out}}} = 1$ of granularity $h_{\text{out}} = 1/L_{\text{out}}$. Then, consider the same nonlinear system introduced in the previous section

$$\mathbf{f}(\vec{\mathbf{a}}, \vec{\mathbf{b}}) = \begin{pmatrix} \mathbf{a}_0 - \mathbf{x} \\ \mathbf{a}_1 - \mathbf{d}(\mathbf{a}_0, \mathbf{b}_1) \\ \vdots \\ \mathbf{a}_{L_{\text{out}}} - \mathbf{d}(\mathbf{a}_{L_{\text{out}}-1}, \mathbf{b}_{L_{\text{out}}}) \\ \mathbf{b}_0 - \mathbf{q}(\mathbf{a}_0, \mathbf{b}_1) \\ \vdots \\ \mathbf{b}_{L_{\text{out}}-1} - \mathbf{q}(\mathbf{a}_{L_{\text{out}}-1}, \mathbf{b}_{L_{\text{out}}}) \\ \mathbf{b}_{L_{\text{out}}} + \frac{1}{2N}\partial_{\mathbf{z}}\mathcal{E}(\mathbf{a}_{L_{\text{out}}}, \mathbf{y}) \end{pmatrix} \tag{3.27}$$

involving the coupled forward-backward solution operators $\mathbf{d}$ and $\mathbf{q}$. Following Gander et al. [2020], we propose to approximate

$$\mathbf{J}(\vec{\mathbf{a}}, \vec{\mathbf{b}}) \approx \hat{\mathbf{J}}_{\text{SE}}(\vec{\mathbf{a}}, \vec{\mathbf{b}}), \tag{3.28}$$

$\hat{\mathbf{J}}_{\text{SE}}(\vec{\mathbf{a}}, \vec{\mathbf{b}})$ being the Jacobian when the symplectic Euler discretization scheme is employed on the coarse grid. Essentially, we approximate the true Jacobian with one of lower cost, where in each subinterval of the coarse grid the problem is solved in one step with symplectic Euler. Note that, as we remarked in the previous section, the computation of this Jacobian is easy since employing $\hat{\mathbf{d}}_{\text{SE}}$ and $\hat{\mathbf{q}}_{\text{SE}}$ effectively decouples the solution operators. In order to make the problem fully discrete, consider the i-th coarse interval $[T_{i-1}, T_i]$ and introduce a fine grid $T_{i-1} = t_{i,0} < t_{i,1} \ldots < t_{i,L_{\text{in}}} = T_i$ of granularity $h_{\text{in}} = 1/L_{\text{in}}$. Then, we may solve the local problems on the fine grid with, e.g., Algorithm 3.1.2. The resulting procedure is reported[2] in Algorithm 3.1.3. We remark that, with this setup, the computation of the local problems may be parallelized. Furthermore, the latter involve solving linear systems of size $\mathcal{O}(dNL_{\text{in}})$, while the outer problem on the coarse grid requires the solution of linear systems of size $\mathcal{O}(dNL_{\text{out}})$. This, then, enables a better control on the computational complexity of the algorithm with respect to the total layers in the network $L = L_{\text{out}}L_{\text{in}}$.

[2]For ease of notation we keep the solution to the local problem implicit and focus only on the outer parameters.

Algorithm 3.1.3: Time-parallel multi-grid (TPMG) algorithm

Input: A sample $\mathbf{x}$, observations $\mathbf{y}$, number of coarse grid layers L_{out}, number of coarse grid layers L_{in}, initial solutions $\vec{\mathbf{a}}^{(0)}, \vec{\mathbf{b}}^{(0)}$, kernel $\mathcal{K}$, maximum iterations M, tolerance tol, line search parameters ω, τ and α, regression basis functions $f_1, \ldots, f_m$.

Output: Approximation function $\hat{g}$.

Set iteration counter $k \leftarrow 0$ and initialize residual $r \leftarrow \left\| \mathbf{f}(\vec{\mathbf{a}}^{(0)}, \vec{\mathbf{b}}^{(0)}) \right\|_2$;

while $r > tol$ *and* $k \leq M$ **do**

  Solve local problems on time intervals $[T_{i-1}, T_i]$, $i = 1, \ldots, L_{\text{out}}$;

  Compute $\mathbf{f}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)})$;

  Compute approximated Jacobian $\mathbf{J}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)})$;

  Compute regression parameters $c^{(k)} = \arg\min_{c \in \mathbb{R}^m} \left\| \mathbf{V}[\mathbf{z}^{(k)}_{L_{\text{out}}}]c - \mathbf{y} \right\|_2^2$;

  Compute descent direction by solving

$$\mathbf{J}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}) \begin{pmatrix} \Delta\vec{\mathbf{a}}^{(k)} \\ \Delta\vec{\mathbf{b}}^{(k)} \end{pmatrix} = -\mathbf{f}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}); \tag{3.29}$$

  Compute approximated expected decrease

$$m \leftarrow 2\mathbf{f}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)})^\top \mathbf{J}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}) \begin{pmatrix} \Delta\vec{\mathbf{a}}^{(k)} \\ \Delta\vec{\mathbf{b}}^{(k)} \end{pmatrix} \tag{3.30}$$

  of squared residual ;

  Set $t \leftarrow \tau m$, line search counter $s \leftarrow 0$ and initialize $\omega^{(0)} \leftarrow \omega$;

  while

$$\left\| \mathbf{f}(\vec{\mathbf{a}}^{(k)} + \omega^{(j)}\Delta\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)} + \omega^{(j)}\Delta\vec{\mathbf{b}}^{(k)}) \right\|_2^2 > \left\| \mathbf{f}(\vec{\mathbf{a}}^{(k)}, \vec{\mathbf{b}}^{(k)}) \right\|_2^2 + \omega^{(j)} t \tag{3.31}$$

  do

    Set $\omega^{(j+1)} \leftarrow \alpha\omega^{(j)}$;

    Update counter $j \leftarrow j + 1$;

  end

  Update solution

$$\begin{pmatrix} \vec{\mathbf{a}}^{(k+1)} \\ \vec{\mathbf{b}}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \vec{\mathbf{a}}^{(k)} \\ \vec{\mathbf{b}}^{(k)} \end{pmatrix} + \omega^{(j)} \begin{pmatrix} \Delta\vec{\mathbf{a}}^{(k)} \\ \Delta\vec{\mathbf{b}}^{(k)} \end{pmatrix}; \tag{3.32}$$

  Update residual $r \leftarrow \left\| \mathbf{f}(\vec{\mathbf{a}}^{(k+1)}, \vec{\mathbf{b}}^{(k+1)}) \right\|_2$ and counter $k \leftarrow k + 1$;

end

Compute the surrogate function $\hat{g}$ from $\vec{\mathbf{a}}^{(k)}$, the solutions to the local problem at iteration k and $\vec{\mathbf{b}}^{(k)}$ and $c^{(k)}$.

3.1.4 Riemannian gradient descent

Let us now introduce the training algorithm inspired by the Riemannian gradient flow introduced in Section 2.2.3. In particular, we turn to the time discretization. Typically, one considers Riemannian gradient descent, instead of the ideal gradient flow, for practical applications. The problem of discretizing time in Riemannian optimization is that the updates lead the solution to escape the manifold the optimization takes palace on (in our case the manifold $\mathcal{M}$ of diffeomorphisms). The solution of Riemannian gradient descent is to exploit so-called retractions, which essentially at each step project the updated solution back onto the manifold. We refer the reader to Chapter 4 of Boumal [2022] for further details. In our setting, however, staying in the manifold during optimization entails the solution of an ODE integration problem which, clearly, can not happen and necessarily needs to be discretized. With this in mind, then, our strategy is to allow for the optimization algorithm to take the solution out of the manifold, and regard the continuous-time problem just as an idealized version of the discrete one. The hope is that the two solutions will not differ by much. With the notation of Section 2.2.3, this means that we consider the following discrete dynamics

$$\phi_{l+1} = \phi_l - h_l \mathrm{grad}(\mathcal{J}^N)(\phi_l). \tag{3.33}$$

Note that we will refer to this scheme, though somewhat improperly, as Riemannian gradient descent nonetheless. The iteration index l should remark that, in this setting, the layers and optimization steps coincide. In a way, we think of the neural network itself as an optimizer which, through its layers, modifies the inputs in order to minimize the loss functional. The choice of step size is left up to a backtracking line-search strategy. The latter can be seen as an adaptive choice of the time discretization granularity. More precisely, we choose an initial value for h_l and shrink it by a factor of α until

$$\mathcal{J}^N(\phi_l - h_l \mathrm{grad}(\mathcal{J}^N)(\phi_l)) < \mathcal{J}^N(\phi_l) - h_l \tau \left\| \mathrm{grad}(\mathcal{J}^N)(\phi_l) \right\|_{\mathcal{H}}^2, \tag{3.34}$$

τ being the usual scaling parameter set to 10^{-4}. Here $\left\|\mathrm{grad}(\mathcal{J}^N)(\phi_l)\right\|_{\mathcal{H}}^2$ measures the expected decrease in $\mathcal{J}^N$ when moving along direction $\mathrm{grad}(\mathcal{J}^N)(\phi_l)$. Note that, even though we have presented Riemannian gradient descent focusing on the maps ϕ_l, as explained in Section 2.2.3 we only need to track the trajectories of the training particles, so that all of the above involves finite-dimensional computations only. The complete algorithm is shown in Algorithm 3.1.4.

The appeal of this algorithm stems from its simplicity. The controls are decoupled across layers and, hence, each iteration is extremely cheap to compute with respect to the previous proposed algorithms. Indeed, it involves a simple gradient computation and a matrix-vector multiplication of size $\mathcal{O}(dN)$. The drawback comes from the fact that we do not control the necessary number of layers of the network in order to achieve a sufficient amount of decrease in the objective functional. Then, if convergence is slow, this may result in networks with many layers which are expensive to evaluate. On a positive note, this scheme allows for a very cheap validation strategy for the number of optimization steps. Indeed, one can employ Algorithm 3.1.4 with some number of iterations L to get

$(\mathbf{z}_0, \ldots, \mathbf{z}_{L-1})$, $(\boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_{L-1})$, $(c_0, \ldots, c_L)$ and, hence, the maps $\phi_1, \ldots, \phi_L$. Then, given a validation sample $\mathbf{y}^{\text{val}} := (g(x_{\text{val}}^{(1)}), \ldots, g(x_{\text{val}}^{(N_{\text{val}})}))$ of size N_{val}, it is sufficient to compute

$$\mathbf{z}_l^{\text{val}} = (\phi_l(x_{\text{val}}^{(1)}), \ldots, \phi_l(x_{\text{val}}^{(N_{\text{val}})})), \tag{3.35}$$

and the associated loss

$$\left\| \mathbf{V}[\mathbf{z}_l^{\text{val}}] c_l - \mathbf{y}^{\text{val}} \right\|_2^2, \tag{3.36}$$

for each layer $l = 1, \ldots, L$. This allows to choose the layer $L^\star$ which achieves the smallest validation loss.

Algorithm 3.1.4: Riemannian gradient descent (RGD) algorithm

Input: A sample $\mathbf{x}$, observations $\mathbf{y}$, kernel $\mathcal{K}$, maximum number of layers L, line search parameters h, τ and α, regression basis functions $f_1, \ldots, f_m$.

Output: Approximation function $\hat{g}$.

Set $\mathbf{z}_0 = \mathbf{x}$;
for $l = 0, \ldots, L-1$ **do**
  Compute the Euclidean gradient $\boldsymbol{\beta}_l = \partial_{\mathbf{z}} \hat{\mathcal{J}}^N(\mathbf{z}_l)$;
  Compute the Riemannian gradient $\mathbf{v}_l = \mathbf{K}[\mathbf{z}_l]\boldsymbol{\beta}_l$;
  Compute regression parameters $c_l = \arg\min_{c \in \mathbb{R}^m} \|\mathbf{V}[\mathbf{z}_l]c - \mathbf{y}\|_2^2$;
  Set $h_l = h$;
  while $\hat{\mathcal{J}}^N(\mathbf{z}_l - h_l \mathbf{v}_l) > \hat{\mathcal{J}}^N(\mathbf{z}_l) - h_l \tau \left\| \text{grad}(\mathcal{J}^N)(\phi_l) \right\|_{\mathcal{H}}^2$ **do**
    Update $h_l \leftarrow \alpha h_l$;
  end
  Set $\mathbf{z}_{l+1} = \mathbf{z}_l - h_l \mathbf{v}_l$;
end
Compute regression parameters $c_L = \arg\min_{c \in \mathbb{R}^m} \|\mathbf{V}[\mathbf{z}_L]c - \mathbf{y}\|_2^2$;
Compute the surrogate function $\hat{g}$ from $(\mathbf{z}_0, \ldots, \mathbf{z}_{L-1})$, $(\boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_{L-1})$ and c_{L-1}.

In order to further reduce the cost per iteration, we propose a stochastic Riemannian optimization strategy as well. More specifically, we consider a mini-batch version of the Riemannian gradient descent algorithm just introduced. The essence of a mini-batch approach is to, at each iteration, randomly split the data into B subsets (the mini-batches) and then perform a gradient descent step where the loss functional is computed with the mini-batch data only, for each of the mini-batches. More in detail, let $\sigma : \{1, \ldots, N\} \to \{1, \ldots, N\}$ denote a random permutation. Then, fix iteration index l (which we drop for notational convenience), denote $N_B := N/B$ the batch size and consider the mini-batch of observations

$$\mathbf{z}^{(b)} := (z^{(\sigma(bN_B+1))}; \ldots; z^{(\sigma((b+1)N_B))}), \quad \mathbf{y}^{(b)} := (y^{(\sigma(bN_B+1))}; \ldots; y^{(\sigma((b+1)N_B))}), \tag{3.37}$$

for $b = 0, \ldots, B-1$. Furthermore, define the mini-batch discrete functional

$$\hat{\mathcal{J}}^{N_B}(\mathbf{z}^{(b)}, \mathbf{y}^{(b)}) = \min_{f \in \mathcal{V}_m} \left\{ \frac{1}{N_B} \sum_{i=1}^{N_B} \left(f(z^{(\sigma(bN_B+i))}) - y^{(\sigma(bN_B+i))} \right)^2 \right\}. \tag{3.38}$$

Then, instead of one full gradient step, we take B mini-batch gradient steps, namely we update the particle positions as

$$\mathbf{z}_{b+1} = \mathbf{z}_b - h_b \mathbf{K}[\mathbf{z}_b, \mathbf{z}_b^{(b)}]\boldsymbol{\beta}^{(b)}, \quad b = 0, \dots, B-1. \tag{3.39}$$

where

$$\boldsymbol{\beta}^{(b)} := \partial_{\mathbf{z}} \hat{\mathcal{J}}^{N_B}(\mathbf{z}_b^{(b)}, \mathbf{y}^{(b)}) \tag{3.40}$$

Essentially we substitute one big gradient step with B mini-batch updates. The resulting scheme is presented in Algorithm 3.1.5. Note that we will refer to it, though improperly, as stochastic Riemannian gradient descent. We want to stress the fact that the result of employing Algorithm 3.1.5 with L steps and B mini-batches is a network with LB total layers. However, due to the fact that each control is in a N_B dimensional subspace (as opposed to an N dimensional one), the cost to evaluate the SRGD model is actually of the same order of an RGD model with L layers.

Algorithm 3.1.5: Stochastic Riemannian gradient descent (SRGD) algorithm

Input: A sample $\mathbf{x}$, observations $\mathbf{y}$, kernel $\mathcal{K}$, maximum number of layers L, batch size B line search parameters h, τ and α, regression basis functions $f_1, \dots, f_m$.

Output: Approximation function $\hat{g}$.

Set $\mathbf{z}_{0,0} = \mathbf{x}$;
for $l = 0, \dots, L-1$ **do**
  Compute regression parameters $c_l = \arg\min_{c \in \mathbb{R}^m} \|\mathbf{V}[\mathbf{z}_{l,0}]c - \mathbf{y}\|_2^2$;
  Compute a random permutation $\sigma : \{1, \dots, N\} \to \{1, \dots, N\}$;
  for $b = 0, \dots, B-1$ **do**
    Compute the Euclidean mini-batch gradient $\boldsymbol{\beta}_l^{(b)} = \partial_{\mathbf{z}} \hat{\mathcal{J}}^{N_B}(\mathbf{z}_{l,b}^{(b)}, \mathbf{y}^{(b)})$;
    Compute the Riemannian mini-batch gradient $\mathbf{v}_{l,b} = \mathbf{K}[\mathbf{z}_{l,b}, \mathbf{z}_{l,b}^{(b)}]\boldsymbol{\beta}_l^{(b)}$;
    Set $h_{l,b} = h$;
    while
     $\hat{\mathcal{J}}^{N_B}((\mathbf{z}_{l,b} - h_{l,b}\mathbf{v}_{l,b})^{(b)}, \mathbf{y}^{(b)}) > \hat{\mathcal{J}}^{N_B}(\mathbf{z}_{l,b}^{(b)}, \mathbf{y}^{(b)}) - h_{l,b}\tau \left(\boldsymbol{\beta}_l^{(b)}\right)^\top \mathbf{K}[\mathbf{z}_{l,b}^{(b)}]\boldsymbol{\beta}_l^{(b)}$
     do
      Update $h_{l,b} \leftarrow \alpha h_{l,b}$;
    end
    Set $\mathbf{z}_{l,b+1} = \mathbf{z}_{l,b} - h_{l,b}\mathbf{v}_{l,b}$;
  end
  Set $\mathbf{z}_{l+1,0} = \mathbf{z}_{l,B}$;
end
Compute regression parameters $c_L = \arg\min_{c \in \mathbb{R}^m} \|\mathbf{V}[\mathbf{z}_{L,0}]c - \mathbf{y}\|_2^2$;
Compute the surrogate function $\hat{g}$ from $(\mathbf{z}_{0,0}, \dots, \mathbf{z}_{L-1,B-1})$, $(\boldsymbol{\beta}_0^{(0)}, \dots, \boldsymbol{\beta}_{L-1}^{(B-1)})$ and $c_0, \dots, c_L$.

3.1.5 Riemannian damped Newton

In this section we specify the damped Newton method introduced in Section 2.2.4. The rationale for discretizing time is the same as the one for Riemannian gradient descent. In particular, all the observations about letting the solution get out of the manifold still apply. Let us remark that the magnitude of damping is chosen to be proportional to the magnitude of the Riemannian gradient, namely

$$\omega_l = \omega_0 \left\| \operatorname{grad}(\mathcal{J}^N)(\phi_l) \right\|_{\mathcal{H}}. \tag{3.41}$$

The procedure is detailed in Algorithm 3.1.6 with the notation of Section 2.2.4. Note that the same validation strategy proposed for Riemannian gradient descent allows for the computation of the optimal number of optimization steps $L^\star$.

Algorithm 3.1.6: Riemannian damped Newton (RDN) algorithm

Input: A sample $\mathbf{x}$, observations $\mathbf{y}$, kernel $\mathcal{K}$, maximum number of layers L, initail damping ω_0, line search parameters h, τ and α, regression basis functions $f_1, \dots, f_m$.

Output: Approximation function $\hat{g}$.

Set $\mathbf{z}_0 = \mathbf{x}$;

for $l = 0, \dots, L-1$ **do**

  Compute the Euclidean gradient $\mathbf{p}_l = \partial_{\mathbf{z}} \hat{\mathcal{J}}^N(\mathbf{z}_l)$;

  Compute the Riemannian gradient $\mathbf{g}_l = \mathbf{K}[\mathbf{z}_l]\mathbf{p}_l$ and $\omega_l = \omega_0 \mathbf{p}_l^\top \mathbf{g}_l$;

  Compute regression parameters $c_l = \arg\min_{c \in \mathbb{R}^m} \|\mathbf{V}[\mathbf{z}_l]c - \mathbf{y}\|_2^2$;

  Solve

$$(\mathbf{K}'[\mathbf{z}_l, \mathbf{p}_l] + \mathbf{K}[\mathbf{z}_l]\mathbf{H}[\mathbf{z}_l] + \omega_l I)\mathbf{v}_l = -\mathbf{g}_l; \tag{3.42}$$

  Compute

$$\mathbf{v}_l' = -\frac{1}{\omega_l}\left((\mathbf{K}''[\mathbf{z}_l, \mathbf{p}_l] + \mathbf{K}'[\mathbf{z}_l, \mathbf{p}_l]^\top \mathbf{H}[\mathbf{z}_l])\mathbf{v}_l + \mathbf{K}'[\mathbf{z}_l, \mathbf{p}_l]^\top \mathbf{p}_l \right); \tag{3.43}$$

  Solve

$$\begin{pmatrix} \mathbf{K}[\mathbf{z}_l] & \mathbf{K}'[\mathbf{z}_l, \mathbf{p}_l] \\ \mathbf{K}'[\mathbf{z}_l, \mathbf{p}_l]^\top & \mathbf{K}''[\mathbf{z}_l, \mathbf{p}_l] \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_l^1 \\ \boldsymbol{\beta}_l^2 \end{pmatrix} = \begin{pmatrix} \mathbf{v}_l \\ \mathbf{v}_l' \end{pmatrix}; \tag{3.44}$$

  Set $h_l = h$;

  while $\hat{\mathcal{J}}^N(\mathbf{z}_l - h_l \mathbf{v}_l) > \hat{\mathcal{J}}^N(\mathbf{z}_l) - h_l \tau \left((\boldsymbol{\beta}_l^1)^\top \mathbf{K}[\mathbf{z}_l]\boldsymbol{\beta}_l^1 + (\boldsymbol{\beta}_l^1)^\top \mathbf{K}'[\mathbf{z}_l, \mathbf{p}_l]\boldsymbol{\beta}_l^2 + (\boldsymbol{\beta}_l^2)^\top \mathbf{K}''[\mathbf{z}_l, \mathbf{p}_l]\boldsymbol{\beta}_l^2 \right)$ **do**

    Update $h_l \leftarrow \alpha h_l$;

  end

  Set $\mathbf{z}_{l+1} = \mathbf{z}_l - h_l \mathbf{v}_l$;

end

Compute regression parameters $c_L = \arg\min_{c \in \mathbb{R}^m} \|\mathbf{V}[\mathbf{z}_L]c - \mathbf{y}\|_2^2$;

Compute the surrogate function $\hat{g}$ from $(\mathbf{z}_0, \dots, \mathbf{z}_{L-1})$, $(\mathbf{p}_0, \dots, \mathbf{p}_{L-1})$, $(\boldsymbol{\beta}_0^1, \dots, \boldsymbol{\beta}_{L-1}^1)$, $(\boldsymbol{\beta}_0^2, \dots, \boldsymbol{\beta}_{L-1}^2)$ and c_L.

3.2 Numerical results

In this section we turn to the numerical exploration of the presented training algorithms with simulated regression tasks. In particular, we employ two target functions, namely an arctangent and Gaussian bell, which act as prototypes for a discontinuity and local feature respectively. Note that we can group the presented algorithms with respect to the problem they solve, i.e. the regularized one of Section 2.1 and the unregularized one of Section 2.2. In order to make clear which algorithms/optimization problem we refer to, let us denote the former as Optimal Control (OC) algorithms/optimization problem and the latter as Riemannian optimization (RO) algorithms/optimization problem. Furthermore, note that in the following we are going to use $z_t(x)$ and $\phi_t(x)$ synonymously to denote the diffeomorphism at time t applied to point $x \in \mathbb{R}^d$. Finally, in all our experiments we take as initial condition for the diffeomorphism the identity mapping.

3.2.1 The prototype of a discontinuity

Let us analyze the presented algorithms and the resulting models in the context of the following example. We consider multidimensional arctangent functions, namely

$$g_{d,\lambda}(x) = \arctan(\lambda^\top x), \qquad x \in [-1,1]^d, \tag{3.45}$$

where $\lambda \in \mathbb{R}^d$ is a parameter controlling the angle and steepness of the slope at the origin. The latter, then, controls the complexity of the target function. Indeed, letting $\|\lambda\|_2 \to \infty$ makes the function approximate a discontinuity, as shown in Figure 3.1. For

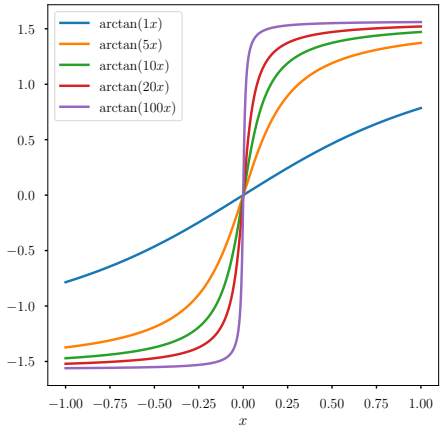

Figure 3.1: Arctangent function $g_{1,\lambda}$ dimensions for different values of $\lambda \in \mathbb{R}$. As λ increases, $\arctan(\lambda x)$ approaches a discontinuity at $x = 0$.

these experiments we employ the RKHS $\mathcal{H}$ generated by the simple separable Gaussian kernel, namely

$$\mathcal{K}(x, x') = e^{r\|x-x'\|_2^2} I, \tag{3.46}$$

where $r > 0$ is the shape parameter. We then approximate $g_{d,\lambda}$ via $y^{(i)} = g_{d,\lambda}(x^{(i)})$, $i = 1,2,\ldots,N$ noiseless observations in $[-1,1]^d$. The latter are obtained with a Quasi-Monte

Carlo design. In particular, we choose to work with the Halton sequence. Furthermore, we mostly employ two kinds of regression space $\mathcal{V}_m$, which we denote $\mathcal{V}^1$ and $\mathcal{V}^{10}$. The former denotes the space corresponding to linear regression, namely

$$\mathcal{V}^1 = \{1, x_1, \dots, x_d\}, \tag{3.47}$$

while the latter denotes the space $\mathcal{V}^1$ plus all monomials of degree less or equal to 10 in the last component, namely

$$\mathcal{V}^2 = \mathcal{V}^1 \cup \left\{x_d^2, \dots, x_d^{10}\right\}. \tag{3.48}$$

Training analysis

Let us now analyze the performance of the presented training methods. The first question we want to address is if the proposed methodologies outperform the preconditioned gradient method, given that some of them exploit second order information. We test this on $g_{1,10}$ for simplicity and report results in Figure 3.2. The hyperparameters setup used for this experiment is instead reported in Table 3.1.

Algorithm	N	L	r	γ	L_{in}	L_{out}
PGD	10	25	5	10^{-4}	-	-
TP	10	25	5	10^{-4}	-	-
TPMG	10	25	5	10^{-4}	5	5
RGD	10	1000	5	-	-	-
RDN	10	100	5	-	-	-

Table 3.1: Network hyperparameters used to approximate the target function $g_{1,10}$ for the various algorithms.

Let us first detail the way the TPMG algorithm is employed in practice for this experiment. First, we solve the problem on the coarse grid (i.e. with L_{out} layers) with the TP algorithm. Note that we also record this residual relevant to the solution of the coarse problem in the left plot of Figure 3.2 (green line, up to around iteration 25). Then, we use this as the initialization of the TPMG algorithm, while also fixing the regression parameters (in some sense the TPMG adds the missing layers). This corresponds to the rapid ascent in the residual (green line, at around iteration 25) since the one relevant to the TPMG algorithm working on L layers is being recorded. The first observation is that, among the OC algorithms, the second order methods improve upon the PGD in terms of speed at which a good enough solution is found. This is particularly evident from the plot of the residuals which shows that, within 500 iterations, it remains at around 10^{-1} for the PGD, while it converges rapidly to zero both for the TP and TPMG algorithms.

The Riemannian optimization algorithms show a similar trend, namely the second order information is beneficial both in terms of speed of convergence, as expected, and in terms generalization power of the computed solution. Furthermore, it is interesting to notice that, due to the lack of explicit regularization, these algorithms are in principle

able to make the test error arbitrarily small. This, especially when a richer regression space is employed, appears to improve (sometimes drastically, as in the case of the RDN) the performance as seen by the reported test errors.

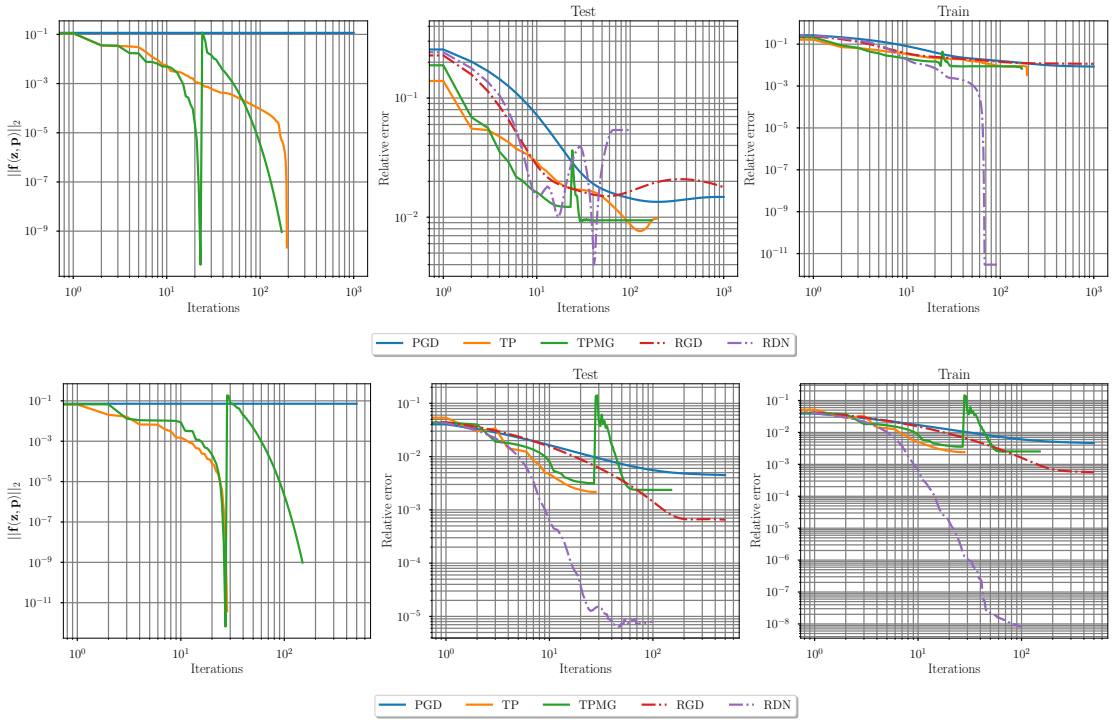

Figure 3.2: Training performance of the presented methodologies for the approximation of $g_{1,10}$ with $N = 10$ observations. The employed regression spaces are $\mathcal{V}^1$ (top) and $\mathcal{V}^2$ (bottom).

This, of course, suggests that we should further decrease the regularization in the explicitly regularized problem the OC algorithms solve. However, these are ill-conditioned with respect to this parameter, as it is shown in Figure 3.3, which displays the condition number of the Jacobian matrix $\hat{\mathbf{J}}_{\text{SE}}$ of the TP algorithm through the iterations for different values of the regularization parameter γ. Part of this behaviour is due to the fact that, whenever linear regression is employed and, hence, all of the non-linearity of the problem must be addressed by the diffeomorphism, the problem is itself ill conditioned. This is the result of the diffeomorphism making the training points very close to each other in order to mimic the shape of the arctangent function. One way this behaviour may be dampened is by enriching the regression space with polynomials of higher degree which should help in solving the steep gradient of the target function. In Figure 3.3 it is also shown how, at the solution, the condition number of the kernel matrix $\mathbf{K}[\mathbf{z}_l]$ explodes faster over time as the regression space size diminishes.

We further argue that it is precisely this ill conditioning effect which is responsible for the slow convergence speed of the RGD. In order to experimentally investigate this, in Figure 3.4 it is shown the performance, along with some training diagnostics, of the

SRGD when employing the linear regression space $\mathcal{V}^1$ to approximate $g_{1,10}$ with $N = 20$ observations, across different numbers of batches N_B (note that the RGD is recovered when $N_B = 1$). The condition number shown refers to the kernel matrix defining the subspace in which the descent direction is taken, namely $\mathbf{K}[\mathbf{z}_{l,b}^{(b)}]$ where $\mathbf{z}_{l,b}^{(b)}$ denotes the chosen batch at time $t = (lB+b)/(LB)$. The plot suggests that improving the conditioning of the subspace containing the gradient benefits the convergence rate as it results in gradients of larger magnitude. The SRGD successfully achieves this simply by sub-sampling the training points. This observation yields an unusual interpretation of stochastic gradient descent as a way to improve the conditioning of an optimization problem. However, employing a stochastic optimization approach with richer regression spaces becomes unstable since the regression solution is accurate only if a sufficient number of samples is provided.

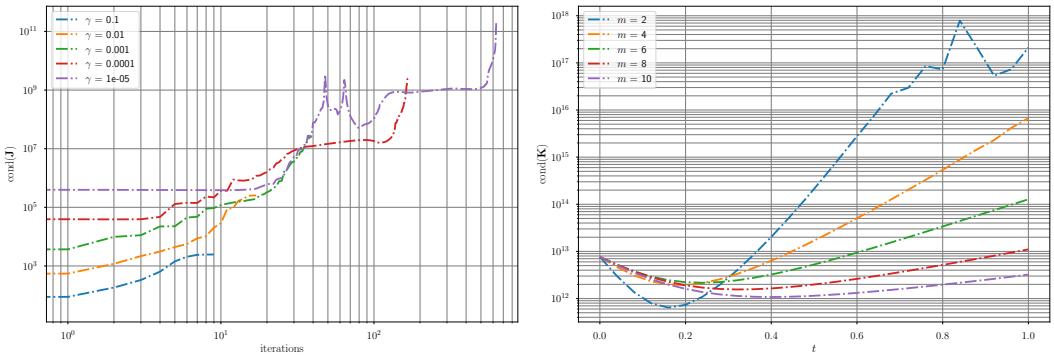

Figure 3.3: Left: Condition number of the matrix $\hat{\mathbf{J}}_{\mathrm{SE}}$ of the TP algorithm for different regularization parameters γ. The experiment was carried out in dimension $d = 1$ with a complexity parameter $\lambda = 10$ and $N = 10$ training points. Right: Condition number of the matrix $\mathbf{K}[\mathbf{z}_t]$ for different sizes m of the regression space. The experiment was carried out in dimension $d = 1$ with a complexity parameter $\lambda = 10$ and $N = 20$ training points.

Diffeomorphism analysis

The first experiment on the diffeomorphism we perform regards its representation in $v : [0,1] \to \mathcal{H}$, namely the control, for the OC problem with $\gamma = 10^{-5}$. In Figure 3.5 it is shown the norm of the control $\|v_t\|_{\mathcal{H}}^2$ over time for $L = 10$ and $L = 100$, where the target function is again $g_{1,10}$ and the number of observations is $N = 10$. Recall that in the continuous time problem $\|v_t\|_{\mathcal{H}}^2$ is supposed to remain constant in time as it coincides with the Hamiltonian of the the system. In the discrete setting, Proposition 2.1.8 predicts fluctuations in $\|v_t\|_{\mathcal{H}}^2$ of the order $\mathcal{O}(L^{-1})$. This is indeed what the plot shows as we have fluctuation of $\mathcal{O}(10^{-1})$ and $\mathcal{O}(10^{-2})$ respectively.

It is then interesting to explore the convergence of the discrete-in-time solution to the continuous one. This is shown in Figure 3.6. This plot was constructed taking as an approximation of the true solution, which we denote ϕ^∞ and the associated vector field

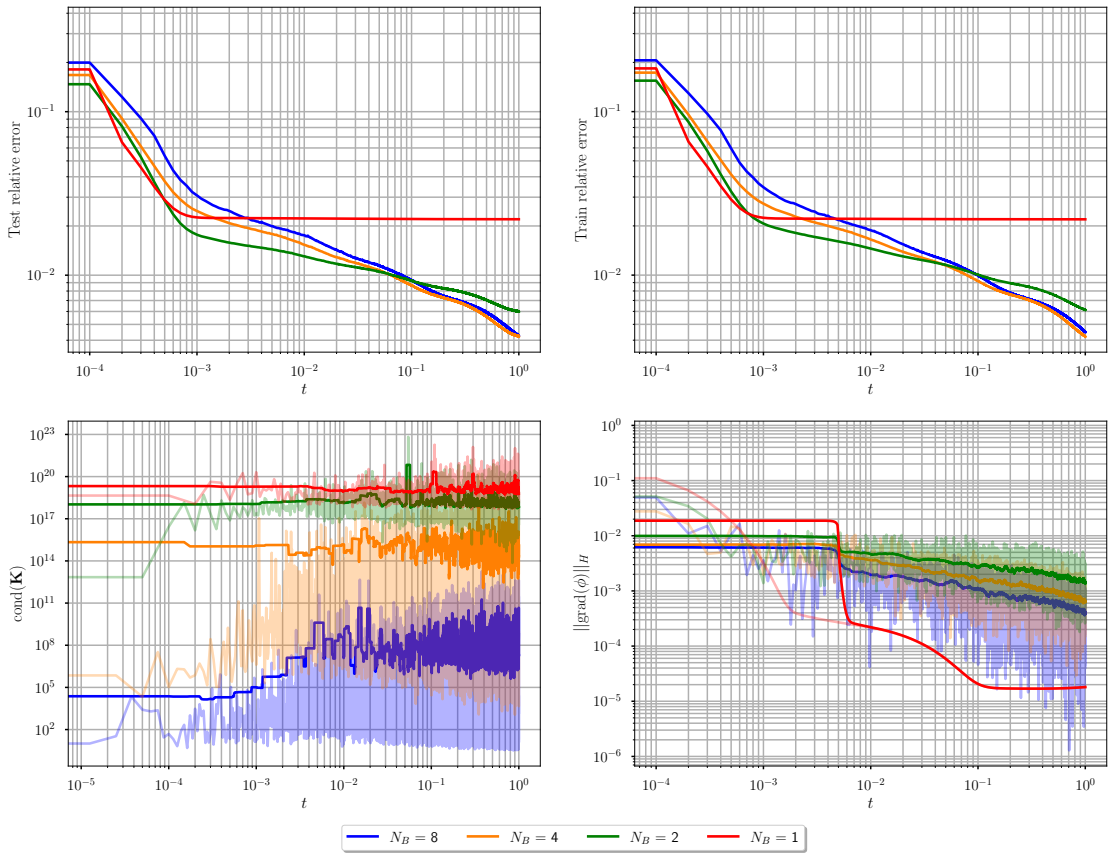

Figure 3.4: This experiment was carried out for target function $g_{1,10}$ and $N = 20$ of training points, across different numbers N_B of batches. Top: generalization (left) and training (right) errors over time. Bottom: condition number of the matrix $\mathbf{K}[\mathbf{z}_t^{\text{batch}}]$ over time (left) and RKHS norm of the stochastic gradient over time (right). Note that in the bottom plots the true data is transparent, while the opaque lines are smoothed versions in order to improve readability.

v^∞, one with $L = 200$ layers. Then, we tracked both of the errors

$$\int_0^1 (\phi^L - \phi^\infty)^2 d\mu, \qquad \int_0^1 \left\| v_t^L - v_t^\infty \right\|_{\mathcal{H}}^2 dt, \tag{3.49}$$

ϕ^L and v^L being, respectively, the map and vector fields solutions with L layers to the regression problem. First, this corroborates the theoretical convergence results presented in Section 2.1.4. Furthermore, it is interesting to notice that the L^2_μ convergence in the map seems faster than the one in $L^2([0,1];\mathcal{H})$.

Let us now interpret the role of the diffeomorphism in dimensions $d = 1$ and $d = 2$. For this, we first examine the solution to the OC problem. We employ the linear regression space $\mathcal{V}^1$ and then compare it with the solution found for a the richer regression space $\mathcal{V}^2$. The rest of the network hyperparameters for these experiments are summed up in

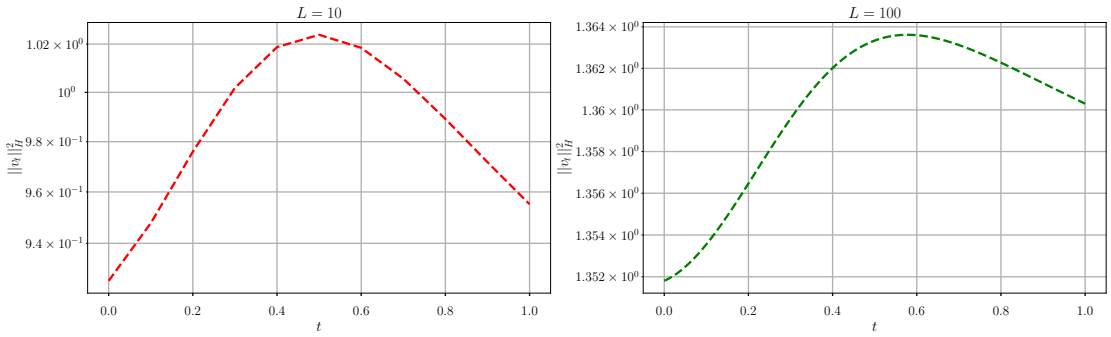

Figure 3.5: RKHS norm of the control $\|v_t\|^2_{\mathcal{H}}$ over time. This experiment was carried out in dimension $d = 1$ with a complexity parameter $\lambda = 10$ and $N = 10$ number of training points, both for $L = 10$ and $L = 100$ layers.

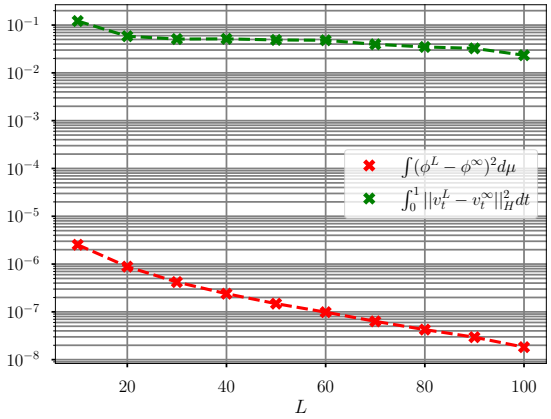

Figure 3.6: This experiment was carried out with $N = 10$ particles in $d = 1$ dimensions with $\lambda = 10$. The plot shows the convergence of the solution of the discrete problem with L layers towards the true solution, both at the level of diffeomorphisms (i.e. $\int_{[0,1]}(\phi^L - \phi^\infty)^2 d\mu$ in red), and at the level of representing vector fields (i.e. $\int_0^1 \|v_t^L - v_t^\infty\|^2_{\mathcal{H}} dt$, in green).

Table 3.2. Figure 3.7 reports the results for dimension $d = 1$. Not surprisingly, using linear regression makes the diffeomorphism approximate the shape of the target function directly. In order to achieve this the flow takes the particles and concentrates them at the extremes of the interval, as it is evident from the density plot over time. The richer regression space makes the burden on the diffeomorphism much lighter. This results in a map closer to the identity, which just barely stretches the middle portion of the interval in order to match the steepness of the target. In Figure 3.8 the two dimensional problem with linear regression is shown. The target function is aligned with the second axis, hence the diffeomorphism mimics the one-dimensional behavior along it. It is interesting, however, that at the beginning of the transformation a rotational motion appears, meaning that the

diffeomorphism is getting the first variable involved in the approximation as well. When the regression space $\mathcal{V}^2$ is employed, shown in Figure 3.9, the diffeomorphism shows a less clear interpretation due to the presence of the higher degree polynomials. In particular, the symmetry which characterizes the vector field is lost, and more "local" phenomena of either expansion or contraction appear. However, it is interesting that at time $t = 1$ the vector field aligns with the x_2 axis.

As far as the RO algorithms are concerned, the results for the linear regression case are shown if figures 3.10 and 3.11, which refer to the RGD and RDN respectively. It is interesting that no rotational motion appears in the fields. This is because moving the points along the x_1 axis has no effect on the loss. Indeed, the function is constant along x_1. Then, the field concentrates the points at the extremes of the interval with respect to x_2, mimicking what happens in one dimension. Also, there seem to not be any significant difference in the behaviour of the driving vector field between the RGD and RDN training strategies. When the richer regression space $\mathcal{V}^2$ is employed, we observe a similar behaviour as far as the vector fields having no component in the direction x_1 (although the vector field for the RDN slightly diverges from this trend close to time $t = 1$). The main difference from the linear regression case is that the "stretching" behaviour is no more centered around the subspace identified by $x_2 = 0$, which is where the fast transient of $g_{2,(0;10)}$ lies. Instead, it seems to not only happen around more general curves in space, but more than one at the same time (see for instance time $t = 1$ and time $t = 0.6$ in Figure 3.12 and time $t = 0.2$ in Figure 3.13).

	$d = 1$				$d = 2$			
Algorithm	N	L	r	γ	N	L	r	γ
OC	10	15	5	10^{-5}	100	6	5	10^{-6}
RGD	-	-	-	-	100	1000	5	-
RDN	-	-	-	-	100	100	5	-

Table 3.2: Network hyperparameters. Here OC stands for the solution of the OC problem with any algorithm (although these experiments were performed with the TP algorithm). These are the same regardless of the employed regression space.

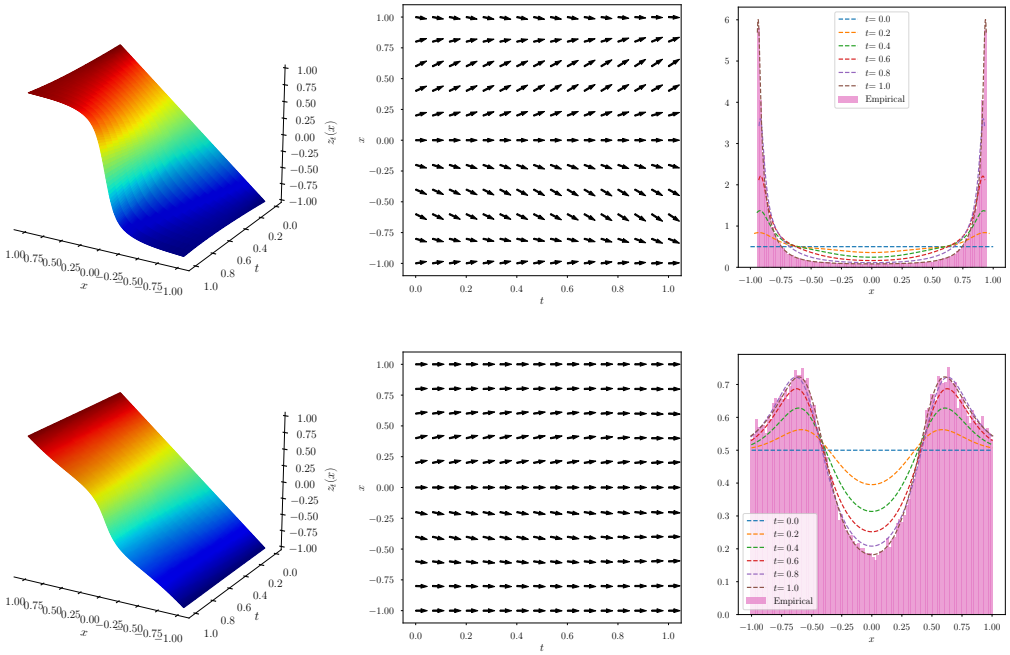

Figure 3.7: Illustration for target function $g_{1,10}$ and OC problem. Left: vector field $(1, v_t(x))$ driving the state $(t, z_t(x))$ over time. Center: diffeomorphism $\phi_t(x)$ over time and space. Right: density $\mu_{\phi_t}(x)$ over time and space.

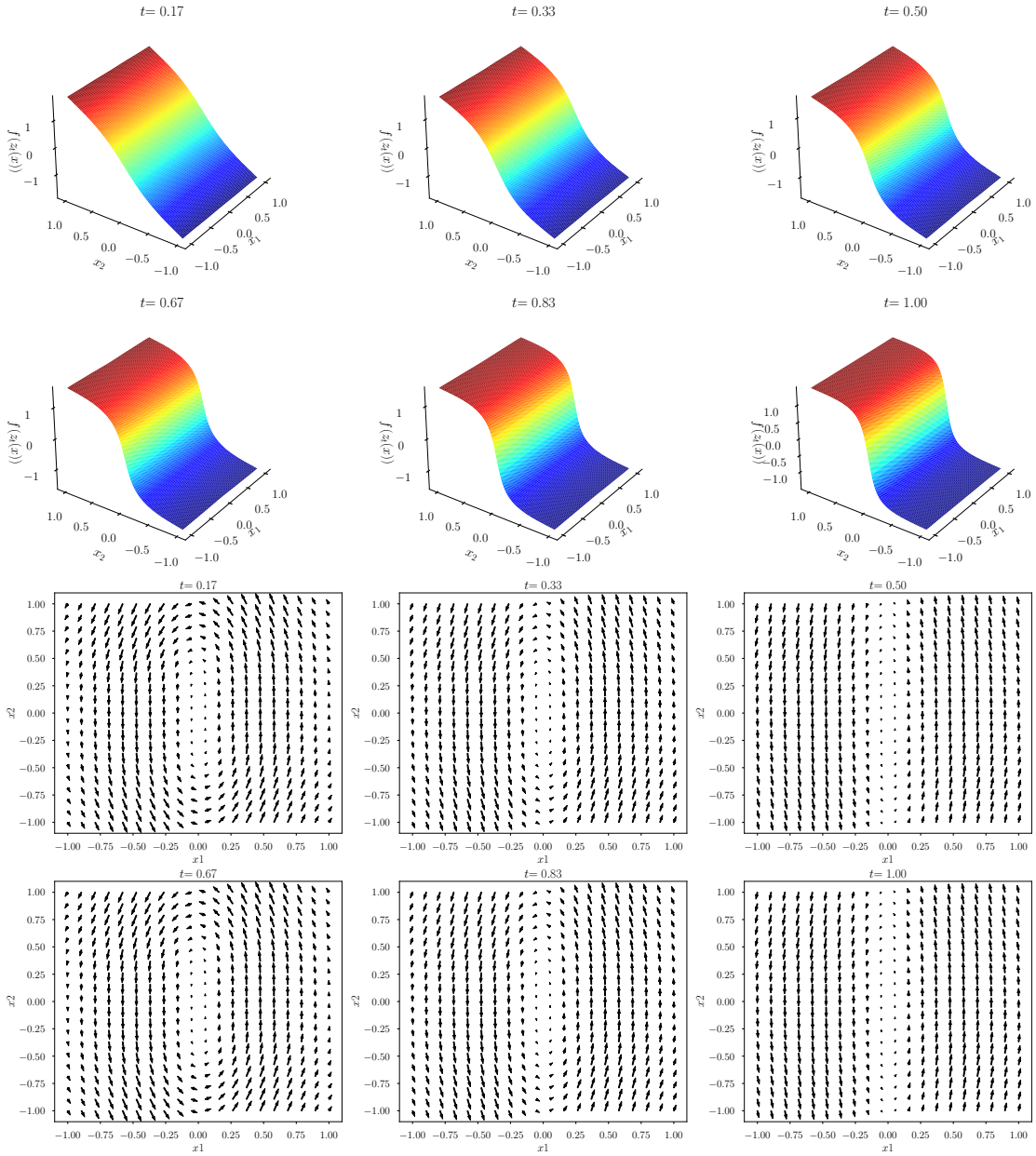

Figure 3.8: Illustration for target function $g_{2,(0;10)}$ and OC problem with regression space $\mathcal{V}^1$. Top: prediction $f(\phi_t(x))$ over time and space (note that the regression parameters used at time t are the optimal ones for the remapped points $(\phi_t(x^{(1)}), \ldots, \phi_t(x^{(N)}))$). Bottom: vector field $v_t(x)$ driving the state $z_t(x)$ over time.

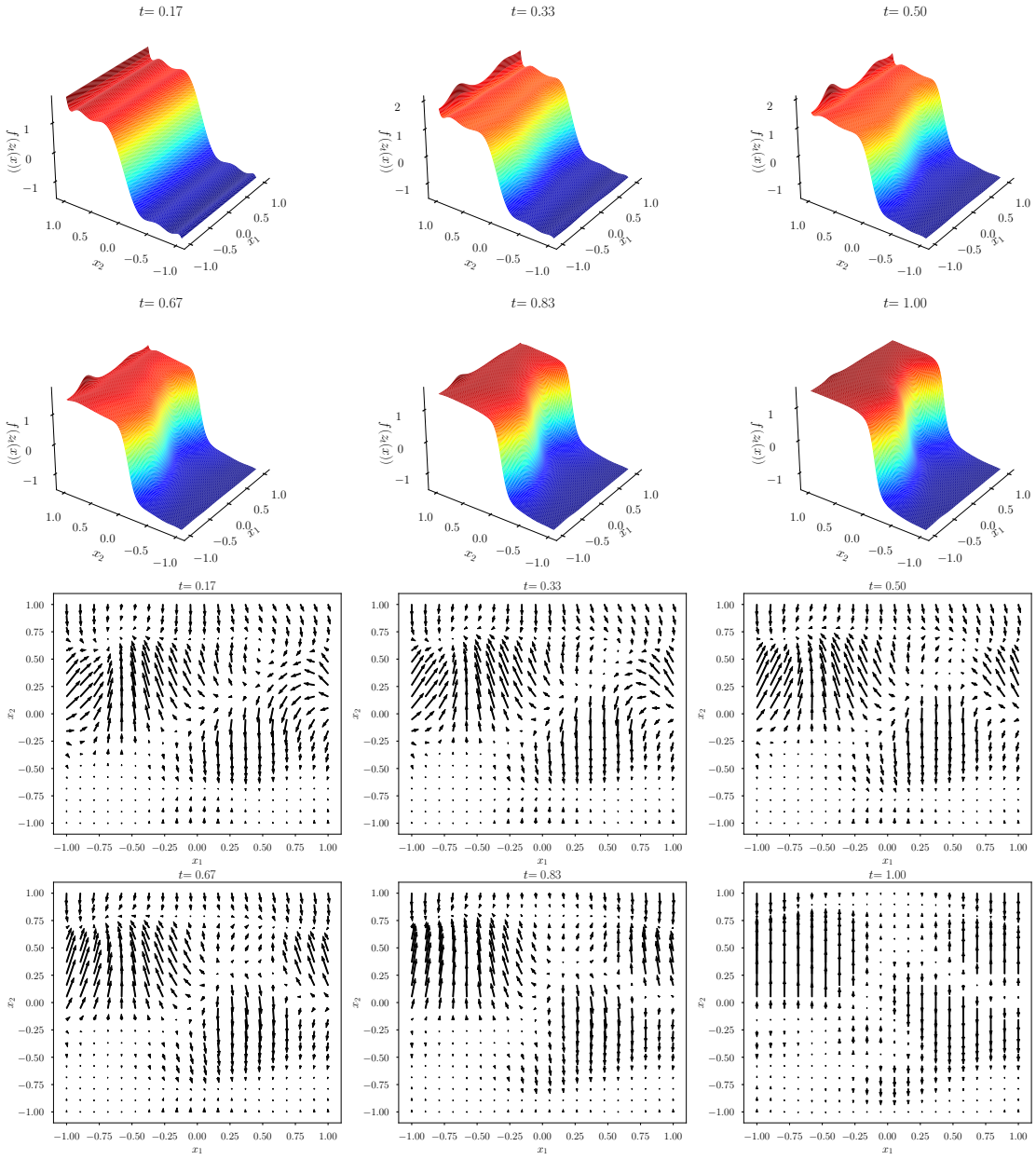

Figure 3.9: Illustration for target function $g_{2,(0;10)}$ and OC problem with regression space $\mathcal{V}^2$. Top: prediction $f(\phi_t(x))$ over time and space (note that the regression parameters used at time t are the optimal ones for the remapped points $(\phi_t(x^{(1)}), \ldots, \phi_t(x^{(N)}))$). Bottom: vector field $v_t(x)$ driving the state $z_t(x)$ over time.

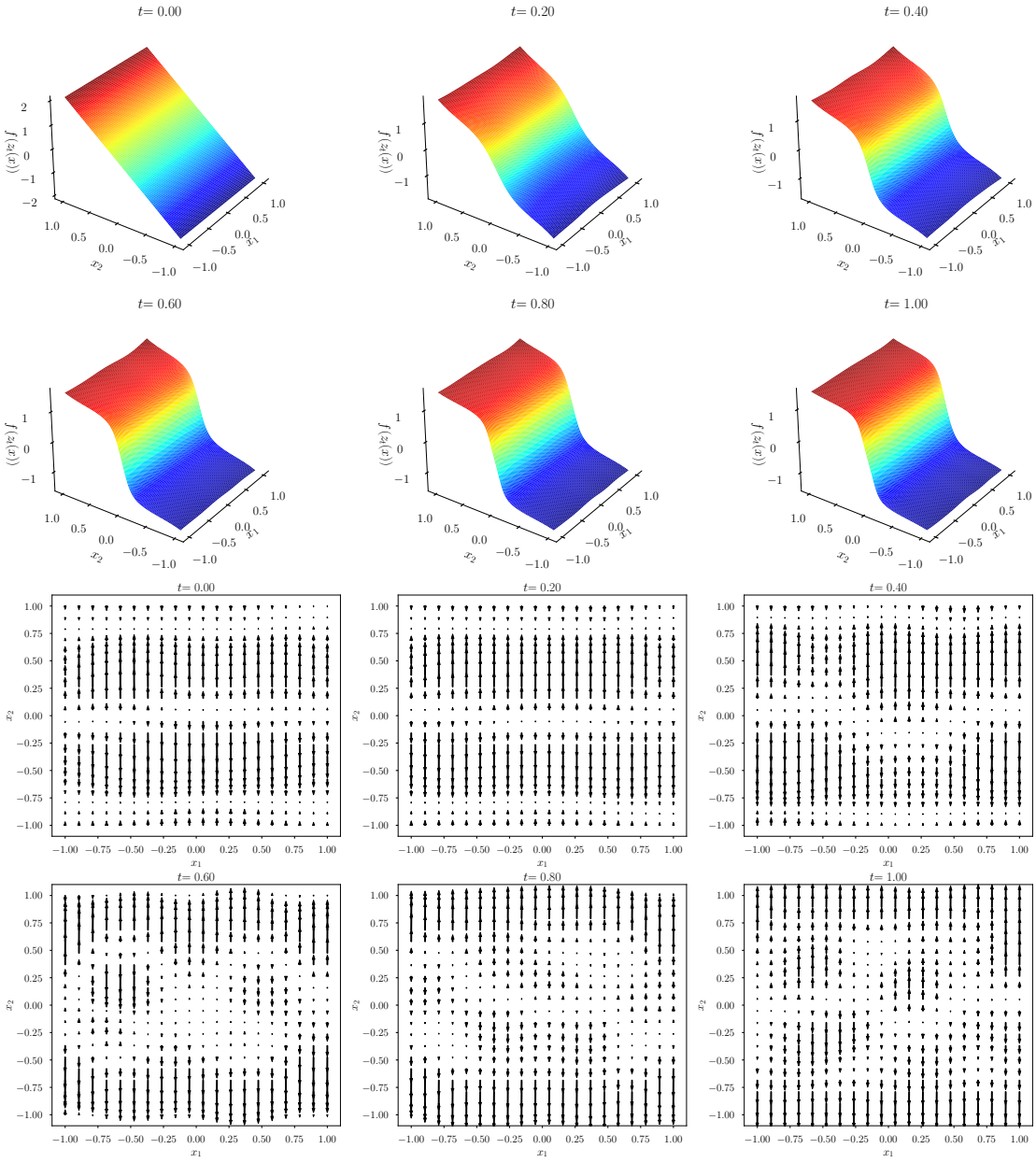

Figure 3.10: Illustration for target function $g_{2,(0;10)}$ and RO problem with regression space $\mathcal{V}^1$, solved via RGD. Top: prediction $f(\phi_t(x))$ over time and space (note that the regression parameters used at time t are the optimal ones for the remapped points $(\phi_t(x^{(1)}), \ldots, \phi_t(x^{(N)}))$). Bottom: vector field $v_t(x)$ driving the state $z_t(x)$ over time.

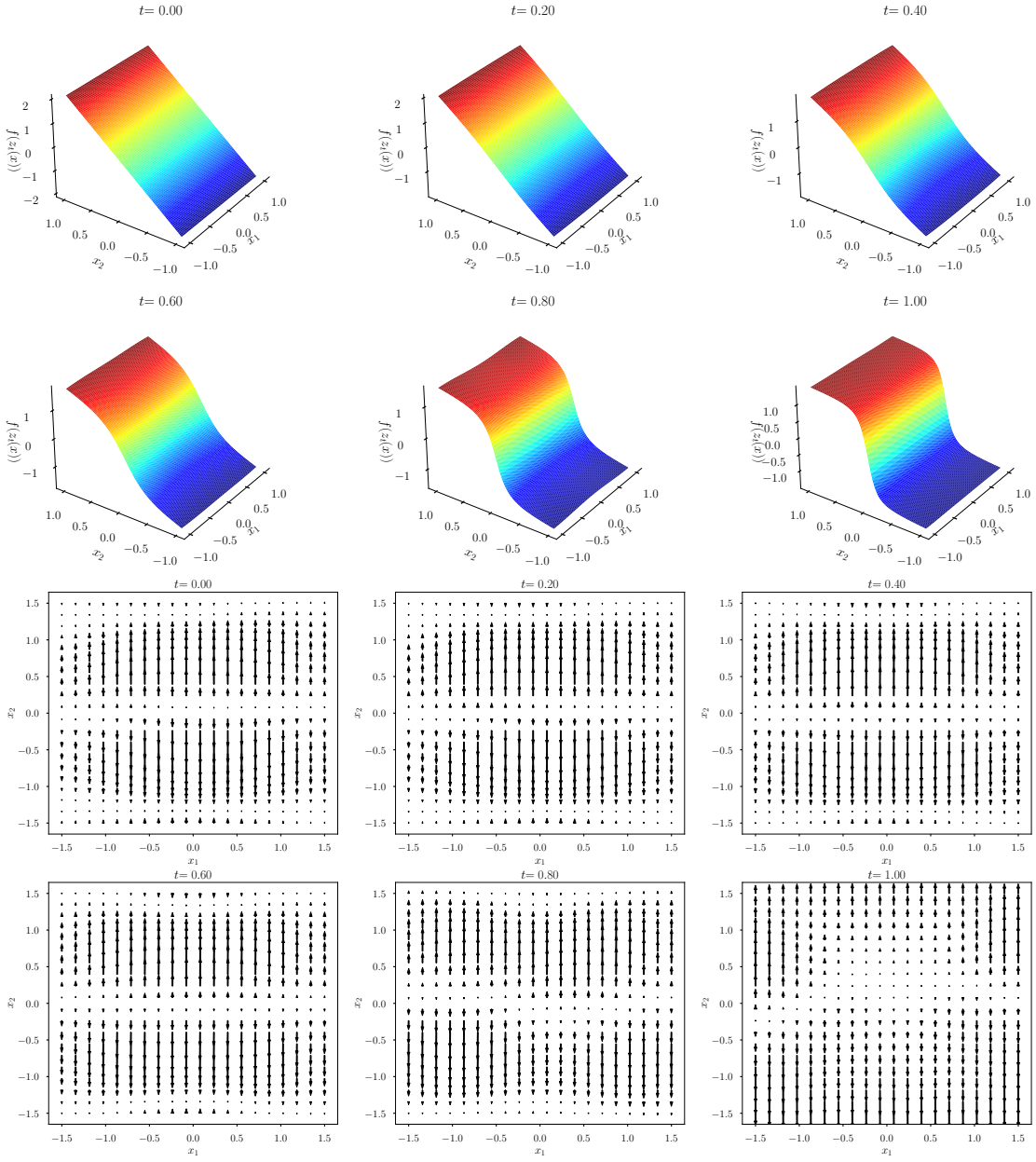

Figure 3.11: Illustration for target function $g_{2,(0;10)}$ and RO problem with regression space $\mathcal{V}^1$, solved via RDN. Top: prediction $f(\phi_t(x))$ over time and space (note that the regression parameters used at time t are the optimal ones for the remapped points $(\phi_t(x^{(1)}), \ldots, \phi_t(x^{(N)}))$). Bottom: vector field $v_t(x)$ driving the state $z_t(x)$ over time.

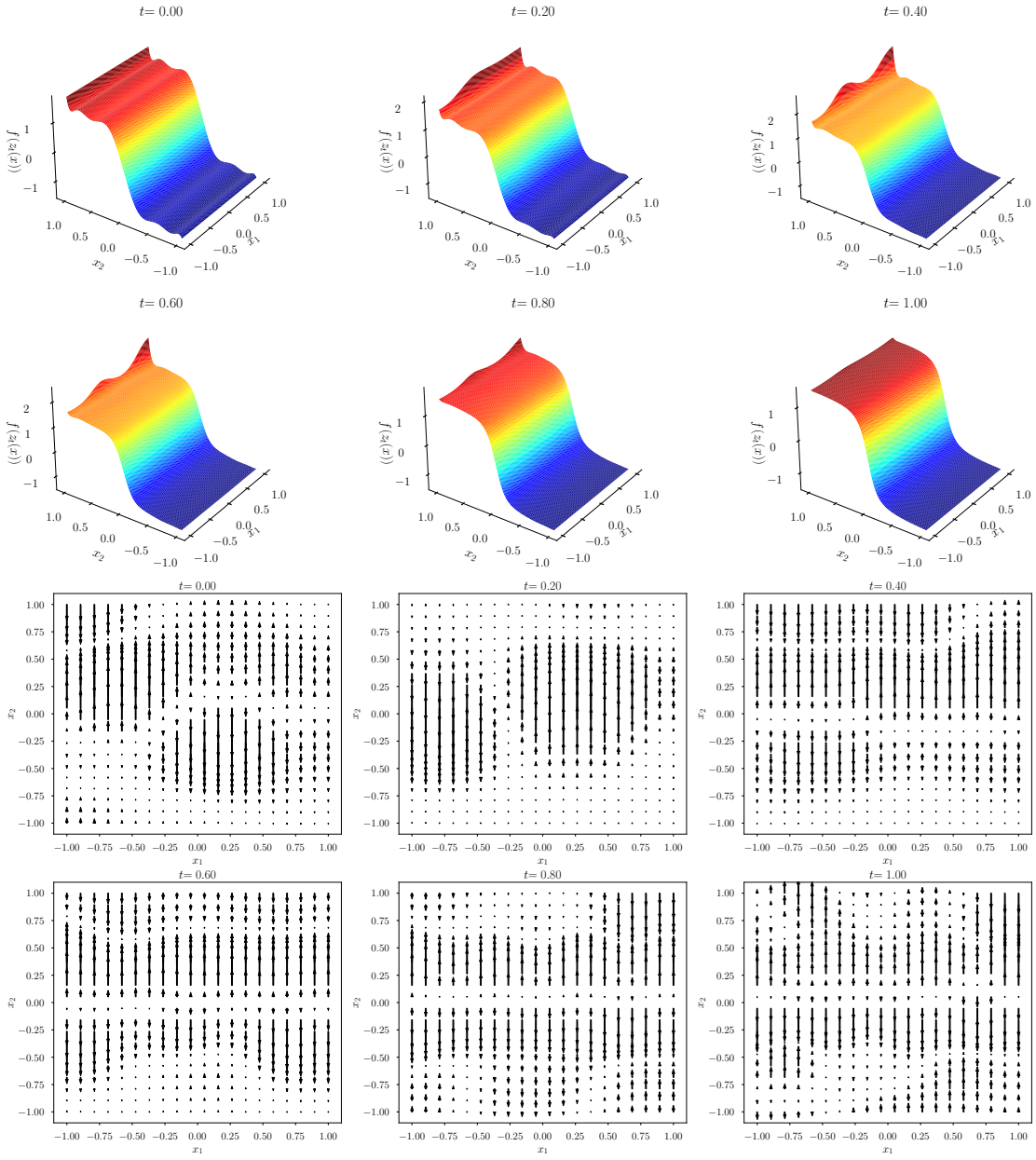

Figure 3.12: Illustration for target function $g_{2,(0;10)}$ and RO problem with regression space $\mathcal{V}^2$, solved via RGD. Top: prediction $f(\phi_t(x))$ over time and space (note that the regression parameters used at time t are the optimal ones for the remapped points $(\phi_t(x^{(1)}), \ldots, \phi_t(x^{(N)}))$). Bottom: vector field $v_t(x)$ driving the state $z_t(x)$ over time.

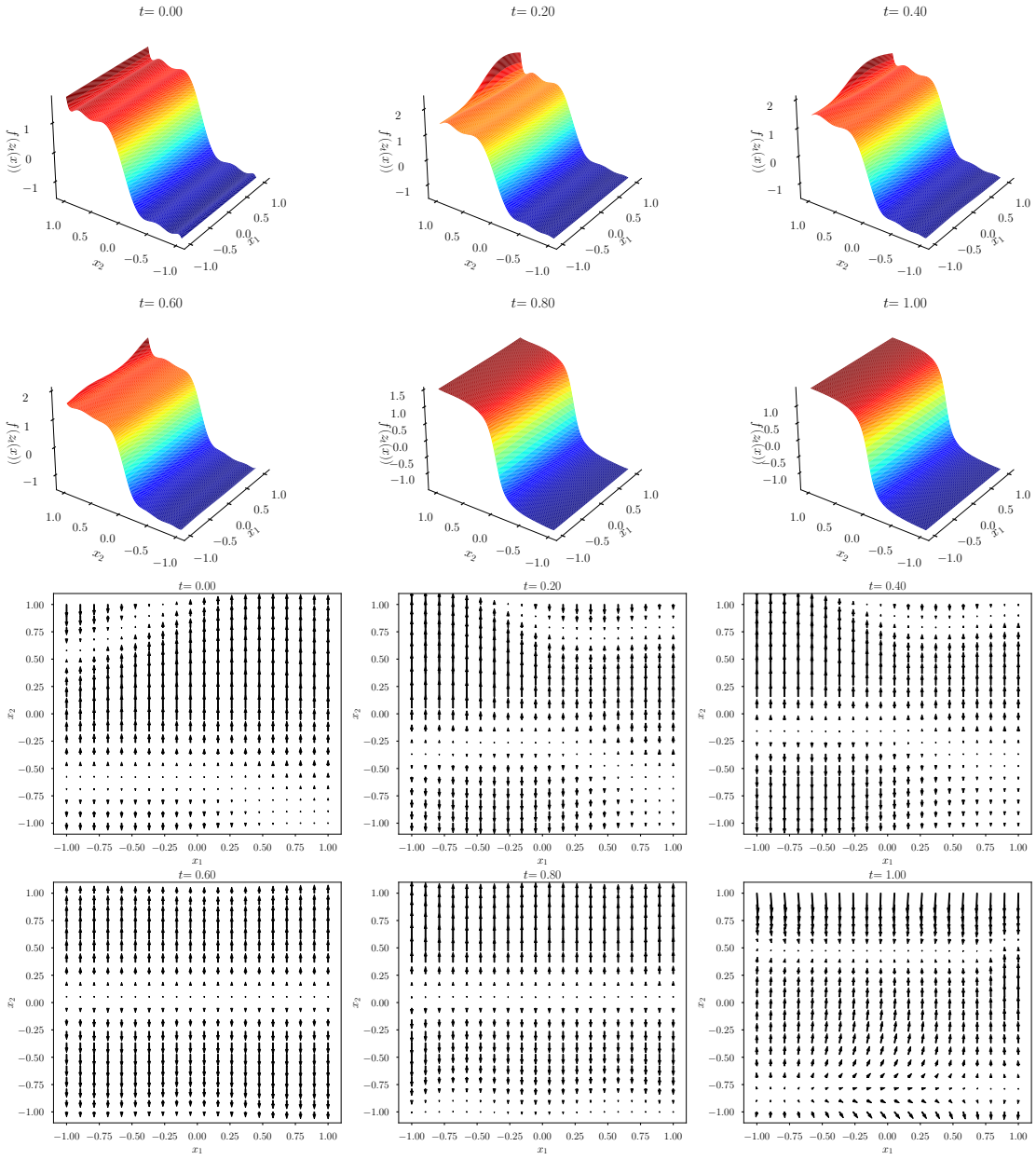

Figure 3.13: Illustration for target function $g_{2,(0;10)}$ and RO problem with regression space $\mathcal{V}^2$, solved via RDN. Top: prediction $f(\phi_t(x))$ over time and space (note that the regression parameters used at time t are the optimal ones for the remapped points $(\phi_t(x^{(1)}), \ldots, \phi_t(x^{(N)}))$). Bottom: vector field $v_t(x)$ driving the state $z_t(x)$ over time.

Order of convergence

Next, we focus on the speed of convergence to the target function the OC, RGD and RDN models. Table 3.3 reports the relevant hyperparameters for the experiments. Note that the proposed models are compared against kernel ridge regression (see Section 1.2.2) with Gaussian kernel $k(x,x') = e^{-r\|x-x'\|_2^2}$. The KRR hyperparameters are optimized with a validation approach over the following grid[3]:

$$(\gamma, r) \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\} \times \{0.1, 0.5, 1, 5, 10, 15\}. \tag{3.50}$$

Furthermore, we always report the relative error for the polynomial regression (PR) in the original space as well. That is, we consider the approximation function $f^\star$ solving the least squares problem

$$f^\star \in \underset{f \in \mathcal{V}_m}{\arg\min} \quad \sum_{i=1}^{N} \left(f(x^{(i)}) - y^{(i)} \right)^2. \tag{3.51}$$

This way it is possible to determine the effect of performing the regression in the mapped space $\phi_1([-1,1]^d)$ as opposed to the original space $[-1,1]^d$.

	$d=1$			$d=2$		
Algorithm	L	r	γ	L	r	γ
OC	10	5	10^{-5}	10	5	10^{-5}
RGD	1000	5	-	1000	5	-
RDN	100	5	-	100	1	-

Table 3.3: Network hyperparameters. Here OC stands for the solution of the OC problem with any algorithm (although these experiments were performed with the TP algorithm). These are the same regardless of the employed regression space.

First, let us consider again the target functions $g_{1,10}$ and $g_{2,(0;10)}$ with regression space $\mathcal{V}^1$, the results for which are shown in Figure 3.14. The first observation is that, among the proposed algorithms, the OC solution is the best performing one in $d=1$ with order of convergence $\mathcal{O}(N^{3.59})$, while in $d=2$ it is the RDN which prevails. Interestingly enough, the latter improves the convergence rate from dimension $d=1$ to $d=2$. The RGD shows a similar trend in that its performance is very poor in $d=1$, suggesting that 1000 iterations are not nearly enough in order to find a good solution, while in $d=2$ it performs like the OC solution. Finally let us remark that (apart from the RDN in $d=1$) the algorithms manage to outperform the KRR by a large enough margin.

Let us now consider the target functions $g_{1,10}$ with regression space $\mathcal{V}^2$. The results are reported in Figure 3.15. The clear takaway from this experiment is that RGD benefits greatly from the richer regression space as it features an order of convergence of

[3]Recall that γ denotes the regularization parameter for KRR, not to be confused with the one relevant to the OC problem.

$\mathcal{O}(N^{-7.58})$, much higher than the one of the other two approaches. Note also the drastic improvement with respect to performing the regression in the original space. The issue with the OC problem is that it is very hard to solve, as it was explained and showed by previous experiments. As it does not seem to drastically outperform the RGD, while being outperformed by the RDN instead, we choose to carry on the convergence analysis based on the RO algorithms only. Then, we continue with the study of the target function $g_{2,(0;10)}$, again with regression space $\mathcal{V}^2$. The results in this setting corroborate the observations made for the one dimensional case. However, note that, despite being the superior approach nonetheless, the order of convergence of the RDN deteriorates significantly with the added dimension.

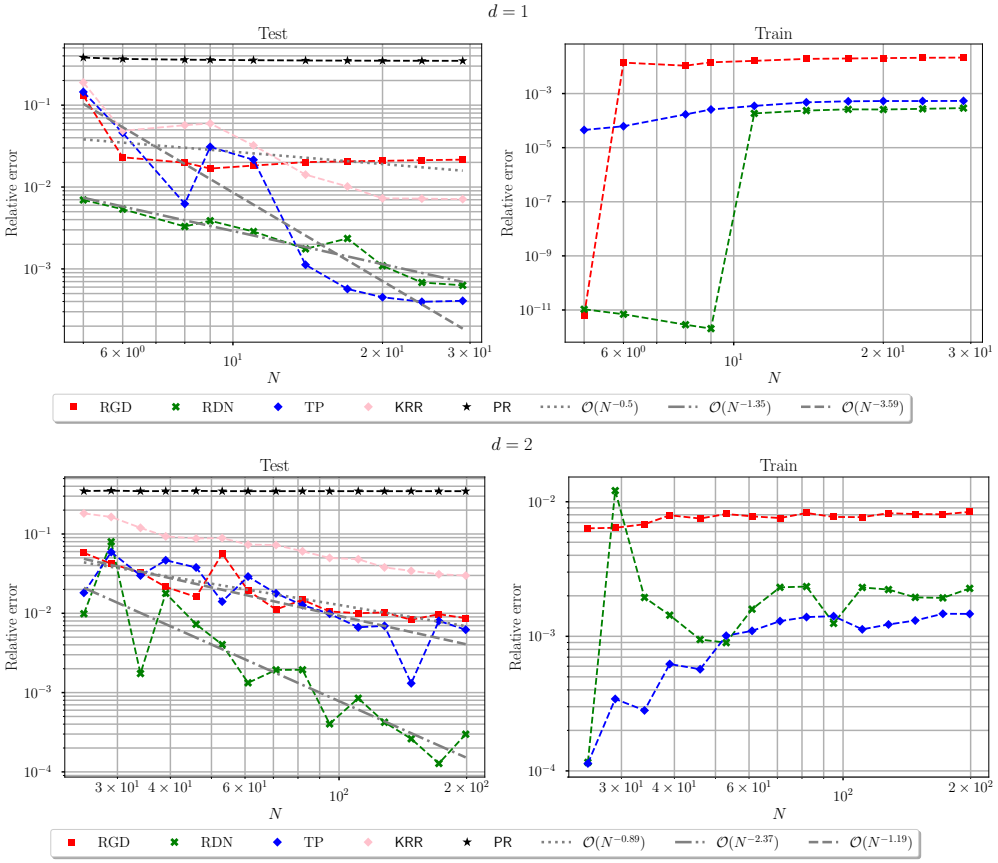

Figure 3.14: Plot of the test (left) and train (right) relative errors for target functions $g_{1,10}$ (top) and $g_{2,(0;10)}$ (bottom) with regression space $\mathcal{V}^1$, across different numbers of observations N.

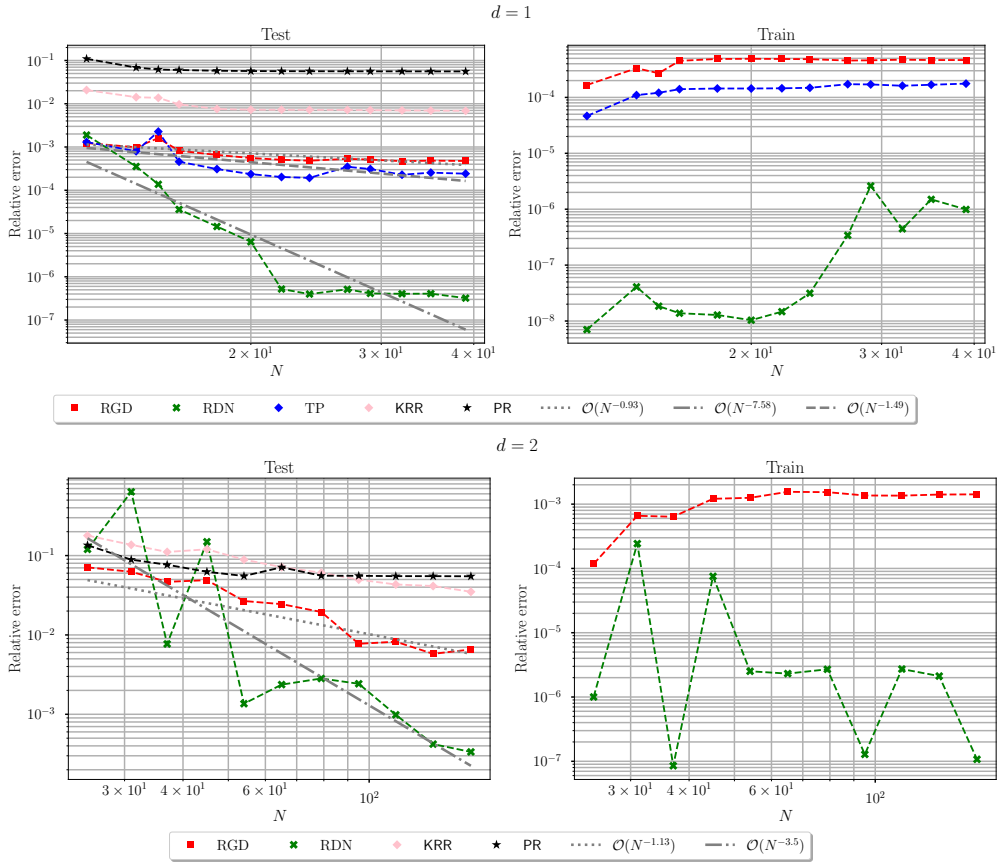

Figure 3.15: Plot of the test (left) and train (right) relative errors for target functions $g_{1,10}$ (top) and $g_{2,(0;10)}$ (bottom) with regression space $\mathcal{V}^2$, across different numbers of observations N.

3.2.2 A local feature

The second example we consider consists of multidimensional Gaussian functions, namely

$$g_{d,\lambda}(x) = e^{-\lambda\|x\|^2}, \qquad x \in [-1,1]^d, \tag{3.52}$$

where $\lambda \in \mathbb{R}^d$ is a precision parameter controlling the shape of the multidimensional bell. The latter, then, controls the complexity of the target function since $\|\lambda\| \to \infty$ makes the function approximate a spike-like function centered around the origin, as shown in Figure 3.16. For these experiments we employ the RKHS $\mathcal{H}$ generated by the separable

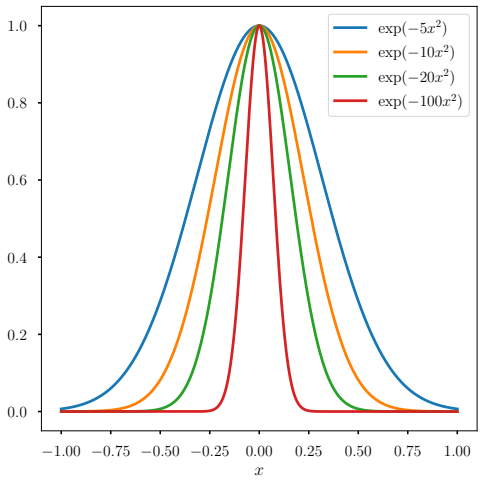

Figure 3.16: Gaussian bell function in $d = 1$ dimensions for different values of $\lambda \in \mathbb{R}$. As λ increases, $e^{(-\lambda x^2)}$ approaches a spike-like function centered in $x = 0$.

Gaussian kernel in (3.46). We then approximate $g_{d,\lambda}$ via $y^{(i)} = g_{d,\lambda}(x^{(i)})$, $i = 1,2,\dots,N$ noiseless observations, the points $x^{(i)}$, $i = 1,\dots,N$ being generated in $[-1,1]^d$ via a QMC approach (the Halton sequence, in particular). Furthermore, we take $\mathcal{V}_m$ to be the set of polynomials of total degree less or equal than 10, namely

$$\mathcal{V}_m = \{x_1^{\alpha_1} \dots x_d^{\alpha_d} \mid \alpha \in \mathbb{N}^d, \sum_{i=1}^{d} \alpha_i \leq 10\} \tag{3.53}$$

Furthermore, as we have assessed that the more robust and successful training algorithms seem to be the RO ones, we restrict the analysis to those only.

Diffeomorphism analysis

Given the rich regression space, here we are interested in which transformation the diffeomorphism realizes. We address the question with similar plots to what was shown in the previous section, concentrating on the two-dimensional case. Note that with the regression space design introduced just before, the dimension is $m = 66$. As in all our experiments, we took $L = 1000$ and $L = 100$ for the RGD and RDN respectively, while

the complexity parameter is set to $\lambda = 20$. The results are shown in figures 3.17 and 3.18. First, note how within 1000 iterations the RGD was not able to modify the space enough for the regression to closely reproduce the target function. The vector field shows clearly that the space is stretched out from the origin, while simultaneously being compressed along a one-dimensional manifold whose shape reminds that of a rhomboid. The field for the RDN shows a similar trend, however towards time $t = 1$ it slightly diverges from the usual behaviour. The main difference here is that, within 100 iterations, the RDN is able to yield a model which very accurately reproduces the target function. Again, this corroborates the idea that the second order information becomes more important as the regression space gets richer.

Order of convergence

Let us now turn to the analysis of the convergence of the models to the target function. We investigate this in dimensions $d = 1$ and $d = 2$ for complexity parameter $\lambda = 20$. Recall from the previous section that we compare the models with KRR and, moreover, we report the performance of the regression in the original space (that is, taking the diffeomorphism ϕ to be the identity mapping.). Here, however, we have that choosing $r = 20$ in KRR gives exactly the target function. We hence artificially restrict KRR to use a smaller shape parameter. In particular, we tune the KRR hyperparameters considering the grid

$$(\gamma, r) \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\} \times \{0.1, 0.5, 1, 5\}. \tag{3.54}$$

The results are displayed in Figure 3.19. These were obtained with the usual experimental setup of $L = 1000$ and $L = 100$ for the RGD and RDN respectively, and setting the shape parameter to be $r = 5$ for the RGD and $r = 1$ for the RDN. Notice that, in dimension $d = 1$, KRR and the RGD algorithm perform similarly as far as order of convergence is concerned, though RGD seems to converge slightly faster. Again, the RDN instead features a very fast order of convergence. Thing change in dimension $d = 2$ where the RGD is outperformed (if raw approximation power is concerned) by KRR. However, RGD features a much improved order of convergence with respect to the one-dimensional case. Indeed, it is very close to the one of the RDN which, instead, suffered from the extra dimension. Finally, note that these trends agree with the observations made in the previous section for the arctangent function.

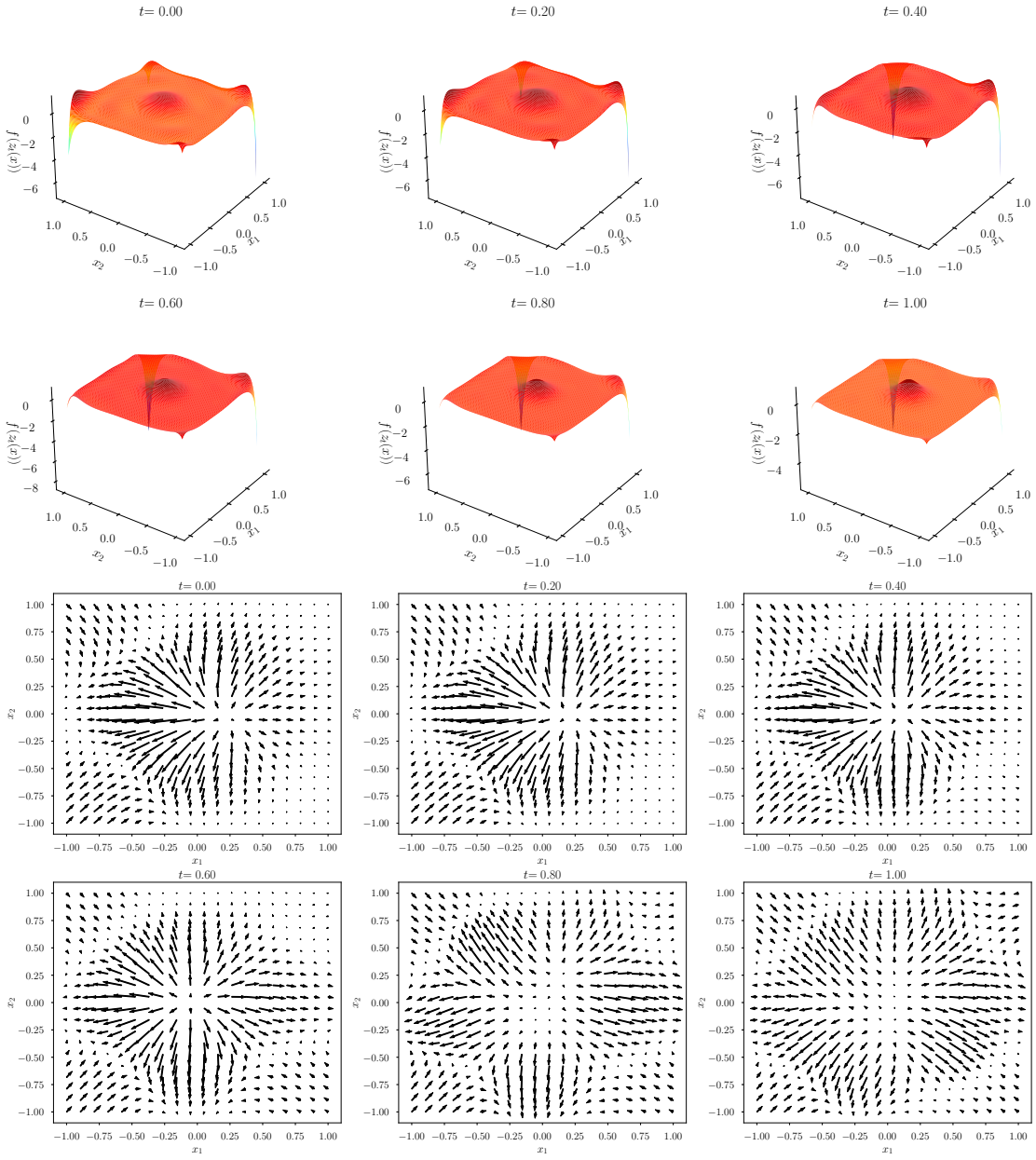

Figure 3.17: Illustration for target function $g_{2,20}$ and RO problem, solved via RGD. Top: prediction $f(\phi_t(x))$ over time and space (note that the regression parameters used at time t are the optimal ones for the remapped points $(\phi_t(x^{(1)}), \ldots, \phi_t(x^{(N)}))$). Bottom: vector field $v_t(x)$ driving the state $z_t(x)$ over time.

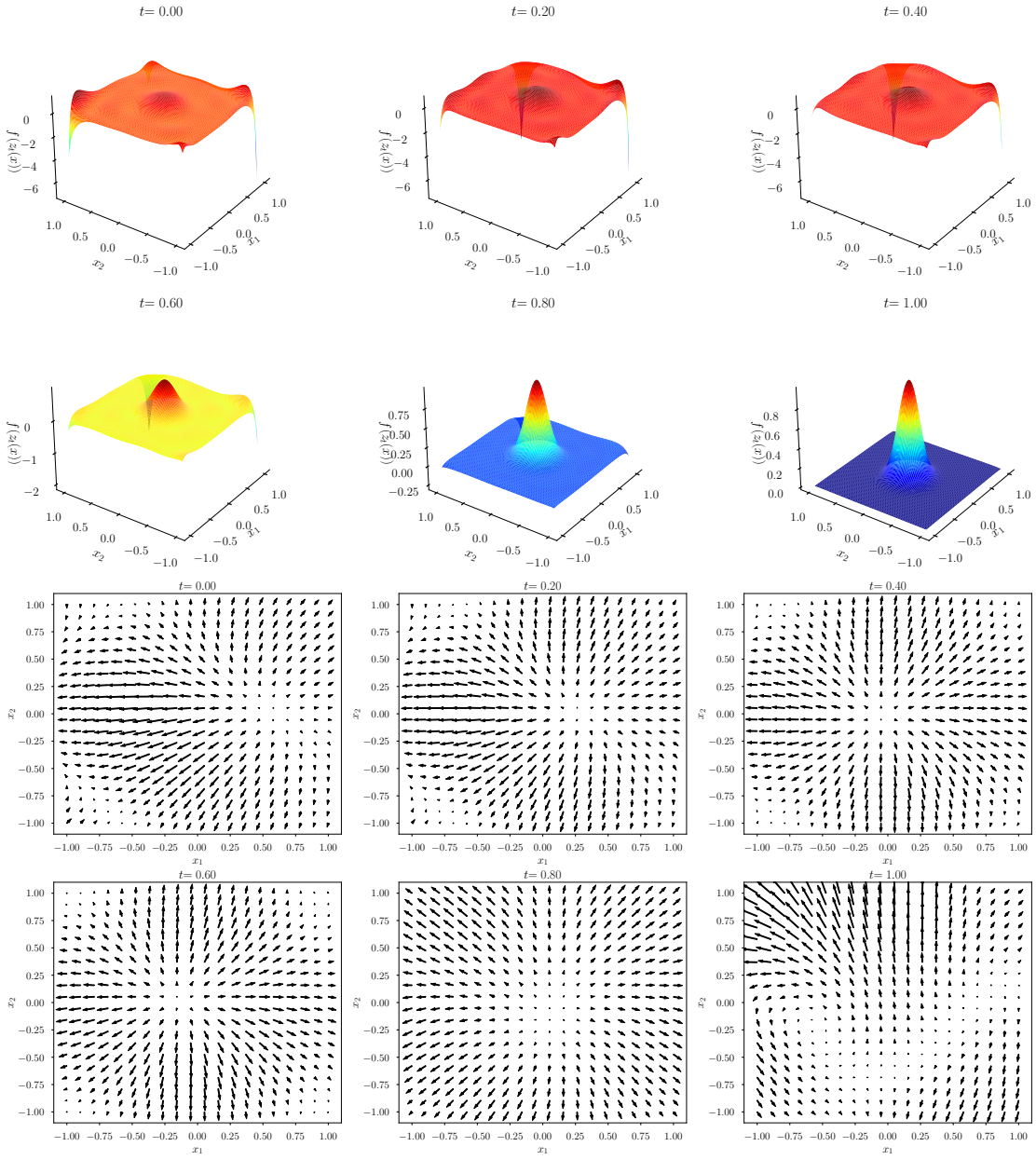

Figure 3.18: Illustration for target function $g_{2,20}$ and RO problem, solved via RDN. Top: prediction $f(\phi_t(x))$ over time and space (note that the regression parameters used at time t are the optimal ones for the remapped points $(\phi_t(x^{(1)}), \ldots, \phi_t(x^{(N)}))$). Bottom: vector field $v_t(x)$ driving the state $z_t(x)$ over time.

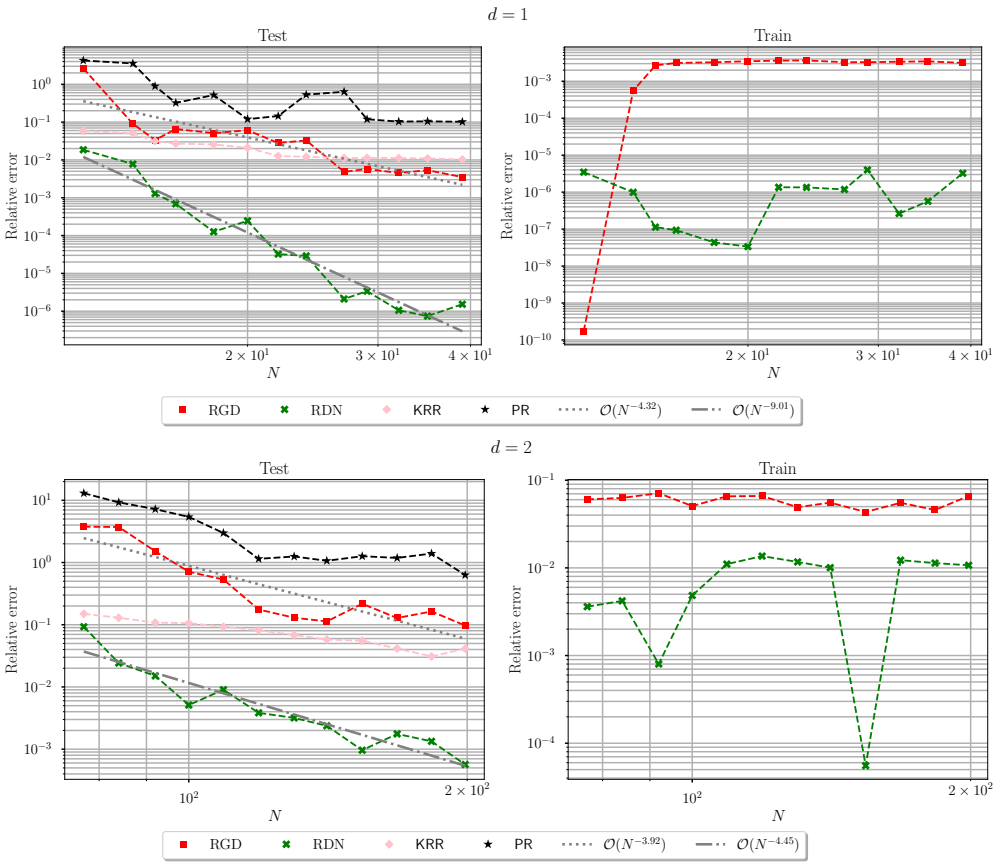

Figure 3.19: Plot of the test (left) and train (right) relative errors for target functions $g_{1,20}$ (top) and $g_{2,20}$ (bottom), across different numbers of observations N.

Conclusions

In this thesis, we have considered Kernel Neural ODEs, namely Neural ODEs where the right-hand side is taken in a fixed reproducing kernel Hilbert space. This structure makes the learning problem particularly amenable to a rigorous mathematical study. In particular, we established that, for the continuous-time setting, under suitable assumptions on the RKHS the regularized learning problem admits minimizers. Furthermore, we characterized these minimizers exploiting the Pontryagin maximum principle, leading to the reduction of the learning problem to a two-point boundary value problem. We then introduced a suitable time discretization and showed L^2 convergence of the minimizers of the discrete-in-time functional to the continuous-time solutions. The algorithms we designed starting by the OC perspective are three. The first is a preconditioned version of standard gradient descent, where the preconditioning matrix is suggested by the Hamiltonian maximization step prescribed by the PMP. The second algorithm is instead inspired by time-parallel ODE integration techniques. This algorithm can alternatively be interpreted as a second order method on the reduced (i.e., where the control has been eliminated) Lagrangian associated to the learning problem. In order to improve scalability with respect to the number of layers of this scheme, we further introduced a multi-grid method. The idea behind this is to decompose in time the original TPBVP into a multitude of sub-TPBVP on a coarse grid. Then, each of these sub-problems may be solved with any discretization and by any algorithm. Finally one has to impose matching conditions in order to regain the solution to the original TPBVP. The resulting non-linear system of equations is then solved via a Quasi-Newton method, where the Jacobian of the system on the matching conditions is approximated by the Jacobian of the coarse problem.

As an alternative approach, we endow the set of diffeomorphisms $\mathcal{M}$ realized by KerODEs with a formal Riemannian manifold structure following the work Duncan et al. [2019]. Then, we recast the learning problem as an optimization problem on $\mathcal{M}$. We then computed the Riemannian gradient of the loss functional and interpreted the KerODE solution to the learning problem as a gradient flow. This approach was then further extended to second order optimization. In particular, we computed the Riemannian Hessian of the loss functional, and showed that, provided a damping term is added, the linear system (in the RKHS) defining the direction of the geometric flow can be reduced to a finite dimensional problem. These results lead to practical layer by layer training algorithms. In particular, we propose a Riemannian gradient descent algorithm and its stochastic version based on the Riemannian gradient flow perspective. Furthermore, exploiting the second order information we develop a Riemannian damped Newton algorithm which at

each iteration involves the solution to a linear system. The peculiarity of the Riemannian optimization point of view on the learning problem is that, in some sense, the KerODE and the optimization problem are identified. In other words, this suggests that a Neural Network optimizes the loss through its layers.

The numerical experiments show that the second order algorithms derived from the Optimal Control perspective outperform the preconditioned gradient method in terms of speed of convergence towards a good solution. However, these are ill conditioned with respect to the explicit regularization parameter. This makes the prediction of noiseless data accurately hard since such regularization must be large enough for the methods to converge. The algorithms inspired by Riemannian optimization are instead a lot more robust in terms of the problems they manage to find good solutions to. In particular, the Riemannian damped Newton performs surprisingly well, particularly when the regression space is rich enough.

The optimal control perspective is very mathematically compelling. Future work directions involve a thorough theoretical investigation into the rate of convergence of the minimizer of the space-discretized problem to the solution of the continuous-time problem. On the algorithmic side, it would instead be interesting to devise other, more robust training strategies. The Riemannian optimization point of view furnishes a compelling interpretation of learning with DNNs which leads to innovative training algorithms. An interesting future research direction involves giving a more solid mathematical foundation to the setting by providing a rigorous treatment of the manifold of diffeomorphisms.

Bibliography

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: a review. *Found. Trends Mach. Learn.*, 4:195–266, 2012.

Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966.

Mohamed-Ali Belabbas. On implicit regularization: Morse functions and applications to matrix factorization. *CoRR*, abs/2001.04264, 2020. URL https://arxiv.org/abs/2001.04264.

J. Frédéric Bonnans and Julien Laurent-Varin. Computation of order conditions for symplectic partitioned runge-kutta schemes with application to optimal control. *Numerische Mathematik*, 103:1–10, 2006.

Nicolas Boumal. An introduction to optimization on smooth manifolds. To appear with Cambridge University Press, Jun 2022. URL https://www.nicolasboumal.net/book.

R Brooks and P Trauber. Van est theorem for groups of diffeomorphisms. *Hadronic J.; (United States)*, 8 1978. URL https://www.osti.gov/biblio/6554436.

Russel E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998. doi: 10.1017/S0962492900002804.

Jesper Carlsson, Moon Kyoung-Sook, Anders Szepessy, Raul Tempone, and Georgios Zouraris. Stochastic differential equations: Models and numerics, 2018. URL https://people.kth.se/~szepessy/Notes2018.pdf. lecture notes.

Claudio Carmeli, Ernesto de Vito, and Alessandro Toigo. Vector valued reproducing kernel hilbert spaces of integrable functions and mercer theorem. *Analysis and Applications*, 04:377–408, 2006.

Benoit Chachuat. Nonlinear and dynamic optimization: From theory to practice. 01 2007.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Kristjanson Duvenaud. Neural ordinary differential equations. *ArXiv*, abs/1806.07366, 2018.

Felipe Cucker and Ding Xuan Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2007. doi: 10.1017/CBO9780511618796.

Gianni Dal Maso. *An Introduction to [gamma]-convergence*. 1993.

Alexander Duncan, N. Nuesken, and Lukasz Szpruch. On the geometry of stein variational gradient descent. *ArXiv*, abs/1912.00894, 2019.

Paul Dupuis, Ulf Grenander, and Michael I. Miller. Variational problems on flows of diffeomorphisms for image matching. *Quarterly of Applied Mathematics*, 56(3):587–600, 1998. ISSN 0033569X, 15524485. URL http://www.jstor.org/stable/43638248.

Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5:1–11, 2017.

Lawrence C. Evans. An introduction to mathematical optimal control theory version 0.2, 2013. lecture notes.

Alessio Figalli and Federico Glaudo. *An Invitation to Optimal Transport, Wasserstein Distances, and Gradient Flows*. 08 2021. ISBN 978-3-98547-010-5. doi: 10.4171/ETB/22.

Martin J. Gander, Félix Kwok, and Julien Salomon. PARAOPT: A parareal algorithm for optimality systems. *SIAM Journal on Scientific Computing*, 42(5):A2773–A2802, September 2020. doi: 10.1137/19M1292291. URL https://hal.archives-ouvertes.fr/hal-02346535.

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

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. 1993.

Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. ISBN 3-540-30663-3; 978-3-540-30663-4. Structure-preserving algorithms for ordinary differential equations.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. URL https://faculty.marshall.usc.edu/gareth-james/ISL/.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *ArXiv*, abs/1605.07648, 2017.

Qianxiao Li, Long Chen, Cheng Tai, and E. Weinan. Maximum principle based algorithms for deep learning. *J. Mach. Learn. Res.*, 18(1):5998–6026, jan 2017. ISSN 1532-4435.

Daniel Liberzon. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2012. ISBN 9780691151878. URL http://www.jstor.org/stable/j.ctvcm4g0s.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/b3ba8f1bee1238a2f37603d90b58898d-Paper.pdf.

Michael A. Nielsen. Neural networks and deep learning, 2018. URL http://neuralnetworksanddeeplearning.com/.

Nikolas Nüsken and D. R. Michiel Renger. Stein variational gradient descent: many-particle and long-time asymptotics, 2021. URL https://arxiv.org/abs/2102.12956.

Felix Otto. The geometry of dissipative evolution equations: the porous medium equation. *Communications in Partial Differential Equations*, 26(1-2):101–174, 2001. doi: 10.1081/PDE-100002243. URL https://doi.org/10.1081/PDE-100002243.

Houman Owhadi. Do ideas have shape? plato's theory of forms as the continuous limit of artificial neural networks. *ArXiv*, abs/2008.03920, 2020.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/pascanu13.html.

Lorenzo Rosasco. Reproducing kernel hilbert spaces, 2010. URL http://www.mit.edu/~9.520/scribe-notes/class03_gdurett.pdf. lecture notes.

Tomas Roubicek. *Nonlinear Partial Differential Equations with Applications*, volume 153. 01 2005. ISBN 3-7643-7293-1. doi: 10.1007/3-7643-7397-0.

Schmid Rudolf. Infinite-dimensional lie groups and algebras in mathematical physics. *Advances in Mathematical Physics*, 2010, 01 2010. doi: 10.1155/2010/280362.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

Lars Ruthotto and Eldad Haber. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 62:352–364, 2019.

Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *The Journal of Machine Learning Research*, 2, 01 2002. doi: 10.1162/153244302760185252.

Ingo Steinwart and Andreas Christmann. Support vector machines. In *Information science and statistics*, 2008.

Ding-Xuan Zhou. Derivative reproducing properties for kernel methods in learning theory. *Journal of Computational and Applied Mathematics*, 220(1):456–463, 2008. ISSN 0377-0427. doi: https://doi.org/10.1016/j.cam.2007.08.023. URL https://www.sciencedirect.com/science/article/pii/S0377042707004657.