



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Classical and Quantum approaches to computational optimal transport

TESI DI LAUREA MAGISTRALE IN
MATHEMATICAL ENGINEERING - INGEGNERIA MATEMATICA

Author: **Juan Carlos Gonzalez Espitia**

Student ID: 10779264

Supervisor: Prof. Giulia Cavagnari

Co-supervisor: Prof. Francesca Ieva

External advisors: Dr Jannis Born (IBM Research), Prof. Dr. Marianna Rapsomaniki (IBM Research)

Academic Year: 2023-24

IN MEMORIAM

*In loving memory of
Jose de Jesús Espitia Moreno,
whose legacy I will carry in my soul forever.*

*This thesis is dedicated as a tribute to his
inspiring life and the indelible mark he made
on our hearts.*

March 18th 2024.

Contents

Contents	iii
Introduction	1
1 Background	3
1.1 Mathematical preliminaries	3
1.2 Optimal Transport	6
1.2.1 Transport of measures	7
1.2.2 Monge and Kantorovich problems	10
1.2.3 Properties of the Optimal Transport	12
1.2.4 Kantorovich duality	14
1.2.5 Wasserstein Spaces	15
1.2.6 Brenier's theorem	17
1.3 Quantum computing	20
1.3.1 Qubits and Quantum Logic	23
2 Methods and developments	29
2.1 Classical convex interpolation	29
2.2 Approximating continuous transport maps from discrete samples of continuous probability measures	32
2.3 Quantum Convex regression	33
3 Results	37
3.1 Variational Quantum Circuit Regression	37
3.2 Convex and non-convex regression	42
3.3 Approximating continuous transport maps from discrete samples	43
3.4 VQLS as linear solver for the convex interpolation problem	47
3.4.1 Decomposition of matrix A into complex linear combinations of tensor products of 2-dimensional unitary matrices	48

3.4.2	Preparation of the quantum state b	49
3.4.3	Hadamard test and optimization of the parameters of the VQC . .	49
4	Discussion	51
4.1	On the impracticality of performing convex regression using Variational Quantum Circuits	51
4.2	On the convenience of Convex Splines Interpolation	53
4.3	On the approximation of continuous transport maps from discrete samples	54
4.4	On the use of the VQLS to perform convex interpolation	56
4.5	On the solution of the OT problem using convex interpolation and VQLSs	57
5	Conclusions	59
	Bibliography	61

Introduction

The Optimal Transport (OT) problem originates in the late 18th century after Gaspard Monge studied the problem of transporting construction materials efficiently from a source to a target location. The richness and potential applicability of OT has captivated the attention of researchers inside and outside mathematics for centuries. Nowadays, OT is widely spread and used in fields like Image processing and Computer Vision, Machine Learning and Data Analysis, Economics and Finance, Neuroscience, Differential geometry, and the list goes on and on. However, despite its popularity and great utility it is still an extremely difficult problem from a numerical perspective due to the challenges of its computational implementations.

From an analytical point of view, we can formulate the OT problem following the Kantorovich formulation:

$$\min_{\gamma \in \Gamma(\mu, \nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} |x - y|^2 d\gamma(x, y). \quad (1)$$

Here μ and ν are given probability measures in \mathbb{R}^d , while $\Gamma(\mu, \nu)$ is the set of admissible couplings/transport plans - that is, the set of all Borel probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ having μ and ν as marginals. The OT problem then corresponds in finding the optimal coupling between μ and ν . Such optimality is achieved when $\gamma \in \Gamma(\mu, \nu)$ is realizing the minimum in Equation 1, thus minimizing the weighted distance between the allocations of mass at x and y - in the supports of μ and ν , respectively - and satisfying the constraint. A major result, at the basis of our work, is Brenier's theorem. It states that, under some regularity on the source measure μ , there exists a unique optimal coupling $\gamma \in \Gamma(\mu, \nu)$, and more importantly, this coupling is induced by a map T acting on μ which is given by the gradient of a convex function. Such a map T is called transport map from μ to ν since, when push-forwarded to μ gives the image measure ν .

In this work, developed in conjunction with the AI for scientific discovery and Quantum Algorithms research groups at IBM Research/IBM Quantum, we explored old and new methods to solve the OT problem in classical and quantum settings. The contributions of this work are the following:

1. A modification of criterion of slope-selection in the Convex Splines interpolation method proposed by [Messac and Sivanandan \[1997\]](#),
2. A new formulation to approximate continuous solutions of transport maps from discrete samples based on a linear formulation using Convex Interpolation.
3. An application of the Variational Quantum Linear Solver algorithm to solve the OT problem.

This work is structured as follows. In Chapter 1 we introduce the essentials of both the theory of Optimal Transport and Quantum Computing. Because both subjects are extremely dense and extensive, we limit to the theoretical minimum for this work to be self-contained, and refer the reader to references with deeper discussion on some matters when needed. In Chapter 2 we introduce the methods used in this work both from a mathematical and algorithmic point of view. In particular we introduce our proposed method for convex interpolation, and convex quantum regression. Chapter 3 contains the main results of the thesis and the implementations of all the methods discussed in Chapter 2. We proceed in Chapter 4 with a discussion of the obtained results and analyze the benefits and drawbacks of our methods. We also describe the setting in which they are potentially beneficial and superior to the classical algorithms. In the last Chapter we finish the work with some concluding remarks and some comments on further future developments.

The result of this thesis, other than the proposal and implementation three algorithms listed before, is the study of the feasibility of solving OT problems under particular conditions using diverse classical and quantum methods. In the classical setting, our method is capable of approximating continuous transport maps from discrete samples. In the quantum setting, we are able to frame and solve OT problems under some extra restrictions and considerations. We detail the benefits and drawbacks of each method and propose some extensions to our current proposals.

1 | Background

1.1. Mathematical preliminaries

Modern formulations of optimal transport theory can be framed in various equivalent ways and on different topological spaces. In the context of this work, we will be considering all metric spaces to be complete and separable, unless explicitly stated. For a more general formulation relying on Polish spaces, [Villani \[2021\]](#) offers a great and comprehensive approach to the problem.

For topological reasons that will not be investigated in the present work, the weak* convergence of Borel finite signed measures - namely, the convergence induced by duality with the space of compactly supported continuous functions - is a crucial and natural way to define convergence of finite signed measures. In this subsection, I define that notion of convergence along with narrow convergence, and introduce a compactness result known as the Prokhorov theorem. Finally, we discuss some aspects and properties of narrowly convergent sequences of probability measures.

The main references for this section, and for Section 1.2 are [Ambrosio et al. \[2005\]](#), [Figalli and Glaudo \[2021\]](#), [Villani et al. \[2009\]](#) and [Villani \[2021\]](#).

From now on, let X, Y be complete and separable metric spaces, and $\mathcal{B}(X)$, $\mathcal{B}(Y)$ their corresponding Borel σ -algebras, that is, the smallest sigma algebras generated by all open sets on X and Y , respectively. We denote as $C_c(X)$ the space of continuous and compactly supported functions on X and $C_0(X)$ the space of continuous functions vanishing at infinity.

Definition 1.1 (Space of signed measures on X). *We denote as $\mathcal{M}(X)$ the space of finite signed measures on $(X, \mathcal{B}(X))$.*

Definition 1.2 (Support of a measure). *Let $\mu \in \mathcal{M}(X)$, we define:*

$$\text{supp}(\mu) = \{x \in X \mid \mu(B_\epsilon(x)) > 0 \forall \epsilon > 0\}.$$

Definition 1.3 (Space of Borel probability measures on X). We denote as $\mathcal{P}(X)$ the space of all probability measures on $(X, \mathcal{B}(X))$.

Remark 1.4. The spaces $\mathcal{M}(X)$ and $\mathcal{P}(X)$ satisfy that:

1. $\mathcal{P}(X) \subset \mathcal{M}(X)$. Indeed, $\mathcal{P}(X)$ is the set of all finite measures μ on $(X, \mathcal{B}(X))$ such that $\mu(X) = 1$
2. $\mathcal{P}(X)$ is a convex subset of $\mathcal{M}(X)$. Indeed, any convex combination of probability measures is a probability measure.
3. Depending on the type of convergence adopted, $\mathcal{P}(X)$ might be closed on $\mathcal{M}(X)$.

Motivated by topological properties of $\mathcal{M}(X)$, we consider the following notion of weak convergence on $\mathcal{P}(X) \subset \mathcal{M}(X)$.

Definition 1.5 (weak* convergence). Let $\{\mu_k\}_{k \in \mathbb{N}}$ be a sequence of Borel probability measures on X . We say that $\{\mu_k\}_{k \in \mathbb{N}}$ weak* converges to some $\mu \in \mathcal{M}(X)$ if

$$\int_X \varphi d\mu_k \rightarrow \int_X \varphi d\mu, \quad \text{as } k \rightarrow +\infty \quad \forall \varphi \in C_c(X),$$

and we denote it by $\mu_k \xrightarrow{*} \mu$.

Observe that $\mathcal{P}(X)$ is not closed on $\mathcal{M}(X)$ with respect to the weak* convergence. We have the following remark.

Remark 1.6. Let $X = \mathbb{R}$, and $\mu_k = \delta_k$ for $k \in \mathbb{Z}$. Then, for any $\varphi \in C_c(\mathbb{R})$:

$$\int_{\mathbb{R}} \varphi d\mu_k = \varphi(k) \rightarrow 0 \text{ as } k \rightarrow +\infty$$

Hence $\mu_k \xrightarrow{*} 0 \notin \mathcal{P}(\mathbb{R})$, even if $0 \in \mathcal{M}(\mathbb{R})$.

And so, the weak* limit of sequences of probability measures, when it exists, is not necessarily a probability measure. To guarantee that the limit is still a probability measure, we introduce the notion of narrow convergence.

Definition 1.7 (Narrow Convergence). Let $C_b(X)$ be the set of continuous bounded func-

tions. We say that $\{\mu_k\}_{k \in \mathbb{N}} \subset \mathcal{P}(X)$ converges to $\mu \in \mathcal{P}(X)$ narrowly if

$$\int_X \varphi \, d\mu_k \rightarrow \int_X \varphi \, d\mu \quad \text{for any } \varphi \in C_b(X)$$

We denote this convergence by $\mu_k \rightarrow \mu$ narrowly .

Since $C_c(X) \subset C_b(X)$ we deduce that the narrow notion of convergence is stronger than the weak* one.

Remark 1.8. *Narrow convergence guarantees that the limit of a sequence of probability measures, if exists, is still a probability measure. Indeed, let $\mu_k \in \mathcal{P}(X)$, $\mu \in \mathcal{M}(X)$ be such that $\mu_k \rightarrow \mu$ narrowly. Then, taking $C_b(X) \ni \varphi \equiv 1$ yields:*

$$\mu_k(X) = \int_X 1 \, d\mu_k \longrightarrow \int_X 1 \, d\mu = \mu(X).$$

That is, $\mu \in \mathcal{P}(X)$.

Now, we look for conditions in order to have sequential compactness results for subsets of $\mathcal{P}(X)$ under narrow convergence.

Definition 1.9 (Tightness of measures). *Let $K \subset \mathcal{P}(X)$ be a family of probability measures. We say that K is tight if for any $\varepsilon > 0$ there exists a compact set $C_\varepsilon \subset X$ such that $\mu(X \setminus C_\varepsilon) \leq \varepsilon$ for any $\mu \in K$.*

Intuitively, we can think that the elements of a tight family of probability measures do not spread out too much.

Theorem 1.10 (Prokhorov). *A family $K \subset \mathcal{P}(X)$ is tight if and only if K is relatively compact for the narrow convergence in $\mathcal{P}(X)$. That is, if and only if for any sequence $\{\mu_k\}_{k \in \mathbb{N}} \subset K$ there exists a subsequence $\{\mu_{k_j}\}_{j \in \mathbb{N}}$ and a probability measure $\mu \in \mathcal{P}(X)$ such that*

$$\mu_{k_j} \rightarrow \mu \text{ narrowly as } j \rightarrow +\infty$$

Definition 1.11 (Lower semicontinuity). *A function $\phi : X \rightarrow [-\infty, \infty]$ is lower semicontinuous if for any $\{x_k\}_{k \in \mathbb{N}} \subset X, x \in X$ such that $x_k \rightarrow x$, it follows that $\phi(x) \leq \liminf_{k \rightarrow \infty} \phi(x_k)$.*

Lemma 1.11.1 (Lower semicontinuity of integrals). *Let $\mu_k \xrightarrow{*} \mu$, and let $\phi : X \rightarrow [0, +\infty]$*

be a lower semicontinuous function. Then,

$$\int_X \phi \, d\mu \leq \liminf_{k \rightarrow +\infty} \int_X \phi \, d\mu_k$$

Thanks to the lower semicontinuity of the integral, it can be proved that if a sequence of probability measures weakly* converges to a probability measure, then the convergence is narrow.

Lemma 1.11.2. *Let $\{\mu_k\}_{k \in \mathbb{N}} \subset \mathcal{P}(X)$, and assume that $\mu_k \xrightarrow{*} \mu$ for some $\mu \in \mathcal{P}(X)$. Then:*

$$\mu_k \rightarrow \mu \quad \text{narrowly}$$

1.2. Optimal Transport

The problem of Optimal Transport is long-standing and has many different flavors. It all started in the late 18th century with Gaspard Monge who addressed the problem of transporting a pile of construction materials from one location to another. In order to find the optimal way of transporting the materials, he introduced a displacement cost as the absolute distance between source and target points. The problem of Monge corresponds to look for a map from the source configuration to the target such that the pre-defined cost is minimized in the transportation process.

A similar problem was later framed by Leonid Kantorovich during the mid 20th century in the context of optimal allocation of resources - problem that granted him the Nobel prize in economics in 1975 along with Tjalling Koopmans -. That is, the problem of how to transport resources from production sites to consumption sites by minimizing a given cost. Unlike Monge, who took a more geometrical approach, Kantorovich applied more modern tools from functional analysis and measure theory to introduce a new notion of distance between probability measures. That distance is naturally connected to the solution of a transportation problem where the cost function is another distance in the *physical* space. Along with the formulation of the problem, he realized that he can frame it in the context of linear programming, and formulated a duality theorem of the utmost importance for subsequent developments.

By the last decade of the 20th century, optimal transport had already gained a significant popularity with applications in diverse fields. Between many other significant contributors, Yann Brenier notably stated a theorem relating optimal transport and convex analysis. He proved that, for quadratic cost functions and certain regularity conditions

on the involved measures, an optimal transport map exists and that it can be written as the gradient of a convex function.

Recently, optimal transport is again growing in popularity, mainly thanks to its applications to geometrical data analysis and machine learning. Notably, during the 21st century, Alessio Figalli - who was awarded the Fields medal for his contributions to optimal transport - has made significant advances in the study of the regularity and stability of optimal transport maps, and in the study of the geometrical and analytical aspects of the problem. On the computational side, Gabriel Peyrè and Marco Cuturi have contributed to the development of numerical methods and algorithms, specially introducing Sinkhorn distances as approximators of Wasserstein distances. In particular, they have a great responsibility in the popularization of the problem in the fields of machine learning, geometrical data analysis and computer vision, due to their study on applications in those fields.

In this section we introduce the theoretical foundations of optimal transport following as reference [Ambrosio et al. \[2005\]](#), [Figalli and Glaudo \[2021\]](#), [Villani et al. \[2009\]](#), and [Villani \[2021\]](#). First we define the image measure as a way of transporting measures using maps, enunciate its properties and importance. Then, we formulate the Monge-Kantorovich problems for general cost functions, and explore some analytical properties of optimal solutions. We then move to Brenier's formulation and finalize with a study of Wasserstein spaces.

1.2.1. Transport of measures

Definition 1.12. *Given X, Y and a Borel map $T : X \rightarrow Y$, let $\mu \in \mathcal{P}(X)$. We define the image measure $T_{\#}\mu$, also known as the push-forward of μ , as:*

$$(T_{\#}\mu)(E) = \mu(T^{-1}(E))$$

for every $E \in \mathcal{B}(Y)$.

Lemma 1.12.1. *$T_{\#}\mu$ is a probability measure on Y .*

Proof. Clearly $(T_{\#}\mu)(\emptyset) = \mu(T^{-1}(\emptyset)) = \mu(\emptyset) = 0$.

Observe that, because T is a Borel map, then for every $E \in \mathcal{B}(Y)$ there exist a $F \in \mathcal{B}(X)$ such that $T^{-1}(E) = F$. Then, $(T_{\#}\mu)(E) = \mu(T^{-1}(E)) = \mu(F) \geq 0$.

To prove σ -additivity, let $\{E_n\}_{n \in I}$ be countable pairwise disjoint sequence of elements in $\mathcal{B}(Y)$. First, we claim that $\{T^{-1}(E_n)\}_{n \in \mathbb{N}} \subset \mathcal{B}(X)$ are disjoint. Indeed, by contradiction,

take $x \in T^{-1}(E_n) \cap T^{-1}(E_m)$, $n, m \in I, n \neq m$, then $T(x) \in E_n \cap E_m$ which is a contradiction. Therefore, by σ -additivity:

$$\begin{aligned} (T_{\#}\mu) \left(\bigcup_{n \in I} E_n \right) &= \mu \left(T^{-1} \left(\bigcup_{n \in I} E_n \right) \right) \\ &= \mu \left(\bigcup_{n \in I} T^{-1}(E_n) \right) \\ &= \sum_{n \in I} \mu(T^{-1}(E_n)) \\ &= \sum_{n \in I} (T_{\#}\mu)(E_n) \end{aligned}$$

Finally, observe that $(T_{\#}\mu)(Y) = \mu(T^{-1}(Y)) = \mu(X) = 1$. That is, $T_{\#}\mu \in \mathcal{P}(Y)$.

□

Remark 1.13. We think of a push-forward operator $(T_{\#}\mu)(A)$ instead of a pull-back operator $(S^{\#}\nu)(E) = \nu(S(E))$ for $S : X \rightarrow Y$, $\nu \in \mathcal{P}(Y)$ because the latter construction doesn't work in general. In fact, note that the image of two disjoint sets in X might coincide, and then $S^{\#}\nu$ will not be σ -additive in disjoint sets.

Lemma 1.13.1. Let $T : X \rightarrow Y, \mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y)$. Then

$$\nu = T_{\#}\mu$$

if and only if, for any $\varphi : Y \rightarrow \mathbb{R}$ Borel and bounded we have

$$\int_Y \varphi(y) d\nu(y) = \int_X \varphi(T(x)) d\mu(x)$$

Proof. (\Leftarrow) Let $E \in \mathcal{B}(Y)$, and $\varphi = \mathbb{1}_E$, that is $\varphi(x) = 1$ for every $x \in E$, and 0 otherwise. Then:

$$\nu(E) = \int_E d\nu(y) = \int_Y \mathbb{1}_E d\nu(y) = \int_{T^{-1}(E)} d\mu(x) = \mu(T^{-1}(E)) = (T_{\#}\mu)(E).$$

(\Leftarrow) For any $E \in \mathcal{B}(Y)$, $\nu(E) = \mu(T^{-1}(E))$ if and only if

$$\int_E d\nu(y) = \int_{T^{-1}(E)} d\mu(x).$$

Observe that:

$$\int_X \mathbb{1}_{T^{-1}(E)}(x) d\mu(x) = \int_{T^{-1}(E)} d\mu(x) = \int_X \mathbb{1}_E(T(x)) d\mu(x)$$

Then, for every $\lambda \in \mathbb{R}$, and by linearity:

$$\int_Y \lambda \mathbb{1}_E(y) d\nu(y) = \int_X \lambda \mathbb{1}_E(T(x)) d\mu(x),$$

which implies that,

$$\int_Y \sum_{i \in I} \lambda_i \mathbb{1}_{E_i}(y) d\nu(y) = \int_X \sum_{i \in I} \lambda_i \mathbb{1}_{E_i}(T(x)) d\mu(x)$$

for any I finite set of indices, $\{\lambda_i\}_{i \in I} \subseteq \mathbb{R}$, $\{E_i\}$ Borel subsets of Y .

It is clear that the desired equality follows whenever φ is a simple function. Therefore, it suffices to prove that any bounded Borel function can be approximated by simple functions. To do so, take $\phi : Y \rightarrow \mathbb{R}$ bounded and Borel map, $M \gg 0$ and for any $i \in \mathbb{Z}$ define: $E_i := \{y \in Y : \frac{i}{M} \leq \varphi(y) < \frac{i+1}{M}\}$. Let $\varphi_M := \sum_{i \in \mathbb{Z}} \frac{i}{M} \mathbb{1}_{E_i}$ (since φ is bounded, $E_i = \emptyset$ for $|i| \gg 1$). Then

$$\|\varphi - \varphi_M\|_{L^\infty} \leq \max_{i \in \mathbb{Z}} \|\varphi - \varphi_M\|_{L^\infty(E_i)} \leq \frac{1}{M},$$

and therefore:

$$\begin{aligned} \left| \int_Y (\varphi - \varphi_M) d\nu \right| &\leq \|\varphi - \varphi_M\|_{L^\infty} \int_Y d\nu \leq \frac{1}{M} \\ \left| \int_X (\varphi(T(x)) - \varphi_M(T(x))) d\mu \right| &\leq \|\varphi - \varphi_M\|_{L^\infty} \int_X d\mu \leq \frac{1}{M} \end{aligned}$$

Hence, since φ_M is simple:

$$\int_Y \varphi_M(y) d\nu(y) = \int_X \varphi_M(T(x)) d\mu(x)$$

and so, by letting $M \rightarrow +\infty$ it follows that $\int_X \varphi(y) d\nu(y) = \int_X \varphi(T(x)) d\mu(x)$. \square

Corollary 1.14. *For any function $\varphi : Y \rightarrow \mathbb{R}$ Borel and bounded, it holds that:*

$$\int_Y \varphi(y) d(T_{\#}\mu)(y) = \int_X (\varphi \circ T)(x) d\mu(x).$$

Lemma 1.14.1. *Let $T : X \rightarrow Y$ and $S : Y \rightarrow Z$ measurable. Then,*

$$(S \circ T)_{\#}\mu = S_{\#}(T_{\#}\mu)$$

Proof. For any $\varphi : Z \rightarrow \mathbb{R}$ Borel and bounded, by Corollary 1.14:

$$\begin{aligned} \int_X \varphi d((S \circ T)_{\#}\mu) &= \int_X (\varphi \circ (S \circ T)) d\mu \\ &= \int_X (\varphi \circ S) \circ T) d\mu \\ &= \int_X (\varphi \circ S) d(T_{\#}\mu) \\ &= \int_X \varphi d(S_{\#}(T_{\#}\mu)) \end{aligned}$$

□

1.2.2. Monge and Kantorovich problems

Let X, Y be separable and complete metric spaces.

Definition 1.15 (Transport map). *Let $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$, and $T : X \rightarrow Y$. We say that T is a transport map from μ to ν if $(T)_{\#}\mu = \nu$.*

Definition 1.16 (Monge problem). *Let $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$, and $c : X \times Y \rightarrow [0, +\infty]$ a lower semicontinuous function. The Monge problem consists in solving:*

$$C_M(\mu, \nu) := \inf \left\{ \int_X c(x, T(x)) d\mu(x) : T_{\#}\mu = \nu \right\}$$

That is, Monge's problem looks for the optimal transport map T such that the cost c of transporting μ to ν is minimized. A major drawback of this formulation, is that, despite being simple, it does not always have a solution. This fact motivates the following remark.

Remark 1.17 (A transportation map might not always exist). *The set $\mathcal{T} = \{T : X \rightarrow Y \mid$*

$T_{\#}\mu = \nu$ } might be empty. Take for instance $x_0 \in X$, $\mu = \delta_{x_0}$, and some $T : X \rightarrow Y$. Using Lemma 1.13.1, for every $\varphi : Y \rightarrow \mathbb{R}$ Borel and bounded, we compute:

$$\int_Y \varphi(y) d(T_{\#}\mu)(y) = \int_X (\varphi \circ T)(x) d\mu(x) = \varphi(T(x_0)),$$

and so $T_{\#}\mu = \delta_{T(x_0)}$. Hence, for $y_1, y_2 \in Y$, if $\nu = \frac{1}{2}\delta_{y_1} + \frac{1}{2}\delta_{y_2}$, there doesn't exist any map T such that $T_{\#}\mu = \nu$.

The possible emptiness of \mathcal{T} makes it impossible for Monge's problem to solve situations where the map has to *split masses*, that is, when it has to map values from single points in X to many different points in Y . Kantorovich's formulation solves this issue by relaxing the hypothesis that the solution of the transport problem is a map, and instead using probability measures on product spaces. Before stating this version of the OT problem, we need to introduce a couple of definitions.

Definition 1.18 (Projection operator). We define $\pi_X : X \times Y \rightarrow X$, and $\pi_Y : X \times Y \rightarrow Y$ projection operators on X and Y , respectively, if for every $(x, y) \in (X, Y)$:

$$\pi_X(x, y) = x \qquad \pi_Y(x, y) = y.$$

Definition 1.19 (Coupling on product spaces). Let $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$, $\Omega = X \times Y$. A measure $\gamma \in \mathcal{P}(\Omega)$ is called a coupling between μ and ν if $(\pi_X)_{\#}\gamma = \mu$ and $(\pi_Y)_{\#}\gamma = \nu$. We denote the set of all possible couplings between μ and ν as $\Gamma(\mu, \nu)$.

That is, a coupling between a pair of measures (μ, ν) is a special measure in the product space $X \times Y$ such that its marginals coincide with μ and ν .

Remark 1.20 (A coupling always exists). Observe that the product measure $\mu \otimes \nu \in \mathcal{P}(X \times Y)$, defined by $(\mu \otimes \nu)(A \times B) = \mu(A)\nu(B)$ for all $A \in \mathcal{B}(X), B \in \mathcal{B}(Y)$, and so, $\gamma = \mu \otimes \nu$ trivially satisfies the condition of Definition 1.19.

A case of significant importance for the development of this work is that of deterministic couplings. It will play a relevant role on behalf of Brenier's theorem 1.35 and its numerical applications.

Definition 1.21 (Deterministic coupling). A coupling $\gamma \in \mathcal{P}(X, Y)$ is said to be deterministic if there exists a Borel function $T : X \rightarrow Y$ such that $\gamma = (Id \times T)_{\#}\mu$, where $Id : X \rightarrow X$ is the identity map, and $\mu := (\pi_X)_{\#}\gamma$.

Remark 1.22. If γ is a deterministic coupling, we say that it is concentrated on the graph

of the Borel function $T : X \rightarrow Y$. Moreover, this map T , if exists, is unique μ -a.e. Notice that we get $(\pi_Y)_\# \gamma = T((\pi_X)_\# \gamma)$

We stress that not all couplings are deterministic.

For an excellent exemplifications of various couplings and their application contexts we refer the reader to Chapter 1 of Villani et al. [2009]. Here we are only concerned about the notion optimal transport coupling as it gives origin to the Kantorovich problem.

Definition 1.23 (Kantorovich problem). Let $\mu \in \mathcal{P}(X), \nu \in \mathcal{P}(Y)$, and $c : X \times Y \rightarrow [0, +\infty]$ be lower semicontinuous. The Kantorovich problem consists in solving

$$C_K(\mu, \nu) = \inf \left\{ \int_{X \times Y} c(x, y) d\gamma(x, y) : \gamma \in \Gamma(\mu, \nu) \right\}$$

Where $\Gamma(\mu, \nu)$ is the set of all couplings from μ to ν , this set is also commonly known as the set of all transportation plans from μ to ν .

Remark 1.24. If $T_\# \mu = \nu$, then $(Id \times T)_\# \mu$ is a deterministic coupling between μ and ν , and so $\gamma \in \Gamma(\mu, \nu)$. Also, Corollary 1.14 entails that

$$\begin{aligned} \int_X c(x, T(x)) d\mu(x) &= \int_X c \circ (Id \times T)(x) d\mu(x) \\ &= \int_X c(x, y) d((Id \times T)_\# \mu) \end{aligned}$$

and so, $C_M(\mu, \nu) \geq C_K(\mu, \nu)$.

Remark 1.25. Let $\gamma \in \Gamma(\mu, \nu)$ and assume that $\gamma = (Id \times S)_\# \mu$ for some bounded Borel map $S : X \rightarrow Y$. Then

$$\begin{aligned} \nu &= (\pi_Y)_\# \gamma = (\pi_Y)_\# (Id \times S)_\# \mu \\ &= (\pi_Y \circ (Id \times S))_\# \mu \\ &= S_\# \mu \end{aligned}$$

Thus, S is a transport map from μ to ν .

Consequently if the coupling has a graph structure then it yields a transportation map.

1.2.3. Properties of the Optimal Transport

First observe that the transport plans are tight and closed under narrow convergence.

Lemma 1.25.1. *The set $\Gamma(\mu, \nu)$ is tight in $X \times Y$ according to definition 1.9. Moreover, if $\{\gamma_k\}_k \subset \Gamma(\mu, \nu)$ are such that $\gamma_k \rightarrow \gamma$ narrowly as $k \rightarrow \infty$, then $\gamma \in \Gamma(\mu, \nu)$.*

Proof. Observe that the singleton $\{\mu\} \subset \mathcal{P}(X)$ is compact for the narrow convergence in $\mathcal{P}(X)$, thus by Theorem 1.10, $\{\mu\}$ is tight. Consequently, for all $\varepsilon > 0$, there exists a compact set $K_\varepsilon \subset X$ such that $\mu(X \setminus K_\varepsilon) \leq \frac{\varepsilon}{2}$. Under the same argument for ν , we see that $\{\nu\}$ is tight in $\mathcal{P}(Y)$ and thus there exists a compact set $\tilde{K}_\varepsilon \subset Y$ such that $\nu(Y \setminus \tilde{K}_\varepsilon) \leq \frac{\varepsilon}{2}$. Define the set $\hat{K}_\varepsilon := K_\varepsilon \times \tilde{K}_\varepsilon \subset X \times Y$. This set is clearly compact. Then, for any $\gamma \in \Gamma(\mu, \nu)$, we have

$$\begin{aligned} \gamma\left((X \times Y) \setminus \hat{K}_\varepsilon\right) &= \gamma\left(\left[(X \setminus K_\varepsilon) \times Y\right] \cup \left[X \times (Y \setminus \tilde{K}_\varepsilon)\right]\right) \\ &\leq \gamma\left((X \setminus K_\varepsilon) \times Y\right) + \gamma\left(X \times (Y \setminus \tilde{K}_\varepsilon)\right) \\ &= \int_{X \times Y} \mathbb{1}_{X \setminus K_\varepsilon}(x) d\gamma(x, y) + \int_{X \times Y} \mathbb{1}_{Y \setminus \tilde{K}_\varepsilon}(y) d\gamma(x, y) \\ &= \int_X \mathbb{1}_{X \setminus K_\varepsilon}(x) d\mu(x) + \int_Y \mathbb{1}_{Y \setminus \tilde{K}_\varepsilon}(y) d\nu(y) \\ &= \mu(X \setminus K_\varepsilon) + \nu(Y \setminus \tilde{K}_\varepsilon) \\ &\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon. \end{aligned}$$

This proves that $\Gamma(\mu, \nu)$ is tight in $X \times Y$.

On the other hand, let $\{\gamma_k\} \subset \Gamma(\mu, \nu)$, and assume that $\gamma_k \rightarrow \gamma$ narrowly. Then, for any $\varphi \in C_b(X)$:

$$\int_X \varphi(x) d\mu(x) = \int_{X \times Y} \varphi(x) d\gamma_k(x, y) \rightarrow \int_{X \times Y} \varphi(x) d\gamma(x, y),$$

which means that $(\pi_X)_\# \gamma = \mu$. In turn, let $\psi \in C_b(Y)$, then:

$$\int_Y \psi(y) d\nu(y) = \int_{X \times Y} \psi(y) d\gamma_k(x, y) \rightarrow \int_{X \times Y} \psi(y) d\gamma(x, y),$$

and so $(\pi_Y)_\# \gamma = \nu$. That is, $\Gamma(\mu, \nu)$ is closed under narrow convergence. \square

Theorem 1.26 (Existence of an optimal coupling). *Let $c : X \times Y \rightarrow [0, +\infty]$ be a lower semicontinuous function, $\mu \in \mathcal{P}(X), \nu \in \mathcal{P}(Y)$. Then, there exists a coupling $\bar{\gamma} \in \Gamma(\mu, \nu)$ such that $\bar{\gamma}$ is the minimizer of the Kantorovich problem (Definition 1.23).*

Proof. Let us define:

$$\alpha := \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{X \times Y} c(x, y) d\gamma(x, y).$$

If $\alpha = +\infty$, there is nothing to prove. If instead $\alpha < \infty$, let $\{\gamma_k\}_{k \in \mathbb{N}} \subset \Gamma(\mu, \nu)$ be a minimizing sequence of c , that is,

$$\int_{X \times Y} c(x, y) d\gamma_k(x, y) \rightarrow \alpha \quad \text{as } k \rightarrow +\infty.$$

Because $\{\gamma_k\}_{k \in \mathbb{N}} \subset \Gamma(\mu, \nu)$, the sequence is tight (see Lemma 1.25.1). Then, by Prokhorov's Theorem 1.10, there exists a subsequence $\{\gamma_{k_j}\}_{j \in \mathbb{N}}$ such that $\gamma_{k_j} \rightarrow \bar{\gamma}$ narrowly as $j \rightarrow +\infty$. Observe that c is positive and lower semi-continuous, hence by Lemma 1.11.1, we have that

$$\alpha = \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{X \times Y} c(x, y) d\gamma(x, y) = \liminf_{j \rightarrow +\infty} \int_{X \times Y} c(x, y) d\gamma_{k_j}(x, y) \geq \int_{X \times Y} c(x, y) d\bar{\gamma}(x, y),$$

as $j \rightarrow +\infty$.

On the other hand, again by Lemma 1.25.1, the sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ is also closed under narrow convergence hence $\bar{\gamma} \in \Gamma(\mu, \nu)$. Then,

$$\alpha = \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{X \times Y} c(x, y) d\gamma(x, y) \leq \int_{X \times Y} c(x, y) d\bar{\gamma}(x, y)$$

From both inequalities, it follows that $\bar{\gamma}$ is the minimizer of the Kantorovich problem. \square

1.2.4. Kantorovich duality

So far we have been tackling Optimal Transport as a minimization problem, looking for either a map - in Monge's formulation - or a plan - in Kantorovich's formulation - that minimizes the total cost of transporting elements from a source probability measure μ to a target probability measure ν . In particular, in Kantorovich's formulation, we minimize the linear functional $\gamma \rightarrow \int_{X \times Y} c(x, y) d\gamma(x, y)$ under the convex constraints $(\pi_X)_\# \gamma = \mu$ and $(\pi_Y)_\# \gamma = \nu$. It is well known that this type of problem admits a dual formulation where instead of looking for the minimum on the original - or primal - problem we look for a maximum in a new problem such that its value is equivalent to that of the primal problem.

Definition 1.27 (Dual problem). *Let $\mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y)$. We define the dual*

problem as:

$$\sup_{(\varphi, \psi) \in \mathcal{A}_c} \left\{ \int_X \varphi(x) d\mu(x) + \int_Y \psi(y) d\nu(y) \right\}$$

where \mathcal{A}_c is the set of pairs (φ, ψ) such that $\varphi \in L^1_\mu(X)$, $\psi \in L^1_\nu(Y)$ and for μ -a.e. $x \in X$ and ν -a.e. $y \in Y$:

$$\varphi(x) + \psi(y) \leq c(x, y)$$

Theorem 1.28 (Kantorovich duality). *Let $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$, and $c : X \times Y : [0, +\infty]$ a lower semicontinuous cost function. Then:*

$$\inf_{\gamma \in \Gamma(\mu, \nu)} \int_{X \times Y} c(x, y) d\gamma(x, y) = \sup_{(\varphi, \psi) \in \mathcal{A}_c} \left\{ \int_X \varphi(x) d\mu(x) + \int_Y \psi(y) d\nu(y) \right\}.$$

Proof. We refer the author to Remark 2.6.7 of Figalli and Glaudo [2021] for the proof of this theorem. \square

Observe that Kantorovich duality changes the optimization problem from $\Gamma(\mu, \nu)$ to $L^1(X) \times L^1(Y)$ which is easier in general. There are other equivalent formulations of Kantorovich duality that involve different spaces of functions - see theorem 5.10 from Villani et al. [2009]. In particular, motivated by the computational implementations of optimal transport we are highly interested in the version of Kantorovich duality employing concave and convex functions and its relationship with cyclic monotonicity. To be concise, we avoid to report here the expression of this cited alternative duality formula. However, we still state and use one of its most powerful repercussions, the Brenier theorem, in Section 1.2.6.

1.2.5. Wasserstein Spaces

Wasserstein spaces provide a natural setting for studying optimal transport problems. The Wasserstein distance quantifies the cost of transporting mass between probability measures, and Wasserstein spaces serve as the domain for these measures. Properties of Wasserstein spaces, such as completeness and convexity, are essential for formulating and analyzing optimal transport problems. Wasserstein spaces also are crucial for the problem of gradient flows, especially for the study of evolution of energy functionals.

In what is next, we introduce the notion of Wasserstein distance, Wasserstein space, and

mention a couple of interesting properties. We take (X, d) a complete and separable metric space, later we will consider the case $X = \mathbb{R}^d$ endowed with the euclidean metric.

Definition 1.29 (Wasserstein distance). *Let $p \in [1, +\infty)$, and d a metric in X . For any two probability measures μ, ν on X , we define:*

$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu, \nu)} \int_{X \times Y} d(x, y)^p d\gamma(x, y) \right)^{\frac{1}{p}}$$

Notice that finding $W_p(\mu, \nu)$ corresponds to solve the Kantorovich problem with $X = Y$ and $c(x, y) = d(x, y)^p$.

If we restrict $W_p(\cdot, \cdot)$ to a subset of $\mathcal{P}(X) \times \mathcal{P}(X)$ on which it takes finite values, then we construct the Wasserstein spaces, and $W_p(\cdot, \cdot)$ turns out to be a distance on these spaces.

Definition 1.30 (Wasserstein spaces). *Let $p \in [1, +\infty)$. The Wasserstein space of order p is defined as:*

$$P_p(X) := \left\{ \mu \in \mathcal{P}(X) : \int_X d(x_0, x)^p d\mu(x) < \infty \right\},$$

where $x_0 \in X$ is arbitrary. This space does not depend on the choice of the point x_0 . Additionally, we denote by

$$m_p(\mu) := \int_X d(x_0, x)^p d\mu(x)$$

the p -th order moment of μ .

Consequently, the Wasserstein space of order p is the space of probability measures on X with finite moments of order p . As mentioned, W_p is a metric space on $P_p(X)$, so that we say that $\{\mu_n\}_n \subset P_p(X)$ converges strongly to $\mu \in P_p(X)$ if $W_p(\mu, \mu_n) \rightarrow 0$ as $n \rightarrow \infty$.

Theorem 1.31 (Strong convergence on $(P_p(X), W_p)$). *Let $\mu \in P_p(X)$ and $\{\mu_n\}_{n \in \mathbb{N}} \subset P_p(X)$. Then $W_p(\mu, \mu_n) \rightarrow 0$ as $n \rightarrow +\infty$ if and only if both of the following hold:*

1. $\mu_n \rightarrow \mu$ narrowly as $n \rightarrow +\infty$.
2. $\int_X d(x_0, x)^p d\mu_n(x) \rightarrow \int d(x_0, x)^p d\mu(x)$.

1.2.6. Brenier's theorem

Definition 1.32 (Subdifferential). *Let $\varphi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ convex. We define its subdifferential at $x \in \mathbb{R}^d$ as:*

$$\partial\varphi(x) := \{y \in \mathbb{R}^d \mid \forall z \in \mathbb{R}^d : \varphi(z) \geq \varphi(x) + \langle y, z - x \rangle\}.$$

Also, we define:

$$\partial\varphi := \bigcup_{x \in \mathbb{R}^d} \{x\} \times \partial\varphi(x) \subset \mathbb{R}^d \times \mathbb{R}^d.$$

It can be proven that if φ is differentiable at $x \in \mathbb{R}^d$, then $\partial\varphi(x) = \{\nabla\varphi(x)\}$.

Justified by Rockefellar's theorem (see Figalli and Glaudo [2021, Theorem 2.4.2]), we consider the following definition.

Definition 1.33. *A set $S \subseteq \mathbb{R}^d \times \mathbb{R}^d$ is said to be cyclically monotone if and only if there exists a convex function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ such that $S \subset \partial\varphi$.*

Definition 1.34 (Legendre transform). *Let $\varphi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ (with $\varphi \not\equiv +\infty$) be a convex function. We define its Legendre's transform $\varphi^* : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ as:*

$$\varphi^*(y) := \sup_{x \in \mathbb{R}^d} \{x \cdot y - \varphi(x)\}$$

Proposition 1.1 (Properties of Legendre's transform). *The following properties hold:*

1. $\varphi(x) + \varphi^*(y) \geq x \cdot y$ for all $x, y \in \mathbb{R}^d$
2. $\varphi(x) + \varphi^*(y) = x \cdot y$ if and only if $y \in \partial\varphi(x)$.

Kantorovich duality provides a rich framework to study optimal transport from analytical, geometrical and numerical perspectives. The special case of quadratic cost functions is remarkably important. This cost appears naturally in the contexts of physics, economy, and statistical inference among many others, and so it is natural that it has a special result. Brenier's theorem provides conditions for existence and uniqueness of optimal transport plans of quadratic cost functions. It also sets the OT problem in the framework of convex optimization.

Theorem 1.35 (Brenier). *Let $X = Y = \mathbb{R}^d$ and $c(x, y) = \frac{|x-y|^2}{2}$. Suppose that μ and ν*

have finite second moments:

$$\int_{\mathbb{R}^d} |x|^2 d\mu(x) + \int_{\mathbb{R}^d} |y|^2 d\nu(y) < +\infty$$

and that $\mu \ll dx$, that is μ is absolutely continuous with respect to the Lebesgue measure on \mathbb{R}^d . Then, there exists a unique optimal transport plan $\bar{\gamma} \in \Gamma(\mu, \nu)$. In addition, $\bar{\gamma} = (Id \times T)_{\#}(\mu)$, and $T = \nabla\phi$ for some convex function ϕ .

Proof. Let us start by observing that the quadratic cost is non-negative and continuous, hence lower-semicontinuous. Therefore, by Theorem 1.26 there exists an optimal transport plan $\bar{\gamma}$. It can be proven that $\text{supp}(\bar{\gamma}) \subset \partial\varphi$ for some convex function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ - that is, $\text{supp}(\bar{\gamma})$ is cyclically monotone. By Proposition 1.1, $\varphi(x) + \varphi^*(y) = x \cdot y$ for y in the subdifferential of φ at x . This implies that

$$\varphi(x) + \varphi^*(y) = x \cdot y \quad \text{for } (x, y) \in \text{supp}(\bar{\gamma}).$$

In particular, $(\varphi(x), \varphi^*(y))$ is finite $\bar{\gamma}$ -a.e., which implies that $\varphi(x)$ is finite μ -a.e. Also, since $\mu \ll dx$ by Alexandrov's theorem -see [Rockafellar, 2015] Theorem 25.5-, φ is differentiable μ -a.e.

Now, let $A \subset \mathbb{R}^d$ be a μ -zero measure set such that φ is differentiable everywhere in $\mathbb{R}^d \setminus A$. Taking $\bar{x} \in \mathbb{R}^d \setminus A$ and suppose that $(\bar{x}, \bar{y}) \in \text{supp}(\bar{\gamma})$. Then

$$\begin{aligned} \varphi(\bar{x}) + \varphi^*(\bar{y}) &= \bar{x} \cdot \bar{y}, \\ \varphi(z) + \varphi^*(\bar{y}) &\geq z \cdot \bar{y} \quad \forall z \in \mathbb{R}^d, \end{aligned}$$

which implies that

$$\Phi_{\bar{x}}(z) := \varphi(z) - \varphi(\bar{x}) - \langle \bar{y}, z - \bar{x} \rangle \geq 0, \quad .$$

Thanks to the differentiability of φ at \bar{x} , $\Phi_{\bar{x}}$ is also differentiable at \bar{x} . Hence, since $\Phi_{\bar{x}}$ has a minimum at \bar{x} , we deduce that

$$0 = \nabla\Phi_{\bar{x}}(\bar{x}) = \nabla\varphi(\bar{x}) - \bar{y} \quad \Rightarrow \quad \bar{y} = \nabla\varphi(\bar{x}).$$

for all $\bar{x} \in \mathbb{R}^d \setminus A$ and $(\bar{x}, \bar{y}) \in \text{supp}(\bar{\gamma})$. And because $\bar{\gamma}(A \times \mathbb{R}^d) = \mu(A) = 0$, then

$$(x, y) = (x, \nabla\varphi(x)) \quad \bar{\gamma}\text{-a.e.}$$

Thus, for any function $F \in C_b(\mathbb{R}^d \times \mathbb{R}^d)$ we have

$$\begin{aligned} \int_{\mathbb{R}^d \times \mathbb{R}^d} F(x, y) d\bar{\gamma}(x, y) &= \int_{\mathbb{R}^d \times \mathbb{R}^d} F(x, \nabla\varphi(x)) d\bar{\gamma}(x, y) \\ &= \int_{\mathbb{R}^d} F(x, \nabla\varphi(x)) d\mu(x) \\ &= \int_{\mathbb{R}^d \times \mathbb{R}^d} F(x, y) d((\text{Id} \times \nabla\varphi)_\# \mu)(x, y), \end{aligned}$$

Consequently $\bar{\gamma} = (\text{Id} \times \nabla\varphi)_\# \mu$, is a transport plan, and so, as desired, there exist a solution to the Kantorovich problem, and, as a byproduct, also the Monge's problem.

Now, by contradiction, assume that the optimal transport plan is not unique, let $\bar{\gamma}_1$ and $\bar{\gamma}_2$ be two different optimal plans, and define $\tilde{\gamma} = \frac{\bar{\gamma}_1 + \bar{\gamma}_2}{2}$. Observe that

$$\begin{aligned} \int_{\mathbb{R}^d \times \mathbb{R}^d} |x - y|^2 d\tilde{\gamma}(x, y) &= \int_{\mathbb{R}^d \times \mathbb{R}^d} |x - y|^2 d\left(\frac{\bar{\gamma}_1 + \bar{\gamma}_2}{2}\right)(x, y) \\ &= \frac{1}{2} \left(\int_{\mathbb{R}^d \times \mathbb{R}^d} |x - y|^2 d\bar{\gamma}_1(x, y) + \int_{\mathbb{R}^d \times \mathbb{R}^d} |x - y|^2 d\bar{\gamma}_2(x, y) \right) \\ &= C^*(\mu, \nu), \end{aligned}$$

that is, $\tilde{\gamma}$ is also an optimal transport plan. And so, for any $\psi \in C_b(\mathbb{R}^d)$, it holds that

$$\begin{aligned} \int_{\mathbb{R}^d \times \mathbb{R}^d} \psi(x) d\tilde{\gamma}(x, y) &= \int_{\mathbb{R}^d \times \mathbb{R}^d} \psi(x) d\left(\frac{\bar{\gamma}_1 + \bar{\gamma}_2}{2}\right)(x, y) \\ &= \frac{1}{2} \left(\int_{\mathbb{R}^d \times \mathbb{R}^d} \psi(x) d\bar{\gamma}_1(x, y) + \int_{\mathbb{R}^d \times \mathbb{R}^d} \psi(x) d\bar{\gamma}_2(x, y) \right) \\ &= \int_{\mathbb{R}^d} \psi d\mu(x), \end{aligned}$$

that is $(\pi_X)_\#(\tilde{\gamma}) = \mu$. Analogously $(\pi_Y)_\#(\tilde{\gamma}) = \nu$.

So, we have proven that being $\bar{\gamma}_1, \bar{\gamma}_2, \tilde{\gamma}$ optimal transport plans, there exist three convex functions $\varphi_1, \varphi_2, \bar{\varphi}$ such that:

1. $\bar{\gamma}_1 = (\text{Id} \times \nabla\varphi_1)$. Thus $(x, y) = (x, \nabla\varphi_1)$ $\bar{\gamma}_1$ -a.e.
2. $\bar{\gamma}_2 = (\text{Id} \times \nabla\varphi_2)$. Thus $(x, y) = (x, \nabla\varphi_2)$ $\bar{\gamma}_2$ -a.e.
3. $\tilde{\gamma} = (\text{Id} \times \nabla\bar{\varphi})$. Thus $(x, y) = (x, \nabla\bar{\varphi})$ $\tilde{\gamma}$ -a.e.

And so, combining the first and the third equations with the second and the third, we

obtain:

$$\begin{aligned} (x, \nabla\varphi_1) \bar{\gamma}_1\text{-a.e.} &\Rightarrow (x, \nabla\varphi_1) = (x, \nabla\bar{\varphi}) \mu\text{-a.e.} \\ (x, \nabla\varphi_2) \bar{\gamma}_2\text{-a.e.} &\Rightarrow (x, \nabla\varphi_2) = (x, \nabla\bar{\varphi}) \mu\text{-a.e.} \end{aligned}$$

Thus, we conclude that $(x, \nabla\varphi_1) = (x, \nabla\varphi_2) \mu\text{-a.e.}$ which means that $\nabla\varphi_1 = \nabla\varphi_2$, or equivalently, that $\bar{\gamma}_1 = \bar{\gamma}_2$. Contradiction. Hence, the optimal plan is unique. \square

We now present two important corollaries for the computation of optimal transports using numerical methods. Their proofs are simple and we do not write them in this text. We refer the reader to Figalli and Glaudo [2021] Corollaries 2.4.10 and 2.4.11 for further details.

Corollary 1.36. *Under the assumptions of Brenier's theorem:*

1. *There exists a unique optimal transport map: $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that $T_{\#}\mu = \nu$. Also, $T = \nabla\varphi$ for some convex function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$.*
2. *If $S_{\#}\mu = \nu$ and $S = \nabla\phi \mu\text{-a.e.}$ for some ϕ convex, then S is the unique optimal transport map from μ to ν .*

Corollary 1.37. *Under the assumptions of Brenier's theorem, assume also that $\nu \ll dx$. Let $\nabla\varphi$ be the optimal transport map from μ to ν , and let $\nabla\psi$ be the optimal transport map from ν to μ . Then $\nabla\varphi$ is invertible $\mu\text{-a.e.}$, its inverse is unique $\nu\text{-a.e.}$ and is given by $\nabla\psi$.*

1.3. Quantum computing

The 20th century was undoubtedly remarkable because of the accelerated technological and scientific success in diverse fields of science and engineering. The theory of relativity, quantum mechanics, the birth of the computer, the invention of the transistor, space exploration, material sciences, and many others defined a giant leap for humanity. Notably, besides being of the utmost importance for understanding physical phenomena at minuscule scales, quantum mechanics faced the scientific community with a complete change in the physics paradigm. Even Einstein was skeptical and somehow reluctant to accept a probabilistic quantum theory. In a famous letter to Max Born in 1926, he said:

"Quantum mechanics is certainly imposing. But an inner voice tells me that it is not yet a real thing. The theory says a lot, but does not really brings us any closer to the secret

of the 'old one'. I, at any rate, am convinced that He [God] does not throw dice..." (Cite Born-Einstein letters, 52)

Famously, in 1935, the now-known EPR paper (Einstein-Podolski-Rosen) entitled "Can Quantum-Mechanical description of physical reality be considered complete?" highlighted the incompleteness of quantum mechanics, especially regarding quantum entanglement or "spooky action at a distance" as Einstein called it. However, despite the doubts, the quantum theory was correct yet tremendously counter intuitive. Interestingly, Einstein was awarded the 1921 Nobel Prize in Physics for "his services to theoretical physics, and especially for this discovery of the photoelectric effect," a foundational result of quantum physics.

In 1936, Alan Turing wrote his seminal paper "On Computable Numbers," proposing a Universal Computing Machine: a machine able to perform any algorithmically describable computation. Notably, the notion of Lambda calculus introduced by Alonzo Church and the Church-Turing thesis established theoretical limits of computation. That is, there are computational tasks that cannot be solved algorithmically, and therefore, there exists at least one class of problems for which no Turing machine can solve them.

In the 1940s, von Neumann's theoretical and practical contributions to the ENIAC project and the transistor development by Bardeen, Brattain, and Shockley enabled the design of fully digital computers as the TX-0 during the 1950s at MIT. Further technological refinements have made it possible, up to now, to reduce the transistors' dimension while increasing the machines' computing power. The pace of the development was so rapid that Gordon Moore conjectured in 1965 that the number of transistors in an integrated circuit roughly doubles every two years. This empirical law, Moore's law, imposes significant technological challenges to fabricating new devices. We are slowly approaching a limit where building smaller transistors is not feasible anymore. This situation makes it imperative to look for new computational paradigms that can beat the most influential classical computers available.

The end of the decade was crucial for what we now know as information theory. It was in 1948 that Shannon published "A Mathematical Theory of Communication," in which he formalized the notions of information and communication channels and studied the flow of information through noisy and noiseless channels. Significantly, he showed that information could be protected using error-corrected codes. These concepts are fundamental to understanding how information is transmitted and how it can be protected as it propagates through imperfect channels. Today, the theory of error correction is well developed, and almost all, if not all, modern communications are error-blind using an

error-correcting protocol.

Quantum computing is a rich and interdisciplinary field. It is a convergent point of, among others, quantum physics, information theory, and computer science. By the 1990s, those fields were already in a mature stage where it was natural to ask questions like: Would it be possible to leverage the effects of quantum mechanics to enable new, possibly faster ways of computing? Richard Feynmann was one of the first to speculate about the need for quantum computers to simulate quantum systems. In "simulating physics with computers" [Feynman, 2018], Feynmann says:

[...] Can a quantum system be probabilistically simulated by a classical (probabilistic, I'd assume) universal computer? [...] No! This is called the hidden variable problem: it is impossible to represent the results of quantum mechanics with a classical universal device. [...] And I'm not happy with all the analyses that go with just the classical theory, because nature isn't classical, dammit, and if you want to make simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look easy.

Building upon Feynman's ideas, David Deutsch proposed a notion of Universal Quantum Computer [Deutsch, 1985], a machine that ideally could solve computational problems hard or impossible for classical computers. From then on, and until the 2010's, most of the advances in quantum computing were mainly theoretical establishing quantum counterparts to classical computing algorithms. Some crucial algorithms proved to be faster than their classical counterparts. With an exponential speedup the Deutsch-Jozsa algorithm (1992) to identify constant or balanced patterns in long binary strings, and Shor's algorithm (1994) to factor large numbers. In turn, with a quadratic speedup, Grover's algorithm is an efficient algorithm to search in unsorted databases. Some other algorithms widely used nowadays are the Quantum Fourier Transform and the Quantum Phase Estimation.

In this section, we briefly introduce some notions of quantum computing. The first part of the section will be devoted to define the concepts of qubit, quantum logic and quantum circuits. The main reference for this part will be [Nielsen and Chuang, 2001]. Then, we will discuss some algorithms to which we refer to the original papers as main reference. These algorithms are the Quantum Fourier Transform (QFT); Quantum Phase Estimation (QPE); the Harrow–Hassidim–Lloyd (HHL) algorithm; and the Variational Quantum Linear Solver (VQLS). Although quantum mechanical effects are the principal phenomena involved in quantum computing, in this section we shall focus in the algorithmic properties

of quantum circuits rather than in their physics.

1.3.1. Qubits and Quantum Logic

In classical computing, the bit - or binary digit - is the smallest unit of information and a realization of a classical two-level system. It encodes some characteristics of a classical system into two possible values 0 and 1. A standard realization of a bit employs electric signals and assigns the value of 0 when a device reads an electric voltage below a certain threshold, and 1 when the reading is above it. Analogously, a quantum bit - also quantum binary digit or simply qubit - is the smallest unit of **quantum** information. It follows the same physical principle of a bit in the sense that it encodes some characteristics of a **quantum** two-level system. It is important to make emphasis on the word quantum, because although a bit is also the realization of a two-level system - we conceived a bit as a system with two states: voltage above the threshold and voltage below the threshold -, the differences between classical and quantum mechanics establish sharp differences between bits and qubits. The most crucial difference is that instead of being either in state 0 or 1, a qubit can exist as a constrained linear combination of both states.

Here we follow the mathematical formulation of quantum mechanics introduced by [Dirac \[1981\]](#) and [Von Neumann \[2018\]](#). We start with a the following fundamental axiom.

Axiom 0. *To any quantum mechanical system belongs at least one complex Hilbert Space \mathcal{H} in which the quantum theory of that system can be formulated.*

In the previous axiom, the phrase “*in which the quantum theory of the system can be formulated*” refers to the fact that the chosen Hilbert space should be able to represent - or rather encode - the characteristics of the physical system.

The reason why quantum mechanics can be formulated in Hilbert Spaces is far from intuitive and in some pathological scenarios can lead to theoretical contradictions - see [\[Bratteli and Robinson, 2012\]](#). Nowadays, more modern and sophisticated mathematical frameworks to describe quantum mechanics include C^* -algebras and category theory. We recommend the interested reader to refer to [\[Gieres, 2000\]](#) and [\[Heunen and Vicary, 2019\]](#) for deeper discussions in the matter.

Throughout this work, we consider $\mathcal{H} = (\mathbb{C}^n, \langle \cdot | \cdot \rangle)$, where $\langle \cdot | \cdot \rangle : \mathbb{C}^n \times \mathbb{C}^n \rightarrow \mathbb{C}$ will always denote the standard Euclidean inner product in \mathbb{C}^n . Notably, two-level systems are represented by $(\mathbb{C}^2, \langle \cdot | \cdot \rangle)$. Here we denote the inner product as $\langle \cdot | \cdot \rangle$ instead of $\langle \cdot, \cdot \rangle$ because we adopt the standard Dirac’s bra-ket notation - see [\[Dirac, 1939\]](#).

Definition 1.38 (Bra-Ket notation). Let \mathcal{H} be a suitable Hilbert space to represent the physical problem. We denote any element $\psi \in \mathcal{H}$ as $|\psi\rangle$ and will call it **ket**. Also, we write the **bra** of $\psi \in \mathcal{H}$ as $\langle\psi|$ and define it as the conjugate transpose of $|\psi\rangle$, that is $|\psi\rangle^\dagger = (|\psi\rangle^*)^T = \langle\psi|$. The inner product between $\varphi, \psi \in \mathcal{H}$ will be written as $\langle\varphi|\psi\rangle = \langle\psi|\varphi\rangle^*$ and will be indistinctly referred to as the the **bracket** between $|\varphi\rangle$ and $|\psi\rangle$.

After setting the Hilbert space of the physical problem, we can proceed to describe its elements. The properties of quantum systems are summarized in a set of postulates known as *the postulates of quantum mechanics*.

Postulate 1 (Quantum states). At each time instant t , we can find a ket $|\psi(t)\rangle \in \mathcal{H}$ representing the physical characteristics of a quantum mechanical system. We say that the ket $|\psi(t)\rangle$ is the quantum state of the system at time t .

Let us now focus in the case of two-level systems. We have said that, in that setting, $\mathcal{H} = (\mathbb{C}^2, \langle\cdot|\cdot\rangle)$. Then, we can find an orthonormal basis for \mathcal{H} , that is a set $\{|\phi_1\rangle, |\phi_2\rangle\} \subset \mathcal{H}$ such that $\mathcal{H} = \text{span}\{|\phi_1\rangle, |\phi_2\rangle\}$, $\| |\phi_1\rangle \| = \| |\phi_2\rangle \| = 1$, and $\langle\phi_1|\phi_2\rangle = 0$. The exact form of that basis is indifferent to us, and so, we relabel the two elements of the basis as $|\phi_1\rangle \equiv |0\rangle$ and $|\phi_2\rangle \equiv |1\rangle$. Additionally, in vector notation we write:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

We have chosen this basis transformation to guarantee that the new basis is orthonormal with respect to the inner product of \mathcal{H} , and that it still spans \mathcal{H} . Indeed,

$$\begin{aligned} \langle 0|1\rangle &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \\ \| |0\rangle \|^2 = \langle 0|0\rangle &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1 \\ \| |1\rangle \|^2 = \langle 1|1\rangle &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1 \end{aligned}$$

Thus, any arbitrary ket in \mathcal{H} can be written as a complex linear combination of elements of its basis:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle,$$

where $\alpha, \beta \in \mathbb{C}$.

This basis is known as the computational basis.

Definition 1.39 (Computational basis). We call the set $\{|0\rangle, |1\rangle\}$ computational basis of \mathcal{H} , and $\mathcal{H} = \text{Span}\{|0\rangle, |1\rangle\}$.

There is a particular subset $\mathcal{Q} \subset \mathcal{H}$ that has a concrete physical meaning. This subspace is known as **qubit space** and its elements as **qubits**. In more general settings, for arbitrary separable Hilbert spaces, we refer to those concepts as state space and quantum state, respectively.

Definition 1.40 (Qubit). Let $|\varphi\rangle \in \mathcal{H}$. We say that $|\varphi\rangle$ is a qubit if there exist $\alpha, \beta \in \mathbb{C}$ such that $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$, and $\langle\varphi|\varphi\rangle = 1$.

Remark 1.41. From the last condition of Definition 1.40, we identify $\langle\varphi|\varphi\rangle$ as the square of the norm the norm induced by the inner product - and denote as $\|\varphi\|$. Therefore, we require the state $|\varphi\rangle$ to be normalized. Equivalently we can ask for the coefficients α, β to satisfy $|\alpha|^2 + |\beta|^2 = 1$. Indeed:

$$1 = \langle\varphi|\varphi\rangle^2 = (\alpha^*\langle 0| + \beta^*\langle 1|)(\alpha|0\rangle + \beta|1\rangle) = \alpha^*\alpha\langle 0|0\rangle + \beta^*\beta\langle 1|1\rangle = |\alpha|^2 + |\beta|^2.$$

Having defined the underlying Hilbert space and its elements, we can ask ourselves: what kind of operations can be applied to those mathematical objects? Do they all have a physical meaning? Are they measurable experimentally? These questions lead us to the notions of quantum operators.

Definition 1.42 (Quantum Operator). Let $A : \mathcal{H} \rightarrow \mathcal{H}$. We say that A is a quantum operator if and only if $A \in B(\mathcal{H})$, where $B(\mathcal{H})$ is the set of linear and bounded operators from \mathcal{H} to \mathcal{H} over the field \mathbb{C} .

Remark 1.43. Because Hilbert spaces in quantum mechanics are separable, they are isometrically isomorphic to l^2 . Hence, in $\mathcal{H} = \mathbb{C}^2$ we can write maps A as 2×2 matrices acting on $|\varphi\rangle \in \mathcal{H}$. More generally, for any finite dimensional Hilbert space of dimension N , quantum operators can be represented by $N \times N$ matrices with complex entries.

The coefficients α and β have a fundamental role in quantum mechanics. They are known as **probability amplitudes** because of their connection to the measurement problem. Measuring in quantum mechanics is a complete subject on its own. Here we don't under- mine into the details, but for the interested reader, there is a very detailed and rigorous discussion on the matter in [Peres, 1997] and [Wiseman and Milburn, 2009]. However,

for our purposes it is enough to say that after we have defined a basis for the problem's Hilbert space, it is possible to devise a machine able measure to physical properties of the system in that basis. Of course, in physical systems $|0\rangle$ and $|1\rangle$ represent physical quantities, for example:

- Magnetization
- Spin
- Charge.

The main result of measurement theory is that, after measuring the system, it will result in one of the two basis states, that is, the experimenter will measure either $|0\rangle$ or $|1\rangle$. If the procedure is repeated for many different but identically prepared systems, the probability that they will measure $|0\rangle$ will be close to $|\alpha|^2$, and the probability that they will measure $|1\rangle$ will be close to $|\beta|^2$. Provided that the procedure is repeated infinitely many times, by means of the law of large numbers the probabilities of measuring each state will converge to $|\alpha|^2$ and $|\beta|^2$ surely - in the statistical sense of sure convergence. Therefore, they are constrained to satisfy that:

$$|\alpha|^2 + |\beta|^2 = 1.$$

Motivated by this fact α and β are referred to as **probability amplitudes**.

Definition 1.44 (Observable). *A quantum operator B associated to a measurable physical quantity is known as an observable. Observables have the additional mathematical property of being self-adjoint operators.*

On behalf of the spectral theorem, if $B \in \mathcal{B}(\mathbb{C}^2)$, then it is diagonalizable. Moreover, if we apply B to a quantum system, the experimenter will register in the measuring device one of the eigenvalues of B , and then can identify the quantum state with the one corresponding to that eigenvalue.

A remarkably useful set of observables is that of the Pauli matrices.

Definition 1.45 (Pauli matrices). *We define Pauli matrices $\sigma_X, \sigma_Y, \sigma_Z \in \mathbb{C}^{2 \times 2}$ as:*

$$\sigma_X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Definition 1.46 (Quantum gate). *A quantum gate U is a unitary quantum operator, i.e. an operator such that $UU^\dagger = \mathbb{I}$, with \mathbb{I} the identity operator. Such gates can be single qubit gates acting on one qubit at a time, and in such case $U \in \mathbb{C}^{2 \times 2}$; or they can be k -qubit*

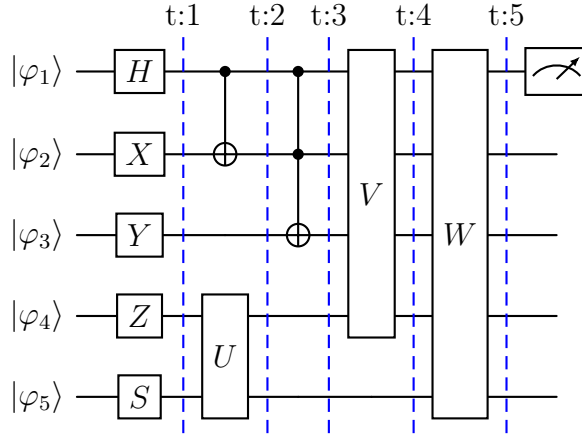


Figure 1.1: Schematic of a generic Quantum Circuit (QC). Each line represents a different qubit. We refer to those lines as wires. The wires serve to keep track of the quantum operations applied to each particular qubit at each time instant. In this example, at each time $\tau \in \{1, 2, 3, 4, 5\}$ we apply different τ -qubit gates represented as boxes or connecting vertical lines. The boxes labeled with X, Y, Z, H , and S represent the quantum Gates previously defined. The vertical lines are depictions of τ -qubit controlled X Gates with \bullet the control qubits, and \oplus the target. The boxes labeled with U, V , and W are arbitrary 2, 3, and 4 qubit gates. The time is discrete and increases from left to right. The last step is a measurement of $|\varphi_1\rangle$ represented with a Gauge.

gates, acting on k qubits simultaneously. In the latter $U \in \mathbb{C}^{2^k \times 2^k}$.

From now on, we will refer to the Pauli matrices $\{\sigma_X, \sigma_Y, \sigma_Z\}$ as $\{X, Y, Z\}$. In particular X is known as the NOT gate because $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$. These matrices are crucial for the development of the quantum circuits discussed in the Methods and Results sections. Another particular set important set of gates is $\{H, X, S, \text{CNOT}\}$, where:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad \text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1.1)$$

The relevance of the last sets of gates is that it is universal in the following sense:

Definition 1.47 (Universal set of gates). *A universal set of gates is a minimal collection of quantum gates such that, when combined, allow for the approximation of any arbitrary unitary transformation on a quantum system up to any degree of precision.*

Finally, we can introduce the notion of quantum circuit.

Definition 1.48 (Quantum circuit). *A Quantum Circuit (QC) is a collection of qubits to which some quantum gates are applied respecting a certain temporal order. We depict an example of a generic Quantum Circuit in Figure 1.1.*

Remark 1.49. *We denote a particular operation in a specific wire by sub-indexing the corresponding operator with the index of the wire on which it acts.*

Following the previous remark, the circuit in Figure 1.1 up to time 1 can be written as

$$H_1 X_2 Y_3 Z_4 S_5 |\varphi_1\rangle |\varphi_2\rangle |\varphi_3\rangle |\varphi_4\rangle |\varphi_5\rangle.$$

For the sake of simplicity, we indistinctly write

$$H_1 X_2 Y_3 Z_4 S_5 |\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5\rangle.$$

2 | Methods and developments

We have seen that under some settings Brenier's Theorem 1.35 allows us to compute transportation maps as gradients of convex functions. In Amos et al. [2017] the authors propose a Neural Network (NN) architecture known as Input Convex Neural Network (ICNN) and they prove that it is a convex function learner, that is, that regardless the input data, the output will always be a convex function. We want to link ICNNs and the Optimal Transport (OT) via convex optimization. Indeed, exploiting the functional properties of a ICNN, we fit the weights of the neural network to retrieve the optimal convex function leading to the optimal transportation map between two measures on $\mathcal{P}(\mathbb{R})$.

2.1. Classical convex interpolation

Splines interpolation solves the problem of finding a smooth curve connecting - equivalently: interpolating - a set of points. Let $(x_0, s_0), (x_1, s_1) \in \mathbb{R}^2$, with $x_0 < x_1$. We want to find $s : \mathbb{R} \rightarrow \mathbb{R}$ smooth such that

$$s(x_0) = s_0 \qquad s(x_1) = s_1$$

Additionally, we set $s'_0, s'_1 \in \mathbb{R}$ to be the derivatives of $s(\cdot)$ at each corresponding point x_0 or x_1 .

The case where the interpolating function is polynomial of third degree is known as Cubic Splines and has been largely studied in the literature. Using the following parametrization: $\xi = \frac{x-x_0}{x_1-x_0}, \lambda = x_1 - x_0$, the result for the problem is:

$$s(\xi, \lambda) = (1 - 2\xi^2 + 2\xi^3)s_0 + (3\xi^2 - 2\xi^3)s_1 + \lambda(\xi - 2\xi^2 + \xi^3)s'_0 + \lambda(-\xi^2 + \xi^3)s'_1.$$

However, the problem of cubic splines is that it produces functions that are not convex in general. Since we are interested in finding only convex functions satisfying the constraints, if they exist, we manually impose a convexity on the function s . Namely, we require positivity of the second derivative $\frac{d^2s}{dx^2}$. It is important to point out that such function

might not exist, and so, besides solving the problem we need to find the valid ranges of parameters for which the problem is solvable. Using a convenient parametrization of the second derivative introduced in [Messac and Sivanandan, 1997], we write:

$$\frac{d^2s(x)}{dx^2} := \lambda^{n-2} \left(a \left(\xi(x) - \frac{1-\gamma}{2} \right)^{n-2} + b \left(\xi(x) - \frac{1+\gamma}{2} \right)^{n-2} \right),$$

with $n \in \mathbb{N}$, $n \geq 4$, $0 \leq \gamma \leq 1$, $a, b > 0$. And so, integrating twice with respect to x , we find the solution of the new constrained problem as:

$$s(x) = \lambda^n \left(\frac{a}{n(n-1)} \left(\xi(x) - \frac{1-\gamma}{2} \right)^n + \frac{b}{n(n-1)} \left(\xi(x) + \frac{1-\gamma}{2} \right)^n \right) + c\xi(x) + d,$$

with the coefficients a, b, c, d being the solution to:

$$Ax = \begin{bmatrix} \alpha_0 & \beta_0 & 0 & 1 \\ \beta_0 & \alpha_0 & 1 & 1 \\ \alpha_1 & -\beta_1 & 1 & 0 \\ \beta_1 & -\alpha_1 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ s'_0 \\ s'_1 \end{bmatrix} = s, \quad (2.1)$$

where we have defined:

$$\begin{aligned} \alpha_0 &= \lambda^n \bar{\alpha}_0, & \bar{\alpha}_0 &= \frac{1}{n(n-1)} \left(\frac{1-\gamma}{2} \right)^n, \\ \beta_0 &= \lambda^n \bar{\beta}_0, & \bar{\beta}_0 &= \frac{1}{n(n-1)} \left(\frac{1+\gamma}{2} \right)^n, \\ \alpha_1 &= \lambda^{n-1} \bar{\alpha}_1, & \bar{\alpha}_1 &= \frac{-1}{(n-1)} \left(\frac{1-\gamma}{2} \right)^{n-1}, \\ \beta_1 &= \lambda^{n-1} \bar{\beta}_1, & \bar{\beta}_1 &= \frac{1}{(n-1)} \left(\frac{1+\gamma}{2} \right)^{n-1}. \end{aligned}$$

The previous development requires to set the order n of the convex spline curve. The value of that parameter defines the degree of the approximating polynomial and consequently, the shape of the interpolating function. Because we need to guarantee that the second derivative of s is positive, we require $n \geq 4$ and even. Additionally, when $\gamma = 1$ we follow the procedure from Messac and Sivanandan [1997]. We define

$$\bar{s} = \frac{s_1 - s_0}{\lambda}, \quad n_1 = \frac{s'_1 - s'_0}{\bar{s} - s'_0}, \quad n_2 = \frac{s'_1 - s'_0}{s'_1 - \bar{s}},$$

and select a proper order n_c evaluating:

$$n_c = \begin{cases} 4, & \text{if } \min(n_1, n_2) \geq 4 \text{ or } \max(n_1, n_2) \leq 4 \\ \min\{2k \geq \max(n_1, n_2) : k \in \mathbb{N}\} & \text{otherwise.} \end{cases}$$

The positivity constraint of the second derivative imposes conditions on the possible values for the slopes and displacements of the data points. This means that given a dataset $\{(x_1, s_1), (x_2, s_2), \dots, (x_K, s_K)\}$, the valid slopes will be restricted as:

$$\frac{\phi_1 \bar{s} - \phi_3 s'_1}{\phi_2} < s'_0 < \frac{\phi_1 \bar{s} - \phi_2 s'_1}{\phi_3} \quad (2.2)$$

$$\frac{\phi_1 \bar{s} - \phi_2 s'_0}{\phi_3} < s'_1 < \frac{\phi_1 \bar{s} - \phi_3 s'_0}{\phi_2}, \quad (2.3)$$

with $\phi_1 = \bar{\beta}_1 - \bar{\alpha}_1$, $\phi_2 = \bar{\beta}_0 - \bar{\alpha}_1 - \bar{\alpha}_0$, and $\phi_3 = \bar{\alpha}_0 + \bar{\beta}_1 - \bar{\beta}_0$.

The previous conditions are crucial for defining our algorithmic implementation of convex splines interpolation that consists in a modification to the SAM algorithm from [Messac and Sivanandan, 1997].

Now assume that instead of two points, we have a set of K different points that we want to connect with a globally convex function, if it exists. We say that the data set will have K **knots** $\{x_0, x_1, \dots, x_{K-1}\}$, and K **images** $\{s_0, s_1, \dots, s_{K-1}\}$. To compute the resulting curve we need to specify the slopes in the boundary points of each sub-domain $\{[x_k, x_{k+1}]\}_{k=0}^{K-1}$. There are many different valid approaches to specify those slopes. For example, Messac and Sivanandan [1997] uses the Slope Averaging Method (SAM), were they define:

$$\lambda_k = x_{k+1} - x_k, \quad \bar{s}_k = \frac{s_{k+1} - s_k}{\lambda_k}, \quad \mu_{-k} = \frac{\lambda_{k+1}}{\lambda_k + \lambda_{k+1}}, \quad \mu_{+k} = \frac{\lambda_k}{\lambda_k + \lambda_{k+1}}.$$

And impose the slope at each point as,

$$\begin{cases} s'_1 = (1 - \sigma \operatorname{sign}(\bar{s}_1)) \bar{s}_1, & 0.1 \leq \sigma \leq 0.2 \\ s'_k = \mu_{-k} \bar{s}_{k-1} + \mu_{+k} \bar{s}_{k+1}, & 2 \leq k \leq K-1 \\ s'_K = (1 + \sigma \operatorname{sign}(\bar{s}_{K-1})) \bar{s}_{K-1}, & 0.1 \leq \sigma \leq 0.2 \end{cases}$$

Here we propose a simpler approach that we call **Slope Weighted Averaging Method**, SWAM for short. For for $k \in \{1, 2, \dots, K-1\}$, we define the delta displacements and

empirical slopes as:

$$\begin{aligned} \Delta^+(k) &= x_{k+1} - x_k, & \hat{s}^+(k) &= \begin{cases} \frac{s_{k+1} - s_k}{\Delta^+(k)} & \text{if } \Delta^+ > 0 \\ 0 & \text{if } \Delta^+ = 0 \end{cases} \\ \Delta^-(k) &= x_k - x_{k-1}, & \hat{s}^-(k) &= \begin{cases} \frac{s_k - s_{k-1}}{\Delta^-(k)} & \text{if } \Delta^- > 0 \\ 0 & \text{if } \Delta^- = 0 \end{cases} \end{aligned}$$

And, then, we set the slope at each point as

$$\begin{cases} s'_1 = (1 - \sigma \operatorname{sign}(\hat{s}_1^+)) \hat{s}_1^+ \\ s'_k = \frac{\Delta^+(k) \hat{s}^+(k) + \Delta^-(k) \hat{s}^-(k)}{\Delta^-(k) + \Delta^+(k)}, & 2 \leq k \leq K - 1 \\ s'_K = (1 + \sigma \operatorname{sign}(\hat{s}_K^-)) \hat{s}_K^- \end{cases}$$

where, again, $0.1 \leq \sigma \leq 0.2$, and we assume $\hat{s}^+(k) \geq \hat{s}^-(k)$.

In short, the SWAM algorithm approximates the slopes at each point of the dataset with the weighted sum of the neighboring empirical slopes.

However, despite being simple and efficient, the SWAM method has the great disadvantage of providing slopes that can lie outside the allowed ranges defined in equations (2.2)-(2.3). We correct the invalid values of s'_1 by choosing the boundary point in equation (2.3) that is the closest to the SWAM estimation.

To find the multi-segment convex interpolating curve it is only left to solve the $K - 1$ independent linear systems using the previously estimated slopes and orders.

2.2. Approximating continuous transport maps from discrete samples of continuous probability measures

We assume that $\mu, \nu \in \mathcal{P}(\mathbb{R})$ are distributions satisfying the assumptions of Brenier's Theorem 1.35. Assume that we cannot access them directly, but instead we sample to obtain two empirical measures. We note as $\mathcal{D}_\mu := \{(x_1, s_1), (x_2, s_2), \dots, (x_K, s_K)\}$ and $\mathcal{D}_\nu := \{(y_1, t_1), (y_2, t_2), \dots, (y_L, t_L)\}$ the dataset defying the empirical distributions of μ and ν , respectively. In particular, $x_k \in \operatorname{supp}(\mu)$, $k = 1, 2, \dots, K$; $y_l \in \operatorname{supp}(\nu)$, $l = 1, 2, \dots, L$; and s_k, t_l are their respective masses. Under this setting, the existence of a unique transport map T from μ to ν is guaranteed on behalf of Theorem 1.35.

In this section we explore a numerical method to approximate the continuous map φ inducing such transport as $T = \nabla\phi$. We rely in a solution of the (discrete) Optimal Transport problem from $\hat{\mu}$ and $\hat{\nu}$ -see [Peyré et al., 2019]- to obtain a (discrete) transport map \hat{T} between $\hat{\mu}$ and $\hat{\nu}$. Our goal is to retrieve the actual continuous map $T : X \rightarrow Y$ being the optimal transport map between μ and ν , using \hat{T} and exploiting the convexity properties of φ . To do so, we integrate naively the discrete OT map using the following algorithm.

The result of the naive integration is a function -see [Davis and Rabinowitz, 2007] Chapter 2.1- $\hat{\varphi} : \text{supp}(X) \rightarrow \text{supp}(Y)$ approximating the actual convex function φ such that $T = \nabla\phi$. Being φ a convex function, we approximate it using by means of our convex interpolation algorithm applied to $\hat{\varphi}$ following the method described in 2.1.

2.3. Quantum Convex regression

In the classical case, we discussed that one possible approach to perform convex regression is through ICNNs Amos et al., 2017. It might seem natural to translate the ICNN architecture into quantum by transforming the classical logic into its quantum counterpart and operating to get the desired result. By doing so, we would obtain a Quantum Input Convex Neural Network (qICNN for short). However, we discovered that this approach has profound obstacles:

- **ICNNs require to implement layers with positive weights:** Linear combinations of convex functions with positive coefficients are still convex. This is the rationale behind the selection of positive weights in ICNN architectures. Indeed, when we choose convex activation functions, e.g. ReLu, we will get convex outputs. However, in qICNNs there is no similar result. Because activation function in Quantum Circuits can be understood as rotations in a certain complex Hilbert Space, there is no equivalent property that allows us to produce convex functions by constraining the network parameters.
- **ICNNs require to implement recurrent connections:** It is well known that the no-cloning theorem avoid us from copying quantum states without damaging its quantum properties [Bužek and Hillery, 1996]. Therefore we cannot implement recurrent connections copying the information from the first layer of the qICNN into consecutive layers. A workaround for this is re-inputing the data in each layer, however, as pointed out by Schuld et al. [2021] and discussed next in this section. This approach affects the circuit's expressivity.

- **Preparing arbitrary states using QC is difficult:** The mathematical constraints on quantum operators - see section 1.3 - make it harder to design flexible quantum circuits to fit arbitrary functions. For example, consider the set of gates $\{H, X, S, CNOT\}$ defined in Equation 1.1, which is universal. Therefore, applying a suitable sequence of these operations we are able to prepare any quantum state. However, the number required operations could be so high that would make the procedure imprecise and demanding due to induction of errors at each rotation.

These divergences between the Classical and Quantum logic prevent us from constructing a direct way to move from ICNN to qICNNs.

Since the gates are unitary operations acting on the elements of \mathcal{H} , then it is very difficult to devise a quantum circuit able to represent any arbitrary mathematical function with ease. Moreover, it is even tougher to represent structured functions exhibiting, for example, convexity. This is problem is commonly known in the literature as the *expressivity problem*.

When dealing with representation of arbitrary functions using quantum circuits, there are several steps involved: first, the data need to be properly encoded in a quantum space, then the encoded quantum state needs to be operated with a quantum circuit, last, the result have to be decoded into the original domain. In practical terms, even though the application of a certain quantum algorithm could be very efficient, it is important to consider the costs of encoding and decoding the data. Indeed, many algorithms struggle when dealing with real datasets, and the so called *quantum advantage* disappears. A well known result by Schuld et al. [2021] proves that the expressive power of a variational quantum model is limited by the data-encoding strategy. They write:

$$f(x) = \sum_{\omega \in \Omega} c_{\omega} e^{i\omega x}$$

where the set of accesible functions is determined by the data encoding protocol.

We now introduce a quantum routine for executing regression.

HHL Algorithm

The HHL algorithm [Harrow et al., 2009] aims for solving linear systems of equations.

Solving a linear system of equations aims to find $x \in \mathbb{C}^n$ such that $Ax = b$, where $A \in \mathbb{C}^{n \times n}$, $b \in \mathbb{C}^n$ are known. There are many different methods for solving this problem. Between then, some solution techniques have the following computational complexities:

1. **Gaussian elimination:** $O(n^3)$
2. **Fast matrix multiplication:** $O(n^w), w < 2.372864$

The quantum version of the problem assumes that we know a normalized state $|b\rangle$, and a matrix A . Then, we look for a quantum state $|\tilde{x}\rangle$ that resembles the solution x^* to $Ax = b$ up to a precision ε , that is, if $|x\rangle$ encodes x^* , then

$$\| |x\rangle - |\tilde{x}\rangle \| \leq \varepsilon.$$

There are some assumptions on the system:

1. We assume that A is s -sparse - i.e. each row has at most $s \ll n$ nonempty rows - and can be accessed by two oracles:
 - $O_A : |i, j\rangle|0\rangle \mapsto |i, j\rangle|A(i, j)\rangle$ (retrieves matrix elements)
 - $O_{A,loc} : |j\rangle|\ell\rangle \mapsto |j\rangle|\nu(j, \ell)\rangle$ (locates the nonzero elements)
 where the l -th non-zero element in row j is in the column $\nu(j, l)$.
2. We need to be able to efficiently prepare copies of the state $|b\rangle$ without errors. This fact makes it impossible to run the algorithm in real hardware today.
3. The condition number - the ration between the biggest and the smallest eigenvalue of A - has to be finite.

To implement the algorithm, we use 3 subroutines.

1. Hamiltonian simulation (of sparse matrices)
2. Quantum Fourier Transform/Quantum Phase Estimation
3. Amplitude amplification

For a detailed discussion in how these methods are implemented we refer the reader to [Nielsen and Chuang \[2001\]](#).

It turns out that the previous assumptions are too restrictive for real problems. However, there are alternative approximate methods like the so-called Variational Quantum Linear Solver (VQLS) introduced by [Bravo-Prieto et al. \[2023\]](#) that circumvent some of these difficulties.

3 | Results

In this section, we present out results coming from the application of techniques and methods introduced in the previous chapter.

3.1. Variational Quantum Circuit Regression

We evaluated first the feasibility and capabilities of performing regression for arbitrary functions using quantum circuits. We employed the architecture suggested by [Schuld et al. \[2021\]](#), and depicted in Figure 3.1. This consists in a Variational Quantum Circuit with two alternating components:

1. **Fully entangling circuit (FEC)**: A quantum architecture devised to generate entanglement between the qubits to favor expressivity and introduce non-linearities. The circuit has two sub-blocks: (1) individual rotations around the 3-axis of the Bloch sphere followed by entangling operations induced by CNOT gates between **adjacent** qubits; and (2) individual rotations around the 3-axis of the Bloch sphere followed by entangling operations induced by CNOT gates between qubits **separated by a distance 2**.
2. **Encoding block (EB)**: A rotation of x radians around the x -axis of the bloch sphere, where x is a particular datum.

These two routines are used to ensemble the Variational Quantum Circuit VQC used for regression in this work. That circuit is composed by n alternating **FCE** + **EB** layers with an extra **FEC** block at the end. In Figure 3.2 we show the diagram and explain the final circuit used to perform the regression task.

To benchmark the circuit's capabilities, we fitted the following function defined by a partial Fourier sum:

$$f(x) = \sum_{j=-k}^k c_j \exp(ix)$$

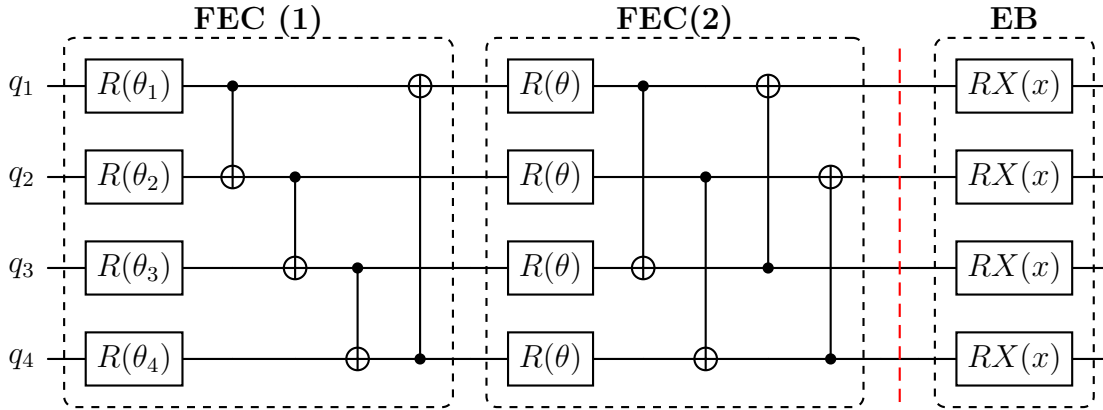


Figure 3.1: Depiction of the quantum gates used in the quantum regression circuit. The picture shows a circuit with four qubits, however, following the same logic the same architectures can be generalized to an arbitrary number of qubits. The sub-blocks **FEC(1)** followed by **FEC(2)** form the routine known as fully entangling circuit **FEC**; while **EB** is itself the encoding routine.

The model was trained semi-classically by computing the gradient of the Circuit with the so called shift rule [Mitarai et al., 2018], and evaluated in the training dataset using the circuit depicted in Figure 3.2. We applied Stochastic Gradient Descend (SGD) to minimize the Mean Squared Error (MSE) as objective function with a learning rate of $\gamma = 0.1$. We trained over a maximum of 100 training epochs and used early-stopping with a patience of 3, and an epsilon of 10^{-3} . The training and test points were sampled uniformly from the domain $[-6, 6]$. No sample used for training was used to compute the reported MSE, whose value corresponds to that of the test data set. In total 140 points were used for training and 60 for testing. We trained 6 different models considering all the possible combinations for number of 1 and 4 qubits; and for a number of layers $n = 1, 10, 20$. The results are reported in Figure 3.3.

After that first model we fitted a second model for the convex function $f(x) = x^4 + 30(x - 5)^2$. The training was performed on a data-set of 140 points $\{x, f(x)\}$ uniformly sampled from the interval $[-1, 1]$. Additionally, 60 samples, obtained in the same way, were used for validation. As in the previous case, we selected as objective function the MSE and as the classical optimization routine the Adam Optimizer with learning rate 0.01. The models were fitted for 10, 20, 30 and 40 layers, and for 2, 4, and 8 qubits. In Table 3.1 we report some average metric over 10 training cycles.

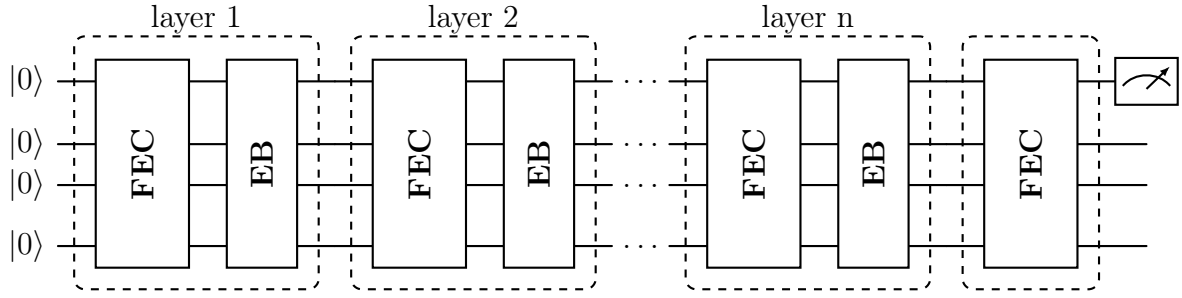


Figure 3.2: Quantum Regression Circuit used for the regression task. The circuit is composed by n alternating layers **FEC+EB** with an extra **FEC** at the end. Only the first register is measured. The figure shows a circuit with 4 qubits, however the same architecture can be generalized to any number of qubits. When the circuit has only 1 qubit we called it serial circuit, and when it has more than 1 qubit we call it parallel circuit.

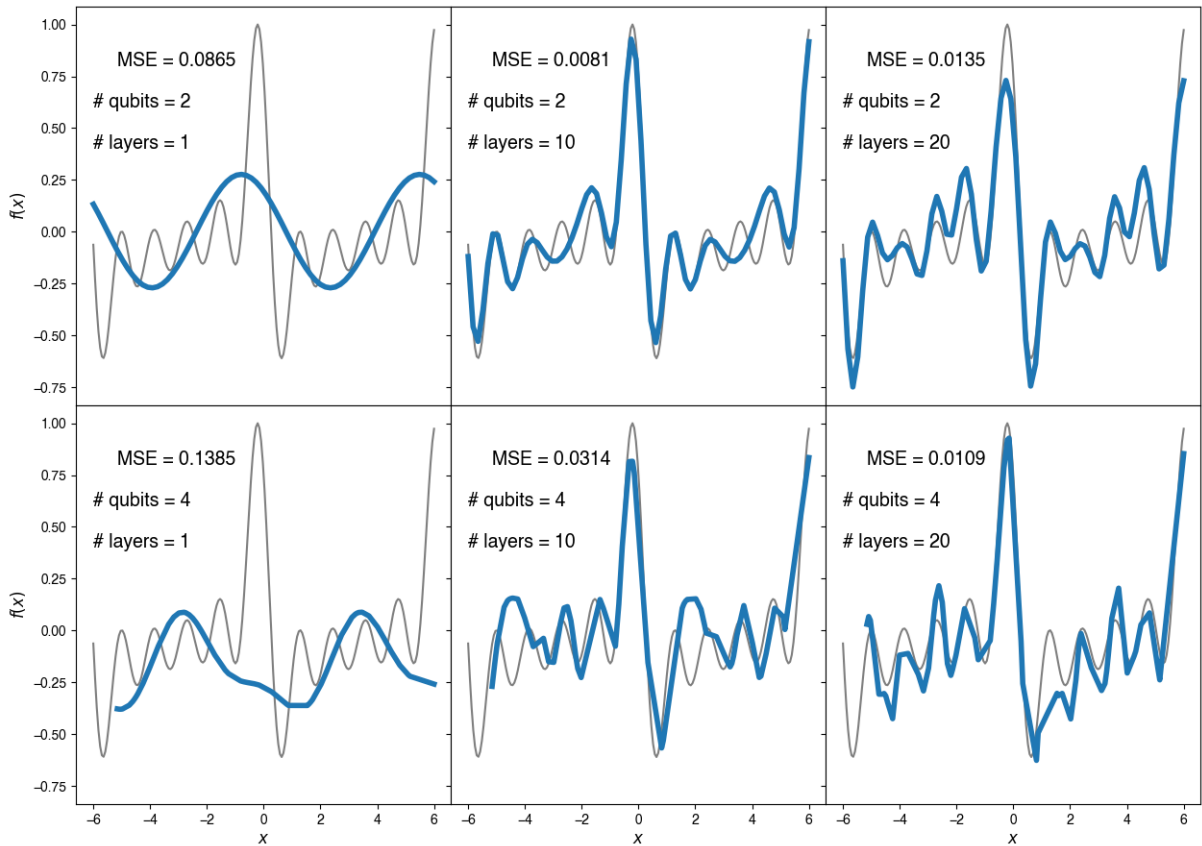


Figure 3.3: Results of the regression of the curve $f(x) = \sum_{j=-k}^k c_j \exp(ix)$ using the quantum circuit of Figure 3.2 for different configurations of number of qubits and number of layers. The model with 1 qubit is commonly known as series model, while the one with 4 qubits is known as parallel model. The blue curve represents the learned function while the gray curve is the real one.

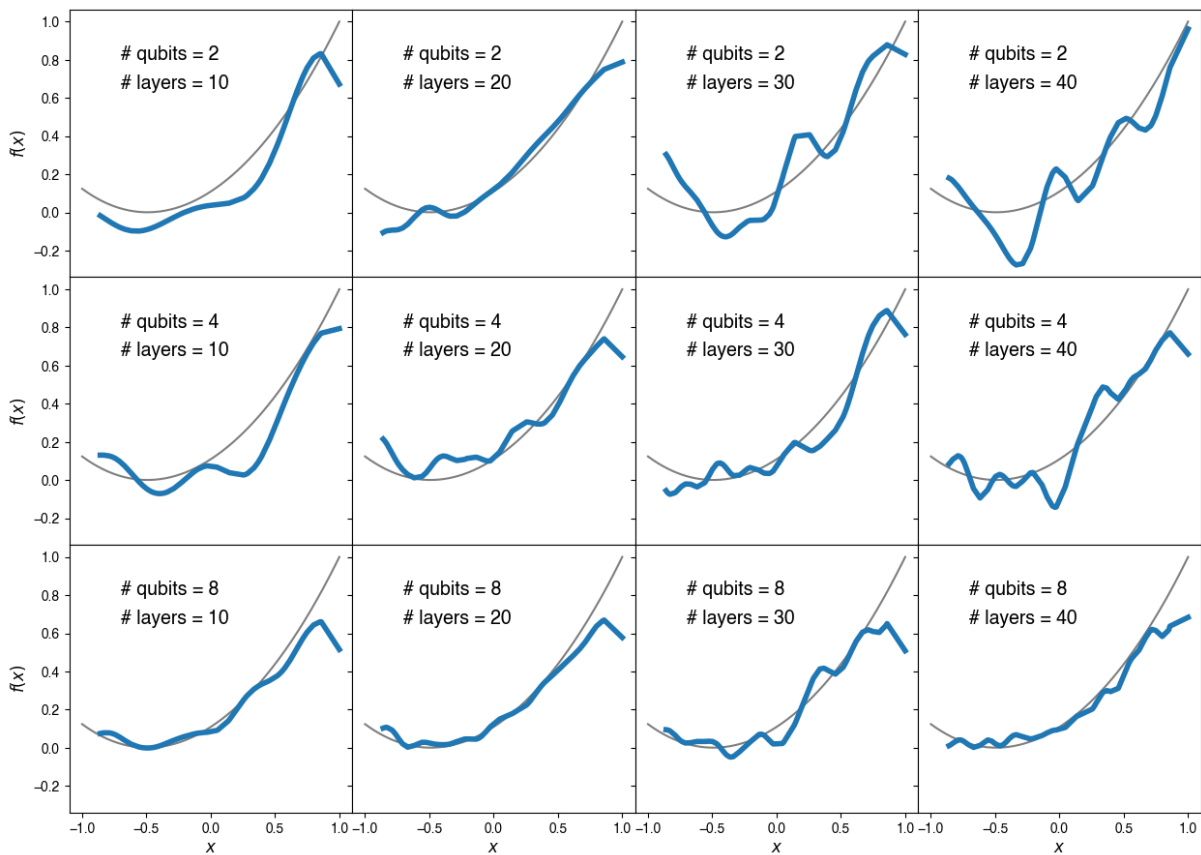


Figure 3.4: Results of the regression of the curve $f(x) = x^4 + 30(x-5)^2$ using the quantum circuit of Figure 3.2 for different configurations of number of qubits and number of layers. The blue curve represents the learned function while the gray curve is the real one.

Number of layers	Number of qubits	Number of training parameters	Average number of training epochs	Average training time (normalized)	MSE	
					(Train)	(Test)
10	2	132	22.5 ± 8.3	0.164385362	0.011053 ± 0.006713	0.010284 ± 0.005757
	4	264	11.6 ± 5.5	0.164657455	0.006912 ± 0.005138	0.007966 ± 0.006567
	8	528	14 ± 1.9	0.403475861	0.005128 ± 0.000804	0.003036 ± 0.000481
20	2	252	23.4 ± 5.4	0.327635447	0.007206 ± 0.007285	0.008727 ± 0.00876492
	4	504	12.8 ± 5.7	0.350342656	0.005485 ± 0.002231	0.005752 ± 0.002243
	8	1098	10.5 ± 1.4	0.584378766	0.004136 ± 0.000463	0.003065 ± 0.000380
30	2	372	23.3 ± 8.6	0.475293989	0.008708 ± 0.007256	0.009841 ± 0.009449
	4	744	13.9 ± 3.3	0.561262009	0.005922 ± 0.002113	0.006849 ± 0.002874
	8	1488	8.7 ± 1.1	0.723566268	0.004649 ± 0.0009449	0.003662 ± 0.000553
40	2	492	28.9 ± 7.4	0.78457688	0.006270 ± 0.004460	0.005976 ± 0.004460
	4	984	15.7 ± 3.5	0.84252967	0.005338 ± 0.001878	0.004143 ± 0.001928
	8	1968	9.2 ± 2.9	1	0.004949 ± 0.001907	0.00381942 ± 0.001397

Table 3.1: Simulation results for the convex function $f(x) = x^4 + 30(x - 5)^2$ using the quantum circuit of Figure 3.2 for different configurations of number of qubits and number of layers. We report different metrics averaged over 10 trainings. The time is normalized with respect to the highest training time, that corresponds to the model with 8 qubits and 40 layers. The average time is shown without standard deviation because its small value.

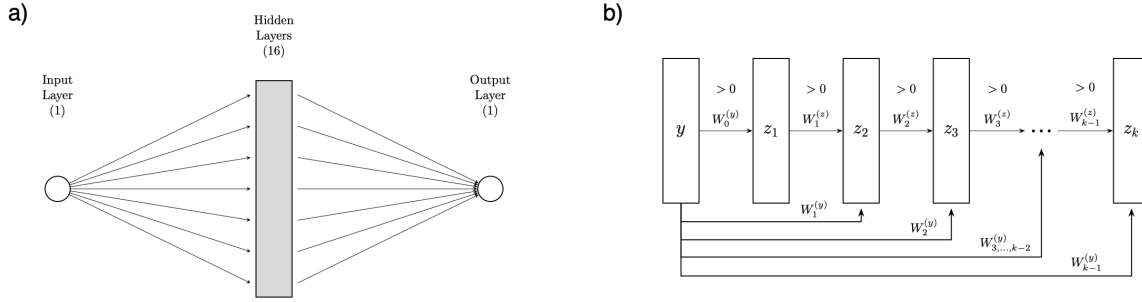


Figure 3.5: On the left we depict a single layer Feedforward Neural Network (FNN). It consists of one input and one output layer with 1 neuron each; and one hidden layer with 16 neurons with ReLu activation functions. We chose this one as our benchmark implementation on behalf of the Universal Approximation Theorem [Cybenko, 1989]. On the right we depict the architecture of an Input Convex Neural Network as introduced by Amos et al. [2017]. It consists of a constrained multi-layer FNN with residual connections. Observe that the weights $\{W_1^{(z)}, W_2^{(z)}, \dots, W_{k-1}^{(z)}\}$ are positive to ensure the convexity of the output.

In Table 3.1, we show the training error for 2, 4, and 8 qubits, and 10, 20, 30, and 40 layers.

3.2. Convex and non-convex regression

We bench-marked the methods developed in Chapter 2. In particular we did so for three different settings:

1. A **convex function**: $f(x) = 0.4x^2 + 0.5x^4 - 1.6x^6 + 2.5x^{10}$ for $x \in [-1, 1]$
2. A **non-convex function**: $f(x) = -(0.4x^2 + 0.5x^4 - 1.6x^6 + 2.5x^{10}) + 1.8$ for $x \in [-1, 1]$
3. A **non-convex function**: $f(x) = \frac{x}{2} \sin(x)$ for $x \in [0, 6]$.

In particular, we applied The Convex Splines method developed in Section 2.1, and compared it against the Input Convex Neural Network [Amos et al., 2017], and a Feedforward Neural Network. We now detail the procedure employed for each method.

We performed the Convex Splines interpolation using our SWAM method with $\gamma = 1$, and $\sigma = 0.1$. For the Feedforward Neural Network (FNN) we used an architecture of only one hidden layer of 16 neurons. To train the network we applied Stochastic Gradient Descend (SGD) to minimize the MSE with a learning rate 0.01. We used 20 000 epochs to train the model. The Input Convex Neural Network consisted of 2 hidden layers with

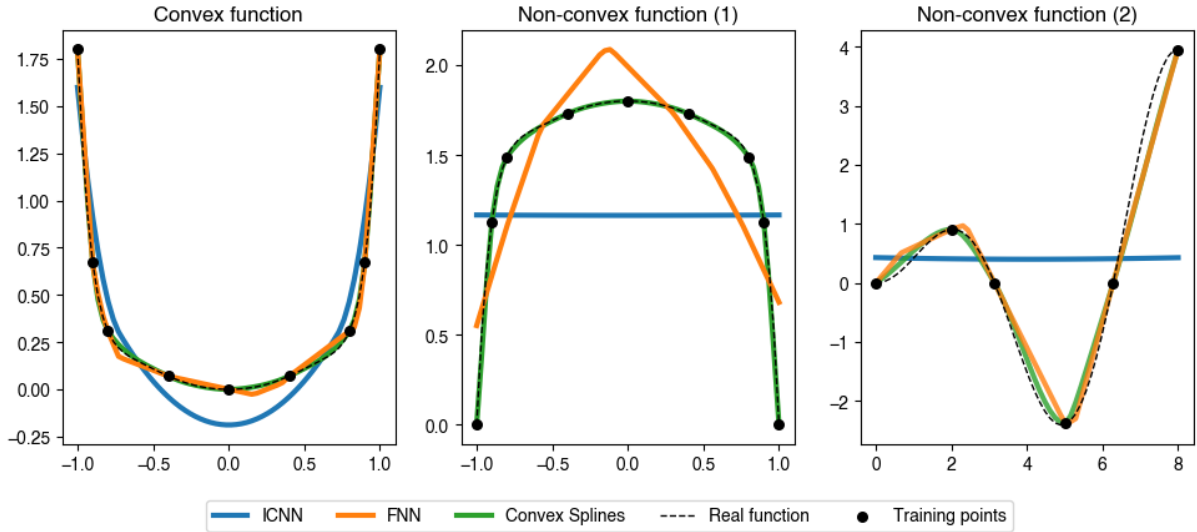


Figure 3.6: Comparison of ICNN, FNN, and convex splines for the regression of three different functions. On the left we plot a (strictly) convex function in $[-1, 1]$ defined by $f(x) = 0.4x^2 + 0.5x^4 - 1.6x^6 + 2.5x^{10}$. In the center we see a nowhere convex function in $[-1, 1]$ defined by $f(x) = -(0.4x^2 + 0.5x^4 - 1.6x^6 + 2.5x^{10}) + 1.8$. Finally, in the right we see the function $f(x) = \left(\frac{x}{2}\right) \sin(x)$ for $x \in [0, 8]$ that is only convex in $[-\text{approximately-}[0, 1.076] \cup [3.643, 6.578]$.

64 neurons each, and sparse connections of layers with positive weights. We detail those architectures in Figure 3.5. The training hyper-parameters were the same as the FNN.

3.3. Approximating continuous transport maps from discrete samples

In this section we approximate the optimal (continuous) transport maps T from μ to ν using empirical source and target distributions. To avoid ambiguities, we refer as $\hat{\mu}$ and $\hat{\nu}$, to the empirical source and target distributions, respectively.

Following the combinatorial approach detailed in [Peyré et al., 2019] we computed the optimal (discrete) transport map \tilde{T} from $\hat{\mu}$ to $\hat{\nu}$. As an initial benchmark we selected as source distribution a truncated Gaussian distribution supported in the interval $[0, 1]$ with mean 0.45 and standard deviation 0.1. Then we proceeded as follows.

1. We selected as the transport map T the discrete exponential function evaluated in the sample points of the source distribution $\hat{\mu}$. Then we computed $\hat{\nu}$ by push-forwarding T to $\hat{\mu}$. This distribution is shown in orange in Figure 3.7a.

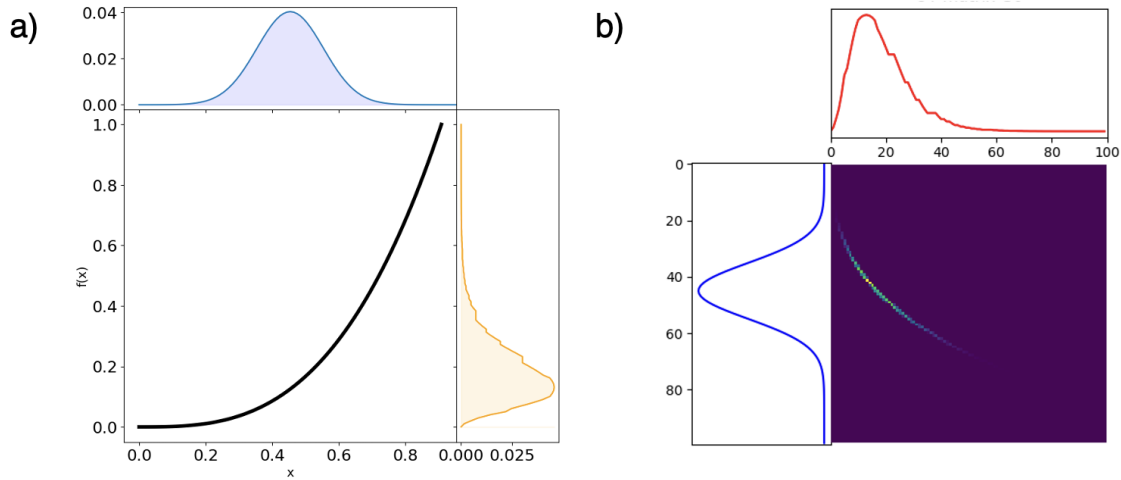


Figure 3.7: Optimal Transport between a source and target distributions solved by different computational methods. On the left we computed the target distribution ν -orange plot on the right axis of figure a)- from 1000 samples of the given blue truncated Gaussian distribution μ with mean 0.45, standard deviation $\sigma = 0.1$, and supported in $x \in [-1, 1]$. In black we see the transport map $T(x) = \exp(x)$. On the right, we see the result of the numerical transportation plan in which the given data are the previous distributions μ and ν , while the transport is computed by the Python library POT. This plot is a color map and it is flipped and inverted with respect to the figure on the right.

2. We now know that the optimal transport map from $\hat{\mu}$ to $\hat{\nu}$ is the discrete exponential function. Therefore, we computed such map using the Python Library POT (Python Optimal Transport). We can see the resulting transport in Figure 3.7b.

Finally, as a further experiment, we generated a target distribution from a sampled given source distribution by using the transport map T obtained as the gradient of the convex function given in Section 3.2 item 1. The resulting distribution is plotted in orange on Figure (3.8).

With the (optimal) discrete transport map shown in Figure 3.8 we then wanted to retrieve the continuous transport map T . To do so we integrated naively as, described in Section 2.2, and obtained the black dots on the right side of Figure 3.9. We then interpolated using our Convex Splines algorithms to get an approximation of f . After taking the numerical gradient of the orange function thus obtained, we retrieve the continuous blue line of Figure 3.9a as a continuous approximation of $T = \nabla f$.

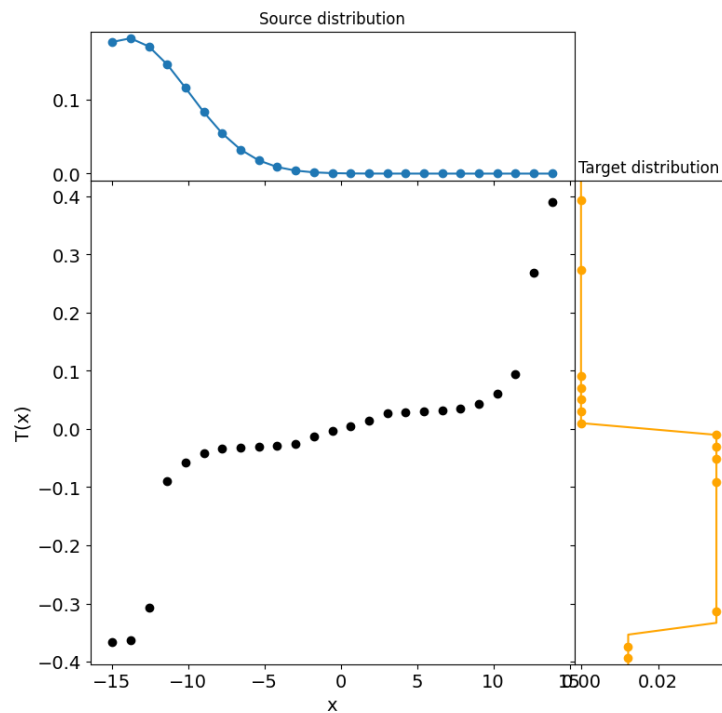


Figure 3.8: Discrete Source (blue) and target (orange) distributions and their associated discrete transport plan depicted in black. The source points are uniformly from the truncated Gaussian distribution with mean -14 and standard deviation 4 and support in $[-15, 15]$. The transport map is $\nabla\varphi$, where φ is the convex function from Figure 3.6. The orange points are the transported images.

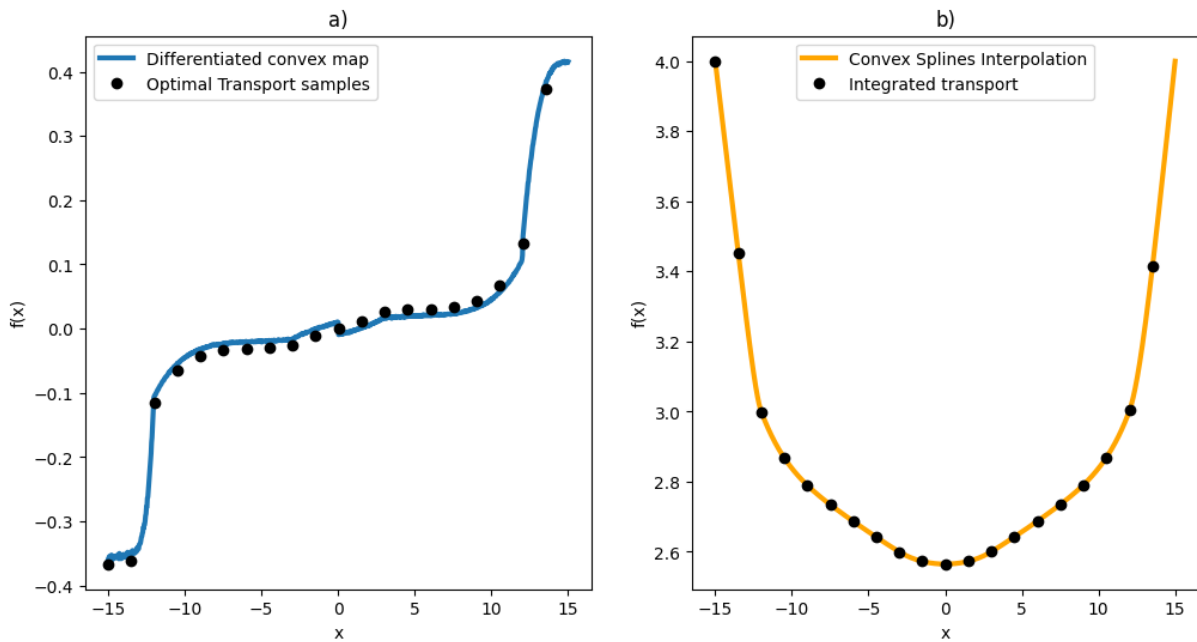


Figure 3.9: Result of retrieving the continuous transport plan T from discrete samples \hat{T} . We start with the black points on the left, that are the result of solving the discrete optimization problem between μ and ν by combinatorial methods. Then, after naive integration, we get the black points in b) to which we apply our convex splines procedure whose result is shown in orange. Finally, by differentiating the orange curve, we obtain the continuous transport map depicted in blue on the left.

3.4. VQLS as linear solver for the convex interpolation problem

Finally, we explored the solution to the convex interpolation problem formulated as a quantum linear system. We do it using the Convex Splines Interpolation method introduced Section 2.1 with our SWAM algorithm -from now on we will refer to this as our Convex Splines method- by relying on the Variational Quantum Linear Solver (VQLS) defined by Bravo-Prieto et al. [2023]. In this framework, given a matrix A , a known vector b , and an unknown vector x , instead of solving $Ax = b$, the algorithm tries to prepare a quantum state $|x(\theta)\rangle = V(\theta)|0\rangle$ such that it corresponds to the classical solution of the problem. In particular, here we solve:

$$|\varphi(\theta)\rangle = \frac{1}{C}A|x\rangle \approx |b\rangle,$$

where C is a normalization constant introduced to guarantee the that $|x(\theta)\rangle$ the resulting quantum state is properly normalized.

Therefore, we are interested in constructing a Variational Quantum Circuit (VQC) $V(\theta)$ such that the similarity between $\varphi(\theta)$ and $|b\rangle$ is maximized, or alternatively, that:

$$C = 1 - |\langle b|\varphi(\theta)\rangle|^2,$$

is minimized. And so, the optimal parameters for the VQC are:

$$\hat{\theta} = \frac{1}{2} \left[1 - \frac{1}{n} \frac{\langle 0|V^\dagger(\theta)U|0\rangle \langle 0|U^\dagger AV|0\rangle}{\langle 0|VA^\dagger AV|0\rangle} \right],$$

with U a unitary operator such that $U|0\rangle = |b\rangle$.

The implemented procedure consists in several steps:

- Decomposition of A as a product of unitary matrices.
- Preparation of $|b\rangle$
- Hadamard test and optimization of the parameters of the variational quantum circuit.

We depict the circuit in Figure 3.10.

As a proof of concept we execute the algorithm for the linear system of an individual spline segment with $s = \left(0, \frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}\right)$. The corresponding A matrix is the following:

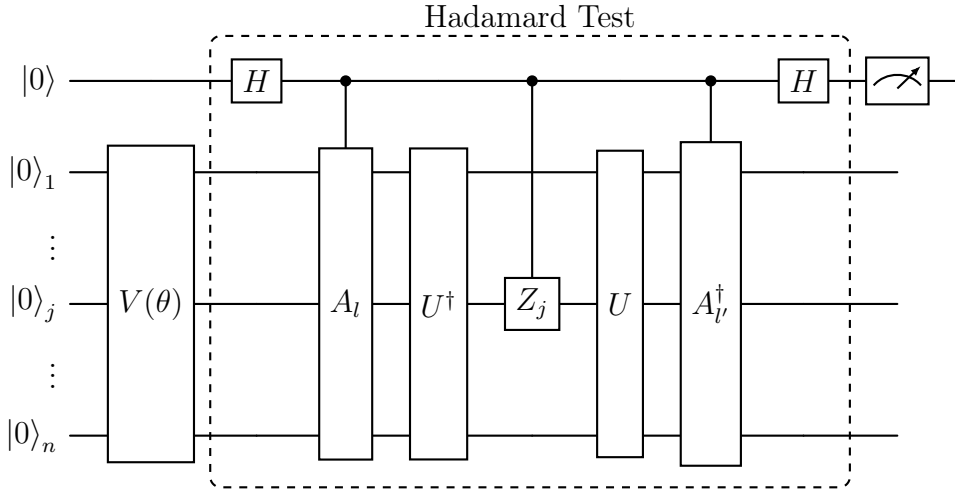


Figure 3.10: Hadamard test used to estimate the real part of $\langle 0|V^\dagger A_l^\dagger Z_j A_l V|0\rangle$. For the imaginary part, it is enough to add a phase shift of $-\pi/2$ after the first Hadamard Gate in the top qubit.

$$A = \begin{bmatrix} 0 & 1.33 & 0 & 1 \\ 1.33 & 0 & 2 & 1 \\ 0 & -2.66 & 1 & 0 \\ 2.66 & 0 & 1 & 0 \end{bmatrix}$$

3.4.1. Decomposition of matrix A into complex linear combinations of tensor products of 2-dimensional unitary matrices

We first decomposed A into linear combinations of tensor products of 2×2 unitary matrices as required by the algorithm. In particular we chose our set of unitaries to be $G = \{\mathbb{1}, X, Y, Z\}$, that is, the Pauli Gates and the Identity gate. We can observe that $\mathbb{G} = G \otimes G$, where \otimes represents the tensor product in $\mathbb{C}^2 \times \mathbb{C}^2$, is a basis of the space of 4×4 matrices. Then, we projected A into that basis using the standard projection procedure from linear algebra. The resulting coefficients are reported next:

$$c = \{0.25, 0.91\bar{6}, 0., 0.25, 0.25, 0.75, 1.58\bar{3}i, -0.25, \\ -0.25i, -0.75i, -1.08\bar{3}, 0.25i, -0.25, 0.41\bar{6}, -0.25i, -0.25\}.$$

And so, we can write $A = \sum_{k=1}^{16} c_k \mathbb{G}_k$, and we can apply the operations as different wires in the VQC 3.10.

3.4.2. Preparation of the quantum state b

We encoded the vector $b = \frac{1}{\sqrt{2}}(0, 1, 0, 1)$ -that corresponds to s in our classical development of the Convex Splines Interpolation- into the quantum state $\frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)$, preparing the quantum circuit X_1H_0 . Indeed:

$$X_1H_0|00\rangle = X_1\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle),$$

where X_1 and H_0 are follow the convention from Remark 1.49.

3.4.3. Hadamard test and optimization of the parameters of the VQC

We trained the Circuit in Figure 3.10 semi-classically, calculating the gradient of the circuit using the shift rule [Mitarai et al., 2018] and evaluated it in the input data using that Quantum Circuit. The weights were optimized via Stochastic gradient descend with a learning rate of 0.8.

Because of the dimensionality of the problem we use 2 qubits to encode the 4×4 matrix A , and an extra auxiliary qubit to perform the Hadamard test. For each iteration we took 10^4 simulated shots. The final MSE obtained was $1.043 \cdot 10^{-3}$.

4 | Discussion

In this section, we comment and discuss the results from the previous Chapter 3.

4.1. On the impracticality of performing convex regression using Variational Quantum Circuits

It is well known that Variational Quantum circuits (VQC) implemented in current state of the art devices suffer from different drawbacks. On the one hand, there is the issue of expressivity. Indeed, [Schuld et al. \[2021\]](#) shows that for VQCs of the type shown in [Figure 3.2](#), the class of represented functions is determined by the encoding protocol. Additionally, they proved that the class of represented functions can be written as:

$$f(x) = \sum_{\omega \in \Omega} c_{\omega} e^{i\omega x},$$

where the space Ω is determined by the particular data encoding protocol, and that the coefficients c_{ω} are determined by the architecture of the circuit - in our case, the number of alternating layers and the FEC.

This fact possesses a significant challenge to perform quantum regression and, in particular, to perform convex quantum regression. On the one hand, one desires the Quantum Circuits to have good expressive power, that is, to be able to represent a wide class of functions, which requires to design architectures where Ω is dense and large. However that implies the construction of large circuits with intricate encoding protocols that are challenging to implement and prone to errors. On the other hand, the fact that the function $f(x)$ is expressed as a Fourier sum, imposes heavy restriction on the valid domain, as well as in the coefficients and frequencies of the sum.

To explore what is the effect of the Fourier-sum-like behaviour of the functions obtained via VQCs, in [Figure 3.3](#) we depict the capabilities of the Quantum Circuit to learn the function $f(x) = \sum_{j=-5}^5 c_j \exp(ix)$. As the theory suggests, there should exist a circuit capable of approximating such function perfectly, since it comes from a Fourier sum.

Despite the fact that we explored only six different configurations, it is noticeable how the resulting function imitate the oscillating pattern of the target function. Indeed, for the circuit with 2 qubits and 10 layers, we see the best fit with a MSE of 0.0081. The rest of the curves, except the ones for circuits with only one layer, also fit the target function at a good level approximation. This verifies how the chosen architecture is adequate to perform regression task with classical data.

Moving forward, we now analyze those regression capabilities for the particular task of convex objective functions. In Figure 3.4 we use different circuit configurations to fit the function $f(x) = x^4 + 30(x - 5)^2$ in the interval $[-1, 1]$, domain in which it is convex. We again observe oscillating patterns of the predicted function as a consequence of to the Fourier-sum-like behaviour described in the previous paragraph. Notably, none of the approximating curves is convex despite the fact that the target function does have that property. It is interesting to observe that increasing the number of qubits and the number of layers, simultaneously reduces the amplitude of the oscillations, as expected. Unfortunately, in real applications it is not feasible to build reliable large and deep circuits to damp the oscillations.

The reason why building deep circuits -that is, with architectures with many consecutive layers- is not feasible, is twofold. On the one hand, we see in Table 3.1 that increasing the number of layers or the number of qubits increases the training time, as expected. However, that increment in computational time is not necessarily reflected in an increment on the approximating power of the circuit. Indeed, observe that the best result on the test set was obtained by the configuration with 8 qubits and 30 layers, instead of that with 40 layers. This result should not be surprising under the light of the Fourier sum written previously. The depth of circuit is augmenting the set of accessible coefficients c_ω , whose effect might not be as beneficial as increasing the set of accessible frequencies ω . On the other hand, if we look back to Figure 3.4, we observe that increasing the number of layers while keeping fixed the number of qubits could be detrimental for the final prediction. The reason for this has to deal with the gradient landscape. Increasing the number of layers also increases the number of training parameters which might result in the presence of Barren Plateaus [McClean et al., 2018].

The presence of Barren Plateaus is the phenomenon in which the gradient landscape becomes almost flat when the dimensionality of the parameters is very large. That could happen either when the circuits are deep or when the number of qubits is large. In those almost-flat landscapes the cost function -that is the function being optimized- becomes exponentially small and makes it difficult for the training algorithm to find the minimum of the cost function. This phenomenon is similar to the vanishing gradient present in

conventional Neural Networks (NN) but in a more critical way due to the exponential decay of the gradient. While in classical NNs strategies like weight initialization, batch normalization or residual connections might be useful, in Variational Quantum Circuits there is no one-fits-all way to solve the problem. This is an open problem in Quantum Machine Learning and an area of active research right now.

Motivated by these reasons, we can safely assure that performing convex regression for **arbitrary** convex functions is impractical due to the inner properties of the quantum circuits. Here we make emphasis in the word arbitrary, because there are works studying different QCs architectures to exploit different types of functional symmetries -like for example rotation symmetries in Meyer et al. [2023]- however, they are bespoke to solve a very narrow class of problems. With that result in mind, we decided to take two alternative approaches. The first one is to make use of the output of the VQC in Figure 3.2 and to project it into the closest convex function numerically - via the Python library `cvxpy`. The second one, to change the framework of the problem and to frame it as a linear solution problem in order to use techniques like the HHL algorithm or the VQLS algorithm that are presumed to have a real quantum advantage against their classical counterparts. We take this second approach by implementing the so called Convex Spline Interpolation.

4.2. On the convenience of Convex Splines Interpolation

At first sight it might appear that substituting the problem of convex regression with that of convex interpolation is too restrictive. However, if we consider our particular setting for which the optimal transport maps are gradients of convex functions, the new framework becomes natural to consider.

In Figure 3.6 we depict the results of the application of our convex splines interpolation algorithm. As expected, we see that it achieves the task of interpolating points with convex and smooth functions. Moreover, thanks to our particular computational implementation, we see that in the case of non-convex functions -middle and right figures- it still performs the interpolation task but in the intervals for which it can interpolate with a convex function, it does it. This gives a little of freedom for the numerical implementations of the solution of the OT problem that is discussed later.

From the same figure, we see some advantages of our method in comparison with the FNN and the ICNN. A remarkable point is that being a linear system that includes the

convexity constrains, it does not need any training, and therefore does not suffer from the optimization problem experienced by the FNN and ICNN. Indeed, in Figure 3.6, in the case of a convex objective function, we see that both algorithms deviate considerably from the real function due to the fact that the training set is limited, and that they cannot extract from it the necessary information to train properly the data-set. However, it is clear that the ICNN outperforms the FNN because it is constrained to produce a convex function. The situation changes when the fitting function is an everywhere non-convex function. In that situation, the convexity constraint enforces the ICNN to be a constant taking the value of the mean of the images of the training points, while the FNN approximates better the dataset. Remarkably, our convex splines method also achieves the task by interpolating the points with a continuous function. Lastly, in the third scenario, we see how the ICNN fails to fit a convex function and still takes the mean value of the training dataset, while the FNN and the convex splines interpolation achieve the task satisfactorily. The greatest difference between the FNN and the convex splines in this setting is that the FNN does not enforce neither convexity nor smoothness. Instead, the convex splines interpolator is a smooth curve that enforces convexity in particular in the regions between the 4th and 6th training points.

4.3. On the approximation of continuous transport maps from discrete samples

Trying to solve the continuous OT problem from sampled data is a goal that has been actively pursued in applied and computational OT because people usually have access to discrete samples rather than to the full continuous distributions. A common strategy found in the literature -see for example [Bunne et al., 2023]- is to tackle the problem by relying in the usage of Brenier’s Theorem 1.35 and so to search for a convex function inducing the optimal transport map. This framework has the advantage of framing the problem in the context of convex optimization for which there are diverse out-of-the-box numerical packages available. However, most of the works relying on this approach overlook the fact that empirical distributions do not satisfy the assumptions of the theorem. Indeed, it does not work for atomic measures -which is just the case of empirical distributions from data samples- because they are not absolutely continuous with respect to the Lebesgue measure.

Our approach to compute continuous OT plans from discrete samples consists in the following steps:

1. To solve the discrete OT problem as a problem of combinatorial optimization as detailed in Chapter 2.
2. To numerically integrate the optimal transport map to obtain samples from the continuous convex function inducing the plan.
3. To interpolate those points using our convex splines method.
4. To differentiate the continuous curve to obtain the continuous OT plan associated to the sampling.

Proceeding in such way, we see how in Figure 3.7 we compare our combinatorial framework against the solution provided by the Python library POT. Unfortunately, the only comparison that we can do between (3.7a) and (3.7b) is visual because POT does not provide the optimal map, but a color plot depicting the transported mass on each point of the grid. However, the visual inspection exercise allow us -provided that we rotate and scale the plots to make them match, a qualitative confirmation of the validity of the combinatorial approach. Many more test were conducted, however, we display only this one as representative.

Having verified the validity of the OT solution, we can now apply it to a real case. In Figure 3.6, we defined the convex function on the left. That function corresponds to a Optimal Transport plan between a pair of probability measures. In Figure 3.8 we show samples from those measures and solve the OT problem obtaining the discrete optimal map depicted in black. We numerically integrate that map to obtain the black points in Figure 3.9b. As we know that the obtained Optimal Transport plan is the gradient of a convex function, we apply our convex splines interpolation method to find the solid curve in the same plot, which coincides with convex function on Figure 3.6. After taking the derivative of the continuous curve, we obtain the continuous line from Figure 3.8 which approximates the original discrete transport plan.

A major advantage of this result is that relying in the formulation of Section 2.1, we only need to formulate a suitable linear problem and solve it with any linear method. This change of perspective allows us to establish a direct link between our solution method and quantum algorithms via Variational Quantum Linear Solvers (VQLSs).

4.4. On the use of the VQLS to perform convex interpolation

We selected a Variational Quantum Linear Solver as our routine because we want our method to be implementable in current NISQ and ISQ devices. The main challenge about this implementation is the preparation of the quantum state $|b\rangle$. We selected a particular case where the dataset has only 3 points: $\mathcal{D} = \{(-2, -\frac{1}{\sqrt{2}}), (0, 0), (2, \frac{1}{\sqrt{2}})\}$. The reason for that is the simplicity of preparing the quantum state encoding the linear system for the convex splines method. Those states can be easily prepared with a simple product of X , Z , and H operators, which allow us to compute the solution.

As described in the result section, and reported with the corresponding error metric we successfully applied the method to this case applying the VQLS circuit depicted in Figure 3.10. However, it is important to take into account different considerations that might limit the utility of considering this approach for general cases:

1. **Normalization conditions on the input:** In order to transform the vector $s = [s_0, s_1, s'_0, s'_1]$ from the linear system $Ax = s$ -see Equation 2.1- into the quantum state $|b\rangle$, it is mandatory to guarantee the vector is normalized. Indeed, from the quantum theory, for $|b\rangle$ to be a valid quantum state, this condition is essential. At the end of the quantum encoding we want to obtain $|b\rangle = [\tilde{s}_0, \tilde{s}_1, \tilde{s}'_0, \tilde{s}'_1]$. Normalizing s might alter the scale of the matrix A -or the vector x - and that fact should be considered when decoding the solution state $|x\rangle$.
2. **Knowledge of the explicit encoding circuit U :** As seen in Circuit 3.10, U and U^\dagger make part of the Hadamard test, and therefore we need to be able to prepare them accordingly. For non-trivial values of the components of $|b\rangle$ this might not be an easy task, since preparing a quantum circuit involves the application of a sequence of unitary rotations of $|0\rangle$ along the surface of the Bloch's sphere.
3. **Easy preparation of U :** As mentioned in the previous literal, the quantum encoding of an arbitrary vector s might be extremely difficult. On top of that difficulty, U should be easily realizable, or equivalently, it has to be a shallow and absent of many complicated entanglement operations between qubits or multi-qubit gates. This is because, as we discussed previously, deep circuits are prone to errors and might damage the quality of the solution. The same applies for multi-qubit gates in the current NISQ and ISQ devices.
4. **Semiclassical solution:** Even if our architectures requires preparing and evalu-

ating a quantum circuit, the training of the variational component is done classically via stochastic gradient descent.

4.5. On the solution of the OT problem using convex interpolation and VQLSs

In Section 4.3 we discussed how to solve the OT problem using our proposed method for convex interpolation. In Section 4.4 we discussed how to perform convex interpolation by solving the linear system using Variational Quantum Circuits. In this section we discuss how to combine both methods and the possible benefits and drawbacks of doing so.

With the OT framework defined in Section 4.3 it is straightforward to realize that it suffices to replace the classical linear solver for the Variational Quantum Linear Solver (VQLS). The potential benefit of doing so is clear in case of large datasets, that is in setting with many interpolation segments. The complexity of the VQLS scales logarithmically with the number of segments $\approx \mathcal{O}(\log(K))$ -see [Bravo-Prieto et al., 2023] for a detailed discussion on the complexity of the algorithm- which is advantageous with respect to the classical algorithms for solution of classical systems that scale at least like $\mathcal{O}(K^2)$ for algorithms like LU decomposition or Cholesky decomposition.

However, it is important to note that the potential quantum speedups might disappear in practice when the preparation of the state U is not simple. Another important factor to consider is that if the circuit U is not explicitly defined, there is no way to solve the problem using this method. This last fact restricts the applicability of the method in general setting and makes it only suitable for states where a simple encoding protocol is known.

As the result of this section we conclude that even if this method of solution is possible it is not practical for many real-life scenarios.

5 | Conclusions

Throughout this work we have studied how to solve the OT problem in classical and quantum settings. We have shown how, despite the difficulties, it is possible to approximate continuous Optimal Transport plans from discrete samples using convex splines interpolation, and how Variational Quantum Circuits can be incorporated to that framework via the Variational Quantum Linear Solver (VQLS). As we discussed in Section 4, the quantum usage of this method is impractical in general settings, albeit possible. And claimed that for large datasets whose associated s vector could be easily encoded into a quantum state $|b\rangle$ the VQLS approach might be beneficial thanks to its logarithmic scaling in the number of segments that is superior to the quadratic scaling of common linear solver algorithms.

As a continuation of this work, we propose the following extensions and further developments:

1. **To explore different data encoding mechanisms:** In this work we encoded classical data into quantum states using X-rotations. An optimal first extension to this work would involve exploring different encoding protocols like Hamiltonian encoding or Quantum Error Correction Codes. These modifications can be beneficial to the expressive power of the Quantum Circuits, but most likely will also impact the computational complexity of the calculations and will make training slower. Therefore, it is necessary to investigate the effect of different encoding, and their resource impact.
2. **To use a combinatorial optimization quantum algorithm to solve the discrete OT problem:** Maybe the easiest way to incorporate quantum into the solution of the discrete OT problem is via a combinatorial quantum optimization framework. Since in this scenario the search space is discrete and finite, we can rely on a Quantum Approximate Optimization Algorithm (QAOA) as the main solving routine. It is not granted that using a QAOA would represent any computational advantage against classical algorithms, but it is worth exploring this direction and looking for optimizing different architectures in order to decrease the complexity of

the algorithms and increase their accuracy.

3. **To train a Variational Quantum Circuit with loss penalizing non-convex predictions:** An alternative and potentially beneficial way to solve the convex regression problem using quantum circuits is to incorporate a penalization term in the loss function of architectures like (3.2) accounting for the convexity of the outcome function. Despite being an easy task in classical Neural Networks, in quantum this would imply the design of a quantum circuit whose goal is to test convexity. As for today this approach has not been explored and there are no related works in the literature.

A rigorous development of the last points will greatly contribute to answering the question about the feasibility of the usage of quantum circuits to solve OT problems.

Bibliography

- L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.
- B. Amos, L. Xu, and J. Z. Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR, 2017.
- O. Bratteli and D. W. Robinson. *Operator algebras and quantum statistical mechanics: Volume 1: C*-and W*-Algebras. Symmetry Groups. Decomposition of States*. Springer Science & Business Media, 2012.
- C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles. Variational quantum linear solver. *Quantum*, 7:1188, 2023.
- C. Bunne, S. G. Stark, G. Gut, J. S. Del Castillo, M. Levesque, K.-V. Lehmann, L. Pelkmans, A. Krause, and G. Rättsch. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11):1759–1768, 2023.
- V. Bužek and M. Hillery. Quantum copying: Beyond the no-cloning theorem. *Physical Review A*, 54(3):1844, 1996.
- G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.
- P. J. Davis and P. Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.
- D. Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.
- P. A. M. Dirac. A new notation for quantum mechanics. In *Mathematical proceedings of the Cambridge philosophical society*, volume 35, pages 416–418. Cambridge University Press, 1939.

- P. A. M. Dirac. *The principles of quantum mechanics*. Number 27. Oxford university press, 1981.
- R. P. Feynman. Simulating physics with computers. In *Feynman and computation*, pages 133–153. CRC Press, 2018.
- A. Figalli and F. Glaudo. *An invitation to optimal transport, Wasserstein distances, and gradient flows*. 2021.
- F. Gieres. Mathematical surprises and dirac’s formalism in quantum mechanics. *Reports on Progress in Physics*, 63(12):1893, 2000.
- A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.
- C. Heunen and J. Vicary. *Categories for Quantum Theory: an introduction*. Oxford University Press, 2019.
- J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):4812, 2018.
- A. Messac and A. Sivanandan. A new family of convex splines for data interpolation. *Computer aided geometric design*, 15(1):39–59, 1997.
- J. J. Meyer, M. Mularski, E. Gil-Fuster, A. A. Mele, F. Arzani, A. Wilms, and J. Eisert. Exploiting symmetry in variational quantum machine learning. *PRX Quantum*, 4(1):010328, 2023.
- K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.
- M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*, volume 2. Cambridge university press Cambridge, 2001.
- A. Peres. *Quantum theory: concepts and methods*, volume 72. Springer, 1997.
- G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
- R. T. Rockafellar. *Convex Analysis:(PMS-28)*. Princeton university press, 2015.
- M. Schuld, R. Sweke, and J. J. Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

- C. Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.
- C. Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.
- J. Von Neumann. *Mathematical foundations of quantum mechanics: New edition*, volume 53. Princeton university press, 2018.
- H. M. Wiseman and G. J. Milburn. *Quantum measurement and control*. Cambridge university press, 2009.

