

REPORT ON PROGRESS

## Machine learning & artificial intelligence in the quantum domain: a review of recent progress

To cite this article: Vedran Dunjko and Hans J Briegel 2018 *Rep. Prog. Phys.* **81** 074001

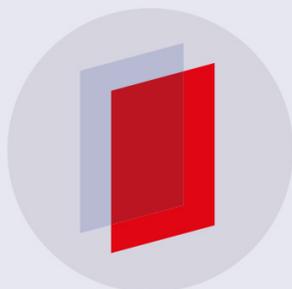
View the [article online](#) for updates and enhancements.

### Related content

- [Science in the Arena: Doping in sport](#)  
B Baker
- [Quantum-enhanced deliberation of learning agents using trapped ions](#)  
V Dunjko, N Friis and H J Briegel
- [Automated Car Park Management System](#)  
J P Fabros, D Tabañag, A Espra et al.

### Recent citations

- [Reinforcement learning for semi-autonomous approximate quantum eigensolver](#)  
F Albarrán-Arriagada *et al*
- [Quantum Fisher information matrix and multiparameter estimation](#)  
Jing Liu *et al*
- [Visual assessment of multi-photon interference](#)  
Fulvio Flamini *et al*



**IOP | ebooks™**

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

## Report on Progress

# Machine learning & artificial intelligence in the quantum domain: a review of recent progress

Vedran Dunjko<sup>1,2</sup>  and Hans J Briegel<sup>1,3</sup> 

<sup>1</sup> Institute for Theoretical Physics, University of Innsbruck, Innsbruck 6020, Austria

<sup>2</sup> Max Planck Institute of Quantum Optics, Garching 85748, Germany

<sup>3</sup> Department of Philosophy, University of Konstanz, Konstanz 78457, Germany

E-mail: [vedran.dunjko@mpq.mpg.de](mailto:vedran.dunjko@mpq.mpg.de) and [hans.briegel@uibk.ac.at](mailto:hans.briegel@uibk.ac.at)

Received 29 September 2017, revised 14 February 2018

Accepted for publication 5 March 2018

Published 19 June 2018



CrossMark

Corresponding Editor Professor Maciej Lewenstein

### Abstract

Quantum information technologies, on the one hand, and intelligent learning systems, on the other, are both emergent technologies that are likely to have a transformative impact on our society in the future. The respective underlying fields of basic research—quantum information versus machine learning (ML) and artificial intelligence (AI)—have their own specific questions and challenges, which have hitherto been investigated largely independently. However, in a growing body of recent work, researchers have been probing the question of the extent to which these fields can indeed learn and benefit from each other. Quantum ML explores the interaction between quantum computing and ML, investigating how results and techniques from one field can be used to solve the problems of the other. Recently we have witnessed significant breakthroughs in both directions of influence. For instance, quantum computing is finding a vital application in providing speed-ups for ML problems, critical in our ‘big data’ world. Conversely, ML already permeates many cutting-edge technologies and may become instrumental in advanced quantum technologies. Aside from quantum speed-up in data analysis, or classical ML optimization used in quantum experiments, quantum enhancements have also been (theoretically) demonstrated for interactive learning tasks, highlighting the potential of quantum-enhanced learning agents. Finally, works exploring the use of AI for the very design of quantum experiments and for performing parts of genuine research autonomously, have reported their first successes. Beyond the topics of mutual enhancement—exploring what ML/AI can do for quantum physics and vice versa—researchers have also broached the fundamental issue of quantum generalizations of learning and AI concepts. This deals with questions of the very meaning of learning and intelligence in a world that is fully described by quantum mechanics. In this review, we describe the main ideas, recent developments and progress in a broad spectrum of research investigating ML and AI in the quantum domain.

Keywords: quantum computing, artificial intelligence, machine learning, quantum information processing

(Some figures may appear in colour only in the online journal)

**Contents**

1. Introduction.....	2	6. Quantum enhancements for machine learning.....	38
1.1. Quantum mechanics, computation and information processing.....	3	6.1. Learning efficiency improvements: sample complexity.....	38
1.2. Artificial intelligence and machine learning.....	6	6.1.1. Quantum PAC learning.....	39
1.2.1. Learning from data: machine learning.....	7	6.1.2. Learning from membership queries.....	40
1.2.2. Learning from interaction: reinforcement learning.....	8	6.2. Improvements in learning capacity.....	41
1.2.3. Intermediary learning settings.....	9	6.2.1. Capacity from amplitude encoding.....	41
1.2.4. Putting it all together: the agent–environment paradigm.....	10	6.2.2. Capacity via quantized Hopfield networks.....	41
1.3. Miscellanea.....	11	6.3. Run-time improvements: computational complexity.....	42
2. Classical background.....	11	6.3.1. Speed-up via adiabatic optimization.....	43
2.1. Methods of machine learning.....	11	6.3.2. Speed-ups in circuit architectures.....	45
2.1.1. Artificial neural networks for supervised and unsupervised learning.....	12	7. Quantum learning agents and elements of quantum artificial intelligence.....	51
2.1.2. Support vector machines for supervised learning.....	14	7.1. Quantum learning via interaction.....	51
2.1.3. Other models.....	16	7.2. Quantum agent–environment paradigm for reinforcement learning.....	55
2.2. Mathematical theories of supervised and inductive learning.....	17	7.2.1. Agent–environment-based classification of quantum machine learning.....	57
2.2.1. Computational learning theory.....	18	7.3. Towards quantum artificial intelligence.....	57
2.2.2. Vapnik–Chervonenkis theory.....	19	8. Outlook.....	58
2.3. Basic methods and theory of reinforcement learning.....	21	Acknowledgments.....	60
3. Quantum mechanics, learning and artificial intelligence.....	24	References.....	60
4. Machine learning applied to (quantum) physics.....	25		
4.1. Hamiltonian estimation and metrology.....	25		
4.1.1. Hamiltonian estimation.....	26		
4.1.2. Phase estimation settings.....	26		
4.1.3. Generalized Hamiltonian estimation settings.....	27		
4.2. Design of target evolutions.....	28		
4.2.1. Off-line design.....	28		
4.2.2. On-line design.....	28		
4.3. Controlling quantum experiments, and machine-assisted research.....	29		
4.3.1. Controlling complex processes.....	29		
4.3.2. Learning how to experiment.....	31		
4.4. Machine learning in condensed-matter and many-body physics.....	31		
5. Quantum generalizations of machine learning concepts.....	32		
5.1. Quantum generalizations: machine learning of quantum data.....	33		
5.1.1. State discrimination, state classification and machine learning of quantum data.....	33		
5.1.2. Computational learning perspectives: quantum states as concepts.....	35		
5.2. (Quantum) learning and quantum processes.....	36		

**1. Introduction**

Quantum theory has influenced most branches of the physical sciences. This influence ranges from minor corrections to profound overhauls, particularly in fields dealing with sufficiently small scales. In the second half of the last century, it became apparent that genuine quantum effects can also be exploited in engineering-type tasks, where such effects enable features which are superior to those achievable using purely classical systems. The first wave of such engineering gave us, for example, the laser, transistors and nuclear magnetic resonance devices. The second wave, which gained momentum in the 1980s, constitutes a broad-scale, albeit not fully systematic, investigation of the potential of utilizing quantum effects for various types of tasks which, at a fundamental level, deal with the processing of information. This includes the research areas of cryptography, computing, sensing and metrology, all of which now share the common language of quantum information science. Often, the research into such interdisciplinary programs was exceptionally fruitful. For instance, quantum computation, communication, cryptography and metrology are now mature, well-established and impactful research fields which have, arguably, revolutionized the way we think about information and its processing.

In recent years, it has become apparent that the exchange of ideas between quantum information processing and the fields of artificial intelligence and machine learning has its own genuine questions and promises. Although such lines of research are only now receiving a broader recognition, the very first ideas were present already at the early days of quantum computing, and we have made an effort to fairly acknowledge such visionary works.

In this review we aim to capture research at the interplay between machine learning, artificial intelligence and quantum mechanics in its broad scope, with a reader with a physics background in mind. To this end, we dedicate a comparatively large amount of space to classical machine learning and artificial intelligence topics, which are often sacrificed in physics-oriented literature, while keeping the quantum information aspects concise.

The structure of the paper is as follows. In the remainder of this introductory section 1, we give quick overviews of the relevant basic concepts of the fields of quantum information processing, machine learning and artificial intelligence. We finish the introduction with a list of abbreviations and a comment on notation. Subsequently, in section 2 we delve deeper into chosen methods, technical details and the theoretical background of the classical theories. The selection of topics here is not necessarily balanced, from a classical perspective. We place an emphasis on elements which either appear in subsequent quantum proposals, which can sometimes be somewhat exotic, or on aspects which can help put the relevance of the quantum results into proper context. Section 3 briefly summarizes the topics that will be covered in the quantum part of the review. Sections 4–7 cover the four main topics we survey and constitute the central body of the paper. We finish with an outlook section 8.

**Remark.** The overall objective of this survey is to give a broad, ‘bird’s-eye’ account of the topics which contribute to the development of various aspects of the interplay between quantum information sciences and machine learning and artificial intelligence. Consequently, this survey does not necessarily present all the developments in a fully balanced fashion. Certain topics, which are in their very early stages of investigation, yet important to the nascent research area, have been given what is perhaps a disproportionate level of attention, compared to more developed themes. This is, for instance, particularly evident in section 7, which aims to address the topics of quantum artificial intelligence beyond mainstream data analysis applications of machine learning. While this topic is relevant for a broad perspective on the emerging field, it has only been broached by very few authors, including the authors of this review and their collaborators. The more extensively explored topics of, e.g., quantum algorithms for machine learning and data mining, quantum computational learning theory or quantum neural networks have been addressed in more focused recent reviews (Wittek 2014a, Schuld *et al* 2014b, Biamonte *et al* 2016, Arunachalam and de Wolf 2017, Ciliberto *et al* 2017).

## 1.1. Quantum mechanics, computation and information processing

**Executive summary:** Quantum theory leads to many counterintuitive and fascinating phenomena, including the results of the field of quantum information processing and, in particular, quantum computation. This field studies the intricacies of quantum information, its communication, processing and use. Quantum information admits a plethora of phenomena which do not occur in classical physics. For instance, quantum information cannot be cloned—this restricts the types of processing that are possible for general quantum information. Other aspects lead to advantages, as has been shown for various communication and computation tasks: for solving algebraic problems, reduction of sample complexity in black-box settings, sampling problems and optimization. Even restricted models of quantum computation, amenable for near-term implementations, can solve interesting tasks. Machine learning and artificial intelligence tasks can, as components, rely on the solving of such problems, leading to an advantage.

Quantum mechanics, as commonly presented in quantum information, is based on few simple postulates: (1) the pure state of a quantum system is given by a unit vector  $|\psi\rangle$ <sup>4</sup> in a complex Hilbert space, (2) closed system pure state evolution is generated by a Hamiltonian  $H$ , specified by the linear Schrödinger equation  $i\hbar\frac{\partial}{\partial t}|\psi\rangle = H|\psi\rangle$ , (3) the structure of composite systems is given by the tensor product and 4) projective measurements (observables) are specified by, ideally, non-degenerate Hermitian operators, and the measurement process changes the description of the observed system from state  $|\psi\rangle$  to an eigenstate  $|\phi\rangle$ , with probability given by the Born rule  $p(\phi) = |\langle\psi|\phi\rangle|^2$  (Nielsen and Chuang 2011). While the full theory still requires the handling of subsystems and classical ignorance<sup>5</sup>, the few mathematical axioms of pure-state closed-system theory already give rise to quintessentially quantum phenomena, like superpositions, no-cloning, entanglement and others, most of which stem from just the linearity of the theory. Many of these properties re-define how researchers in quantum information perceive what information is, but also have a critical functional role in, say, quantum enhanced-cryptography, communication, sensing and other applications. Some of the most fascinating consequences of quantum theory are, arguably, captured by the field of quantum information processing (QIP), and in particular quantum computation (QC), which is most relevant to our purposes.

<sup>4</sup> More precisely, a ray, or a one-dimensional projector onto  $|\psi\rangle$  in the same Hilbert space.

<sup>5</sup> This requires the more general and richer formalism of density operators, and leads to generalized measurements, completely positive evolutions, etc.

QC has revolutionized the theory and implementation of computation. This field originated from the observations by Manin (1980) and Feynman (1982) that the calculation of certain properties of quantum systems, as they evolve in time, may be intractable, while the quantum systems themselves, in a manner of speaking, do perform that hard computation by merely evolving. From these early ideas QC has proliferated and, indeed, the existence of quantum advantages which are offered by scalable universal quantum computers have been demonstrated in many settings. Perhaps most famously, quantum computers have been shown to have the capacity to efficiently solve algebraic computational problems which are believed to be intractable for classical computers. This includes the well known problems of factoring large integers and computing the discrete logarithms (Shor 1997), but also many others such as Pell equation solving, some non-Abelian hidden subgroup problems and others; see e.g. Childs and van Dam (2010) and Montanaro (2016) for a review. Related to this, nowadays we also have access to a growing collection of quantum algorithms<sup>6</sup> for various linear algebra tasks, as given in e.g. Harrow *et al* (2009), Childs *et al* (2015) and Reberntrost *et al* (2016b), which may offer speed-ups.

Quantum computers can also offer improvements in many optimization and simulation tasks, for instance, computing certain properties of partition functions (Poulin and Wocjan 2009), simulated annealing (Crosson and Harrow 2016), solving semidefinite programs (Brandão and Svore 2017), performing approximate optimization (Farhi *et al* 2014) and, naturally, in the tasks of simulating quantum systems (Georgescu *et al* 2014).

Advantages can also be achieved in terms of the efficient use of sub-routines and databases. This is studied using oracular models of computation, where the quantity of interest is the number of calls to an *oracle*, a black-box object with a well-defined set of input–output relations which, abstractly, stands in for a database, sub-routine, or any other information processing resource (see figure 1). The canonical example of a quantum advantage in this setting is the Grover’s search algorithm (Grover 1996) which achieves a provably optimal quadratic improvement in unordered search (where the oracle is the database). Similar results have been achieved in a plethora of other scenarios, such as spatial search (Childs and Goldstone 2004), search over structures (including various quantum-walk-based algorithms (Kempe 2003, Childs *et al* 2003, Reitzner *et al* 2012)), NAND (Childs *et al* 2009) and more general boolean tree evaluation problems (Zhan *et al* 2012), as well as more recent

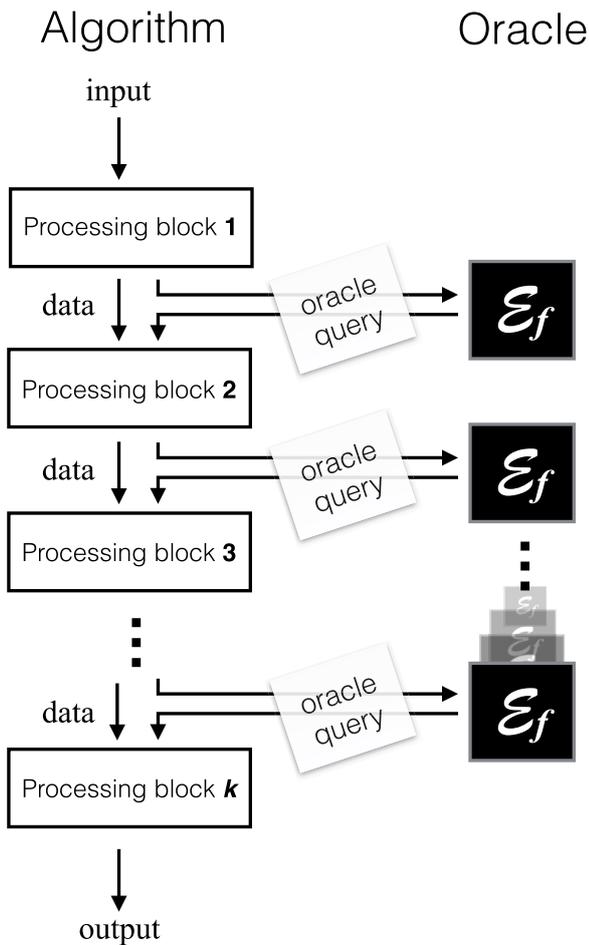
‘cheat sheet’ technique results (Aaronson *et al* 2016) leading to better-than-quadratic improvements. Taken a bit more broadly, oracular models of computation can also be used to model communication tasks, where the goal is to reduce communication complexity (i.e. the number of communication rounds) for some information exchange protocols (de Wolf 2002). Quantum computers can also be used for solving sampling problems. In sampling problems the task is to produce a sample according to an (implicitly) defined distribution, and they are important for both optimization and (certain instances of) algebraic tasks<sup>7</sup>.

For instance, Markov Chain Monte Carlo methods, arguably the most prolific set of computational methods in natural sciences, are designed to solve sampling tasks, which in turn, can often be used to solve other types of problems. For instance, in statistical physics, the capacity to sample from Gibbs distributions is often the key tool to compute properties of the partition function. A broad class of quantum approaches to sampling problems focuses on quantum enhancements of such Markov Chain methods (Temme *et al* 2011, Yung and Aspuru-Guzik 2012). Sampling tasks have been receiving an ever increasing amount of attention in the QIP community, as we will comment on shortly. Quantum computers are typically formalized in one of a few standard *models of computation*, many of which are, computationally speaking, equally powerful<sup>8</sup>. Even if the models are computationally equivalent, they are conceptually different. Consequently, some are better suited, or more natural, for a given class of applications. Historically, the first formal model, the quantum Turing machine (Deutsch 1985), was preferred for theoretical and computability-related considerations. The quantum circuit model (Nielsen and Chuang 2011) is standard for algebraic problems. The measurement-based QC (MBQC) model (Raussendorf and Briegel 2001, Briegel *et al* 2009) is, arguably, best-suited for graph-related problems (Zhao *et al* 2016), multi-party tasks, distributed computation (Kashefi and Pappa 2016) and blind QC (Broadbent *et al* 2009). Topological QC (Freedman *et al* 2002) was an inspiration for certain knot-theoretic algorithms (Aharonov *et al* 2006), and is closely related to algorithms for topological error-correction and fault tolerance.

<sup>7</sup> Optimization and computation tasks can be trivially regarded as special cases of sampling tasks, where the target distribution is (sufficiently) localized at the solution.

<sup>8</sup> Various notions of ‘equally powerful’ are usually expressed in terms of algorithmic reductions. In QIP, typically, the computational model B is said to be at least as powerful as the computational model A if any algorithm of complexity  $O(f(n))$  (where  $f(n)$  is some scaling function, e.g. ‘polynomial’ or ‘exponential’) defined for model A can be efficiently (usually this means in polynomial time) translated to an algorithm for B which solves the same problem and whose computational complexity is  $O(\text{poly}(f(n)))$ . Two models are then equivalent if A is as powerful as B and B is as powerful as A. Which specific reduction complexity we care about (polynomial, linear, etc) depends on the setting: e.g. for factoring polynomial reductions are interesting, since there seems to be an exponential separation between classical and quantum computation. In contrast, for search, the reductions need to be sub-quadratic to maintain a quantum speed-up, since only a quadratic improvement is achievable.

<sup>6</sup> In this review it makes sense to point out that the term ‘quantum algorithm’ is a bit of a misnomer, as what we really mean is ‘an algorithm for a quantum computer’. An algorithm—an abstraction—cannot *per se* be ‘quantum’, and the term quantum algorithm could also have meant e.g. ‘algorithm for describing or simulating quantum processes’. Nonetheless, this term, in the sense of ‘algorithm for a quantum computer’ is commonplace in QIP, and we use it in this sense as well. The concept of ‘quantum machine learning’ is, however, still ambiguous in this sense and, depending on the authors, can easily mean ‘quantum algorithm for machine learning’, or ‘machine learning applied to QIP’.



**Figure 1.** Oracular computation and query complexity: a (quantum) algorithm solves a problem by intermittently calling a black-box subroutine, defined only via its input–output relations. Query complexity of an algorithm is the number of calls to the oracle which the algorithm will perform.

The adiabatic QC model (Farhi *et al* 2000) is constructed with the task of ground-state preparation in mind, and is thus well-suited to optimization problems (Heim *et al* 2017).

Research into QIP also produced examples of interesting restricted models of computation: models which are in all likelihood not universal for efficient QC but can still solve tasks which seem hard for classical machines. Recently, there has been an increasing interest in such models, specifically in the linear optics model, the so-called low-depth random circuits model and the commuting quantum circuits model<sup>9</sup>. In Aaronson and Arkhipov (2011) it was shown that the linear optics model can efficiently produce samples from a distribution specified by the permanents of certain matrices, and it was proven (barring certain mathematical conjectures, which are, however, plausible) that classical

<sup>9</sup>Other restricted models exist, such as the one-clean-qubit model (DQC1) where the input comprises only one qubit in a pure state and others are maximally mixed. This model can be used to compute a function—the normalized trace of a unitary specified by a quantum circuit—which seems to be hard for classical devices.

computers cannot reproduce the samples from the same distribution in polynomial time. Similar claims have been made for low-depth random circuits (Boixo *et al* 2016, Bravyi *et al* 2017) and commuting quantum circuits, which comprise only commuting gates (Shepherd and Bremner 2009, Bremner *et al* 2017). Critically, these restricted models can be realized to sufficient size to allow for a demonstration of computations which the most powerful classical computers currently available cannot achieve, with near-term technologies. This milestone, referred to as *quantum supremacy* (Preskill 2012, Lund *et al* 2017) has been receiving a significant amount of attention in recent times. A lively class of research here focuses on the computational capacity of quantum devices with a restricted architecture, often restricted depth. Here much of the computational work is delegated to the classical machine, which optimizes the parameters of the (shallow) circuit. Two related and prominent examples in this direction are the quantum approximate optimization algorithm (Farhi *et al* 2014) and the Quantum Variational Eigensolver (Peruzzo *et al* 2014, McClean *et al* 2016). A table listing some of the more common restricted and universal models of quantum computation, with typical applications is given in figure 2.

Another highly active field in QIP concentrates on (analog) quantum simulations, with applications in quantum optics, condensed-matter systems and quantum many-body physics (Georgescu *et al* 2014). Many, if not most, of the above mentioned aspects of QC are finding a role in quantum machine learning applications.

Next, we briefly review basic concepts from the classical theories of artificial intelligence and machine learning.

**Executive summary:** The field of artificial intelligence (AI) incorporates various methods, which are predominantly focused on solving problems which are hard for computers, yet seemingly easy for humans. Perhaps the most important class of such tasks pertain to learning problems. Various algorithmic aspects of learning problems are tackled by the field of machine learning, which evolved from the study of pattern recognition in the context of AI. Modern machine learning addresses a variety of learning scenarios, dealing with learning from data, e.g. supervised (data classification), and unsupervised (data clustering) learning, or from interaction, e.g. reinforcement learning. Modern AI states, as its ultimate goal, the design of an intelligent agent which learns and thrives in unknown environments. Artificial agents that are intelligent in a general, *human* sense must have the capacity to tackle all the individual problems addressed by machine learning and other more specialized branches of AI. They will, presumably, require a complex combination of techniques.

List of models (BQP-complete)	applications (not exclusive)
QTM	theory
QCircuits	algorithms
MBQC	distributed computing
Topological	knot-theoretic problems
Adiabatic	optimization problems
List of models (restricted)	applications
DQC1	computing trace of unitary
Linear Optics	sampling
Shallow Random Q. Circuits	sampling
Commuting Q. Circuits	sampling
RestrictedAdiabatic	optimization tasks

**Figure 2.** Computational models.

### 1.2. Artificial intelligence and machine learning

In its broadest scope, the modern field of AI encompasses a wide variety of sub-fields. Most of these sub-fields deal with the understanding and abstracting of aspects of various human capacities which we would describe as intelligent, and attempt to realize the same capacities in machines. The term ‘AI’ was coined at Dartmouth College conferences in the 1956 (Russell and Norvig 2009), which were organized to develop ideas about machines that can think, and the conferences are often cited as the birthplace of the field. The conferences aimed to ‘find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves’<sup>10</sup>. The history of the field has been turbulent, with strong opinions on how AI should be achieved. For instance, over the course of its first 30 years, the field has crystallized into two main competing and opposite viewpoints (Eliasmith and Bechtel 2006) on how AI may be realized: *computationalism*—holding that that the mind functions by performing purely formal operations on symbols, in the manner of a Turing machine; see e.g. Newell and Simon (1976), and *connectionism*—which models mental and behavioral phenomena as the emergent processes of interconnected networks of simple units, mimicking the biological brain; see e.g. Medler (1998)). Aspects of these two viewpoints still influence approaches to AI. Irrespective of the underlying philosophy, for the larger part of the history of AI the realization of ‘genuine AI’ was, purportedly, perpetually ‘a few years away’—a feature often attributed also to quantum computers by critics of the field. In the case of AI, such run-away optimism had a severe calamitous effect on the field, in multiple instances, especially in the context of funding (leading to periods now dubbed ‘winters of AI’). By the late 90s the reputation of the field was low and, even in hindsight, there was no consensus on the reasons why AI failed to produce human-level intelligence. Such factors played a vital role in the fragmentation of the field into various sub-fields which focused on specialized tasks, often appearing under different names.

<sup>10</sup> Paraphrased from McCarthy *et al* (1955).

A particularly influential perspective of AI, often called *nouvelle* or *embodied* AI, was advocated by Brooks, which posited that intelligence emerges from (simple) embodied systems which learn through interaction with their environments (Brooks 1990). In contrast to standard approaches of the time, Nouvelle AI insists on learning, rather than having properties pre-programmed, and on the embodiment of AI entities, as opposed to abstract entities like chess playing programs. To a physicist, this perspective that intelligence is embodied is reminiscent of the viewpoint that *information is physical*, which had been ‘the rallying cry of quantum information theory’ (Steane 1998). Such embodied approaches are particularly relevant in robotics where the key issues involve perception (the capacity of the machine to interpret the external world using its sensors, which includes computer vision, machine hearing and touch), motion and navigation (critical in, e.g., automated cars). Related to human-computer interfaces, AI also incorporates the field of natural language processing which includes language understanding—the capacity of the machine to derive meaning from natural language—and language generation—the ability of the machine to convey information in a natural language.

Other general aspects of AI pertain to a few well-studied capacities of intelligent entities (Russell and Norvig 2009). For instance, automated planning is related to decision theory<sup>11</sup> and, broadly speaking, addresses the task of identifying strategies (i.e. sequences of actions) which need to be performed in order to achieve a goal, while minimizing (a specified) cost.

Already the simple class of so-called *off-line* planning tasks, where the task, cost function and the set of possible actions are known beforehand, contains genuinely hard problems, e.g. it includes, as a special case, the NP-complete<sup>12</sup> traveling salesman problem (TSP); for illustration see figure 3<sup>13</sup>.

In modern times, TSP itself would no longer be considered a genuine AI problem, but it serves to illustrate how already very specialized, simple sub-sub-tasks of AI may be hard. More general planning problems also include *on-line* variants, where not everything is known beforehand (e.g. TSP but where the ‘map’ may fail to include all the available roads—or roads may effectively disappear due to traffic jams or rerouting—and one simply has to actually travel to find good strategies). On-line planning overlaps with reinforcement learning, discussed later in this section. Closely related to planning is the capacity of intelligent entities for problem solving. In technical literature, problem solving is distinguished from planning by a lack of additional structure in the problem, usually assumed in planning—in other words, problem solving is more general and typically more

<sup>11</sup> Not to be confused with decision problems, studied in algorithmic complexity.

<sup>12</sup> Roughly speaking, NP is the class of decision (yes, no) problems whose solutions can be efficiently verified by a classical computer in polynomial time. NP-complete problems are the hardest problems in NP in the sense that any other NP problem can be reduced to an NP complete problem via polynomial-time reductions. Note that the exact solutions to NP-complete problems are believed to be intractable even for quantum computers.

<sup>13</sup> Figure 3 has been modified from [https://commons.wikimedia.org/wiki/File:TSP\\_Deutschland\\_3.png](https://commons.wikimedia.org/wiki/File:TSP_Deutschland_3.png).



**Figure 3.** TSP example: finding the shortest route visiting the largest cities in Germany.

broadly defined than planning. The lack of structure in general problem solving establishes a clear connection to (also unstructured) searching and optimization: in the setting of no additional information or structure, problem solving is the search for the solution to a precisely specified problem. While general problem solving can be, theoretically, achieved by a general search algorithm (which can still be subdivided into classes such as depth-first, breath-first, depth-limited search etc), more often there is structure to the problem, in which case *informed* search strategies—often called *heuristic search strategies*—will be more efficient (Russell and Norvig 2009). Human intelligence, to no small extent, relies on knowledge. We can accumulate knowledge, reason over it and use it to come to the best decisions, for instance in the context of problem solving and planning. An aspect of AI tries to formalize such logical reasoning, knowledge accumulation and knowledge representation, often relying on formal logic, most often first-order logic.

A particularly important class of problems central to AI, and related to knowledge acquisition, involves the capacity of the machine to learn through experience. This feature was emphasized already in the early days of AI, and the derived field of machine learning (ML) now stands as arguably the most successful aspect (or spin-off) of AI, which we will address in more detail.

### 1.2.1. Learning from data: machine learning.

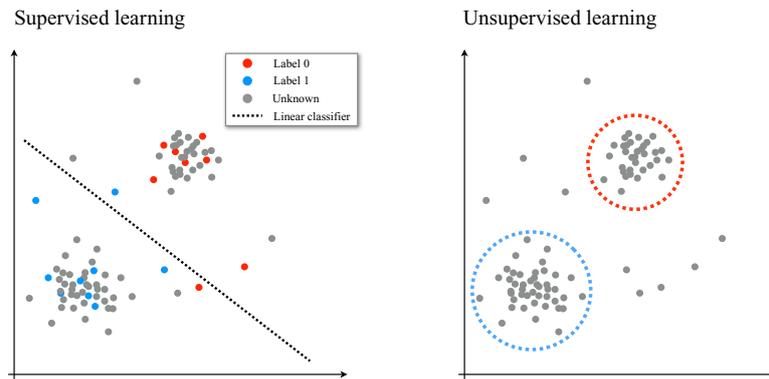
Stemming from the traditions of pattern recognition, such as recognizing handwritten text, and statistical learning theory (which places ML ideas in a rigorous mathematical framework), ML, broadly speaking, explores the construction of algorithms that can learn from and make predictions about data. Traditionally, ML deals with two main learning settings: supervised and unsupervised learning, which are closely related to data analysis and data mining-type tasks

(Shalev-Shwartz and Ben-David 2014), see figure 4. A broader perspective (Alpaydin 2010) on the field also includes reinforcement learning (Sutton and Barto 1998), which is closely related to learning as is realized by biological intelligent entities. We shall discuss reinforcement learning separately.

In broad terms, supervised learning deals with learning-by-example: given a certain number of labeled points (so-called *training set*)  $\{(x_i, y_i)\}_i$  where  $x_i$  denote data points, e.g.  $N$  – dimensional vectors, and  $y_i$  denote labels (e.g. binary variables, or real values), the task is to infer a ‘labeling rule’  $x_i \mapsto y_i$  which allows us to guess the labels of previously *unseen data*, that is, beyond the training set. Formally speaking, we deal with the task of inferring the conditional probability distribution  $P(Y = y|X = x)$  (more specifically, generating a labeling function which, perhaps probabilistically, assigns labels to points) based on a certain number of samples from the joint distribution  $P(X, Y)$ . For example, we could be inferring whether a particular DNA sequence belongs to an individual who is likely to develop diabetes. Such an inference can be based on the datasets of patients whose DNA sequences had been recorded, along with the information on whether they actually developed diabetes. In this example, the variable  $Y$  (diabetes status) is binary and the assignment of labels is not deterministic, as diabetes also depends on environmental factors. Another example could include two real variables, where  $x$  is the height from which an object is dropped and  $y$  the duration of the fall. In this example, both variables are real-valued and (in vacuum) the labeling relation will be essentially deterministic. In unsupervised learning, the algorithm is provided just with the data points without labels. Broadly speaking, the goal here is to identify the underlying distribution, or structure, and other informative features in the dataset. In other words, the task is to infer properties of the distribution  $P(X = x)$ , based on a certain number of samples, relative to a user-specified guideline or rule. Standard examples of unsupervised learning are clustering tasks, where data-points are supposed to be grouped in a manner which minimizes within-group mean-distance, while maximizing the distance between the groups. Note that the group membership can be thought of as a label, and so this also corresponds to a labeling task, but lacks ‘supervision’: examples of correct labelings. In basic examples of such tasks the number of expected clusters is given by the user, but this too can be automatically optimized.

Another class of unsupervised learning tasks includes feature extraction and dimensionality reduction, critical in combating the so-called curse of dimensionality. The curse of dimensionality refers to problems which stem from the fact that the raw representations of real-life data often occupy very high dimensional spaces. For instance, a standard resolution one-second video-clip at standard refresh frequency, capturing events which are extended in time, maps to a vector in  $\sim 10^8$  dimensional space<sup>14</sup>, even though the relevant information it carries (say a license-plate number of a speeding car filmed) may be significantly smaller. More generally, intuitively it is clear that, since geometric volume scales exponentially

<sup>14</sup> Each frame is approximately.  $10^6$  dimensional, as each pixel constitutes one dimension, multiplied by 30 frames required for the one-second clip.



**Figure 4.** Supervised (in this case, best linear classifier) and unsupervised learning (here clustering into two most likely groups and outliers) illustrated.

with the dimension of the space it is in, the number of points needed to capture (or *learn*) general features of an  $n$  – dimensional object will also scale exponentially. In other words, learning in high dimensional spaces is exponentially difficult. However, very often, the data-points which we can observe in reality (i.e. the effective support of the distribution  $P(X)$ ) lie in a sub-manifold of a substantially lower dimension. Hence, a means of dimensionality reduction from raw representation space (e.g. moving car clips) to the relevant feature space (e.g. license-plate numbers) is a necessity in any real-life scenario.

These approaches map the data-points to a space of significantly reduced dimension, while attempting to maintain the main features—the relevant information—of the structure of the data. Feature extraction can also be understood as a labeling process, where the points in the reduced space correspond to labels<sup>15</sup>. A typical example of a linear dimensionality example technique is, e.g., principal component analysis. In practice, such algorithms also constitute an important step in data pre-processing for other types of learning and analysis.

Furthermore, unsupervised learning also includes *generative models*. In this flavor of learning, the effective task is to specify a model, in this case a specification of a probability distribution, which best matches the observed data-points drawn from  $P(X)$ . Ideally, such a model well-approximates  $P(X)$  itself and can be used to generate new samples. As a comment on the nomenclature, it should be pointed out that the relationship between various flavors of learning is often a matter of perspective. For instance, generative models stand in contrast to *discriminative models*, which, as the name suggests, discriminate, i.e. they associate labels to points. From this perspective, most of supervised learning neatly fits in the discriminative paradigm, whereas unsupervised learning contains both discriminative (clustering) and generative aspects. On the other hand, both supervised discriminative and unsupervised generative models can be understood as data-fitting problems: in supervised settings, the goal is to identify the best classifying function with respect to how the labels in  $Y$  are correlated to the data-points in  $X$ , in the labeled training set (or the distribution  $P(X, Y)$ ). Thus, the problem is to

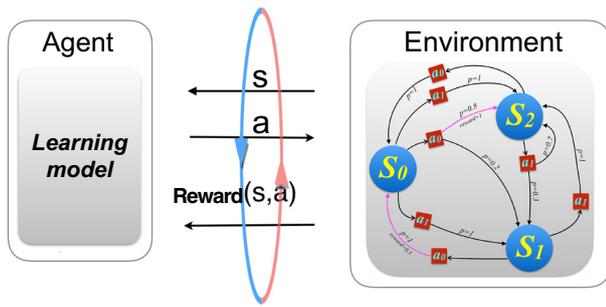
<sup>15</sup> Similarly, clustering can also be understood as feature extraction where the target feature is the specification of the cluster a given data-point belongs to.

identify the best approximation of the conditional distribution  $P(Y|X)$ . Generative models aim to capture the whole distribution (which may be multivariate or not). In both approaches, the actual task, in operational terms, boils down to identifying which element from the model set best fits the observed data. The model set is known as the set of hypotheses, and may consist of functions (e.g. neural networks, or hyperplanes in support vector machines) or parametrizations of distributions (e.g. Boltzmann machines, or more general graphical models), depending on whether the setting is discriminative or generative, respectively. For more details on the various models for ML, see section 2.1.

ML provides a plethora of methods which help us to understand data better, and in an automated fashion. As humanity is amassing data at an exponential rate (insideBIGDATA 2017) such methods may offer the only sustainable route to gaining new knowledge about the world we live in—as long as we can provide the computing power to match the growth of the datasets.

**1.2.2. Learning from interaction: reinforcement learning.** Reinforcement learning (RL) (Sutton and Barto 1998, Russell and Norvig 2009) is, traditionally, the third canonical category of ML. Partially due to the relatively recent prevalence of (un) supervised methods in the contexts of the pervasive data mining and big data analysis topics, many modern textbooks on ML focus on these methods. RL strategies have mostly remained reserved to the robotics and AI communities. Lately, however, the surge of interest in adaptive and autonomous devices, robotics and AI have increased the prominence of RL methods.

One recent celebrated result which relies on the extensive use of standard ML and RL techniques in conjunction is that of AlphaGo (Silver *et al* 2016), a learning system which mastered the game of Go, and achieved, arguably, superhuman performance, easily defeating the best human players. This result is notable for multiple reasons, including the fact that it illustrates the potential of learning machines over special-purpose solvers in the context of AI problems: while specialized devices which relied on programming over learning (such as Deep Blue) could surpass human performance in chess, they failed to do the same for the more complicated game of Go, which has a notably larger space of strategies. The learning



**Figure 5.** An agent interacts with an environment by exchanging percepts and actions. In RL rewards can be issued. Basic environments are formalized by Markov decision processes (inset in Environment). Environments are reminiscent of oracles, see figure 1, in that the agent only has access to the input–output relations. Further, figures of merit for learning often count the number of interaction steps, which is analogous to the concept of query complexity.

system AlphaGo achieved this many years ahead of typical predictions.

In subsequent recent works, it was further demonstrated that superhuman performance in Go can be achieved without relying on human-generated data, or supervised learning. In particular, the system relied on self-play and predominantly on RL methods (Silver *et al* 2017b). Such a more general RL approach lead to high flexibility and resulted in a system which simultaneously plays Go, chess and shogi (also known as Japanese chess) (Silver *et al* 2017a) at superhuman level. Such recent results have prompted a noticeable increase of interest in RL techniques, e.g. RL was listed as one of the ten breakthrough technologies of 2017 by MIT technology review (Review 2017).

The distinction between RL and other data-learning ML methods is particularly relevant from a quantum information perspective, which will be addressed in more detail in section 7.2.

RL constitutes a broad learning setting, formulated within the general agent–environment paradigm (AE paradigm) of AI (Russell and Norvig 2009). Here, we do not deal with a static database, but rather an interactive *task environment*. The learning agent (or, a learning algorithm) learns through the interaction with the task environment.

As an illustration, one can imagine a robot acting on its environment and perceiving it via its sensors—the percepts being, say, snapshots made by its visual system and actions being, say, movements of the robot—as depicted in figure 5. The AE formalism is, however, more general and abstract. It is also unrestrictive as it can also express supervised and unsupervised settings.

In RL, it is typically assumed that the goal of the process is manifest in a reward function, which, roughly speaking, rewards the agent whenever the agent’s behavior is *correct* (in which case we are dealing with *positive reinforcement*, but other variants of operant conditioning are also used<sup>16</sup>).

<sup>16</sup> More generally, we can distinguish four modes of such operant conditioning: positive reinforcement (reward when correct), negative reinforcement (removal of negative reward when correct), positive punishment (negative reward when incorrect) and negative punishment (removal of reward when incorrect).

This model of learning seems to cover pretty well how most biological agents (i.e. animals) learn: one can illustrate this through the process of training a dog to do a trick by giving it treats whenever it performs well. As mentioned earlier, RL is all about learning how to perform the ‘correct’ sequence of actions given the received percepts, which is an aspect of planning, in a setting which is fully on-line: the only way to learn about the environment is by interacting with it.

**1.2.3. Intermediary learning settings.** While supervised, unsupervised and RL constitute the three broad categories of learning, there are many variations and intermediary settings. For instance, semi-supervised learning interpolates between unsupervised and supervised settings, where the number of labeled instances is very small compared to the total available training set. Nonetheless, even a small number of labeled examples have been shown to improve the bare unsupervised performance (Chapelle *et al* 2010), or, from an opposite perspective, unlabeled data can help with classification when facing a small quantity of labeled examples. In active supervised learning, the learning algorithm can further query the human user, or supervisor, for the labels of particular points which would improve the algorithm’s performance. This setting can only be realized when it is operatively possible for the user to correctly label all the points and may yield advantages when this exact labeling process is expensive. Further, in supervised settings, one can consider so-called inductive learning algorithms which output a classifier function, based on the training data, which can be used to label all possible points. A classifier is simply a function which assigns labels to the points in the domain of the data. In contrast, in transductive learning (Chapelle *et al* 2010) settings, the points that need to be labeled later are known beforehand—in other words, the classifier function is only required to be defined on *a priori* known points. Next, a supervised algorithm can perform lazy learning, meaning that the whole labeled dataset is kept in memory in order to label unknown points (which can then be added), or eager learning, in which case, the (total) classifier function is output (and the training set is no longer explicitly required) (Alpaydin 2010). Typical examples of eager learning are linear classifiers, such as basic support vector machines, described in the next section, whereas lazy learning is exemplified by, e.g., nearest-neighbor methods<sup>17</sup>. Our last example, online learning (Alpaydin 2010), can be understood as either an extension of eager supervised learning, or a special case of RL. Online learning generalizes standard supervised learning, in the sense that the training data is provided sequentially to the learner and used to incrementally update the classifying function. In some variants, the algorithm is asked to classify each point and is given the correct response afterward, and the performance is based on the guesses. The match/mismatch of the guess and the actual label can also be understood as

<sup>17</sup> For example, in  $k$  – nearest neighbor classification, the training set is split into disjoint subsets specified by the shared labels. Given a new point which is to be classified, the algorithm identifies  $k$  nearest neighbor points from the dataset to the new point. The label of the new point is decided by the majority label of these neighbors. The labeling process thus needs to refer to the entire training set.

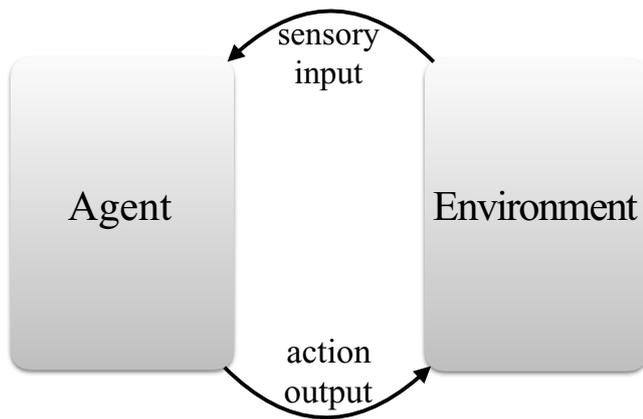


Figure 6. Basic AE paradigm.

a reward, in which case online learning becomes a restricted case of RL.

**1.2.4. Putting it all together: the agent–environment paradigm.** The aforementioned specialized learning scenarios can be phrased in a unifying language, which also enables us to discuss how specialized tasks fit in the objective of realizing true AI. In the modern take on AI (Russell and Norvig 2009), the central concept of the theory is that of an *agent*. An agent is an entity which is defined relative to its environment and which has the capacity to act, that is, *do something*.

In computer science terminology the requirements for something to be an agent (or for something to *act*) are minimal and essentially everything can be considered an agent—for instance, all non-trivial computer programs are also agents.

AI concerns itself with agents which do more—for instance they also perceive their environment, interact with it and learn from experience. AI is nowadays defined<sup>18</sup> as the field which is aimed at designing *intelligent agents* (Russell and Norvig 2009), which are autonomous, perceive their world using sensors, act on it using actuators and choose their activities so as to achieve certain goals—a property which is also called *rationality* in literature.

Agents only exist relative to an environment (more specifically a task environment) with which they interact, constituting the overall AE paradigm, illustrated in figure 6. While it is convenient to picture robots when thinking about agents, they can also be more abstract and virtual, as is the case with computer programs ‘living’ in the internet<sup>19</sup>. In this sense, any learning algorithm for any of the more specialized learning settings can also be viewed as a restricted learning agent, operating in a special type of environment, e.g. a supervised learning environment may be defined by a training phase,

<sup>18</sup> Over the course of its history, AI had many definitions, many of which invoke the notion of an agent, while some older, definitions talk about machines, or programs which ‘think’, ‘have minds’ and so on (Russell and Norvig 2009).

As clarified, the field of AI has fragmented, and many of the sub-fields deal with specific computational problems and the development of computational methodologies useful in AI related problems, for instance ML (i.e. its supervised and unsupervised variants). In such sub-fields with a more pragmatic computational perspective, the notion of agents is not used as often.

<sup>19</sup> The subtle topics of such virtual, yet embodied, agents is touched again later in section 7.1.

where the environment produces examples for the learning agent, followed by a testing phase, where the environment evaluates the agent, and finally the application phase, where the trained and verified model is actually used. The same also obviously holds for more interactive learning scenarios such as the reinforcement-driven mode of learning—RL, we briefly illustrated in section 1.2.2, is natively phrased in the AE paradigm. In other words, all ML models and settings can be phrased within the broad AE paradigm.

Although the field of AI is fragmented into research branches with focus on isolated, specific goals, the ultimate motivation of the field remains the same: the design of true, general AI, sometimes referred to as *artificial general intelligence (AGI)*<sup>20</sup>, that is, the design of a ‘truly intelligent’ agent (Russell and Norvig 2009).

The topic of what ingredients are needed to build AGI is difficult, and there is no consensus.

One perspective focuses on the behavioral aspects of agents. In literature, many features of intelligent behavior are captured by characterizing more specific types of agents: simple reflex agents, model-based reflex agents, goal-based agents, utility-based agents, etc. Each type captures an aspect of intelligent behavior, much like the fragments of the field of ML, understood as a subfield of AI, capture specific types of problems intelligent agents should handle. For our purposes, the most important, overarching aspect of intelligent agents is the capacity to learn<sup>21</sup> and we will emphasize learning agents in particular.

The AE paradigm is particularly well suited for such an operational perspective, as it abstracts from the internal structure of agents and focuses on behavior and input–output relations.

More precisely, the perspective on AI presented in this review is relatively simple: (a) AI pertains to agents which behave intelligently in their environments and (b) the central aspect of intelligent behavior is that of learning. While we, unsurprisingly, do not more precisely specify what intelligent behavior entails, this simple perspective on AI already has non-trivial consequences. The first is that intelligence can be ascertained from the interaction history between the agent and its environment alone. Such a viewpoint on AI is also closely related to behavior-based AI and the ideas behind the Turing test (Turing 1950); it is in line with an embodied viewpoint on AI (see *embodied AI* in section 1.2) and it has influenced certain approaches towards quantum AI, touched on in section 7.3. The second is that the development of better ML and other types of relevant algorithms does constitute genuine progress towards AI, conditioned only on the fact that such algorithms can be coherently combined into a whole agent. It is, however, important to note that actually achieving this integration may be far from trivial. In contrast to such strictly behavioral and operational points of view, an alternative approach towards whole agents (or complete intelligent agents) focuses on agent architectures and cognitive architectures (Russell and Norvig 2009). In this approach to AI the

<sup>20</sup> The field of AGI, under this label, emerged in the mid-2000s and the term is used to distinguish the objective of realizing intelligent agents from the research focusing on more specialized tasks, which are nowadays all labeled AI. AGI is also referred to as strong AI or, sometimes, full AI.

<sup>21</sup> A similar viewpoint, that essentially all AI problems/features map to a learning scenario, is also advocated in Hutter (2005).

emphasis is equally placed not only on intelligent behavior, but also on forming a theory about the structure of the (human) mind. One of the main goals of a cognitive architecture is to design a comprehensive computational model which encapsulates various results stemming from research in cognitive psychology. The aspects which are predominantly focused on understanding human cognition are, however, not central for our take on AI. We discuss this further in section 7.3.

### 1.3. Miscellanea.

#### Abbreviations and acronyms.

Acronym	Meaning	First occurrence
AE paradigm	Agent–environment paradigm	section 1.2.2
AGI	Artificial general intelligence	section 1.2.4
AI	Artificial intelligence	section 1.2
ANN	Artificial neural network	section 2.1.1
BED	Bayesian experimental design	section 4.1.3
BM	Boltzmann machine	section 2.1.1
BQP	Bounded-error quantum polynomial time	section 7.1
CAM	Content-addressable memory	section 2.1.1
CNN	Convolutional neural network	section 2.1.1
COLT	Computational learning theory	section 2.2
DME	Density matrix exponentiation	section 6.3.2
HN	Hopfield network	section 2.1.1
LSTM	Long short-term memory	section 2.1.1
MBQC	Measurement-based quantum computation	section 1.1
MDP	Markov decision process	section 2.3
ML	Machine learning	section 1.2
NN	Neural network	section 2.1.1
NP	Non-deterministic polynomial time	section 1.2
PAC learning	Probably approximately correct learning	section 2.2.1
PCA	Principal component analysis	section 6.3.2
POMDP	Partially observable Markov decision process	section 2.3
PS	Projective simulation	section 2.3
QC	Quantum computation	section 1.1
QIP	Quantum information processing	section 1.1
QUBO	Quadratic unconstrained binary optimization	section 6.3.1
RL	Reinforcement learning	section 1.2.2
rPS	Reflective PS	section 7.1
SVM	Support vector machine	section 2.1.2
VA	Variational autoencoders	section 2.1.1

*Notation.* Throughout this review paper, we have striven to use the notation specified in the reviewed works. To avoid notational chaos, however, we keep the notation consistent within subsections—this means that, within one subsection, we adhere to the notation used in the majority of works if inconsistencies arise.

## 2. Classical background

The main purpose of this section is to provide the background regarding classical ML and AI techniques and concepts which

are either addressed in quantum proposals we discuss in the following sections or are important for the proper positioning of the quantum proposal in the broader learning context. The concepts and models of this section include common models found in classical literature, but also certain more exotic models, which have been addressed in modern quantum ML literature. While this section contains most of the classical background needed to understand the basic ideas of the quantum ML literature, to tame the length of this section certain very specialized classical ML ideas are presented on-the-fly during the upcoming reviews.

We first provide the basics concepts related to common ML models, emphasizing neural networks in section 2.1.1 and support vector machines in section 2.1.2. Following this, in section 2.1.3, we also briefly describe a larger collection of algorithmic methods and ideas arising in the context of ML, including regression models,  $k$ -means/medians and decision trees, but also more general optimization and linear algebra methods which are now commonplace in ML. Beyond the more pragmatic aspects of model design for learning problems, in section 2.2 we provide the main ideas of the mathematical foundations of computational learning theory, which discuss learnability—i.e. the conditions under which learning is possible at all—computational learning theory and the theory of Vapnik and Chervonenkis, which rigorously investigates the bounds on learning efficiency for various supervised settings. Section 2.3 covers the basic concepts and methods of RL.

### 2.1. Methods of machine learning

**Executive summary:** Two particularly famous models in ML are artificial neural networks, inspired by biological brains, and support vector machines, arguably the best understood supervised learning model. Neural networks come in many flavors, all of which model parallel information processing of a network of simple computational units, neurons. Feed-forward networks (without loops) are typically used for supervised learning. Most of the popular deep learning approaches fit in this paradigm. Recurrent networks have loops; this allows, e.g., feeding information from outputs of a (sub)-network back to its own input. Examples include Hopfield networks, which can be used as content-addressable memories, and Boltzmann machines, typically used as generative models in unsupervised learning. These networks are related to Ising-type models, at zero, or finite temperatures, respectively—this sets the grounds for some of the proposals for quantization. Support vector machines classify data in a Euclidean space by identifying best separating hyperplanes, which allows for a comparatively simple theory. The linearity of this model is a feature making it amenable to quantum processing. The power of hyperplane classification can be improved by using kernels which, intuitively, map the data to higher dimensional spaces in a non-linear way. ML naturally goes beyond these two models and includes regression (data-fitting) methods and many other specialized algorithms.

Since the early days of the fields of AI and ML, there have been many proposals on how to achieve the flavors of learning we described above. In what follows we will describe two popular models for ML, specifically artificial neural networks and support vector machines. We highlight that many other models exist and, indeed, in many fields other learning methods (e.g. regression methods) are more commonly used. A selection of such other models is briefly mentioned thereafter, along with examples of techniques which overlap with ML topics in a broader sense, such as matrix decomposition techniques, and which can be used for, e.g., unsupervised learning.

Our choice of emphasis is, in part, again motivated by later quantum approaches and by features of the models which are particularly well-suited for cross-overs with QC.

**2.1.1. Artificial neural networks for supervised and unsupervised learning.** Artificial neural networks (ANNs, or just NNs) are a biologically inspired approach to tackling learning problems. Originating in 1943 (McCulloch and Pitts 1943), the basic component of NNs is the artificial neuron (AN), which is, abstractly speaking, a real-valued function  $AN : \mathbb{R}^k \rightarrow \mathbb{R}$  parametrized by a vector of real, non-negative weights  $(w_i)_i = \mathbf{w} \in \mathbb{R}^k$  and the activation function  $\phi : \mathbb{R} \rightarrow \mathbb{R}$ , given by

$$AN(\mathbf{x}) = \phi \left( \sum_i x_i w_i \right), \text{ with } \mathbf{x} = (x_i)_i \in \mathbb{R}^k. \quad (1)$$

For the particular choice when the activation function is the threshold function, given by  $\phi_\theta(x) = 1$  if  $x > \theta \in \mathbb{R}^+$  and  $\phi_\theta(x) = 0$  otherwise, the AN is called a perceptron (Rosenblatt 1957), and has been studied extensively. Already such simple perceptrons perform classification into subspaces specified by the hyperplane with the normal vector  $\mathbf{w}$  and offset  $\theta$  (see *support vector machines* later in this section).

Note that in ML terminology a distinction should be made between *artificial neurons* (ANs) and *perceptrons*—perceptrons are special cases of ANs, with a fixed activation function, the step function, and a specified update or training rule. ANs in modern times use various activation functions (often the differentiable sigmoid functions) and can use different learning rules. For our purposes, this distinction will not matter. The training of such a classifier or AN for supervised learning purposes consists in optimizing the parameters  $\mathbf{w}$  and  $\theta$  so as to correctly label the training set—there are various figures of merit particular approaches care about and various algorithms that perform such an optimization, which are not relevant at this point. By combining ANs in a network we obtain NNs (if ANs are perceptrons, we usually talk about multi-layered perceptrons). While single perceptrons, or single-layered perceptrons, can realize only linear classification, a three-layered network already suffices to approximate any continuous real-valued function (with precision depending on the number of neurons in the inner, so-called hidden, layer). Cybenko (1989) was the first to prove this for sigmoid activation functions, whereas Hornik generalized this to show that the same holds for all non-constant, monotonically increasing and bounded activation functions (Hornik 1991) soon thereafter. This

shows that if sufficiently many neurons are available, a three-layered ANN can be trained to learn any dataset, in principle<sup>22</sup>. Although this result seems very positive, it comes with the price of a large model complexity, which we discuss in section 2.2.2<sup>23</sup>.

In recent years, it has become apparent that using multiple, sequential, hidden feed-forward layers (instead of one large layer), i.e. deep NNs, may have additional benefits. In particular, convolutional NNs (CNNs) have achieved stellar successes, especially in the fields of vision and pattern recognition. CNNs are, in essence, deep NNs of a particular structure (e.g. the connections between neighboring layers maintain a notion of locality), which is inspired by the biological visual cortex of certain animals (see e.g. Rawat and Wang (2017) and references therein for more information). CNNs have a reduced number of parameters (Poggio *et al* 2017) compared to unrestricted deep NNs. Furthermore, the sequential nature of processing of information from layer to layer can be understood as a feature abstraction mechanism (the layers process the data sequentially, highlighting relevant features at each step) and the analyses of the outputs of the trained layers of the CNN can shed light on the relevant features of the task at hand.

While deep NNs are extremely successful in practice, they also suffer from short-comings. Perhaps the central issue of deep networks follows from the complex nature of the functions they can realize and the convoluted relationship between the network parameters and the realized functions. The high expressivity—their capacity to represent many involved functions—is responsible for their outstanding performance, but also limits what can theoretically be said about generalization performance. Indeed, our lack of understanding of why such networks generalize well is a matter of ongoing debate (Zhang *et al* 2017a). Further, the complex structure of functions realized by deep networks (even CNNs) dramatically limits the interpretability of the model, which is, intuitively, the possibility for high level explanations of the model’s outputs. The relevance of interpretability of learning models is becoming more important as we delegate ever more important decisions to automated systems. For this reason, NNs are often not considered as the go-to approaches in critical systems, where failures may have catastrophic consequences (e.g. automated cars), or when it is simply required to produce a qualified explanation for any given decision. However, the question of what interpretability should entail, and how it can be enforced, is still a matter of cutting-edge research (Lipton 2016, Zhang *et al* 2017b).

One of the main practical disadvantages of such deep networks is the computational cost and computational instabilities in training (see *the vanishing gradient problem* (Hochreiter *et al* 2001)) and also the size of the dataset, which has to be large (Larochelle *et al* 2009). With modern technology and

<sup>22</sup> More specifically, there exists a set of weights doing the job, even though standard training algorithms may fail to converge to that point.

<sup>23</sup> Roughly speaking, models with high model complexity are more likely to ‘overfit’ and it is more difficult to provide guarantees they will generalize well, i.e. perform well beyond the training set.

datasets, both obstacles are becoming less prohibitive, which has led to a minor revolution in the field of ML.

Not all ANNs are feed-forward: recurrent NNs allow for the backpropagation of signals. Particular examples of such networks are so-called Hopfield networks (HNs), Boltzmann machines (BMs) and long short-term memory (LSTM) networks, which are often used for different purposes than feed-forward networks.

In HNs, we deal with one layer, where the outputs of all the neurons serve as inputs to the same layer. The network is initialized by assigning binary values (traditionally  $-1$  and  $1$  are used, for reasons of convenience) to the neurons (more precisely, some neurons are set to fire, and some not), which are then processed by the network, leading to a new configuration. This update can be synchronous (the output values are ‘frozen’ and all the second-round values are computed simultaneously) or asynchronous (the update is done one neuron at a time in a random order). The connections in the network are represented by a matrix of weights  $(w_{ij})_{ij}$ , specifying the connection strength between the  $i$ th and the  $j$ th neuron. The neurons are perceptrons, with a threshold activation function, given by the local threshold vector  $(\theta_i)_i$ . Such a dynamical system, under a few mild assumptions (Hopfield 1982), converges to a configuration (i.e. a bit-string) which (locally) minimizes the energy functional

$$E(\mathbf{s}) = -\frac{1}{2} \sum_{ij} w_{ij} s_i s_j + \sum_i \theta_i s_i, \quad (2)$$

with  $\mathbf{s} = (s_i)_i$ ,  $s_i \in \{-1, 1\}$ , that is, the Ising model. In general, this model has many local minima, which depend on the weights  $w_{ij}$  and the thresholds, which are often set to zero. Hopfield provided a simple algorithm (called Hebbian learning, after D. Hebb for historical reasons (Hopfield 1982)), which enables one to ‘program’ the minima—in other words, given a set of bit-strings  $S$  (more precisely, strings of signs  $+1/-1$ ), one can find the matrix  $w_{ij}$  such that exactly those strings  $S$  are local minima of the resulting functional  $E$ . Such programmed minima are then called *stored patterns*. Furthermore, Hopfield’s algorithm achieved this in a manner which is local (the weights  $w_{ij}$  depend only on the  $i$ th and  $j$ th bits of the targeted strings, allowing parallelizability), incremental (one can modify the matrix  $w_{ij}$  to add a new string without having to keep the old strings in memory) and immediate. Immediateness means that the computation of the weight matrix is not a limiting, but finite process. Violating incrementality would lead to a lazy algorithm (see section 1.2.3), which can be sub-optimal in terms of memory requirements, but often also computational complexity<sup>24</sup>. It was shown that the minima of such a trained network are also attractive fixed-points, with a finite basin of attraction. This means that if a trained network is fed a new string and let run, it will (eventually) converge to a pattern which is closest to it (the distance measure that is used depends on the learning rule, but typically it is the Hamming distance, i.e. the number of entries where the strings disagree). Such a system then forms an

associative memory, also called a content-addressable memory (CAM). CAMs can be used for supervised learning (the ‘labels’ are the stored patterns) and, conversely, supervised learning machinery can be used for CAM<sup>25</sup>. An important feature of HNs is their capacity: how many distinct patterns it can store<sup>26</sup>. For the Hebbian update rule this number scales as  $O(n/\log(n))$ , where  $n$  is the number of neurons, which Storkey (1997) improved to  $O(n/\sqrt{\log(n)})$ . In the meantime, more efficient learning algorithms have been invented (Hillar and Tran 2014). Aside from applications as CAMs, due to the representation in terms of the energy functional in equation (2) and the fact that the running of HNs minimize it, early on they were also considered for tasks of optimization (Hopfield and Tank 1985). The operative isomorphism between HNs and the Ising model, technically, holds only in the case of a zero-temperature system. BMs generalize this. Here, the value of the  $i$ th neuron is set to  $-1$  or  $1$  (called ‘off’ and ‘on’ in literature, respectively) with probability

$$p(i = -1) = (1 + \exp(-\beta \Delta E_i))^{-1}, \quad \text{with } \Delta E_i = \sum_j w_{ij} s_j + \theta_i, \quad (3)$$

where  $\Delta E_i$  is the energy difference between the configurations with the  $i$ th neuron being on or off, assuming the connections  $\mathbf{w}$  are symmetric and  $\beta$  is the inverse temperature of the system. In the limit of infinite running time, the network’s configuration is given by the (input-state invariant) Boltzmann distribution over the configurations, which depends on the weights  $\mathbf{w}$ , local thresholds (weights)  $\theta$  and the temperature. BMs are typically used in a generative fashion, to model and sample from (conditional) probability distributions. In the simplest variant, the training of the network attempts to ensure that the limiting distribution of the network matches the observed frequencies in the dataset. This is achieved by the tuning of the parameters  $\mathbf{w}$  and  $\theta$ . The structure of the network dictates how complicated a distribution can be represented. To capture more complicated distributions, over, say,  $k$ -dimensional data, the BMs have  $N > k$  neurons.  $k$  of them will be denoted as visible units and the remainder are called hidden units, and they capture latent, not directly observable, variables of the system which generated the dataset and which we are in fact modeling. Training such networks consists in a gradient ascent of the log-likelihood of observing the training data in the parameter space. While this seems conceptually simple, it is computationally intractable, in part because it requires accurate estimates of probabilities of equilibrium distributions, which are hard to obtain. In practice, this is mitigated by using restricted BMs, where the hidden and visible units form the partition of a bipartite graph (so only connections between hidden and visible units exist). (Restricted)

<sup>25</sup> For this, one simply needs to add a look-up table connecting labels to fixed patterns.

<sup>26</sup> Reliable storage entails that previously stored patterns will also be recovered without change (i.e. they are energetic local minima of equation (2)), but also that there is a basin of attraction—a ball around the stored patterns with respect to a distance measure (most commonly the Hamming distance) for which the dynamical process of the network converges to the stored pattern. An issue with capacities is the occurrence of spurious patterns: local minima with a non-trivial basin of attraction which were not stored.

<sup>24</sup> The lazy algorithm may have to process all the patterns/data-points, the number of which may be large and/or growing.

BMs have a large spectrum of applications including providing generative models, producing new samples from the estimated distribution; as classifiers, via conditioned generation; as feature extractors, a form of unsupervised clustering; and as building blocks of deep architectures (Larochelle *et al* 2009). BMs can be ‘stacked’ in multiple layers, in analogy to deep NNs to form deep BMs<sup>27</sup>.

The utility of BM-based architectures is mostly limited by the cost of training—for instance, the cost of obtaining equilibrium Gibbs distributions, or by the errors stemming from heuristic training methods such as contrastive divergence (Larochelle *et al* 2009, Bengio and Delalleau 2009, Wiebe *et al* 2014c).

The back-propagating nature of information flow in recurrent NNs can also be used as a means to affect the action of the network on the current data input, depending on the previous input(s). This memory effect was shown to be useful in the processing of time-series, such as natural language processing. For a review of recurrent NNs applied to the learning of sequential data we refer the reader to Lipton *et al* (2015). Arguably the most successful models in this domain comprise LSTM networks (Hochreiter and Schmidhuber 1997). The simplest recurrent networks already have short-term memory capacities: e.g. an ‘identity’ neuron with a back-propagating connection can forward information from a previous step to the subsequent step, as part of input to another neuron (or itself). However, LSTMs are constructed so that the network can be trained to preserve signals (i.e. data points at a given instance in time) for arbitrarily long intervals. Similar results have recently also been obtained by a slightly simpler model of gated recurrent units (Chung *et al* 2014). LSTM and closely related models are currently the go-to methods behind many cutting edge technologies, specifically those pertaining to speech recognition (Wired 2017) and machine translation (Wu *et al* 2016).

Modern advances in NN approaches to ML also involve various combined architectures which are targeted to various shortcomings of the more standard methods. We will just briefly mention a few such approaches relevant for this review. In domain-adversarial NN training (Ganin *et al* 2016), one performs supervised learning, while ensuring that the resulting classifier will also perform well on similar domains. This is achieved by using two datasets, called the source domain (a labeled set) and the target domain (which may be unlabeled), and by training a feature extracting NN such that two competing properties are maintained: from the extracted features, distinction between the domains should be impossible, while correct prediction of the data labelings from the source set should be possible. This is achieved by using two NNs (a domain predictor, which should ultimately fail; and a label classifier, which should not fail). Conceptually related pairings of unsupervised and supervised methods also occur in generative scenarios. In the highly successful *generative adversarial nets* (Goodfellow

*et al* 2014) approach, a generative model is trained to generate new samples based on an input dataset. The generative model is put in opposition to a discriminative model (often a convolutional network), which is trained to distinguish between true examples from the input dataset and the outputs of the generative model. Finally, we mention variational autoencoders (VAs) as one of the most successful approaches for generative models (Kingma and Welling 2013, Doersch 2016). In VAs, very roughly, one combines a compressing encoder (e.g. a feed-forward NN with significantly fewer output than input neurons) with a decoding network, which aims to recover the whole input. Such a structure is often called an autoencoder. In VAs, it is additionally enforced that the distribution output by the encoder (given the dataset), approximately follows the standard normal distribution. If the network is trained to decode correctly while the distribution of the ‘bottleneck layer’ (connecting the encoder and decoder) approximately follows a known easy-to-generate distribution  $P$ , this offers a direct means of using the VA as a generative model—one simply samples from  $P$  and applies the decoder. The performance of VAs is often compared to that of (restricted) BMs. Often mentioned advantages of the VA approach include the fact that they are applicable to models beyond NNs and that the training may be more efficient. For instance, if a VA is built from feed-forward NNs, it can be trained essentially using just the standard and efficient NN machinery.

Novel NN-based architectures have been emerging at an accelerated rate in recent years, in part motivated by the available computing power which allows such complex models to be trained. However, while such more complex models may offer record-setting performances, they certainly lead us further away from interpretable scenarios, where we can understand how and why the system works, in a clean formal way<sup>28</sup>. One model which does allow such a clean treatment is the support vector machine, which we describe next.

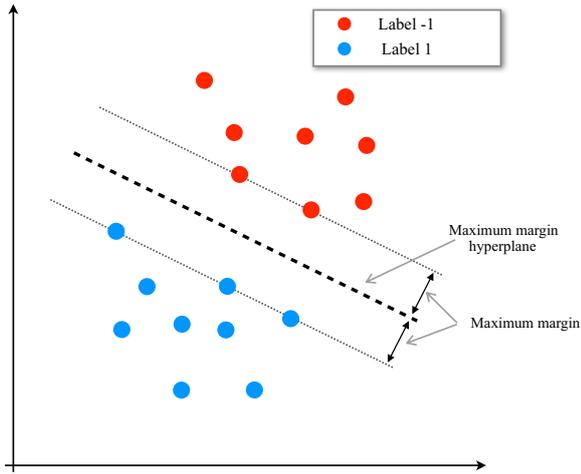
**2.1.2. Support vector machines for supervised learning.** Support vector machines (SVMs) form a family of perhaps best understood approaches for solving classification problems. The basic idea behind SVMs is that a natural way to classify points based on a dataset  $\{\mathbf{x}_i, y_i\}_i$ , for binary labels  $y_i \in \{-1, 1\}$ , is to generate a hyperplane separating the negative instances from the positive ones. Such observations are not new and, indeed, perceptrons, briefly discussed in the previous section, perform the same function.

Such a hyperplane can then be used to classify all points. Naturally, not all sets of points allow this (those that do are called linearly separable), but SVMs are further generalized to deal with sets which are not linearly separable using so-called kernels. Kernels, effectively, realize non-linear mappings of the original dataset to higher dimensions where they may become separable, depending on a few technical conditions<sup>29</sup>,

<sup>27</sup> The graphical representation of deep BMs is similar to that of another well-studied generative model called *deep belief networks*. These networks originated from the study of deep BMs. Technically, they are directed graphical models, which capture dependencies (denoted by edge directionality) between various variables they represent. The effective directedness of the edges stems from a particular training process of the deep BM structure to form this hybrid model. For a comparison and more details on these two related models, we refer the reader to Salakhutdinov and Hinton (2009).

<sup>28</sup> Furthermore, to what extent such very complex models overfit—that is, perform near-perfectly on seemingly arbitrary training sets, but then fail to generalize beyond the training set—remains a vital and mostly unresolved question in the field.

<sup>29</sup> Indeed, this can be supported by hard theory; see Cover’s theorem (Cover 1965).



**Figure 7.** Basic example of an SVM, trained on a linearly separable dataset.

and by allowing a certain degree of misclassification, which leads to so-called ‘soft-margin’ SVMs.

Even in the case where the dataset is linearly separable, there will still be many hyperplanes doing the job. This leads to various variants of SVMs, but the basic variant identifies a hyperplane which: (a) correctly splits the training points and (b) maximizes the so-called margin: the distance of the hyperplane to the nearest point (see figure 7).

The distance of choice is most often the geometric Euclidean distance, which leads to so-called *maximum margin classifiers*. In high-dimensional spaces, in general the maximization of the margin ends in a situation where there are multiple +1 and -1 instances of training data points which are equally far from the hyperplane. These points are called support vectors. The finding of a maximum margin classifier corresponds to finding a normal vector  $\mathbf{w}$  and offset  $b$  of the separating hyperplane, which corresponds to the optimization problem

$$\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (4)$$

$$\text{such that } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \quad (5)$$

The formulation above is actually derived from the basic problem by noting that we may arbitrarily and simultaneously scale the pair  $(\mathbf{w}, b)$  without changing the hyperplane. Therefore, we may always choose a scaling such that the realized margin is 1, in which case, the margin corresponds to  $\|\mathbf{w}\|^{-1}$ , which simply maps a maximization problem to a minimization problem as above. The square ensures the problem is stated as a standard quadratic programming problem. This problem is often expressed in its Lagrange dual form, which reduces to

$$(\alpha_1^*, \dots, \alpha_N^*) = \operatorname{argmin}_{\alpha_1, \dots, \alpha_N} \left( \sum_i \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \right) \quad (6)$$

$$\text{such that } \alpha_i \geq 0 \text{ and } \sum_i \alpha_i y_i = 0, \quad (7)$$

where the solution of the original problem is given by

$$\mathbf{w}^* = \sum_i y_i \alpha_i \mathbf{x}_i. \quad (8)$$

In other words, we have expressed  $\mathbf{w}^*$  in the basis of the data vectors, and the data vectors  $\mathbf{x}_i$  for which the corresponding coefficient  $\alpha_i$  is non-zero are precisely the support vectors. The offset  $b^*$  is easily computed having access to one support vector of, say, an instance +1, denoted  $\mathbf{x}^+$ , by solving  $\mathbf{w}^* \cdot \mathbf{x}^+ + b^* = 1$ .

The class of a new point  $\mathbf{z}$  can also be computed directly using the support vectors via the following expression

$$\mathbf{z} \mapsto \operatorname{sign} \left( \sum_i y_i \alpha_i \mathbf{x}_i \cdot \mathbf{z} + b^* \right). \quad (9)$$

The dual representation of the optimization problem is convenient when dealing with kernels. As mentioned, a way of dealing with data which is not linearly separable is to first map all the points into a higher-dimensional space via a non-linear function  $\phi: \mathbb{R}^m \rightarrow \mathbb{R}^n$ , where  $m < n$  is the dimensionality of the data points. As we can see, in the dual formulation, the data points only appear in terms of inner products  $\mathbf{x}_i \cdot \mathbf{x}_j$ . This leads to the notion of the *kernel function*  $k$  which, intuitively, measures the similarity of the points *in the larger space*, typically defined by  $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ . In other words, to train the SVM according to a non-trivial kernel  $k$ , induced by the non-linear mapping  $\phi$ , the optimization line in equation (6) will be replaced with  $\operatorname{argmin}_{\alpha_1, \dots, \alpha_N} \left( \sum_i \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \right)$ . The offset is computed analogously, using just one application of  $\phi$ . The evaluation of a new point is given in the same way with  $\mathbf{z} \mapsto \operatorname{sign} \left( \sum_i y_i \alpha_i k(\mathbf{x}_i, \mathbf{z}) + b^* \right)$ . In other words, the data points need not be explicitly mapped via  $\phi$ , as long as the map-inducing inner product  $k(\cdot, \cdot)$  can be computed more effectively. The choice of the kernel is critical in the performance of the classifier, and the finding of good kernels is non-trivial and often solved by trial and error.

While increasing the dimension of the extended space (co-domain of  $\phi$ ) may make data points more linearly separable (i.e. fewer mismatches for the optimal classifier), in practice they will not be fully separable (and furthermore, increasing the kernel dimension comes with a cost which we elaborate on later). To resolve this, SVMs allow for misclassification, with various options for measuring the ‘amount’ of misclassification, inducing a penalty function. A typical approach to this is to introduce so-called ‘slack variables’  $\xi_i \geq 0$  to the original optimization task, so:

$$\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}, b} \left( \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i \right) \quad (10)$$

$$\text{such that } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i. \quad (11)$$

If the value  $\xi_i$  of the optimal solution is between 0 and 1, the point  $i$  is correctly classified, but is within the margin, and  $\xi_i > 1$  denotes a misclassification. The (hyper)parameter  $C$  controls the relative importance we place on minimizing the

margin norm, versus the importance we place on misclassification. Interestingly, the dual formulation of the above problem is near identical to the hard-margin setting discussed thus far, with the small difference that the parameters  $\alpha_i$  are now additionally constrained with  $\alpha_i \leq C$  in equation (7). SVMs, as described above, have been extensively studied from the perspective of computational learning theory and have been connected to other learning models. In particular, their generalization performance, which, roughly speaking, characterizes how well a trained model<sup>30</sup> will perform *beyond the training set* can be analyzed. This is the most important feature of a classifying algorithm. We will briefly discuss generalization performance in section 2.2.2. We end this short review of SVMs by considering a non-standard variant, which is interesting for our purposes as it has been beneficially quantized. SVMs as described are trained by finding the maximal margin hyperplane. Another model, called least-squares SVM (LS-SVM), takes a regression (i.e. data-fitting) approach to the problem, and finds a hyperplane which, essentially, minimizes the least square distance of the vector of labels, and the vector of distances from the hyperplane, where the  $i$ th entry of the vector is given with  $(\mathbf{w} \cdot \mathbf{x}_i + b)$ . This is effected by a small modification of the soft-margin formulation:

$$\mathbf{w}_{\text{LS}}^* = \operatorname{argmin}_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i^2 \quad (12)$$

$$\text{such that } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1 - \xi_i, \quad (13)$$

where the only two differences are that the constraints are now equalities and the slack variables are squared in the optimization expression. This seemingly innocuous change causes differences in performance, but also in the training. The dual formulation of the latter optimization problem reduces to a linear system of equations:

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1}_N & \Omega + \gamma^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix}, \quad (14)$$

where  $\mathbf{1}$  is an ‘all ones’ vector,  $Y$  is the vector of labels  $y_i$ ,  $b$  is the offset and  $\gamma$  is a parameter depending on  $C$ .  $\boldsymbol{\alpha}$  is the vector of the Lagrange multipliers yielding the solution. This vector again stems from the dual problem which we omitted due to space constraints, and which can be found in Suykens and Vandewalle (1999). Finally,  $\Omega$  is the matrix collecting the (mapped) ‘inner products’ of the training vectors so  $\Omega_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ , where  $k$  is a kernel function, in the simplest case, just the inner product. The training of LS-SVMs is thus simpler (and particularly convenient from a quantum algorithms perspective), but the theoretical understanding of the model, and its relationship to the well-understood SVMs, is still a matter of study, with few known results (see e.g. Ye and Xiong (2007)).

<sup>30</sup> In ML, the term model is often overloaded. Most often it refers to a classification system which has been trained on a dataset, and in that sense it ‘models’ the actual labeling function. Often, however, it will also refer to a class of learning algorithms (e.g. the SVM learning model).

**2.1.3. Other models.** While NNs and SVMs constitute two popular approaches for ML tasks (in particular, supervised learning), many other models exist, suitable for a variety of ML problems. Here we very briefly list and describe some such models which have also appeared in the context of quantum ML. While classification typically assigns discrete labels to points, in the case where the labeling function has a continuous domain (say the segment  $[0, 1]$ ) we are dealing with function approximation tasks, often dealt with by using regression techniques. Typical examples here include linear regression, which approximates the relationship of points and labels with a linear function, most often minimizing the least-squares error. More broadly, such techniques are closely related to data-fitting, that is, fitting the parameters of a parametrized function so as to best fit observed (training) data. The *k-nearest neighbor* algorithm is an intuitive classification algorithm which, given a new point, considers the  $k$  nearest training points (with respect to a metric of choice) and assigns the label by a majority vote (if used for classification), or by averaging (in the case of regression, i.e. continuous label values). The mutually related *k-means* and *k-medians* algorithms are typically used for clustering: the  $k$  specifies the number of clusters, and the algorithm defines them in a manner which minimizes the within-cluster distance to the mean (or median) point.

Another method for classification and regression optimizes decision trees, where each dimension or entry (or more generally a feature<sup>31</sup>) of the new data point influences a move on a *decision tree*. The depth of the tree is the length of the vector (or number of features) and the degree of each node depends on the possible number of distinct features/levels per entry<sup>32</sup>. The vertices of the tree specify an arbitrary feature of interest, which can influence the classification result, but most often they consider the overlaps with geometrical regions of the data point space. Decision trees are in principle maximally expressive (can represent any labeling function), but very difficult to train without constraints.

More generally, classification tasks can be treated as the problem of finding a hypothesis  $h : \text{Data} \rightarrow \text{Labels}$  (in ML, the term hypothesis is essentially synonymous with the term classifier, also called a learner) which is from some family  $H$ , which minimizes error (or *loss*) under some loss function. For instance, the hypotheses realized by SVMs are given by the hyperplanes (in the kernel space), and in neural nets they are parametrized by the parameters of the nets: geometry, thresholds, activation functions, etc. In addition to loss terms, the

<sup>31</sup> Features, however, have a more generic meaning in the context of ML. A data vector is a vector of features, where what a feature is depends on the context. For instance, features can be simply values at particular positions, or more global properties: e.g. a feature of data vectors depicting an image may be ‘contains a circle’, and all vectors corresponding to pictures with circles have it. Even more generically, features pertain to observable properties of the objects the data points represent (‘observable’ here simply means that the property can be manifested in the data vector).

<sup>32</sup> For instance, we can classify humans, parrots, bats and turtles by binary features *can\_fly* and *is\_mammal*. E.g. choosing the root *can\_fly* leads to the branch *can\_fly = no* with two leaves decided by *is\_mammal = yes* pinpointing the human, whereas *is\_mammal = no* would specify the turtle. Parrots and bats would be distinguished by the same feature in the *can\_fly = yes* subtree.

minimization of which is called empirical risk minimization, ML applications benefit from adding an additional component to the objective function: the regularization term, the purpose of which is to penalize complex functions which could otherwise lead to poor generalization performance; see section 2.2.2. The choices of loss functions, regularization terms and classes of hypotheses lead to different particular models, and training corresponds to optimization problems given by the choice of the loss function and the hypothesis (function) family. Furthermore, it has been shown that essentially any learning algorithm which requires only convex optimization for training leads to poor performance under noise. Thus non-convex optimization is necessary for optimal learning (see e.g. Long and Servedio (2010) and Manwani and Sastry (2011)).

An important class of meta-algorithms for classification problems are boosting algorithms. The basic idea behind boosting algorithms is the highly non-trivial observation, first proven via the seminal AdaBoost algorithm (Freund and Schapire 1997), that multiple weak classifiers, which perform better than random on distinct parts of the input space, can be combined into an overall better classifier. More precisely, given a set of (weak) hypotheses/classifiers  $\{h_j\}$ ,  $h_j : \mathbb{R}^n \rightarrow \{-1, 1\}$ , under certain technical conditions, there exists a set of weights  $\{w_i\}$ ,  $w_i \in \mathbb{R}$ , such that the composite classifier of the form  $hc_w(\mathbf{x}) = \text{sign}(\sum_i w_i h_i(\mathbf{x}))$  performs better. Interestingly, a single (weak) learning model can be used to generate the weak hypotheses needed for the construction of a better composite classifier—one which, in principle, can achieve arbitrarily high success probabilities, i.e. a *strong learner*. The first step of this process is achieved by altering the frequencies at which the training labeled data points appear—one can effectively alter the distributions over the data (in a black-box setting, these can be obtained by, e.g., rejection sampling methods). The training of one and the same model on such differentially distributed datasets can generate distinct weak learners, which emphasize distinct parts of the input space. Once such distinct hypotheses are generated, optimization of the weight  $w_i$  of the composite model is performed. In other words, weak learning models can be boosted<sup>33</sup>.

Aside from the broad classes of approaches to solve various ML tasks, ML is also often conflated with specific computational tools which are used to solve them. A prominent example of this is the development of algorithms for optimization problems, especially those arising in the training of standard learning models. This includes, e.g., particle swarm optimization, genetic and evolutionary algorithms and even variants of stochastic gradient descent.

ML also relies on other methods including linear algebra tools, e.g. matrix decomposition methods, such as singular value decomposition, QR-, LU- and other decompositions, derived methods such as principal component analysis and various techniques from the field of signal analysis (Fourier, Wavelet, Cosine and other transforms). The latter set of techniques serves to reduce the effective dimension of the dataset

and helps combat the curse of dimensionality. The optimization, linear algebra and signal processing techniques and their interplay with quantum information is an independent body of research with enough material to deserve a separate review, and we will only reflect on these methods when needed.

## 2.2. Mathematical theories of supervised and inductive learning

**Executive summary:** Aside from proposing learning models, such as NNs or SVMs, learning theory also provides formal tools to identify the limits of learnability. No Free Lunch theorems provide sobering arguments that naïve notions of ‘optimal’ learning models cannot be obtained, and that all learning must rely on some prior assumptions. Computational learning theory relies on ideas from computational complexity theory to formalize many settings of supervised learning, such as the task of approximating or identifying an unknown (boolean) function—a concept—which is just the binary labeling function. The main question of the theory is the quantification of the number of invocations of the black box—i.e. of the function (or of the oracle providing examples of the function’s values on selected inputs)—needed to reliably approximate the (partially) unknown concept to the desired accuracy. In other words, computational learning theory considers the sample complexity bounds for various learning settings, specifying the concept families and type of access. The theory of Vapnik and Chervonenkis, or simply VC theory, stems from the tradition of statistical learning. One of the key goals of the theory is to provide theoretical guarantees on generalization performance. This is what is asked for in the following question: given a learning machine trained on a dataset of size  $N$ , stemming from some process, with a measured *empirical risk* (error on the training set) of some value  $R$ , what can be said about its future performance on other data points which may stem from the same process? One of the key results of VC theory is that this can be answered with the help of a third parameter—the model complexity of the learning machine. Model complexity, intuitively, captures how complicated the functions are which the learner can learn: the more complicated the model, the higher the chance of ‘overfitting’ and, consequently, the weaker the guarantees on performance beyond the training set. Good learning models can control their model complexity, leading to a learning principle of structural risk minimization. The art of ML is a juggling act, balancing sample complexity, model complexity and the computational complexity of the learning algorithm.<sup>34</sup>

<sup>33</sup> It should be mentioned that the above description only serves to illustrate the intuition behind boosting ideas. In practice, various boosting methods have distinct steps, e.g. they may perform the required optimizations in differing orders, using training phases in parallel, etc, which is beyond the needs of this review.

<sup>34</sup> While the dichotomies between sample complexity and computational complexity are often considered in the literature, the authors first learned of the trichotomic setting, including model complexity, from Wittek (2014b). Examples of such balancing and its failures can be observed in sections 5.1.2 and 6.1.1.

Although the modern increased interest in ML and AI is mostly due to applications, aspects of ML and AI do have strong theoretical backgrounds. Here we focus on such foundational results, which clarify what learning is and which investigate the questions of what learning limits are. We will very briefly sketch some of the basic ideas.

The first collection of results, called No Free Lunch theorems, place seemingly pessimistic bounds on the conditions under which learning is at all possible (Wolpert 1996). No Free Lunch theorems are, essentially, a mathematical formalization of Hume’s famous problem of induction (Hume 1739, Vickers 2016), which deals with the justification of inductive reasoning. One example of inductive reasoning occurs during generalization. Hume points out that, without *a priori* assumptions, concluding any property concerning a class of objects based on any number of observations<sup>35</sup> is not justified.

In a similar vein, learning based on experience cannot be justified without further assumptions: expecting that a sequence of events leads to the same outcome as it did in the past is only justified if we assume a uniformity of nature. The problems of generalization and of uniformity can be formulated in the context of supervised learning and RL, with (not uncontroversial) consequences (see NFL (2018)). For instance, one of the implications is that the expected performance of any two learning algorithms beyond the training set must be equal if one uniformly averages over all possible labeling functions, and analogous statements hold for RL settings—in other words, without assumptions on environments/datasets, the expected performance of any two learning models will be essentially the same, and two learning models cannot be meaningfully compared in terms of performance without making statements about the task environments in question. In practice, however, we always have *some* assumptions on the dataset and environment: for instance the principle of parsimony (i.e. Occam’s razor), asserting that simpler explanations tend to be correct, prevalent in science, suffices to break the symmetries required for NFLs to hold in their strongest form; see Lattimore and Hutter (2011), Hutter (2010) and Ben-David *et al* (2011).

No review of theoretical foundations of learning theory should circumvent the works of Valiant and the general computational learning theory (COLT), which stems from a computer science tradition, initiated by Valiant (1984), and the related VC theory of Vapnik and Chervonenkis, developed from a statistical viewpoint (Vapnik 1995). Roughly speaking, COLT investigates the theoretical limits of learning algorithms for various classes of learning scenarios. Thus, in a typical COLT scenario, one fixes and precisely mathematically specifies an environment or problem class, and characterizes the performance of an optimal learning algorithm. In contrast, VC theory focuses more on settings where the class of learning models (algorithms) is fixed whereas the problem classes are mostly uncharacterized, barring the training samples. The statements of efficiency or quality of performance are then attributed to the specified class of learning models (e.g. a choice of NNs, or some

choice of SVMs), based on empirical data—the performance on the samples. This corresponds to the settings one encounters when ML, in the sense of a data analysis tool, is applied in practice. The two theories thus offer complementary perspectives on learning.

We present the basic ideas of these theories in no particular order.

**2.2.1. Computational learning theory.** COLT can be understood as a rigorous formalization of supervised learning which stems from a computational complexity theory tradition. The most famous model in COLT is that of *probably approximately correct* (PAC) learning. We will explain the basic notions of PAC learning on a simple example: optical character recognition. Consider the task of training an algorithm to decide whether a given image (given as a black and white bitmap) of a letter corresponds to the letter ‘A’, by supplying a set of examples and counterexamples: a collection of images. Each image  $\mathbf{x}$  can be encoded as a binary vector in  $\{0,1\}^n$  (where  $n = \text{height} \times \text{width}$  of the image).

Assuming that there exists a univocally correct assignment of label 0 (not ‘A’) and 1 to each image implies there exists a characteristic function  $f : \{0,1\}^n \rightarrow \{0,1\}$  which discerns letters A from other images. Such an underlying characteristic function (or, equivalently, the subset of bit-strings for which it attains value ‘1’) is, in COLT, called a concept. Any (supervised) learning algorithm will first be supplied with a collection of  $N$  examples  $(\mathbf{x}_i, f(\mathbf{x}_i))_i$ . In some variants of PAC learning, it is assumed that the data points  $(\mathbf{x})$  are drawn from some distribution  $D$  attaining values in  $\{0,1\}^n$ . Intuitively, this distribution can model the fact that, in practice, the examples that are given to the learner stem from its interaction with the world, which specifies what kinds of ‘A’s we are more likely to see<sup>36</sup>. PAC learning typically assumes *inductive* settings, meaning that the learning algorithm, given a sample set  $S_N$  (comprising  $N$  identically independently distributed samples from  $D$ ) outputs a hypothesis  $h : \{0,1\}^n \rightarrow \{0,1\}$  which is, intuitively, the algorithms ‘best guess’ for the actual concept  $f$ . The quality of the guess is measured by the total error (also known as loss, or regret),

$$\text{err}_D(h_{S_N}) = \sum_{\mathbf{x}} P(D = \mathbf{x}) |h_{S_N}(\mathbf{x}) - f(\mathbf{x})|, \quad (15)$$

averaged according to the same (training) distribution  $D$ , where  $h_{S_N}$  is the hypothesis the (deterministic) learning algorithm outputs given the training set  $S_N$ . Intuitively, the larger the training set is ( $N$ ), the smaller the error will be, but this also depends on the actual examples (and thus  $S_N$  and  $D$ ). PAC theory concerns itself with probably ( $\delta$ ), approximately ( $\epsilon$ ) correct learning, i.e. with the following expression:

$$P_{S_N \sim D^N} [\text{err}_D(h_{S_N}) \leq \epsilon] \geq 1 - \delta, \quad (16)$$

where  $S \sim D$  means  $S$  was drawn according to the distribution  $D$ . The above expression is a statement certifying that

<sup>35</sup> An exception to this would be the uninteresting case when the class was finite and all instances had been observed.

<sup>36</sup> For instance, in modern devices, the devices are (mostly) trained for the handwriting of the owner, which will most of the time be distinct from other people’s handwriting, although the device should in principle handle any (reasonable) handwriting.

the learning algorithm, having been trained on the dataset sampled from  $D$ , will, except with probability  $\delta$ , have a total error below  $\epsilon$ . We say a concept  $f$  is  $(\epsilon, \delta)$ -learnable under distribution  $D$  if there exist a learning algorithm and an  $N$  such that equation (16) holds, and simply learnable if it is  $(\epsilon, \delta)$ -learnable for all choices of  $(\epsilon, \delta)$ . The functional dependence of  $N$  on  $(\epsilon, \delta)$  (and on the concept and distribution  $D$ ) is called the sample complexity. In PAC learning, we are predominantly concerned with identifying tractable problems, so a concept/distribution pair  $f, D$  is PAC-learnable if there exists an algorithm for which the sample complexity is polynomial in  $\epsilon^{-1}$  and  $\delta^{-1}$ . These basic ideas are generalized in many ways. First, in the case where the algorithm cannot output all possible hypotheses, but only a restricted set  $H$  (e.g. the hypothesis space is smaller than the total concept space), we can look for the best case solution by substituting the actual concept  $f$  with the optimal choice  $h^* \in H$  which minimizes the error in (15) in all the expressions above. Second, we are typically not interested in just distinguishing the letter ‘A’ from all other letters, but rather recognizing all letters. In this sense, we typically deal with a concept class (e.g. ‘letters’), which is a set of concepts, and it is (PAC) learnable if there exists an algorithm for which each of the concepts in the class are (PAC) learnable. If, furthermore, the same algorithm also learns for all distributions  $D$ , then the class is said to be (distribution-free) learnable.

COLT contains other models, generalizing PAC. For instance, concepts may be noisy or stochastic. In the agnostic learning model, the labeled examples  $(\mathbf{x}, y)$  are sampled from a distribution  $D$  over  $\{0, 1\}^n \times \{0, 1\}$ , which also models probabilistic concepts<sup>37</sup>. Furthermore, in agnostic learning we define a set of concepts  $C \subseteq \{c|c : \{0, 1\}^n \rightarrow \{0, 1\}\}$  and, given  $D$ , we can identify the best deterministic approximation of  $D$  in the set  $C$ , given with  $\text{opt}_C = \min_{c \in C} \text{err}_D(c)$ . The goal of learning is to produce a hypothesis  $h \in C$  which performs not much worse than the best approximation  $\text{opt}_C$ , in the PAC sense—the algorithm is a  $(\epsilon, \delta)$ -agnostic learner for  $D$  and  $C$  if, given access to samples from  $D$ , it outputs a hypothesis  $h \in C$  such that  $\text{err}_D(c) \leq \epsilon + \text{opt}_C$ , except with probability  $\delta$ .

Another common model in COLT is the exact learning from membership queries model (Angluin 1988), which is, intuitively, related to active supervised learning (see section 1.2.3). Here, we have access to an oracle, a black box, which outputs the concept value  $f(x)$  when queried with an example  $x$ . The basic setting is exact, meaning we are required to output a hypothesis which, with a bounded probability (say 3/4), makes no errors whatsoever. In other words, this is PAC learning where  $\epsilon = 0$ , but we get to choose which examples we are given, adaptively, and  $\delta$  is bounded away from 1/2. The figure of merit usually considered in this setting is query complexity, which denotes the number of calls to the oracle the learning algorithm uses, and is for most intents and purposes

synonymous with sample complexity<sup>38</sup>. This, in spirit, corresponds to an active supervised learning setting.

Much of PAC learning deals with identifying examples of interesting concept classes which are learnable (or proving that relevant classes are not), but other more general results exist connecting this learning framework. For instance, we can ask whether we can achieve a finite-sampling universal learning algorithm: that is, an algorithm that can learn any concept, under any distribution, using some fixed number of samples  $N$ . The No Free Lunch theorems we mentioned previously imply that this is not possible: for each learning algorithm (and  $\epsilon, \delta$ ) and any  $N$  there is a setting (concept/distribution) which requires more than  $N$  samples to achieve  $(\epsilon, \delta)$ -learning.

Typically, the criterion for a problem to be learnable assumes that there exists a classifier whose performance is essentially arbitrarily good—that is, it assumes the classifier is strong. The boosting result in ML, already touched upon in section 2.1.3, shows that settling on weak classifiers, which perform only slightly better than random classification, does not generate a different concept of learnability (Schapire 1990).

Classical COLT has also been generalized to deal with concepts with continuous ranges. In particular, so called  $p$ -concepts have range in  $[0, 1]$  (Kearns and Schapire 1994). The generalization of the entire COLT to deal with such continuous-valued concepts is not without problems, but nonetheless some of the central results, for instance quantities which are analogs of the VC dimension, and analogous theorems relating this to generalization performance, can still be provided (see Aaronson (2007) for an overview given in the context of the learning of quantum states discussed in section 5.1.1).

COLT is closely related to the statistical learning theory of Vapnik and Chervonenkis (VC theory) which we discuss next.

**2.2.2. Vapnik–Chervonenkis theory.** The statistical learning formalism of Vapnik and Chervonenkis was developed over the course of more than 30 years and in this review we are forced to present just a chosen aspect of the total theory, which deals with generalization performance guarantees. In the previous paragraph on PAC learning, we have introduced the concept of total error, which we will refer to as (total) risk. It is defined as the average over all the data points, which is, for a hypothesis  $h$ , given by  $R(h) = \text{error}(h) = \sum_{\mathbf{x}} P(D = \mathbf{x}) |h(\mathbf{x}) - f(\mathbf{x})|$  (we are switching notation to maintain consistency with the literature of differing communities). However, this quantity cannot be evaluated in practice, as in practice we only have access to the training data. This leads us to the notion of the empirical risk given by

$$\hat{R}(h) = \frac{1}{N} \sum_{\mathbf{x} \in \mathcal{S}_N} |h(\mathbf{x}) - f(\mathbf{x})|, \quad (17)$$

<sup>38</sup> When the oracle allows non-trivial inputs, one typically talks about query complexity. Sample complexity deals with the question of ‘how many samples’ which suggest the setting where the oracle only produces outputs, without taking inputs. The distinction is not relevant for our purposes and is more often a matter of convention of the research line.

<sup>37</sup> Note that we recover the standard PAC setting once the conditional probability distribution of  $P_D(y|\mathbf{x})$  where the values of the first  $n$  bits (data points) are fixed is Kronecker-delta, i.e. the label is deterministic.

where  $S_N$  is the training set drawn independently from the underlying distribution  $D$ .

The quantity  $\hat{R}(h)$  is intuitive and directly measurable. However, the problem of finding learning models which optimize empirical risk alone is not in itself interesting as it is trivially resolved with a look-up table. From a learning perspective, the more interesting and relevant quantity is the performance beyond the training set, which is contained in the unmeasurable  $R(h)$ , and indeed the task of inductive supervised learning is identifying  $h$  which minimizes  $R(h)$ , given only the finite training set  $S_N$ . Intuitively, the hypothesis  $h$  which minimizes the empirical risk should also be our best bet for the hypothesis which minimizes  $R(h)$ , but this can only make sense if our hypothesis family is somehow constrained, at least to a family of total functions: again, a look-up table has zero empirical risk, yet says nothing about what to do beyond. One of the key contributions of VC theory is to establish a rigorous relationship between the observable quantity  $\hat{R}(h)$ , the empirical risk; the quantity we actually wish to bound  $R(h)$ , the total risk; and the family of hypotheses our learning algorithm can realize. Intuitively, if the function family is too flexible (as is the case with just look-up tables) a perfect fit on the examples says little. In contrast, having a very restrictive set of hypotheses, say just one (which is independent from the dataset/concept and the generating distribution), suggests that the empirical risk is a fair estimate of the total risk (however bad it may be), as nothing has been tailored for the training set. This brings us to the notion of the model complexity of the learning model, which has a few formalizations, and here we focus on the Vapnik–Chervonenkis dimension of the model (VC dimension)<sup>39</sup>.

The VC dimension is an integer number assigned to a set of hypotheses  $H \subseteq \{h|h: \mathbf{S} \rightarrow \{0, 1\}\}$ , (e.g. the possible classification functions our learning algorithm can even in principle be trained to realize), where  $\mathbf{S}$  can be, for instance, the set of bit-strings  $\{0, 1\}^n$ , or, more generally, say, real vectors in  $\mathbb{R}^n$ . In the context of basic SVMs, the set of hypotheses is ‘all hyperplanes’<sup>40</sup>. Consider now a subset  $C_k$  of  $k$  points in  $\mathbb{R}^n$  in general position<sup>41</sup>. These points can attain binary labels in  $2^k$  different ways. The hypothesis family  $H$  is said to shatter the set  $C$  if for any labeling  $\ell$  of the set  $C_k$  there exists a hypothesis  $h \in H$  which correctly labels the set  $C_k$  according to  $\ell$ . In other words, using functions from  $H$  we can learn any labeling function on the set  $C_k$  of  $k$  points in general position perfectly. The VC dimension of  $H$  is then the largest  $k_{\max}$  such that there exists a set  $C_{k_{\max}}$  of points in general position which is shattered (perfectly ‘labelable’ for any labeling) by  $H$ . For instance, for  $n = 2$ , ‘rays’ shatter three points but not 4 (imagine vertices of a square where diagonally opposite vertices share the same label), and in  $n = N$ , ‘hyperplanes’ shatter  $N + 1$  points. While it is beguiling to think that the VC dimension corresponds to the number of free parameters specifying

the hypothesis family, this is not the case<sup>42</sup>. The VC theorem (in one of its variants) (Devroye *et al* 1996) then states that the empirical risk matches total risk up to a deviation which decays in the number of samples, but grows in the VC dimension of the model, more formally:

$$P\left(\hat{R}(h_{S_N}) - R(h_{S_N}) \leq \epsilon\right) = 1 - \delta \quad (18)$$

$$\epsilon = \sqrt{\frac{d(\log(2N/d) + 1)}{N} - \frac{\log(\delta/4)}{N}}, \quad (19)$$

where  $d$  is the VC dimension of the model,  $N$  the number of samples and  $h_{S_N}$  is the hypothesis output by the model given the training set  $S_N$ , which is sampled from the underlying distribution  $D$ . The underlying distribution  $D$  implicitly appears also in the total risk  $R$ . Note that the chosen acceptable probability of incorrectly bounding the true error,  $\delta$ , contributes only logarithmically to the misestimation bound  $\epsilon$ , whereas the VC dimension and the number of samples contribute (mutually inversely) linearly to the square of  $\epsilon$ .

The VC theorem suggests that the ideal learning algorithm would have a low VC dimension (allowing a good estimate of the relationship between the empirical and total risk), while at the same time performing well on the training set. This leads to a learning principle called structural risk minimization. Consider a parametrized learning model (say parametrized by an integer  $l \in \mathbb{N}$ ) such that each  $l$  induces a hypothesis family  $H^l$ , each more expressive than the previous, so  $H^l \subseteq H^{l+1}$ . Structural risk minimization (contrasted to empirical risk minimization which just minimizes empirical risk) takes into account that in order to have (a guarantee on) good generalization performance we need to have both good observed performance (i.e. low empirical risk) and low model complexity. High model complexity induces risk stemming from the structure of the problem, manifested in common issues such as data overfitting. In practice, this is achieved by considering (meta-)parametrized models, like  $\{H^l\}$ , where we minimize a combination of  $l$  (influencing the VC dimension) and the empirical risk associated to  $H^l$ . In practice, this is realized by adding a regularization term to the training optimization, so generically the (unregularized) learning process resulting in  $\operatorname{argmin}_{h \in H} \hat{R}(h)$  is updated to  $\operatorname{argmin}_{h' \in H'} \left(\hat{R}(h) + \operatorname{reg}(l)\right)$ , where  $\operatorname{reg}(\cdot)$  penalizes the complexity of the hypothesis family, or just the given hypothesis.

VC dimension is also a vital concept in PAC learning, connecting the two frameworks. Note first that a concept class  $C$ , which is a set of concepts, is also a legitimate set of hypotheses and thus has a well-defined VC dimension  $d_C$ . Then the sample complexity of  $(\epsilon, \delta)$ –(PAC)–learning of  $C$  is given by  $O((d_C + \log 1/\delta)\epsilon^{-1})$ .

<sup>39</sup> Another popular measure of model complexity is e.g. Rademacher complexity (Bartlett and Mendelson 2003).

<sup>40</sup> Naturally, a non-trivial kernel function enriches the set of hypotheses realized by SVMs.

<sup>41</sup> General position implies that no sub-set of points is co-planar beyond what is necessary, i.e. points in  $S \subset \mathbb{R}^n$  are in general position if no hyperplane in  $\mathbb{R}^n$  contains more than  $n$  points in  $S$ .

<sup>42</sup> The canonical counterexample is the family specified by the partition of the real plane halved by the graph of the two-parametric function  $h_{\alpha, \beta}(x) = \alpha \sin(\beta x)$ , which can be proven to shatter any finite number of points in  $n = 2$ . The fact that the number of parameters of a function does not fully capture the complexity of the function should not be surprising as any (continuous) function over  $k + n$  variables (parameters + dimension) can be encoded as a function over  $1 + n$  variables.

Some of the approaches developed for supervised learning and discriminative models have, to some extent, been extended to the domain of unsupervised learning as well. Such results are much rarer and more specialized and beyond the scope of this review. However, for an example of the types of results that have been established, we refer the interested reader to the following work of Seldin and Tishby (2010).

### 2.3. Basic methods and theory of reinforcement learning

**Executive summary:** While RL, in full generality, studies learning in and from interactive task environments, perhaps the best understood models consider more restricted settings. Environments can often be characterized by Markov Decision Processes. Such environments are characterized by states which can be observed by the agent. The agent can cause transitions from states to states by its actions but the *rules of transitions* are not known beforehand. Some of the transitions are rewarded. The agent learns which actions to perform, given that the environment is in some state, so that it receives the highest value of rewards (expected return), either in a fixed time frame (finite-horizon) or over (asymptotically) long time periods, where future rewards are geometrically depreciated (infinite-horizon). Such models can be solved by estimating action-value functions, which assign expected return to actions given states, for which the agent must explore the space of strategies, but other methods exist. In more general models, the state of the environment need not be fully observable and such settings are significantly harder to solve. RL settings can also be tackled by models from the so-called Projective Simulation framework for the design of learning agents, which exploits physical stochastic processes and a notion of episodic memory. While comparatively new, this model is of particular interest because it offers a natural route for beneficial quantization. Interactive learning methods include models beyond textbook RL, including partially observable settings, which require generalization and more. Such extensions, e.g. generalization, typically require techniques from non-interactive learning scenarios, but also lead to agents with an ever-increasing level of autonomy. In this sense, RL forms a bridge between ML and general AI models.

Broadly speaking, RL deals with the problem of learning how to optimally behave in unknown environments. In the basic textbook formalism we deal with a task environment, which is specified by a Markov decision process (MDP).

MDPs are labeled, directed graphs with additional structures, comprising a discrete and finite set of states  $\mathcal{S} = \{s_i\}$  and of actions  $\mathcal{A} = \{a_i\}$ , which denote the possible states of the environment and the actions the learning agent can perform on it, respectively. A simple three state MDP is illustrated in Figure 8.

The choice of the actions of the agent changes the state of the environment in a manner which is specific to the environment (MDP) and which may be probabilistic. This is captured by a transition rule  $P(s|s', a)$ , denoting the probability of the environment ending up in the state  $s$ , if the action  $a$  had been performed in the state  $s'$ . Technically, this can be viewed as a collection of action-specific Markov transition matrices  $\{P^a\}_{a \in \mathcal{A}}$  that the learner can apply on the environment by performing an action.

These describe the dynamics of the environment conditioned on the actions of the agent. The final component specifying the environment is a reward function  $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \Lambda$ , where  $\Lambda$  is a set of rewards, often binary. In other words, the environment rewards certain transitions<sup>43</sup>. At each time instance, the action of the learner is specified by a policy: a conditional probability distribution  $\pi(a|s)$  specifying the probability of the agent outputting the action  $a$  provided it is in the state  $s$ . Given an MDP, intuitively the goal is finding good policies, i.e. those which yield high rewards. This can be formalized in many non-equivalent ways. Given a policy  $\pi$  and some initial state  $s$  we can, e.g. define a finite-horizon expected total reward after  $N$  interaction steps with  $R_N^s(\pi) = \sum_{i=1}^N r_i$ , where  $r_i$  is the expected reward under policy  $\pi$  at time-step  $i$  in the given environment and assuming we started from the state  $s$ . If the environment is finite and strongly connected<sup>44</sup>, the finite-horizon rewards diverge as the horizon  $N$  grows. However, by adding a geometrically depreciating factor (rate  $\gamma$ ) we obtain an always bounded expression  $R_\gamma(\pi) = \sum_{i=1}^{\infty} \gamma^i r_i$ , called the infinite horizon expected reward (parametrized by  $\gamma$ ), which is more commonly studied in literature. The expected rewards in finite or infinite horizons form the typical figures of merit in solving MDP problems, which come in two flavors. First, in decision theory, or planning (in the context of AI), the typical goal is finding the policy  $\pi^{\text{opt}}$  which optimizes the (in)finite horizon reward in a *given MDP*, formally: given the (full or partial) specification of the MDP  $M$ , solve  $\pi^{\text{opt}} = \text{argmax}_\pi R_{N/\gamma}(\pi)$ , where  $R$  is the expected reward in the finite (for  $N$  steps) or infinite horizon (for a given depreciation  $\gamma$ ) setting, respectively. Such problems can be solved by dynamic and linear programming. In RL (Sutton and Barto 1998), the specification of the environment (the MDP), in contrast, is not given, but rather can be explored by interacting with it dynamically. The agent can perform an action and receive the subsequent state (and perhaps a reward). The ultimate goal here comes in two related (but conceptually different) flavors. One is to design an agent which will over time learn the optimal policy  $\pi^{\text{opt}}$ , meaning the policy can be read out from the memory of the agent/program. Slightly differently, we seek an agent which will, over time gradually alter its policy so as to *act* according to the optimal policy. While in

<sup>43</sup> Rewards can also be probabilistic. This can be modeled by explicitly allowing stochastic reward functions or by extending the state space to include rewarding and non-rewarding instances of states (note the reward depends on current state, action *and* the reached state) in which case the probability of the reward is encoded in the transition probabilities.

<sup>44</sup> In this context this means that the underlying MDP has finite return times for all states, that is, there is a finite probability of going back to the initial state from any state for some sequence of actions.

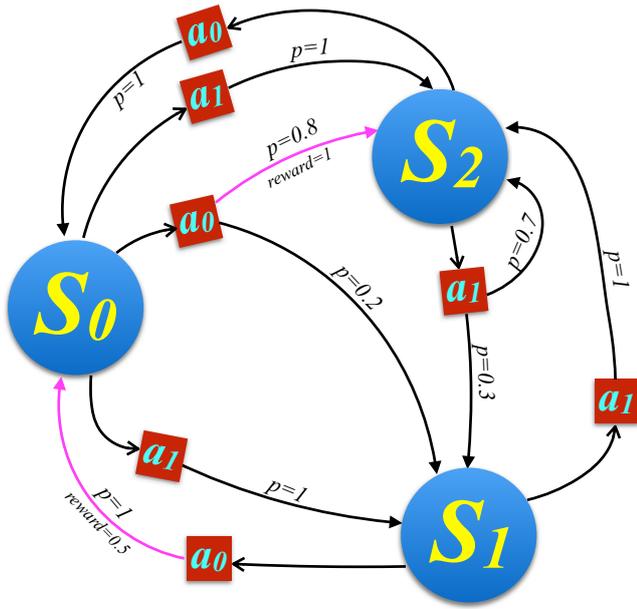


Figure 8. A three state, two-action MDP.

theory these two are closely related, in robotics, for example, these are quite different as the reward rate before convergence (perfect learning) also matters<sup>45</sup>. First of all, we point out that RL problems as given above can be solved reliably whenever the MDP is finite and strongly connected: a trivial solution is to stick to a random policy until a reliable tomography of the environment can be done, after which the problem is resolved via dynamic programming<sup>46</sup>. Often, environments actually have additional structure, so-called initial and terminal states: if the agent reaches the terminal state, it is ‘teleported’ to the fixed initial state. Such structure is called episodic and can be used as a means of ensuring the strong connectivity of the MDP.

One way of obtaining solutions is by tracking so-called value functions  $V_\pi(s) : \mathcal{S} \rightarrow \mathbb{R}$  which assign the expected reward under policy  $\pi$  assuming we start from state  $s$ ; this is done recursively: the value of the current state is the current reward plus the averaged value of the subsequent state (averaged under the stochastic transition rule of the environment  $P(s|a, s')$ ). Optimal policies optimize these functions and this, too, is achieved sequentially by modifying the policy so as to maximize the value functions. This, however, assumes the knowledge of the transition rule  $P(s|a, s')$ . In further development of the theory, it was shown that tracking action-value functions  $Q_\pi(s, a)$ , given by

$$Q_\pi(s, a) = \sum_{s'} P(s'|a, s)(\Lambda(s, a, s') + \gamma V_\pi(s')) \quad (20)$$

<sup>45</sup> These two flavors are closely related to the notions of on-policy and off-policy learning. These labels typically pertain to how the estimates of the optimal policy are internally updated, which may be in accordance with the actual current policy and actions of the agent or independently from the executed action, respectively. For more details see e.g. Sutton and Barto (1998).

<sup>46</sup> If the environment is not strongly connected this is not possible: for instance the first move of the learner may lead to ‘good’ or ‘bad’ regions from which there is no way out, in which case optimal behavior cannot be obtained with certainty.

assigning the value not only to the state, but to the subsequent action as well, can be modified into an online learning algorithm<sup>47</sup>. In particular, the Q-values can be continuously estimated by weighted averaging the current reward (at time step  $t$ ) for an action-value and the estimate of the highest possible Q-value of the subsequent action-value:

$$Q^{t+1}(s_t, a_t) = \underbrace{Q^t(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t}_{\text{learning rate}} \cdot \left( \underbrace{r_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount}} \cdot \underbrace{\max_a Q^t(s_{t+1}, a)}_{\text{estimate of optimal future value}} - \underbrace{Q^t(s_t, a_t)}_{\text{old value}} \right) \quad (21)$$

Note that having access to the optimal Q-values suffices to find the optimal policy—given a state, simply pick an action with the highest Q-value—but the algorithm above says nothing about which policy the agent should employ while learning. In Watkins and Dayan (1992) it was shown that the algorithm specified by the update rule of equation (21), called *Q-learning*, indeed converges to optimal Q values as long as the agent employs any fixed policy which has non-zero probabilities for all actions given any state (the parameter  $\alpha_t$ , which is a function of time, has to satisfy certain conditions and  $\gamma$  should be the  $\gamma$  of the targeted figure of merit  $R_\gamma$ )<sup>48</sup>.

In essence, this result suffices for solving the first flavor of RL, where the optimal policy is ‘learned’ by the agent in the limit, but, in principle, never actually used. The convergence of the Q-learning update to the optimal Q-values and consequently to the optimal behavior has been proven for all learning agents using *greedy-in-the-limit*, *infinite exploration* (GLIE) policies. As the name suggests, such policies in the asymptotic limit perform actions with the highest value estimated<sup>49</sup>.

At the same time, infinite exploration means that, in the limit, all state/action combinations will be tried out infinitely

<sup>47</sup> This rule is inspired by the Bellman optimality equation,  $Q^*(s, a) := \mathbb{E}[R(s, a)] + \gamma \mathbb{E}[\max_{a'} Q^*(s', a')]$ . The expected values appear as  $R(s, a)$  can be stochastic and can be modeled as a random variable, and  $\max_{a'} Q^*(s', a')$  is a random variable as well as it is, implicitly, a function of the stochastic MDP transition rule. This equation has as a solution a fixed point, which is an optimal Q-value function. This equation can be used when the specification of the environment is fully known. Note that the optimal Q-values can be found without actually explicitly identifying an optimal policy.

<sup>48</sup> Q-learning is an example of an off-policy algorithm as the estimate of the future value in equation (21) is not evaluated relative to the actual policy of the agent (indeed, it is not necessarily even defined), but rather relative to the so-called ‘greedy policy’, which takes the action with the maximal value estimate (note the estimate appears with a maximization term).

<sup>49</sup> To avoid any confusion, we have introduced the concept *policy* to refer to the conditional probability distributions specifying what the agent will do given a state. However, the same term is often overloaded to also refer to the specification of the effective policy an agent will use given some state/time-step. For instance, ‘ $\epsilon$ -greedy policies’ refer to behavior in which, given a state, the agent outputs the action with the highest corresponding  $Q$ -value—i.e. acts *greedily*—with probability  $1 - \epsilon$  and produces a random action otherwise. Clearly, this rule specifies a policy at any given time step, given the current Q-value table of the agent. One can also think of *time-dependent* policies, which mean that the policy also explicitly depends on the time-step. An example of a such a time-dependent and a (slowly converging) GLIE policy is an  $\epsilon$ -greedy policy, where  $\epsilon = \epsilon(t) = 1/t$  is a function of the time-step, converging to zero.

many times, ensuring that the true optimal action values are found and that the local minima are avoided. In general, the optimal trade off between these two competing properties, the *exploration of the learning space* and the *exploitation of obtained knowledge*, is quintessential to RL. There are many other RL algorithms which are based on state-value or action-value optimizations, such as SARSA<sup>50</sup>, various value iteration methods, temporal difference methods, etc (Sutton and Barto 1998). More recently, progress has been achieved by using parametrized approximations of state-action-value-functions—a cross-breed between function approximation and RL—which reduces the search space of available Q-functions.

Here the results which use deep learning methods for RL based on value function approximation, leading to the system dubbed deep Q-networks, have been particularly successful (Mnih *et al* 2015). Deep Q-networks also underpin the AlphaGo (Silver *et al* 2016, 2017a, 2017b) system. Utilizing parametric (action-)value functions becomes necessary when the percept/action spaces become large or continuous<sup>51</sup>.

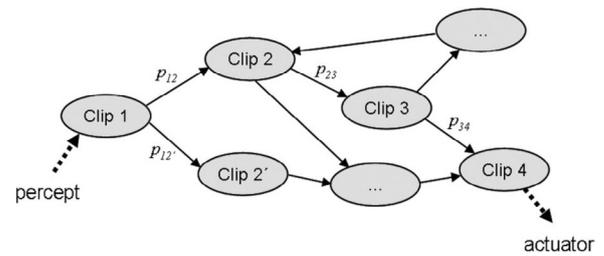
This brings us to a different class of methods which do not optimize (action-)value functions, but rather directly learn complete policies, often by performing an estimate of gradient descent, or other means of optimization in policy space. In such approaches, the policies are often specified indirectly by a comparably small number of parameters. The expected rewards which a given policy yields are evaluated by interacting with the environment, and this empirical objective function can be used to optimize the parameters, and thus the policy. Such policy search methods with parametrization can in some cases lead to faster learning (Peshkin 2001), but may also be a necessity, if the state/action spaces are large. In recent times, the parametrizations of both policies and action-value functions are often NNs. For a survey of methods which combine deep networks with RL machinery, we refer the reader to Arulkumaran *et al* (2017).

Value function and policy search methods can also be combined. A standard example is the so-called *actor-critic framework*, in which the critic—often an action-value function—is used to optimize an actor—a (parametrized) policy. The critic is updated via the interaction with the environment. For example, actor-critic methods are sometimes used to combat the problems which arise with action-value function approximation approaches where the action space is continuous. In this case, computing the action which maximizes the state-action value may be intractable, so the parametrized policy is used as an ansatz. For further details on actor-critic methods we refer the reader to Grondman *et al* (2012).

The theory of RL most often considers scenarios where the environment is Markovian, or, related to this, fully observable. The most common generalization of MDP environments are so-called partially observable MDPs (POMDP), where the underlying MDP structure is extended to include a set of observations  $\mathcal{O}$  and a stochastic function defined with the conditional probability distribution  $P_{\text{POMDP}}(o \in \mathcal{O} | s \in \mathcal{S}, a \in \mathcal{A})$ . The set

<sup>50</sup> SARSA is the acronym for state-action-reward-state-action.

<sup>51</sup> We note that even coarse-graining of the action spaces, or the discretization of the state spaces also constitutes a parametrization of the (otherwise) potentially much larger policy space.



**Figure 9.** Illustration of the structure of the episodic and compositional memory in PS, comprising clips (episodes) and probabilistic transitions. The actuator of the agent performs the action. Adapted from Briegel and De las Cuevas (2012).

of states of the environment are no longer directly accessible to the agent but rather the agent perceives the observations from the set  $\mathcal{O}$ , which indirectly and, in general, stochastically depends on the actual unobservable environmental state, as given by the distribution  $P_{\text{POMDP}}$  and the action the agent took last. POMDPs are expressive enough to capture many real world problems, and are thus a common *world model* in AI, but are significantly more difficult to deal with compared to MDPs<sup>52</sup>.

As mentioned, the setting of POMDPs moves us one step closer to arbitrary environment settings, which is the domain of artificial (general) intelligence<sup>53</sup>.

The context of AGI is often closely related to modern view on robotics, where the structure of what can be observed and what actions are possible stems not only from the nature of the environment, but also (bodily) constraints of the agent: e.g. a robot is equipped with sensors, specifying and limiting what the robot can observe or perceive, and actuators, constraining the possible actions. In such an agent-centric viewpoint, we typically talk about the set of percepts—signals that the agent can perceive—which may correspond to full states, or partial observations, depending on the agent-environment setting—and the set of actions<sup>54</sup>.

This latter viewpoint, that the percept/action structure stems from the physical constitution of the agent and the environment, which we will refer to as an embodied perspective, was one of the starting points of the development of the

<sup>52</sup> For instance, the problem of finding optimal infinite-horizon policies, which was solvable via dynamical programming in the fully observable (MDP) case becomes, in general, uncomputable.

<sup>53</sup> To comment a bit on how RL methods and tasks may be generalized towards general AI, one can consider learning scenarios where one has to combine standard data-learning ML to handle the realistic percept space (which is effectively infinite) with RL techniques. An example of such a successful combination of various ML/RL methods is the famous AlphaGo system (Silver *et al* 2016). Further, one could also consider more general types of interaction, beyond the strict turn-based metronomic model. For instance in *active RL*, the interaction occurs relative to an external clock, which intertwines computational complexity and learning efficiency of the agent (see section 7.1). Further, the interaction may occur in fully continuous time. This setting is also not typically studied in the basic theory of AI, but occurs in the closely related problem of control theory (Wiseman and Milburn 2010), which may be more familiar to physicists. Such generalizations are at the cutting edge of research, including in the classical realm, and are also beyond the scope of this paper.

<sup>54</sup> In this sense, a particular agent/robot may perceive the full state of the environment in some environments (making the percepts identical to states), whereas in other environments the sensors fail to observe everything, in which case the percepts correspond to observations.

<p><i>Applications of ML in quantum physics</i></p> <ol style="list-style-type: none"> <li>(1) Estimation and metrology</li> <li>(2) Quantum control and gate design</li> <li>(3) Controlling quantum experiments, and machine-assisted research</li> <li>(4) Condensed matter and many body physics</li> </ol> <p><i>Quantum enhancements for ML</i></p> <ol style="list-style-type: none"> <li>(1) Quantum perceptrons and neural networks</li> <li>(2) Quantum computational learning theory</li> <li>(3) Quantum enhancement of learning capacity</li> <li>(4) Quantum computational algorithmic speed-ups for learning</li> </ol>	<p><i>Quantum generalizations of ML-type tasks</i></p> <ol style="list-style-type: none"> <li>(1) Quantum generalizations: machine learning of quantum data</li> <li>(2) (Quantum) learning of quantum processes</li> </ol> <p><i>Quantum learning agents and elements of quantum AI</i></p> <ol style="list-style-type: none"> <li>(1) Quantum-enhanced learning through interaction</li> <li>(2) Quantum agent-environment paradigm</li> <li>(3) Towards quantum AI</li> </ol>
--	--

**Figure 10.** Table of topics investigating the overlaps between quantum physics, ML and AI.

projective simulation (PS) model for AI. PS is a physics-inspired model for AI and agency which can be used for solving RL tasks. The centerpiece of the model is the so-called Episodic and Compositional Memory (ECM), which is a stochastic network of clips; see figure 9. Clips are representations of short autobiographical episodes, i.e. the agent’s memories. Using the compositional aspects of the memory, which allows for a rudimentary notion of creativity, the agent can also combine actual memories to generate fictitious clips, which need not correspond to things that have actually occurred, but which are conceivable on the basis of the agent’s previous experience (Briegel 2012). More formally, clips can be defined recursively as either memorized percepts or actions, or otherwise structures (e.g. sequences) of clips. Given a current percept, the PS agent calls its ECM network to perform a stochastic random walk over its clip space (the structure of which depends on the history of the agent) projecting itself into conceivable situations, before committing to an action. Aspects of this model have been beneficially quantized, and also used both in quantum experiments and in robotics, and we will focus more on this model in section 7.1.

*Learning efficiency and learnability for reinforcement learning.* As mentioned in the introduction to this section, No Free Lunch theorems also apply to RL, and any statement about learning requires us to restrict the space of possible environments. For instance, ‘finite-space, time-independent MDPs’ is a restriction which allows perfect learning relative to some of the standard figures of merit, as was first proven by the Q-learning algorithm. Beyond learnability, in more recent times, notions of sample complexity for RL tasks have also been explored, addressing the problem from different perspectives. The theory of sample complexity for RL settings is significantly more involved than for supervised learning, although the very basic desiderata remain the same: how many interaction steps are needed before the agent learns. Learning can naturally mean many things, but most often what is meant is that the agent learns the optimal policy. Unlike supervised learning, RL has an additional temporal dimension in

the definitions of optimality (e.g. finite or infinite horizons), leading to an even broader space of options one can explore. Further details on this important field of research are beyond the scope of this review and we refer the interested reader to e.g. the thesis of Kakade (2003) which also does a good job of reviewing some of the early works and finds sample complexity bounds for RL for many basic settings, or e.g. Lattimore *et al* (2013) and Dann and Brunskill (2015) for some of the newer results.

### 3. Quantum mechanics, learning and artificial intelligence

Quantum mechanics has already had a profound effect on the fields of computation and information processing. However, its impact on AI and learning has, up until very recently, been modest. Although the fields of ML and AI have a strong connection to the theory of computation, these fields are still different, and not all progress in (quantum) computation implies qualitative progress in AI. For instance, although it has been more than 20 years, still the arguably most celebrated result in QC is that of Shor’s factoring algorithm (Shor 1997), which, on the face of it, has no impact on AI<sup>55</sup>. Nonetheless, other, less famous, results may have application to various aspects of AI and learning. The fields of QIP, AI and machine learning have thus, from their early stages, had a careful and tentative interplay, although it is only recently that this line of research has received broader attention. Roughly speaking, we can identify four main directions covering the interplay between ML/AI summarized in figure 10.

Historically speaking, the first contacts between aspects of QIP and learning theory occurred in terms of the direct application of statistics and statistical learning in light of quantum theory, which forms the first line: classical ML applied in

<sup>55</sup> In fact, this is not entirely true—certain proofs of separation between PAC learnability in the quantum and classical model assume hardness of factoring of certain integers (see section 6.1.2).

quantum theory and experiment, reviewed in section 4. In this first topic, ML techniques are applied to classical data (such as the results of measurements) stemming from quantum experiments. The second topic, in contrast, deals with ML over genuinely quantum data: quantum generalization of machine-learning-type tasks, discussed in section 5. This brings us to the topic which has been receiving substantial interest in recent times: whether quantum computers can genuinely help in ML problems, addressed in section 6. The final topic we will investigate considers aspects of QIP which extend beyond ML (taken in a narrow sense), such as generalizations of RL, and which can be understood as stepping-stones towards quantum AI. This is reflected upon in section 7.3.

It is worthwhile to note that there are many possible natural classifications of the comprehensive field we discuss in this review. Our chosen classification is motivated by two subtly differing perspectives on the classification of quantum ML, discussed further in section 7.2.1.

#### 4. Machine learning applied to (quantum) physics

In this section we review works and ideas where ML methods have been either directly utilized, or have otherwise been instrumental, for QIP results. To do so, we are, however, facing the thankless task of specifying the boundaries of what is considered an ML method. In recent times, in part due to its successes, ML has become a desirable key word, and consequently an umbrella term for a broad spectrum of techniques. This includes algorithms for solving genuine learning problems, but also methods and techniques designed for indirectly related problems. From such an all-encompassing viewpoint, ML also includes aspects of (parametric) statistical learning, the solving of black-box (or derivative-free) optimization problems, but also the solving of hard optimization problems in general<sup>56</sup>.

As we do not presume to establish hard boundaries, we adopt a more inclusive perspective. The collection of all works that utilize methods which could conceivably fit in broad-scope ML for QIP applications cannot be covered in one review. Consequently, we place emphasis on pioneering works and works where the authors themselves advertise the ML flavor of used methodologies, thereby emphasizing the potential of such ML/QIP interdisciplinary endeavors.

The use of ML in the context of QIP, understood as above, has been considerable, with an effective explosion of related works in the last few years. ML has been shown to be effective in a great variety of QIP related problems: in quantum signal processing, quantum metrology, Hamiltonian estimation and in problems of quantum control. In recent times, the scope of applications has been significantly extended. ML and involved techniques have also been applied to combating noise in the process of performing quantum computations, problems in condensed-matter and many-body physics and in the design

of novel quantum optical experiments. Such results suggest that advanced ML/AI techniques will play an integral role in quantum labs of the future and, in particular, in the construction of advanced quantum devices and, eventually, quantum computers. In a complementary direction, QIP applications have also engaged many of the methods of ML, showing that QIP may also become a promising proving ground for cutting edge ML research.

Contacts between statistical learning theory (as a part of the theoretical foundations of ML) and quantum theory come naturally due to the statistical foundations of quantum theory. Already the very early theories of quantum signal processing (Helstrom 1969), probabilistic aspects of quantum theory and quantum state estimation (Holevo 1982) and early works (Braunstein and Caves 1994) which would lead to modern quantum metrology (Giovannetti *et al* 2011) included statistical analyses which establish tentative grounds for more advanced ML/QIP interplay. Related early works further emphasize the applicability of statistical methods, in particular maximum likelihood estimation, to quantum tomographic scenarios, such as the tasks of state estimation (Hradil 1997), the estimation of quantum processes (Fiurášek and Hradil 2001) and measurements (Fiurášek 2001) and the reconstruction of quantum processes from incomplete tomographic data (Ziman *et al* 2005)<sup>57</sup>. The works of this type generally focus on physical scenarios where clean analytic theory can be applied. However, in particular in experimental or noisy (thus, realistic) settings, many of the assumptions which are crucial for the pure analytic treatment fail. This leads to the first category of ML applications to QIP which we consider.

##### 4.1. Hamiltonian estimation and metrology

**Executive summary:** Metrological scenarios can involve complex measurement strategies, where e.g. the measurements which need to be performed may depend on previous outcomes. Further, the physical system under analysis may be controlled with the help of additional parameters—so-called controls—which can be sequentially modified, leading to a more complicated space of possibilities. ML techniques can help us find optima in such a complex space of strategies under various constraints, which are often pragmatically and experimentally motivated constraints.

The identifying of properties of physical systems, be they dynamic properties of evolutions (e.g. process tomography) or properties of the states of given systems (e.g. state tomography), is a fundamental task. Such tasks are resolved by various (classical) metrological theories and methods, which can identify optimal strategies, characterize error bounds and which have also been quite generally exported to the quantum realm. For instance, quantum metrology studies the estimation of

<sup>56</sup> Certain optimization problems, such as online optimization problems where information is revealed incrementally and decisions are made before all information is available, are more clearly related to ‘quintessential’ ML problems such as supervised, unsupervised, or reinforcement learning.

<sup>57</sup> Interestingly, such techniques allow for the identification of optimal approximations of *unphysical processes* which can be used to shed light on the properties of quantum operations.

the parameters of quantum systems and, generally, identifies optimal measurement strategies for their estimation. Further, quantum metrology places particular emphasis on scenarios where genuine quantum phenomena yield an advantage over simpler, classical strategies. In the context of quantum metrology, typically, quantum scenarios are characterized by the need for complex and difficult-to-implement quantum devices for their realization.

The specification of optimal strategies, in general, constitutes a problem of planning<sup>58</sup>, for which various ML techniques can be employed. The first examples of ML applications for finding measurement strategies originate from the problem of phase estimation, a special case of Hamiltonian estimation. Interestingly, this simple case already provides a fruitful playground for ML techniques: optimal measurement strategies are relatively easy to find analytically, but are experimentally unfeasible. In turn, if we limit ourselves to a set of ‘simple measurements’, near-optimal results are possible but they require difficult-to-optimize adaptive strategies—the type of problem ML is good for. Hamiltonian estimation problems have also been tackled in more general settings, invoking more complex machinery. We first briefly describe basic Hamiltonian estimation settings and metrological concepts. Then we will delve deeper into these results combining ML with metrology problems.

**4.1.1. Hamiltonian estimation.** The generic scenarios of Hamiltonian estimation, a common instance of metrology in the quantum domain, consider a quantum system governed by a (partially unknown) Hamiltonian within a specified family  $H(\boldsymbol{\theta})$ , where  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)$ , is a set of parameters. Roughly speaking, Hamiltonian estimation deals with the task of identifying the optimal methods (and the performance thereof) for estimating the Hamiltonian parameters.

This amounts to optimizing the choice of initial states (*probe states*) which will evolve under the Hamiltonian and the choice of the subsequent measurements, which uncover the effect the Hamiltonian had, and thus, indirectly, the parameter values<sup>59</sup>. This prolific research area considers many restrictions, variations and generalizations of this task. For instance, one may assume settings in which we either have control over the Hamiltonian evolution time  $t$  or it is fixed so that  $t = t_0$ , which are typically referred to as frequency and phase estimation, respectively. Further, the efficiency of the process can be measured in multiple ways. In a frequentist approach, one is predominantly interested in estimation strategies which, roughly speaking, allow for the best scaling of precision of the estimate, as a function of the number of measurements. The quantity of interest is the

so-called quantum Fisher information, which bounds and quantifies the scaling. Intuitively, in this setting, also called the local regime, many repetitions of measurements are typically assumed. Alternatively, in the Bayesian, or single-shot, regime the prior information, which is given as a distribution over the parameter to be estimated, and its update to the posterior distribution given a measurement strategy and outcome are central objects (Jarzyna and Demkowicz-Dobrzański 2015). The objective here is the identification of preparation/measurement strategies which optimally reduce the average variance of the posterior distribution, which is computed via Bayes’ theorem.

One of the key interests in this problem is that the utilization of arguably genuine quantum features such as entanglement, squeezing etc in the structure of the probe states and measurements may lead to provably more efficient estimation than is possible by so-called classical strategies for many natural estimation problems. Such quantum enhancements are potentially of immense practical relevance (Giovannetti *et al* 2011). The identification of optimal scenarios has been achieved in certain ‘clean’ theoretical scenarios, which are, however, often unrealistic or impractical. It is in this context that ML-flavored optimization and other ML approaches can help.

**4.1.2. Phase estimation settings.** Interesting estimation problems, from an ML perspective, can already be found in the simple examples of a phase shift in an optical interferometer, where one of the arms of an otherwise balanced interferometer contains a phase shift of  $\theta$ . Early on, it was shown that given an optimal probe state with mean photon number  $N$  and an optimal (so-called *canonical*) measurement, the asymptotic phase uncertainty can decay as  $N^{-1}$  (Sanders and Milburn 1995)<sup>60</sup>, known as the Heisenberg limit. In contrast, the restriction to ‘simple measurement strategies’ (as characterized by the authors), involving only photon number measurements in the two output arms, achieves a quadratically weaker scaling of  $\sqrt{N^{-1}}$ , referred to as the standard quantum limit. This was proven in more general terms: the optimal measurements cannot be achieved by the classical post-processing of photon number measurements of the output arms but constitute an involved, experimentally unfeasible POVM (Berry and Wiseman 2000). However, in Berry and Wiseman (2000) it was shown how this can be circumvented by using ‘simple measurements’, provided they can be altered in run-time. Each measurement consists of a photon number measurement of the output arms and is parametrized by an additional controllable phase shift of  $\phi$  in the free arm—equivalently, the unknown phase can be tweaked by a chosen  $\phi$ . The optimal measurement process is an *adaptive strategy*: an entangled  $N$ -photon state is prepared (see e.g. Berry *et al* (2001)), the photons are sequentially injected into the interferometer and photon numbers are measured. At each step, the measurement performed is modified by choosing

<sup>58</sup> More specifically, most metrology settings problems constitute instances of off-line planning, and thus not RL, as the ‘environment specification’ is fully specified—in other words, there is no need to actually run an experiment and the optimal strategies can be found off-line. See section 1.2 for more detail.

<sup>59</sup> Technically, the estimation also involves the use of a suitable estimator function, but these details will not matter.

<sup>60</sup> This is often also expressed in terms of the variance  $(\Delta\theta)^2$ , so as  $N^{-2}$ , rather than the standard deviation.

a different phase shift  $\phi$ , which depends on previous measurement outcomes. In Berry and Wiseman (2000) and Berry *et al* (2001), an explicit strategy was given which achieves the Heisenberg scaling of the optimal order  $O(1/N)$ . However, for  $N > 4$  it was shown this strategy is not *strictly* optimal.

This type of planning is hard as it reduces to the solving of non-convex optimization problems<sup>61</sup>. The field of ML deals with such planning problems as well, and so many optimization techniques have been developed for this purpose. The applications of such ML techniques, specifically particle swarm optimization, were first suggested in the pioneering works of Hentschel and Sanders (2010, 2011) and later in Sergeevich and Bartlett (2012). In subsequent work, perhaps more well-known methods of differential evolution have been demonstrated to be superior and more computationally efficient (Lovett *et al* 2013).

**4.1.3. Generalized Hamiltonian estimation settings.** ML techniques can also be employed in significantly more general settings of quantum process estimation. More general Hamiltonian estimation settings consider a partially controlled evolution given by  $H_C(\theta)$ , where  $C$  is a collection of control parameters of the system. This is a reasonable setting in e.g. the production of quantum devices, which have controls ( $C$ ), but whose actual performance (dependent on  $\theta$ ) needs to be confirmed. Further, since production devices are seldom identical, it is beneficial to even further generalize this setting, by allowing the unknown parameters  $\theta$  to be only probabilistically characterized. More precisely, they are probabilistically dependent on another set of hyperparameters  $\zeta = (\zeta_1, \dots, \zeta_k)$ , such that the parameters  $\theta$  are distributed according to a known conditional probability distribution  $P(\theta|\zeta)$ . This generalized task of estimating the hyperparameters  $\zeta$  thus allows the treatment of systems with inherent stochastic noise, when the influence of noise is understood (given by  $P(\theta|\zeta)$ ). Such very general scenarios are addressed in Granade *et al* (2012), relying on classical learning techniques of Bayesian experimental design (BED) (Loredo 2004), combined with Monte Carlo methods. The details of this method are beyond the scope of this review, but, roughly speaking, BED assumes a Bayesian perspective on the experiments of the type described above. The estimation methods of the general problem (ignoring the hyperparameters and noise, for simplicity, although the same techniques apply) realize a conditional probability distribution  $P(\mathbf{d}|\theta; C)$  where  $\mathbf{d}$  corresponds to experimental data, i.e. measurement outcomes collected in the experiment.

Assuming some prior distribution over hidden parameters ( $P(\theta|C)$ ), the posterior distribution, given experimental outcomes, is given via Bayes' theorem by

$$P(\theta|\mathbf{d}; C) = \frac{P(\mathbf{d}|\theta; C)P(\theta|C)}{P(\mathbf{d}|C)}. \quad (22)$$

The evaluation of the above is already non-trivial, principally because the normalization factor  $P(\mathbf{d}|C)$  includes an integration over the parameter space. Further, of particular interest are scenarios where an experiment is iterated many times. In this case, analogously to the adaptive setting for metrology discussed above, it is beneficial to tune the control parameters  $C$  dependent on the outcomes.

BED (Loredo 2004) tackles such adaptive settings by selecting the subsequent control parameters  $C$  so as to maximize a utility function<sup>62</sup> for each update step. The Bayes updates consist of the computing of  $P(\theta|\mathbf{d}_1, \dots, \mathbf{d}_{l-1}\mathbf{d}_l) \propto P(\mathbf{d}_l|\theta)P(\theta|\mathbf{d}_1, \dots, \mathbf{d}_{l-1})$  at each step. The evaluation of the normalization factor  $P(\mathbf{d}|C)^{-1}$  is, however, also non-trivial, as it includes an integration over the parameter space. In Granade *et al* (2012) this integration is tackled via numerical integration techniques, namely sequential Monte Carlo, yielding a novel technique for robust Hamiltonian estimation.

The robust Hamiltonian estimation method was subsequently expanded to use access to trusted quantum simulators, which forms a more powerful and efficient estimation scheme (Wiebe *et al* 2014b)<sup>63</sup>, which was also shown to be robust to moderate noise and imperfections in the trusted simulators (Wiebe *et al* 2014a). A restricted version of the method of estimation with simulators was experimentally realized in Wang *et al* (2017). More recently, connected to the methods of robust Hamiltonian estimation, Bayesian and sequential Monte Carlo based estimation have further been combined with particle swarm optimization techniques (Stenberg *et al* 2016). There the goal was to achieve reliable coupling strength and frequency estimation in simple decohering systems, corresponding to realistic physical models. More specifically, the studied problem is the estimation of field–atom coupling terms and the mode frequency term in the Jaynes–Cummings model. The controlled parameters are the local field strength and measurements are done via swap spectroscopy.

Aside from using ML to perform partial process tomography of controlled quantum systems, ML can also help in genuine problems of quantum control, specifically, the design of target quantum gates. This forms the next topic.

<sup>61</sup> The non-convexity stems from the fact that the effective input state at each stage depends on previous measurements performed. As the entire interferometer set-up can be viewed as a one-subsystem measurement, the conditional states also depend on unknown parameters and these are used in the subsequent stages of the protocol (Hentschel and Sanders 2010).

<sup>62</sup> The utility function is an object stemming from decision theory and in the case of BED it measures how well the experiment improves our inferences. It is typically defined by the prior–posterior gain of information as measured by the Shannon entropy, although there are other possibilities.

<sup>63</sup> This addition partially circumvents the computation of the *likelihood function*  $P(\mathbf{d}|\theta; C)$ , which requires the simulation of the quantum system and is, in fact, in general intractable.

## 4.2. Design of target evolutions

**Executive summary:** One of the main tasks in quantum information is the design of target quantum evolutions, including quantum gate design. This task can be tackled by quantum control, which studies controlled physical systems where certain parameters can be adjusted during system evolution, or by using extended systems and unmodulated dynamics. Here the underlying problem is an optimization problem, that is, the problem of finding optimal control functions or extended system parameters of a system which is otherwise fully specified. Under realistic constraints these optimization tasks are often non-convex, and so hard for conventional optimizers yet amenable to advanced ML technologies. Target evolution design problems can also be tackled by using feed-back from the actual experimental system, leading to the use of on-line optimization methods and RL.

From a QIP perspective, one of the most important tasks is the design of elementary quantum gates, needed for QC. The paradigmatic approach to this is via quantum control, which aims to identify how control fields of physical systems need to be adapted in time, to achieve desired evolutions. The designing of target evolutions can also be achieved in other settings, e.g. by using larger systems and unmodulated dynamics. In both cases, ML optimization techniques can be used to design optimal strategies off line. However, target evolutions can also be achieved in run-time, by interacting with a tunable physical system, and without the need for the complete description of the system. We first consider off-line settings and briefly comment on the latter on-line settings thereafter.

**4.2.1. Off-line design.** The paradigmatic setting in quantum control considers a Hamiltonian with a controllable (c) and a drift part (dr), e.g.  $H(C(t)) = H_{\text{dr}} + C(t)H_{\text{c}}$ . The free part is modulated via a (real-valued) control field  $C(t)$ . The resulting time-integrated operator  $U = U[C(t)] \propto \exp\left(-i \int_0^T dt H(C(t))\right)$ , over some finite time  $T$ , is a function of the chosen field function  $C(t)$ . The typical goal is to specify the control field  $C(t)$  which maximizes the transition probability from some initial state  $|0\rangle$  to a final state  $|\phi\rangle$ , i.e. to find  $\text{argmax}_C |\langle \phi | U[C(t)] | 0 \rangle|$ <sup>64</sup>. Generically, the mappings  $C(t) \mapsto U[C(t)]$  are highly involved but, nonetheless, empirically it was shown that greedy optimization approaches provide optimal solutions (which is the reason greedy approaches dominate in practice). This empirical observation was later elucidated theoretically (Rabitz *et al* 2004), suggesting that in generic systems local minima do not exist, which leads to easy optimization (see also Russell and Rabitz (2017) for a more up-to-date account). This is good news for experiments, but also suggests that quantum control has no need for advanced ML techniques. However, as is often the case with claims of such generality, the underlying subtle

<sup>64</sup> An example of such additional fields would be controlled laser fields in ion trap experiments, and the field function  $C$  specifies how the laser field strengths are modulated over time.

assumptions are fragile and can often be broken. In particular, greedy algorithms for optimizing the control problem as above can fail, even in the low dimensional case, if we simply place rather reasonable constraints on the control function and parameters. Already for 3-level and 2-qubit systems with constraints on the allowed evolution time  $t$  and the precision of the linearization of the time-dependent control parameters<sup>65</sup>, it is possible to construct examples where greedy approaches fail, yet global (derivative-free) approaches, in particular differential evolution, succeed (Zahedinejad *et al* 2014).

Another example of hard off-line control concerns the design of high fidelity single-shot three-qubit gates<sup>66</sup>, which is in Zahedinejad *et al* (2015) and Zahedinejad *et al* (2016) addressed using a specialized novel optimization algorithm the authors called subspace-selective self-adaptive differential evolution (SuSSADE).

An interesting alternative approach to gate design is by utilizing larger systems. Specifically designed larger systems can naturally implement desired evolutions on a subsystem, without the need for time-dependent control (see QC with always-on interaction (Benjamin and Bose 2003)). In other words, local gates are realized despite the fact that the global dynamics is unmodulated. The non-trivial task of constructing such global dynamics for the Toffoli gate is in Banchi *et al* (2016) tackled by a method which relies upon stochastic gradient descent and draws from supervised learning techniques.

**4.2.2. On-line design.** Complementary to off-line methods, here we assume access to an actual quantum experiment and the identification of optimal strategies relies on on-line feedback. In these cases, the quantum experiment need not be fully specified beforehand. Further, the required methodologies lean towards on-line planning and RL, rather than optimization. In the case where optimization is required, the parameters of optimization are different due to experimental constraints; see Shir *et al* (2012) for an extensive treatment of the topic.

The connections between on-line methods which use feedback from experiments to ‘steer’ systems to desired evolutions have been connected to ML in early works (Bang *et al* 2008, Gammelmark and Imer 2009). These exploratory works deal with generic control problems via experimental feedback and have, especially at the time, remained mostly unnoticed by the community. In more recent times, feedback-based learning and optimization has received more attention. For instance, in Chen *et al* (2014) the authors have explored the applicability of a modified Q-learning algorithm for RL (see section 2.3) on canonical control problems. Further, the potential of RL methods had been discussed in the context of optimal parameter estimation, but also typical optimal control scenarios, in Palittapongarnpim *et al* (2016). In the latter work,

<sup>65</sup> It is assumed that the field function  $C(t)$  describing parameter values as functions of time is step-wise constant, split into  $K$  segments. The larger the value of  $K$  is, the better the approximation of a smooth function is, which would arguably be better suited for greedy approaches.

<sup>66</sup> This includes the Toffoli (and Fredkin) gate which is of particular interest as it forms a universal gate set together with the simple single-qubit Hadamard transform (Shi 2002) (if ancilla qubits are used).

the authors also provide a concise yet extensive overview of related topics and outline a perspective which unifies various aspects of ML and RL in an approach to resolve hard quantum measurement and control problems. In Clausen and Briegel (2016), RL based on PS updates was analyzed in the context of general control-and-feedback problems. Finally, ideas of unified computational platforms for quantum control, albeit without explicit emphasis on ML techniques, had been previously provided in Machnes *et al* (2011).

In the next section, we further coarse-grain our perspective to consider scenarios where ML techniques control various gates and more complex processes, and even help us learn how to do interesting experiments.

#### 4.3. Controlling quantum experiments, and machine-assisted research

**Executive summary:** ML and RL techniques can help us control complex quantum systems, devices and even quantum laboratories. Furthermore, almost as a by-product, they may also help us to learn more about the physical systems and processes studied in an experiment. Examples include adaptive control systems (agents) which learn how to control quantum devices, e.g. how to preserve the memory of a quantum computer, combat noise processes, generate entangled quantum states and target evolutions of interest. Such learning systems may be instrumental in the building of a scalable universal quantum computer. In the process of learning such optimal behaviors even relatively simple artificial agents may also learn, in an implicit, embodied sense, about the underlying physics, which can be used by us to obtain novel insights. In other words artificial learning agents can genuinely help us do research.

The prospects for utilizing ML and AI in quantum experiments have also been investigated for ‘higher-level’ experimental design problems. Here one considers automated machines that control complex processes which e.g. specify the execution of longer sequences of simple gates or the execution of quantum computations. Moreover, it has been suggested that learning machines can be used for, and integrated into, the very design of quantum experiments, thereby helping us in conducting genuine research. We first present two results where ML and RL methods have been utilized to control more complex processes (e.g. generate sequences of quantum gates to preserve memory) and consider the perspectives of machines genuinely helping in research thereafter.

**4.3.1. Controlling complex processes.** The simplest example of involved ML machinery being used to generate control of slightly more complex systems was done in the context of the problem of *dynamical decoupling* for quantum memories. In this scenario, a quantum memory is modeled as a system coupled to a bath (with a local Hamiltonian for the system ( $H_S$ ) and the bath ( $H_B$ ) and decoherence is realized by a coupling term  $H_{SB}$ ; the local unitary errors are captured by  $H_S$ . The evolution

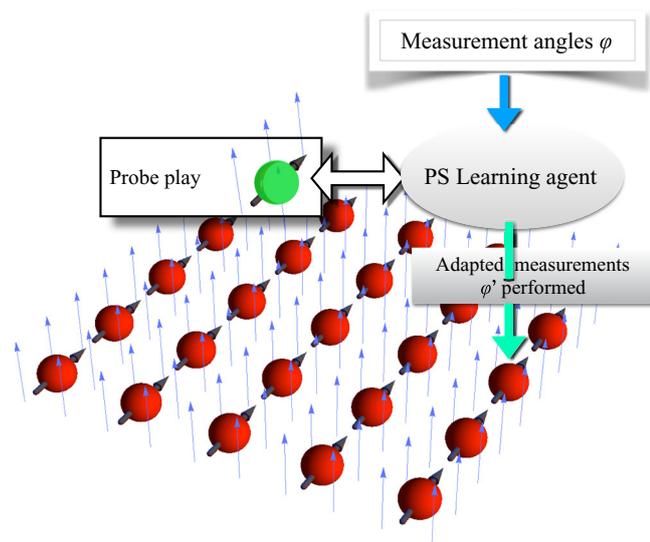
of the total Hamiltonian  $H_{\text{noise}} = H_S + H_B + H_{SB}$  would destroy the contents of the memory, but this can be mitigated by adding a controllable local term  $H_C$  acting on the system alone<sup>67</sup>. Certain optimal choices of the control Hamiltonian  $H_C$  are known. For instance, we can consider the scenario where  $H_C$  is modulated so that it implements instantaneous<sup>68</sup> Pauli- $X$  and Pauli- $Y$  unitary operations, sequentially, at intervals  $\Delta t$ . As this interval, which is also the time of the decoherence-causing free evolution, approaches zero, so  $\Delta t \rightarrow 0$ , this process is known to ensure perfect memory. However, the moment the setting is made more realistic, allowing finite  $\Delta t$  times, the space of optimal sequences becomes complicated. In particular, optimal sequences start depending on  $\Delta t$ , the form of the noise Hamiltonian and total evolution time.

To identify optimal sequences, in August and Ni (2017), the authors employ a recurrent NN, specifically the LSTM (see section 2.1.1 for details), which is trained to generate sequences which minimize final noise. The entire sequences of pulses (Pauli gates) which the networks generated were shown to outperform well-known sequences.

In a substantially different setting, where interaction necessarily arises, the authors studied how AI/ML techniques can be used to make quantum protocols themselves adaptive. Specifically, the authors applied RL methods based on projective simulation (PS) (Briegel and De las Cuevas 2012) (see section 7.1) to the task of protecting QC from local stray fields (Tiersch *et al* 2015). In MBQC (Raussendorf and Briegel 2001, Briegel *et al* 2009), the computation is driven by performing adaptive single-qubit projective measurements on a large entangled resource state, such as the cluster state (Briegel and Raussendorf 2001, Raussendorf and Briegel 2001). In a scenario where the resource state is exposed to a stray field, each qubit undergoes a local rotation. To mitigate this, in Tiersch *et al* (2015), the authors introduce a learning agent which ‘plays’ with a local probe qubit, initialized in, say, the  $+1$  eigenstate of  $\sigma_x$ , denoted by  $|+\rangle$ , learning how to compensate for the unknown field. In essence, given a measurement (preparation), the agent chooses the next measurement direction (effectively guessing its evolution), obtaining a reward whenever a  $+1$  outcome is observed. The agent is thus trained to compensate for the unknown field, and serves as an ‘interpreter’ between desired measurements and the measurements which should be performed in the given setting (i.e. in the given field with given frequency of measurements ( $\Delta t$ )); see figure 11. The problem of mitigating such fixed stray fields could naturally be solved with non-adaptive methods where we use knowledge about the system to solve our problem, by e.g. measuring the field and adapting accordingly, or by using fault-tolerant constructions. From a learning perspective, such direct methods have a few shortcomings which may be worth presenting for didactic purposes. Fault tolerant methods are clearly wasteful, as they fail to gain or utilize any knowledge

<sup>67</sup> For the sake of intuition, a frequent application of  $X$  gates, referred to as bang-bang control, on a system which is freely evolving with respect to  $\sigma_z$  effectively flips the direction of rotation of the system Hamiltonian, effectively undoing its action.

<sup>68</sup> By ‘instantaneous’ we mean that it is assumed that the implementation requires no evolution time, e.g. by using infinite field strengths.



**Figure 11.** The learning agent learns how to correctly perform MBQC measurements in an unknown field.

about the noise processes. In contrast, field estimation methods learn too much, and assume a model of the world. In particular, to compensate for the measured field we need to use quantum mechanics, specifically the Born rule. In contrast, the RL approach is model-free: the Born rule plays no part, and ‘correct behavior’ is learned and established exclusively based on experience. This is conceptually different, but also operatively critical, as model-free approaches allow for more autonomy and flexibility (i.e. the same machinery can be used in more settings without intervention)<sup>69</sup>. Regarding learning too much, one of the basic principles of statistical learning posits that ‘when solving a problem of interest, one should not solve a more general problem as an intermediate step’ (Vapnik 1995), which is intuitive. The problem of the presented setting is ‘how to adapt the measurement settings’, not ‘characterize the stray fields’. While in the present context the information-theoretic content of the two questions may be the same, it should be easy to imagine that if more complex fields are considered full process characterization contains a lot more information than needed to optimally adapt the local measurements. The approaches of Tiersch *et al* (2015) can further be generalized to utilize information from stabilizer measurements (Orsucci *et al* 2016) or, similarly, outcomes of syndrome measurements when codes are utilized (Combes *et al* 2014) (instead of probe states) to similar ends. Addressing somewhat related problems, but using supervised learning methods, the authors in Mavadia *et al* (2017) have also shown how to compensate for qubit decoherence (stochastic evolution) in experiments. Related to the topics of automated optimization of (complex) quantum protocols, in recent works generic optimization algorithms have been exploited to optimize and analyze QKD-type cryptographic protocols in the presence of noise (Krawec 2016, Krawec *et al* 2017). Further,

<sup>69</sup> Indeed, the authors also show that correct behavior can be established when additional unknown parameters are introduced, like time-and-space dependent fields (see Tiersch *et al* (2015) for results), where hand-crafted methods would fail.

in Wigley *et al* (2016), the authors demonstrated how on-line ML optimization can be used to find optimal evaporation ramps for Bose–Einstein condensates production.

Finally, we point out that one of the most exciting applications of ML in the context of quantum experiments is the possibility that advanced ML techniques could significantly mitigate the major obstacles preventing the building of scalable quantum computers. This question can be addressed at various levels. For instance, the aforementioned control problems, relying on adaptive methods for the robust implementations of unitary gates, comprise a ‘lower-level’ aspect. The results of Tiersch *et al* (2015) and August and Ni (2017) also contribute to the overall goal of building large-scale robust quantum computers, but at a higher level—the action space of the agent already assumes access to unitary gates and measurements as primitives. In line with the same program, ML techniques can also help on an even further abstracted level, in helping with hard classical post-processing which may occur in the run-time of a quantum computer. Prominent examples here include ML efforts to optimize the decoders for quantum error correction codes. In this vital area, researchers have explored various ML techniques for a spectrum of issues. For instance, in the pioneering work in Torlai and Melko (2017), the potential of using restricted Boltzmann machines (BMs) (see section 2.1.1) to output near-optimal decoding/correcting strategies based on syndrome outcomes was explored. In follow up works, Varsamopoulos *et al* (2018) and Krastanov and Jiang (2017), feed-forward shallow and deep NN architectures were employed to speed up the decoding process or to achieve performance-beating standard decoding algorithms, respectively. In Baireuther *et al* (2017), the authors proposed an adaptive decoder based on LSTMs (see section 2.1.1). The proposed architecture accounted not only for temporally correlated errors, but allowed the training to be performed based on experimental data alone, i.e. without a reference to an error model. Similar problems have also been considered in the settings of restricted devices. For instance, in Johnson *et al* (2017), the authors have explored device-specific error-correcting encoding and decoding circuits in a variational setting. Here, parametrized circuits of a fixed structure are optimized to perform error correction, with no reference to a particular error correcting code. Such approaches are particularly appealing for near-term devices which can handle only a modest number of qubits and operations of only a limited computational depth. Such variational approaches are similar to how various ML techniques are employed (in particular, they are conceptually close to classical autoencoders; see section 2.1.1). Further, in this setting, the correct solutions may involve very hard optimization steps which form yet another class of entry points for ML methodology to be employed<sup>70</sup>. As the field advances, we are expecting to see a significant increase in the number of applications of ML for the purpose of building large-scale quantum devices.

<sup>70</sup> As highlighted in section 1.1, a similar variational philosophy is also underpinning some of the newer approaches to quantum algorithms with shallow and specialized architectures, see, e.g. Farhi *et al* (2014), Peruzzo *et al* (2014) and McClean *et al* (2016).

**4.3.2. Learning how to experiment.** One of the first examples of applications of RL in QIP appears in the context of experimental photonics, where one of the current challenges lies in the generation of highly entangled, high dimensional, multi-party states. Such states are generated on optical tables, the systematic configuration of which, to generate complex quantum states, is still not fully understood. The search for configurations which are interesting can be mapped to an RL problem, where a learning agent is rewarded whenever it generates an interesting state (in a simulation). In a precursor work by Krenn *et al* (2016), the authors used a feedback-assisted search algorithm to identify previously unknown configurations which generate novel highly entangled states. This demonstrated that the design of novel quantum experiments can also be automated, which can significantly aid in research. This idea, given in the context of optical tables, has subsequently been combined with earlier proposals to employ AI agents in quantum information protocols and as ‘lab robots’ in future quantum laboratories (Briegel 2013). This led to the application of more advanced RL techniques, based on the PS framework, for the tasks of understanding the Hilbert space accessible with optical tables and the autonomous machine-discovery of useful optical gadgets (Melnikov *et al* 2017). Related to the topic of learning new insight from experimenting machines, in Bukov *et al* (2017) the authors consider the problem of preparing target states by means of chosen pulses implementing (a restricted set) of rotations. This is a standard control task, and the authors show that RL achieves respectable and sometimes near-optimal results. However, for our purposes, the most relevant aspects of this work pertain to the fact that the authors also illustrate how ML/RL techniques can be used to obtain new insights in quantum experiments and non-equilibrium physics by circumventing human intuition, which can be flawed. Interestingly, the authors also demonstrate the reverse, i.e. how physics insights can help elucidate learning problems<sup>71</sup>.

#### 4.4. Machine learning in condensed-matter and many-body physics

**Executive summary:** One of the quintessential problems of many-body physics is the identification of phases of matter. A popular overlap between ML and this branch of physics demonstrates that supervised and unsupervised systems can be trained to classify different phases. More interestingly, unsupervised learning can be used to detect phases and even discover order parameters, possibly genuinely leading to novel physical insights. Another important overlap considers the representational power of (generalized) NNs, to characterize interesting families of quantum systems. Both suggest a deeper link between certain learning models, on the one hand, and physical systems, on the other, the scope of which is currently an important research topic.

<sup>71</sup> For instance, the authors investigate the strategies explored by the learning agent, and identify spin-glass-like phase transition in the space of protocols as a function of the protocol duration. This highlights the difficulty of the learning problem.

ML techniques have, over the course of the last 20 years, become an indispensable tool set for many natural sciences which deal with highly complex systems. These include biology (specifically genetics, genomics, proteomics and the general field of computational biology) (Libbrecht and Noble 2015), medicine (e.g. in epidemiology, disease development, etc) (Cleophas and Zwinderman 2015), chemistry (Cartwright 2007) and high-energy and particle physics (Castelvecchi 2015). Unsurprisingly, they have also permeated various aspects of condensed-matter and many-body physics. Early examples of this were proposed in the context of quantum chemistry and density functional theory (Curtarolo *et al* 2003, Snyder *et al* 2012, Rupp *et al* 2012, Li *et al* 2015a), or for the approximation of the Green’s function of the single-site Anderson impurity model (Arsenault *et al* 2014). The interest in connections between NNs and many-body and condensed-matter physics has undergone immense growth since. Some of the results which we cover next deviate from the primary topic of this review, those concerning the overlaps of QIP and ML. However, since QIP and condensed-matter and many-body physics share significant overlaps we feel it is important to at least briefly flesh out the basic ideas. We focus on two of the most fertile research directions in this field of study to emerge in recent times. The first is the problem of learning phases of matter and the detection of phase transitions in physical systems. The second deals with the (surprisingly large) capacities of generative models to capture relevant parts of the Hilbert space.

**Learning phases.** A canonical example is the discrimination of samples of configurations stemming from different phases of matter, e.g. Ising model configurations of thermal states below or above the critical temperature. This problem has been tackled using principal component analysis and nearest neighbor unsupervised learning techniques (Wang 2016) (see also Hu *et al* (2017)). Such methods also have the potential to, beyond just detecting phases, actually identify order parameters (Wang 2016)—in the above case, magnetization. More complicated discrimination problems, e.g. discriminating Coulomb phases, have been resolved using basic feed-forward networks, and CNNs were trained to detect topological phases (Carrasquilla and Melko 2017) as well as phases in fermionic systems on cubic lattices (Ch’ng *et al* 2016). NNs have also been combined with quantum Monte Carlo methods (Broecker *et al* 2016) and with unsupervised methods (van Nieuwenburg *et al* 2017, Ch’ng *et al* 2018) (applied also in Wang (2016)), in both cases to improve classification performance in various systems. It is notable that all these methods prove quite successful in ‘learning’ phases, without any information about the system Hamiltonian. While the focus in this field had mostly been on NN architectures, other supervised methods, specifically kernel methods (e.g. SVMs), had been used for the same purpose (Ponte and Melko 2017)<sup>72</sup>. The large number

<sup>72</sup> Kernel methods may be in some cases advantageous as they can have a higher interpretability: it is often easier to understand the reason behind the optimal model in the cases of kernel methods than with NNs, which also means that learning about the underlying physics may be easier in the cases of kernel methods. However, as clarified in section 2.1.1, the question of what interpretability means and which systems are more likely to be forced to ensure it remains a matter of ongoing debate.

of works dedicated to this problem have in recent times combined various ML methods with various supervised and unsupervised variants of the problem of learning phases of matter. This includes supervised methods, where machines are trained to discriminate phases given labeled examples, and also unsupervised methods (briefly summarized above). It is not difficult to argue that unsupervised approaches may have a greater appeal, as they allow us not only to study systems where phases are quite well understood, but also to recover a fully unknown phase diagram. However, in contrast to pure supervised settings, the unsupervised solutions often utilize hand-crafted pre-processing of data and other types of prior knowledge (Huembeli *et al* 2017). One of the emerging challenges is thus to achieve fully unsupervised extraction of the phase-space diagram, ideally relying only on data which is essentially raw, in the sense that it has not been pre-processed using hand-crafted procedures or extensive prior knowledge. Recent works (Broecker *et al* 2017 and Morningstar and Melko 2017) address this challenge using modified CNNs and deep BMs, respectively. In Huembeli *et al* (2017), the authors utilize cutting-edge techniques from ML, namely domain adversarial NN (DANN) training (see section 2.1.1), to achieve a type of transfer learning—the network is trained, in part, on the part of the parameter space where phases are well-understood (and thus can be labeled) and this knowledge is then applied on the unknown section of the parameter space. The key idea behind DANNs is that the network is trained in such a fashion that the final classifier is forced to be invariant under change of the domain—in other words it cannot distinguish between the data stemming from the two sections of the parameter space. This ensures that only the features specifying the phases influence the classification, and thus ensures correct generalization.

A partial explanation behind the success of neuronal approaches for classifying phases of matter may lie in their form. Specifically, they may have the capacity to encode important properties of physical systems both in the classical and in the quantum case. This motivates the second line of research we mentioned.

*Representational capacities of generative models.* The research in this area is motivated by a simple example. BMs, even in their restricted variant, are known to have the capacity to encode complicated distributions. In the same sense, restricted BMs, extended to accept complex weights (i.e. the weights  $w_{ij}$  in equations (2) and (3)) encode quantum states and the hidden layer captures correlations, both classical and quantum (entanglement). In Carleo and Troyer (2017) it was shown that this approach describes equilibrium and dynamical properties of many prototypical systems accurately: that is, restricted BMs form a useful ansatz for interesting quantum states (called *neural-network quantum states* (NQSs)), where the number of neurons in the hidden layer controls the size of the representable subset of the Hilbert space. This is analogous to how, for instance, the bond dimension controls the scope of the matrix product state ansatz (Verstraete *et al* 2008). This property can also be exploited in order to achieve

efficient quantum state tomography<sup>73</sup> (Torlai *et al* 2017). In subsequent works, the authors have also analyzed the structure of entanglement of NQSs (Deng *et al* 2017) and have provided analytic proofs of the representation power of deep restricted BMs, proving they can e.g. represent ground states of any  $k$ -local Hamiltonians with polynomial-size gaps (Gao and Duan 2017). It is worthwhile to note that the representational power of standard variational representations (e.g. that of the variational renormalization group) had previously been contrasted to those of deep NNs (Mehta and Schwab 2014), with the goal of elucidating the success of deep networks. Related to this, the tensor network (Östlund and Rommer 1995, Verstraete and Cirac 2004) formalism has been used for the efficient description of deep convolutional arithmetic circuits, establishing also a formal connection between quantum many-body states and deep learning (Levine *et al* 2017). In related research, in Glasser *et al* (2018), the authors have established a strong connection between quantum states specified by restricted BMs and classes of tensor-network states: short-range restricted BMs<sup>74</sup> correspond to a class of so-called plaquette states, whereas standard restricted BMs (with no locality constraints) capture the class of so-called string-bond states with a non-local geometry and low bond dimension. The authors also utilize these results to generalize standard NQSs to exactly describe certain lattice fractional quantum Hall states and to well-approximate chiral spin liquid. Such results nicely exemplify the types of advantages we may hope to obtain by combining ML insights with many-body physics.

Very recently, the intersections between ML and many-body quantum physics have also inspired research into ML-motivated entanglement witnesses and classifiers (Ma and Yung 2017, Lu *et al* 2017) and also into furthering the connections between ML and many-body physics, specifically, entanglement theory. These recent results have positioned NNs as one of the most exciting new techniques to be applied in the context of both condensed-matter and many-body physics. In addition, they also show the potential of the converse direction of influence—the application of the mathematical formalism of many-body physics to deepen of our understanding of complex learning models.

## 5. Quantum generalizations of machine learning concepts

The onset of quantum theory necessitated a change in how we describe physical systems, but also a change in our understanding of what information is<sup>75</sup>. Quantum information is a more general concept, and QIP exploits the genuine quantum features for more efficient processing (using quantum computers) and more efficient communication. Such quintessential quantum properties, such as the fact that even pure states cannot be perfectly copied (Wootters and Zurek 1982), are often

<sup>73</sup> This method can be thought of as effectively assigning a prior stating that the analyzed state is well approximated by an NQS.

<sup>74</sup> It should be noted that similar locality restrictions also appear in the structure of CNNs.

<sup>75</sup> Arguably, in the light of the physicalistic viewpoint on the nature of information, which posits that ‘Information is [ultimately] physical’.

argued to be at the heart of many quantum applications, such as cryptography. Similarly, quintessential information processing operations are more general in the quantum world: closed quantum systems can undergo arbitrary unitary evolutions, whereas the corresponding classical closed-system evolutions correspond to the (finite) group of permutations<sup>76</sup>. The majority of ML literature deals with learning from, and about, data—that is, classical information. This section examines the question of what ML looks like when the data (and perhaps its processing) is fundamentally quantum. We will first explore quantum generalizations of supervised learning, where the ‘data points’ are now genuine quantum states. This generates a plethora of scenarios which are indistinguishable in the classical case (e.g. having one or two copies of the same example is not the same!). Next, we will consider another quantum generalization of learning, where quantum states are used to represent the generalizations of unknown concepts in computational learning theory (COLT)—thus we talk about the learning of quantum states. Following this we will present some results on quantum generalizations of POMDPs which could lead to quantum-generalized RL (although this actually just generalizes the mathematical structure).

### 5.1. Quantum generalizations: machine learning of quantum data

**Executive summary:** A significant fraction of the field of ML deals with data analysis, classification, clustering, etc. QIP generalizes standard notions of data to include quantum states. The processing of quantum information comes with restrictions (e.g. no-cloning or no-deleting), but also new processing options. This section addresses the question of how conventional ML concepts can be extended to the quantum domain, mostly focusing on aspects of supervised learning and learnability of quantum systems but also on concepts underlying RL.

One of the basic problems of ML is that of supervised learning, where a training set  $D = \{(\mathbf{x}_i, y_i)\}_i$  is used to infer a labeling rule mapping data points to labels  $\mathbf{x}_i \xrightarrow{\text{rule}} y_i$  (see section 1.2 for more details). More generally, supervised learning deals with the classification of classical data. In the tradition of QIP, data can also be quantum—that is, all quantum states carry, or rather represent, (quantum) information. What can be done with datasets of the type  $\{(\rho_i, y_i)\}_i$ , where  $\rho_i$  is a quantum state? Colloquially it is often said that one of the critical distinctions between classical and quantum data is that quantum data cannot be copied. In other words, having one instance of an example, by abuse of notation denoted  $(\rho_i \otimes y_i)$ , is not generally as useful as having two copies  $(\rho_i \otimes y_i)^{\otimes 2}$ . In contrast, in the case of classification with functional labeling rules, this is the same. The closest classical analog of dealing

with quantum data is the case where labelings are not deterministic or, equivalently, where the conditional distribution  $P(\text{label}|\text{datapoint})$  is not extremal (Dirac). This is the case of classification (or learning) of random variables, or probabilistic concepts, where the task is to produce the best guess label, specifying the random process which ‘most likely’ produced the data point<sup>77</sup>. In this case, having access to two examples in the training phase which are independently sampled from the same distribution is not the same as having two copies of one and the same individual sample—these are perfectly correlated and carry no new information<sup>78</sup>. To obtain full information about a distribution, or random variable, one in principle needs infinitely many samples. Similarly, in the quantum case, having infinitely many copies of the same quantum state  $\rho$  is operatively equivalent to having a classical description of the given state.

Despite similarities, quantum information is still different from mere stochastic data. The precursors of ML-type classification tasks can be identified in the theories of quantum state discrimination, which we briefly comment on first. Next, we review some early works dealing with ‘quantum pattern matching’, which spans various generalizations of supervised settings, and first works which explicitly propose the study of *quantum-generalized ML*. Next, we discuss more general results, which characterize inductive learning in quantum settings. Finally, we present a COLT perspective on learning with quantum data, which addresses the learnability of quantum states.

#### 5.1.1. State discrimination, state classification and machine learning of quantum data.

**State discrimination.** The entry point to this topic can again be traced to seminal works of Helstrom (1969) and Holevo (1982), as the problems of state discrimination can be rephrased as variants of supervised learning problems. In typical state discrimination settings, the task is the identification of a given quantum state (given as an instance of a quantum system prepared in that state), under the promise that it belongs to a (typically finite) set  $\{\rho_i\}_i$ , where the set is fully classically specified. Recall that state estimation, in contrast, typically assumes continuous parametrized families, and the task is the estimation of the parameter. In this sense, discrimination is a discretized estimation problem<sup>79</sup>, and the problems of identifying optimal measurements (under various figures of merit) and success bounds have been considered extensively and continuously throughout the history of QIP (Helstrom 1969, Croke *et al* 2008, Slussarenko *et al* 2017).

**Remark.** Traditional quantum state discrimination can be rephrased as degenerate supervised learning setting for quantum states. Here, the space of ‘data points’ is restrict-

<sup>77</sup> Note that in this setting we do not have the descriptions of the stochastic processes *a priori*—they are to be inferred from the training examples.

<sup>78</sup> In this sense, the no-cloning theorem also applies to classical information: an unknown random variable cannot be cloned. In QIP language this simply means that the no-cloning theorem applies to diagonal density matrices, i.e.  $\rho \not\rightarrow \rho \otimes \rho$ , even when  $\rho$  is promised to be diagonal.

<sup>79</sup> Intuitively, estimation is to discrimination what regression is to classification in the ML world.

<sup>76</sup> Classical evolutions are guaranteed to transform computational basis states (the ‘classical states’) to computational basis states, and closed-system implies the dynamics must be reversible, leaving only permutations.

ed to a finite (or parametrized) family  $\{\rho_i\}_i$ , and the training set contains an effectively infinite number of examples  $D = \{(\rho_i, i)^{\otimes \infty}\}$ ; naturally, this notation is just a short-hand for having the complete classical description of the quantum states<sup>80</sup>. In what follows we will sometimes write  $\rho^{\otimes \infty}$  to denote a quantum system containing the classical description of the density matrix  $\rho$ .

**Quantum template matching—classical templates.** A variant of discrimination, or class assignment task, which is one of the first instances of works which establish explicit connections with ML and discrimination-type problems, is ‘template matching’ (Sasaki *et al* 2001). In this pioneering work, the authors consider discrimination problems where the input states  $\psi$  may not correspond to the (known) *template states*  $\{\rho_i\}_i$  and the correct matching label is determined by the largest Uhlmann fidelity. More precisely, the task is defined as follows: given a classically specified family of template states  $\{\rho_i\}_i$  and given  $M$  copies of a quantum input  $\psi^{\otimes M}$ , output the label  $i_{\text{corr}}$  defined by  $i_{\text{corr}} = \text{argmax}_i \text{Tr} \left[ \sqrt{\sqrt{\psi} \rho_i \sqrt{\psi}} \right]^2$ . In this original work, the authors focus on two-class cases, with pure state inputs, and identify fully quantum and semi-classical strategies for this problem. ‘Fully quantum strategies’ identify the optimal POVM. Semi-classical strategies impose a restriction of measurement strategies to separable measurements, or perform state estimation on the input, a type of ‘quantum feature extraction’.

**Quantum template matching—quantum templates.** In a generalization of the work in Sasaki *et al* (2001), the authors in Sasaki and Carlini (2002) consider the case where instead of having access to the classical descriptions of the template states  $\{\rho_i\}_i$ , we are given access to a certain number of copies,  $K$ . In other words, we are given access to a quantum system in the state  $\bigotimes_i \rho_i^{\otimes K}$ . Setting  $K \rightarrow \infty$  recovers the case with classical templates. This generalized setting introduces many complications, which do not exist in the ‘more classical’ case with classical templates. For instance, classifying measurements now must ‘use up’ copies of template states, as they too cannot be cloned. The authors identify various flavors of semi-classical strategies for this problem. For instance, if the template states are first estimated, we are facing the scenario of classical templates (albeit with error). The classical template setting itself allows semi-classical strategies, where all systems are first estimated, and it allows coherent strategies. The authors find optimal solutions for  $K = 1$  and show that there exists a fully quantum procedure that is strictly superior to straightforward semi-classical extensions.

**Remark.** Quantum template matching problems can be understood as quantum-generalized supervised learning, where the training set is of the form  $\{(\rho_i^{\otimes K}, i)_i\}$ , data beyond the

<sup>80</sup> From an operative and information content perspective, having infinitely many copies is equivalent to having a full classical description: infinite copies are sufficient and necessary for perfect tomography—yielding the exact classical description—whereas having an exact classical description is sufficient and necessary for generating an unbounded copy number.

training set comes from the family  $\{\psi^{\otimes M}\}$  (number of copies is known) and the classes are defined via minimal distance, as measured by the Uhlmann fidelity. The case  $K \rightarrow \infty$  approaches the special case of classical templates. Restricting the states  $\psi$  to the set of template states (restricted template matching) and setting  $M = 1$  recovers standard state discrimination.

**Other known optimality results for (restricted) template matching.** For the restricted matching case, where the input is promised to be from the template set, the optimal solutions for the two-class setting, minimum error figure of merit and uniform priors of inputs have been found in Bergou and Hillery (2005) and Hayashi *et al* (2005) for the qubit case. In Hayashi *et al* (2006) the authors found optimal solutions for the unambiguous discrimination case<sup>81</sup>. An asymptotically optimal strategy in the restricted matching case with finite templates  $K < \infty$  for arbitrary priors and mixed qubit states was later found in Guță and Kotłowski (2010). This work also provides a solid introduction to the topic, a review of quantum analogies for statistical learning and emphasizes connections to ML methodologies and concepts.

Later, in Sentís *et al* (2012), the authors introduced and compared all three strategies—classical estimate-and-discriminate, classical optimal and quantum strategy—for the restricted template matching case with finite templates. Recall that the adjective ‘classical’ here denotes that the training states are fully measured out as the first step—the quantum set is converted to *classical information*, meaning that no quantum memory is further required—and that the learning can be truly inductive. A surprising result is that the intuitive estimate-and-discriminate strategy, which reduces supervised classification to optimal estimation coupled with a (standard) quantum state discrimination problem, is not optimal for learning. Another measurement provides not only better performance, but matches the optimal quantum strategy exactly (as opposed to asymptotically). Interestingly, the results of Guță and Kotłowski (2010) and Sentís *et al* (2012) make opposite claims for *essentially* the same setting: no separation versus separation between coherent (fully quantum) and semi-classical strategies, respectively. This discrepancy is caused by differences in the chosen figures of merit and a different definition of asymptotic optimality (Sentís 2017) and serves as an effective reminder of the subtle nature of quantum learning. Optimal strategies have been subsequently explored in other settings as well, e.g. when the dataset comprises coherent states (Sentís *et al* 2015) and in the cases where an error margin is in an otherwise unambiguous setting (Sentís *et al* 2013).

**Quantum generalizations of (un)supervised learning.** The works of the previous paragraph consider particular families of generalizations of supervised learning problems. The first attempts to classify and characterize what ML could look like

<sup>81</sup> In unambiguous discrimination, the device is allowed to output an ambiguous ‘I do not know’ outcome, but is not allowed to err in the case it does output an outcome. The goal is to minimize the probability of the ambiguous outcome.

in a quantum world from a more general perspective were, however, first explicitly carried out in Aïmeur *et al* (2006). There, the basic object introduced is the database of labeled quantum or classical objects, i.e.  $D_n^K = \{(|\psi_i\rangle^{\otimes K}, y_i)\}_{i=1}^n$ <sup>82</sup>, which may come in copies. Such a database can, in general, then be processed to solve various types of tasks, using classical or quantum processing. The authors propose to characterize quantum learning scenarios in terms of classes, denoted  $L_{\text{goal}}^{\text{context}}$ . Here *context* may denote whether we are dealing with classical or quantum data and whether the learning algorithm is relying on quantum capabilities or not. The *goal* specifies the learning task or goal (perhaps in very broad terms). Examples include  $L_c^c$ , which corresponds to standard classical ML, and  $L_c^q$ , which could mean we use a quantum computer to analyze classical data. The example of template matching classical templates ( $K = \infty$ ) (Sasaki *et al* 2001) considered earlier in this section would be denoted by  $L_q^c$ , and the generalization with finite template numbers  $K < \infty$  would fit in  $L_q^{\otimes K}$ . While the formalism above suggests a focus on supervised settings, the authors also suggest that datasets could be inputs for (unsupervised) clustering. The authors further study quantum algorithms for determining closeness of quantum states<sup>83</sup>, which could be the basic building block of quantum clustering algorithms, and also compute certain error bounds for special cases of classification (state discrimination) using well known results of Helstrom (1969). Similar ideas were used in Lu and Braunstein (2014) for the purpose of defining a quantum decision tree algorithm for data classification in the quantum regime.

The strong connection between quantum-generalized learning theory sketched out in Aïmeur *et al* (2006) and the classical<sup>84</sup> theory of Helstrom (1969) was more deeply explored in Gambs (2008). There, the author computed the lower bounds of sample complexity—in this case the minimal number of copies  $K$ —needed to solve a few types of classification problems. For this purpose the author introduced a few techniques which reduce ML-type classification problems to the settings where the theory (Helstrom 1969) of could be directly applied. These types of results contribute to establishing a deeper connection between the problems of ML and the techniques of QIP.

*Quantum inductive learning.* Recall that inductive, eager learning produces a best guess classifier which can be applied to the entire domain of data points, based on the training set. But, the results of Sasaki and Carlini (2002) discussed in the paragraph on template matching with quantum templates already point to problems with this concept in the quantum realm—the optimal classifier may require a copy of the quantum data points to perform classification, which seemingly prohibits unlimited use. The perspectives of such quantum generalizations of supervised learning in its inductive form were recently

addressed from a broad perspective (Monràs *et al* 2017). Recall that inductive learning algorithms, intuitively, use only the training set to specify a hypothesis (the estimation of the true labeling function). In contrast, in *transductive* learning, the learner is also given the data points for which the labels are unknown. These unlabeled points may correspond to the cross-validation test set or the actual target data. Even though the labels are unknown, they carry additional information from the complete dataset which can be helpful in identifying the correct labeling rule<sup>85</sup>. Another distinction is that transductive algorithms need only label the given points, whereas inductive algorithms need to specify a classifier, i.e. a labeling function, defined on the entire space of possible points. In Monràs *et al* (2017), the authors notice that the property of an algorithm being inductive corresponds to a non-signaling property<sup>86</sup>, which they can use to prove that ‘being inductive’ (i.e. being ‘no signaling’) is equivalent to having an algorithm which outputs a classifier  $h$  based on the training set alone, which is then applied to every training instance. A third equivalent characterization of inductive learning is that the training and testing cleanly separate as phases. While these observations are quite intuitive in the classical case, they are in fact problematic in the quantum world. Specifically, if the training examples are quantum objects, quantum no-cloning, in general, prohibits the applying of a hypothesis function (candidate labeling function)  $h$  arbitrarily many times. This is easy to see since each instance of  $h$  must depend on the quantum data in *some non-trivial way*, if we are dealing with a learning algorithm. Multiple copies of  $h$  would then require multiple copies of (at least parts of) the quantum data.

A possible implication of this would be that, in the quantum realm, inductive learning cannot be cleanly separated into training and testing. Nonetheless, the authors show that the no-signaling criterion, for certain symmetric measures of performance, implies that a separation is asymptotically possible. Specifically, the authors show that for any quantum inductive no-signaling algorithm  $A$  there exists another, perhaps different, algorithm  $A'$ , which does separate in a training and testing phase and which, asymptotically, attains the same performance (Monràs *et al* 2017). Such a protocol  $A'$ , essentially, utilizes a semi-classical strategy. In other words, for inductive settings, classical intuition survives despite the no-cloning theorems.

**5.1.2. Computational learning perspectives: quantum states as concepts.** The previous subsections addressed the topics of classification of quantum states, based on quantum database examples. The overall theory, however, relies on the assumption that there exists a labeling rule which generates such examples and the labeling rule is what is learned. This rule is also known as a concept in COLT (e.g. PAC

<sup>82</sup> Such a dataset can be stored in, or instantiated by, a  $2-n$  partite quantum system, prepared in the state  $\bigotimes_{i=1}^n |\psi_i\rangle^{\otimes K_i} |y_i\rangle$ .

<sup>83</sup> These are based on the SWAP-test (see section 6.3.2), in terms of Uhlmann fidelity

<sup>84</sup> Here we mean classical in the sense of ‘being a classic’, rather than pertaining to classical systems.

<sup>85</sup> For instance, a transductive algorithm may use unsupervised clustering techniques to assign labels, as the whole set is given in advance.

<sup>86</sup> The outcome of the entire learning and evaluation process can be viewed as a probability distribution  $P(\mathbf{y}) = P(y_1 \dots y_k | x_1 \dots x_k; A)$ , where  $A$  is the training set,  $x_1, \dots, x_k$  are the points of the test state and  $y_1 \dots y_k$  the respective labels the algorithm assigns with the probability  $P(\mathbf{y})$ . No signaling implies that the marginal distribution for the  $k$ th test element  $P(y_k)$  only depends on  $x_k$  and the training set, but not on the other test points  $\{x_i\}_{i \neq k}$ .

learning; see section 2.2.1 for details). A reasonable sufficient criterion is whether one can predict the probabilities of the outcomes of any two-outcome measurements on this state, as this already suffices for a full tomographic reconstruction. What would ‘the learning of quantum states’ mean, from this perspective? What does it mean to ‘know a quantum state’? A natural criterion is that one ‘knows’ a quantum state if one can predict the measurement outcome probabilities of any given measurement. In Aaronson (2007), the author addressed the question of the learnability of quantum states in the sense above, where the role of a concept is played by a given quantum state, and ‘knowing’ the concept then equates to the possibility of predicting the outcome probability of a given measurement and its outcome. One immediate distinction from conventional COLT, discussed in section 2.2.1, is that the concept range is no longer binary. However, as we clarified, classical COLT has generalizations with continuous ranges. In particular, so called *p*-concepts have range in  $[0, 1]$  (Kearns and Schapire 1994), and quantities which are analogs of the VC dimension and analogous theorems relating this to generalization performance exist for the *p*-concept case as well (see Aaronson (2007)). Explicitly, the basic elements of such a generalized theory are: domain of concepts  $\mathbf{X}$ , a sample  $\mathbf{x} \in \mathbf{X}$  and the *p*-concept  $f : \mathbf{X} \rightarrow [0, 1]$ . These abstract objects are mapped to central objects of quantum information theory (Aaronson 2007) as follows: the domain of concepts is the set of two-outcome quantum measurements, a sample is a POVM element  $\Pi$ <sup>87</sup> (in short:  $\mathbf{x} \leftrightarrow \Pi$ ) and the *p*-concept to be learned is a quantum state  $\psi$ . The evaluation of the concept/hypothesis on the sample corresponds to the probability  $\text{Tr}[\Pi\psi] \in [0, 1]$  of observing the measurement outcome associated with  $\Pi$  when the state  $\psi$  is measured.

It may be useful to further clarify the connection between the standard data-classification perspectives of supervised learning and the COLT perspective above. In the language of supervised learning we typically talk about classifiers, which partition their domain — or rather, they label the data points. In the setting of learning of quantum states, the classifier is the concept — in this case the quantum state. The quantum state ‘classifies’ the set of quantum POVM elements, according to the probability of observing the corresponding outcome. Specifically, a quantum state  $\rho$  (concept/hypothesis/classifier) assigns the label  $\text{Tr}[\rho\Pi] \in [0, 1]$  to the POVM element (data point)  $\Pi$ . The training set elements for this model are of the form  $(\Pi, \text{Tr}(\rho\Pi))$ , with  $0 \leq \Pi \leq \mathbb{1}$ .

In the spirit of COLT, the concept class ‘quantum states’, is said to be learnable under some distribution  $\mathcal{D}$  over two-outcome generalized measurement elements ( $\Pi$ ) if, for every concept—quantum state  $\rho$ —there exists an algorithm with access to examples of the form  $(\Pi, \text{Tr}(\rho\Pi))$ , where  $\Pi$  is drawn according to  $\mathcal{D}$ , which outputs a hypothesis  $h$  which (approximately) correctly predicts the label  $\text{Tr}(\rho\Pi')$  with high probability when  $\Pi'$  is drawn from  $\mathcal{D}$ . Note that the role of a hypothesis here can simply be played by a ‘best guess’ *classical description* of the quantum state  $\rho$ . The key result

<sup>87</sup> More precisely  $\Pi$  is a positive-semidefinite operator such that  $\mathbb{1} - \Pi$  is positive-semidefinite as well.

of Aaronson (2007) is that quantum states are learnable with sample complexity scaling only *linearly in the number of qubits*<sup>88</sup>, that is, logarithmically in the dimension of the density matrix. In operative terms, if Alice wishes to send an  $n$  qubit quantum state to Bob who will perform on it a two-outcome measurement (and Alice does not know which), she can achieve near-ideal performance by sending ( $O(n)$ ) *classical bits*<sup>89</sup>, which has clear practical but also theoretical importance. In some sense, these results can also be thought of as a generalized variant of Holevo bound theorems (Holevo 1982), limiting how much information can be stored and retrieved in the case of quantum systems. This latter result has thus far been more influential in the contexts of tomography than quantum ML, despite being quite a fundamental result in quantum learning theory. However, for fully practical purposes, the results above come with a caveat. The learning of quantum states is efficient in sample complexity (e.g. number of measurements one needs to perform) but the computational complexity of the reconstruction of the hypothesis is, in fact, likely exponential in the qubit number. Very recently, the efficiency of the reconstruction algorithms for the learning of stabilizer states was also shown in Rocchetto (2017).

## 5.2. (Quantum) learning and quantum processes

**Executive summary:** The notion of quantum learning has been used in the literature to refer to the study of various aspects of ‘learning about’ quantum systems. Beyond the learning of quantum states, one can also consider the learning of quantum evolutions. Here ‘knowing’ is operatively defined as having the capacity to implement the given unitary at a later point—this is similar to how ‘knowing’ in COLT implies that we can apply the concept function at a later point. Finally, as learning can pertain to learning in interactive environments—RL—one can consider the quantum generalizations of such settings. One of the first results in this direction formulates a quantum generalization of POMDPs. Note that as POMDPs form the mathematical basis of RL, the quantum-generalized mathematical object—quantum POMDP—may form a basis of quantum-generalized RL.

*Learning of quantum processes.* The concept of learning is quite diffuse and ‘quantum learning’ has been used in the literature quite often. Not every instance corresponds to generalizations of ‘classical learning’ in a machine or statistical learning sense. Nonetheless, some such works further illustrate the distinctions between the approaches one can employ with access to classical (quantum) tools, while learning about classical or quantum objects.

<sup>88</sup> The dependencies on the allowed inverse error and inverse allowed failure probability are polynomial and polylogarithmic, respectively.

<sup>89</sup> Here we assume Alice can locally generate her states at will. A classical strategy (using classical channels) is thus always possible by having Alice send the outcomes of full state tomography (or equivalently the classical description of the state), but this requires the use of  $O(2^n)$  bits already for pure states.

**Learning unitaries.** For instance ‘quantum learning of unitary operations’ has been used to refer to the task of optimal storing and retrieval of unknown unitary operations, which is a two stage process. In the storing phase, one is given access to a few uses of some unitary  $U$ . In the retrieval phase, one is asked to approximate the state  $U|\psi\rangle$ , given one or a few instances of a (previously fully unknown) state  $|\psi\rangle$ . As in the case of quantum template states (see section 5.1.1), we can distinguish semi-classical prepare-and-measure strategies (where  $U$  is estimated and represented as classical information) from quantum strategies, where the unitaries are applied on some resource state, which is used together with the input state  $|\psi\rangle$  in the retrieval stage. There is no simple universal answer to the question of optimal strategies. In Bisio *et al* (2010), the authors have shown, under reasonable assumptions, the surprising result that optimal strategies are semi-classical. In contrast, in Bisio *et al* (2011), the same question was asked for generalized measurements and the opposite was shown: optimal strategies require quantum memory. See e.g. Sedláč *et al* (2017) for some recent results on probabilistic unitary storage and retrieval, which can be understood as genuinely quantum learning<sup>90</sup> of quantum operations.

**Learning measurements.** The problem of identifying which measurement apparatus one is facing has featured in comparatively fewer works; see e.g. Sedláč and Ziman (2014) for a more recent example. Related to this, we encounter a more learning-theoretical perspective on the topic of learning measurements. In the comprehensive paper Cheng *et al* (2016) (which can serve as a review of parts of quantum ML in its own right), the authors explore the question of the learnability of quantum measurements. This can be thought of as the dual of the task of learning quantum states discussed previously in this section. Here, the examples are of the form  $(\rho, \text{Tr}(\rho E))$  and it is the measurement that is fixed. In this work, the authors compute a number of complexity measures, which are closely related to the VC dimension (see section 2.2.1), for which sample complexity bounds are known. From such complexity bounds one can, for instance, rigorously answer various relevant operative questions, such as how many random quantum probe states we need to prepare on average to accurately estimate a quantum measurement. Complementing the standard estimation problems, here we do not compute the optimal strategy but effectively gauge the information gain of a randomized strategy. These measures are computed for the family of hypotheses/concepts which can be obtained by either fixing the POVM element (thus learning the quantum measurement) or by fixing the state (which is the setting of Aaronson (2007)) and clearly illustrate the power of ML theory when applied in QIP context.

**Foundations of quantum-generalized RL.** The majority of quantum generalizations of ML concepts fit neatly in the domain of supervised learning with a few notable exceptions. In particular, in Barry *et al* (2014), the authors introduce a quantum generalization of partially observable MDPs

(POMDPs), discussed in section 2.3. For the convenience of the reader we give a brief recap of these objects. A fully observable MDP is a formalization of task environments: the environment can be in any number of states  $\mathcal{S}$  which the agent can observe. An action  $a \in \mathcal{A}$  of the agent triggers a transition of the state of the environment—the transition can be stochastic, and is specified by a Markov transition matrix  $P^a$ <sup>91</sup>. Additionally, beyond the dynamics, each MDP comes with a reward function  $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \Lambda$ , which rewards certain state-action-state transitions. In a POMDP, the agent does not see the actual state of the environment, but rather just *observations*  $o \in \mathcal{O}$ , which are (stochastic) functions of the environmental state<sup>92</sup>. Although the exact environmental state of the environment is not directly accessible to the agent, given the full specification of the system the agent can still assign a probability distribution over the state space given an interaction history. This is called a belief state and can be represented as a mixed state (mixing the ‘classical’ actual environmental states) which is diagonal in the POMDP state basis. The quantum generalization promotes the environment belief state to any quantum state defined on the Hilbert space spanned by the orthonormal basis  $\{|s\rangle | s \in \mathcal{S}\}$ . The dynamics of the quantum POMDP are defined by actions which correspond to quantum instruments (superoperators) the agent can apply: to each action  $a$ , we associate the set of Krauss operators  $\{K_o^a\}_{o \in \mathcal{O}}$ , which satisfy  $\sum_o K_o^a \dagger K_o^a = \mathbb{1}$ . If the agent performs the action  $a$  and observes the observation  $o$ , the state of the environment is mapped as  $\rho \rightarrow K_o^a \rho K_o^a \dagger / \text{Tr}[K_o^a \rho K_o^a \dagger]$ , where  $\text{Tr}[K_o^a \rho K_o^a \dagger]$  is the probability of observing that outcome. Finally, rewards are defined via the expected values of action-specific positive operators  $R_a$ , so  $\text{Tr}[R_a \rho]$ , given the state  $\rho$ . In Barry *et al* (2014), the authors have studied this model from the computational perspective of the hardness of identifying the best strategies for the agent, contrasting this setting to classical settings and proving separations. In particular, the complexity of deciding policy existence for finite horizons<sup>93</sup> is the same for the quantum and classical cases<sup>94</sup>. However, a separation can be found with respect to the goal reachability problem, which asks whether there exists a policy (of any length) which, with probability 1, reaches some target state. This separation is maximal—this problem is decidable in the classical case, yet undecidable in the quantum case. While this particular separation may not have immediate consequences for quantum learning, it suggests that there may be other (dramatic) separations with more immediate relevance.

<sup>91</sup> In other words, for any environment state  $s$ , producing an action  $a$  causes a transition to some state  $s'$  with probability  $s'^T P^a \vec{s}$ , where states are represented as canonical vectors.

<sup>92</sup> In general, the observations output can also depend on the previous action of the agent.

<sup>93</sup> That is, given a full specification of the setting, decide whether there exists a policy for the agent which achieves a cumulative reward above some value, in a certain number of states.

<sup>94</sup> This decision problem is already undecidable in the infinite horizon case for the classical problem, and thus trivially undecidable in the quantum case as well.

<sup>90</sup> Quantum in that that which is learned is encoded in a quantum state.

## 6. Quantum enhancements for machine learning

One of the most advertised aspects of quantum ML deals with the question of whether quantum effects can help us solve classical learning tasks more efficiently, ideally mirroring the successes of QC. The very first attempts to apply quantum information techniques to ML problems were made even before the seminal works of Shor (1997) and Grover (1996). Notable examples include the pioneering research into quantum NNs and quantum perceptrons (Lewenstein 1994, Kak 1995), the potential of quantum COLT (Bshouty and Jackson 1998) and early suggestions to exploit the exponential size of the Hilbert space for efficient data-vector representation and classification (Vlasov 1994, 1997). The topic of quantum NNs has undergone sustained growth and development since these early days, exploring various types of questions regarding the interplay of quantum mechanics and NNs. Most of the research in this area is not directly targeted at algorithmic improvements, and hence will be only briefly mentioned here. A fraction of the research into quantum NNs, which was disproportionately more active in the early days, considered the speculative topics of the function of quantum effects in NNs, both artificial and biological (Penrose 1989, Kak 1995). Parts of this research line have focused on concrete models, such as the effect of transverse fields in Hopfield networks (HNs) (Nishimori and Nonomura 1996) and decoherence in models of biological nets (Tegmark 2000), which, it is argued, would destroy any potential quantum effect. A second topic which permeates the research in quantum NNs is concerned with the fundamental question of a meaningful quantization of standard feed-forward NNs. The key question here is finding the best way to reconcile the linear nature of quantum theory with the necessity for non-linearities in the activation function of an NN (see section 2.1.1) and identifying suitable physical systems to implement such a scheme. Early ideas here included giving up on non-linearities *per se* and considering networks of unitaries which substitute layers of neurons (Lewenstein 1994). Another approach exploits non-linearities which stem from measurements and post-selection (arguably first suggested in Kak (1995)). The same issue is addressed by Behrman *et al* (1996) by using a continuous mechanical system where the non-linearity is achieved by coupling the system with an environment<sup>95</sup> in the model system of quantum dots. The purely foundational research into implementations of such networks and analysis of their quantum mechanical features has been and continues to be an active field of research (see e.g. Altaisky *et al* (2017)). For more information on this topic we refer the reader to more specialized reviews (Garman 2011, Schuld *et al* 2014b).

Unlike the research into quantum NNs, which has a foundational flavor, the majority of works studying quantum effects for classical ML problems are specifically focused on identifying improvements. The first examples of quantum advantages in this context were provided in the context of quantum COLT, which is the topic of the first subsection below. In the second subsection we will survey research suggesting the possibilities

of improvement in the capacity of associative memories. The last subsection deals with proposals which address computational run-time improvements in classical learning algorithms, the first of which came out as early as the early 2000s. Here we will differentiate between approaches which focus on quantum improvements in the training phase of a classifier by means of quantum optimization (mostly focused on exploiting near-term technologies and restricted devices) and approaches which build algorithms based on, roughly speaking, quantum parallelism and ‘quantum linear algebra’—which typically assume universal quantum computers and often ‘pre-filled’ databases. It should be noted that the majority of research in quantum ML is focused precisely on this last aspect, and the results here are already quite numerous. We can thus afford to present only a chosen selection of results.

### 6.1. Learning efficiency improvements: sample complexity

**Executive summary:** The first results showing the separation between quantum and classical computers were obtained in the context of oracles and for sample complexity—even the famous Grover’s search algorithm constitutes such a result. Similarly, COLT deals with the learning, i.e. the identification or the approximation, of concepts, which are also nothing but oracles. Thus, quantum oracular computation settings and learning theory share the same underlying framework, which is investigated and exploited in this formal topic. To talk about quantum COLT and improvements, or bounds, on sample complexity, the classical concept oracles are thus upgraded to quantum concept oracles, which output quantum states and/or allow access in superposition.

As elaborated on in section 2.2.1, COLT deals with the problem of learning *concepts*, typically abstracted as boolean functions of bit-strings of length  $n$ , that is,  $c : \{0, 1\}^n \rightarrow \{0, 1\}$ , from input–output relations alone. For intuitive purposes it is helpful to think of the task of optical character recognition (OCR), where we are given a bitmap image (black-and-white scan) of some size  $n = N \times M$ , and a concept may be, say, ‘everything which represents the letter A’ or, more precisely, *the concept* specifying which bitmaps correspond to the bitmaps of the letter ‘A’. Further, we are most often interested in a learning performance for a set of concepts: a concept class  $\mathcal{C} = \{c | c : \{0, 1\}^n \rightarrow \{0, 1\}\}$ —in the context of the running example of OCR, we care about algorithms which are capable of recognizing all letters, and not just ‘A’<sup>96</sup>.

The three typical settings studied in literature are the PAC model, exact learning from membership queries and the agnostic model; see section 2.2.1. These models differ in the type of

<sup>96</sup>Note that the choice and size of the concept class significantly influences the hardness of learning. For instance, if we assume that a given concept can be any boolean function, it is clear that one in principle needs to see all possible (thus exponentially many). In contrast, if the concept class contains only one element, learning is trivial, as the optimal algorithm just outputs the single concept.

<sup>95</sup>Similar ideas were also discussed by Peruš in Peruš (2000).

access to the concept oracle which is allowed. In the PAC model, the oracle outputs labeled examples according to some specified distribution, analogous to basic supervised learning. In the membership queries model, the learner gets to choose the examples, and this is similar to active supervised learning. In the agnostic model, the concept is ‘noisy’, i.e. forms a stochastic function, which is natural in supervised settings (the joint datapoint-label distribution  $P(x, y)$  need not be ‘functional’, i.e. it may have non-zero probabilities for  $P(x, y)$  and  $P(x, y')$ , for some  $x$  and  $y \neq y'$ ); for details we refer the reader to section 2.2.1.

All three models have been treated from a quantum perspective and whether or not quantum advantages are obtainable greatly depends on the details of the settings. Here we give a very succinct overview of the main results, partially following the structure of the recent survey on the topic by Arunachalam and de Wolf (2017).

**6.1.1. Quantum PAC learning.** The first quantum generalization of PAC learning was presented in Bshouty and Jackson (1998), where the quantum example oracle was defined to output coherent superpositions

$$\sum_{\mathbf{x}} \sqrt{p_D(\mathbf{x})} |x, c(x)\rangle, \quad (23)$$

for a given distribution  $D$  over the data points  $x$  for a concept  $c$ . Recall that classical PAC oracles output a sample pair  $(x, c(x))$ , where  $x$  is drawn from  $D$ , which can be understood as copies of the mixed state  $\sum_{\mathbf{x}} p_D(\mathbf{x}) |x, c(x)\rangle \langle x, c(x)|$ , with  $p_D(\mathbf{x}) = P(D = \mathbf{x})$ . The quantum oracle reduces to the standard oracle if the quantum example is measured in the standard (computational) basis. This first pioneering work showed that quantum algorithms with access to such a quantum-generalized oracle can provide more efficient learning of certain concept classes. The authors have considered the concept class of DNF formulas under the uniform distribution: here the concepts are  $s$ -term formulas in disjunctive normal form. In other words, each concept  $c$  is of the form  $c(\mathbf{x}) = \bigvee_I \bigwedge_j (x_I)_j'$ , where  $x_I$  is a substring of  $\mathbf{x}$  associated to  $I$ , which is a subset of the indices of cardinality at most  $s$ , and  $(x_I)_j'$  is a variable or its negation (a literal). An example of a DNF is of the form  $(x_1 \wedge x_3 \wedge \neg x_6) \vee (x_4 \wedge \neg x_8 \wedge x_1) \dots$ , where parentheses (terms) only contain variables or their negations in conjunction (ANDs,  $\wedge$ ), whereas all the parentheses are in disjunction (ORs,  $\vee$ ).

The uniform DNF learning problem (for  $n$  variables and  $\text{poly}(n)$  terms) is not known to be efficiently PAC learnable, but in Bshouty and Jackson (1998) it was proven to be efficiently quantum PAC learnable. The choice of this learning problem was not accidental: DNF learning is known to be learnable in the membership query model, which is described in detail in the next section. The corresponding classical algorithm which learns DNF in the membership query model directly inspired the quantum variant in the PAC case<sup>97</sup>. If the

<sup>97</sup> To provide the minimal amount of intuition, the best classical algorithm for the membership query model heavily depends on Fourier transforms (FT) of certain sets—the authors then use the fact that FT can be efficiently implemented on the amplitudes of the states generated by the quantum oracle using quantum computers. We refer the reader to Bshouty and Jackson (1998) for further details.

underlying distribution over the concept domain is uniform, other concept classes can be learned with a quantum speed-up as well, specifically, so called  $k$ -juntas:  $n$ -bit binary functions which depend only on  $k < n$  bits. In Atıcı and Servedio (2007), Atıcı and Servedio have shown that there exists a quantum algorithm for learning  $k$ -juntas using  $O(k \log(k)/\epsilon)$  uniform quantum examples,  $O(2^k)$  uniform classical examples and  $O(n k \log(k)/\epsilon + 2^k \log(1/\epsilon))$  time. Note that the improvement in this case is not in query complexity but rather in the classical processing, which, for the best known classical algorithm, has complexity at least  $O(n^{2k/3})$  (see Atıcı and Servedio (2007) and Arunachalam and de Wolf (2017) for further details).

Diverging from perfect PAC settings, in Cross *et al* (2015), the authors considered the learning of linear boolean functions<sup>98</sup> under the uniform distribution over the examples. The twist in this work is the assumption of noise<sup>99</sup> which allows for evidence of a classical quantum learnability separation.

In more recent times, it was also shown that *learning with errors*<sup>100</sup>, an important topic in COLT with critical applications in post-quantum cryptography, has been shown to be efficiently learnable given quantum examples (Grilo and Kerenidis 2017).

**Distribution-free PAC.** While the assumption of the uniform distribution  $D$  constitutes a convenient theoretical setting, in reality most often we have few guarantees on the underlying distribution of the examples. For this reason PAC learning often refers to distribution-free learning, meaning learning under the worst case distribution  $D$ . Perhaps surprisingly, it was recently shown that the quantum PAC learning model offers no advantages in terms of sample complexity over the classical model. Specifically, in Arunachalam and de Wolf (2016) the authors show that if  $C$  is a concept class of VC dimension  $d + 1$  then, for every (non-negative)  $\delta \leq 1/2$  and  $\epsilon \leq 1/20$ , every  $(\epsilon, \delta)$ -quantum PAC learner requires  $\Omega(d/\epsilon + \log(d^{-1})/\epsilon)$  samples. The same number of samples, however, is also known to suffice for a classical PAC learner (for any  $\epsilon$  and  $\delta$ ).

A similar result, showing no separation between quantum and classical *agnostic learning* was also proven in Arunachalam and de Wolf (2016)<sup>101</sup>.

<sup>98</sup> The learning of such functions is in QIP circles also known as the (non-recursive) Bernstein–Vazirani problem defined first in Bernstein and Vazirani (1997).

<sup>99</sup> However, the meaning of noise is not exactly the same in the classical and quantum case.

<sup>100</sup> In learning with errors, one is required to learn a hidden vector  $\mathbf{a}$ , given examples of the form  $(\mathbf{x}, \mathbf{x}^T \mathbf{a} + \epsilon)$ , where  $\epsilon$  is an error term drawn from some distribution and all the operations are done within a fixed field. Note that the zero-error version matches the setting of Fourier sampling. The building of cryptographic primitives whose hardness relies on the hardness of the learning with errors had been a common approach to building protocols secure against quantum adversaries. The presented result does not break these approaches but does highlight previously unrecognized issues.

<sup>101</sup> The notions of efficiency and sample complexity in the agnostic model are analogous to those in the PAC model, as is the quantum oracle which provides the coherent samples  $\sum_{\mathbf{x}, y} \sqrt{p_D(\mathbf{x}, y)} |\mathbf{x}, y\rangle$ . See section 2.2.1 for more details.

**Quantum predictive PAC learning.** Standard PAC learning settings do not allow exponential separations between classical and quantum sample complexity of learning, and consequently the notion of learnable concepts is the same in the classical and the quantum case. This changes if we consider weaker learning settings or, rather, a weaker meaning of what it means to learn. The PAC learning setting assumes that the learning algorithm outputs a hypothesis  $h$  with a low error with high confidence. In the classical case, there is no distinction between expecting that the hypothesis  $h$  can be applied once or any arbitrary number of times. However, in the quantum case, where the examples from the oracle may be quantum states, this changes and inductive learning in general may not be possible in all settings; see section 5. In Gavinsky (2012), the author considers a quantum PAC setting where only one (or polynomially few) evaluations of the hypothesis are required, called the Predictive Quantum (PQ) model<sup>102</sup>. In this setting the author identifies a relational concept class (i.e. each data point may have many correct labels) which is not (polynomially) learnable in the classical case, but is PQ learnable under a standard quantum oracle under the uniform distribution. The basic idea is to use quantum states, obtained by processing quantum examples, for each of the testing instances—in other words, the ‘implementation’ of the hypothesis contains a quantum state obtained from the oracle. This quantum state cannot be efficiently estimated, but can be efficiently obtained using the PQ oracle. The concept class and the labeling process are inspired by a distributed computation problem for which an exponential classical–quantum separation had been identified earlier in Bar-Yossef *et al* (2008). This work provides another noteworthy example of the intimate connection between various aspects of QIP—in this case, quantum communication complexity theory—and quantum learning.

**6.1.2. Learning from membership queries.** In the model of exact learning from membership queries, the learner can choose the elements from the concept domain it wishes to be labeled (similar to active learning); however, the task is to identify the concept exactly (no error) except with probability  $\delta < 1/3$ <sup>103</sup>. Learning from membership queries has, in the quantum domain, usually been called *oracle identification*. While quantum improvements in this context are possible, in Servedio and Gortler (2004), the authors show that they are at most low-degree polynomial improvements

in the most general cases. More precisely, if a concept class  $C$  over  $n$ -bits has classical and quantum membership query complexities  $D(C)$  and  $Q(C)$ , respectively, then  $D(C) = O(nQ(C)^3)$ <sup>104</sup>—in other words, improvements in sample complexity can be at most polynomial. Polynomial relationships have also been established for worst-case exact learning sample complexities (so-called  $(N, M)$ -query complexity); see Kothari (2013) and Arunachalam and de Wolf (2017). The above result is in spirit similar to earlier results in Beals *et al* (2001), where it was shown that quantum query complexity cannot provide a better-than-polynomial improvement over classical results, unless structural requirements on the oracle are imposed.

The results so far considered are standard, comparatively simple generalizations of classical learning settings, leading to somewhat restricted improvements in sample complexity. More dramatic improvements are possible if computational (time) complexity is taken into account, or if slightly non-standard generalizations of the learning model are considered. Note that we are not explicitly bringing computational complexity separations into the picture. Rather, under the assumption that certain computation problems are hard for the learner, we obtain a sample complexity separation.

In particular, already in Kearns and Valiant (1994) the authors constructed several classes of Boolean functions in the distribution-free model whose efficient learning (in the sample complexity sense) implies the capacity of factoring so-called Blum integers—a task not known to be solvable classically, but solvable on a quantum computer<sup>105</sup>. Using this observations, Servedio and Gortler have demonstrated classes which are efficiently quantum PAC learnable and classes which are efficiently learnable in the quantum membership query model, but which are not efficiently learnable in the corresponding classical models, unless Blum integers<sup>106</sup> can be efficiently factored on a classical computer (Servedio and Gortler 2004).

<sup>102</sup> In a manner of speaking, to learn a concept, in the PAC sense, implies that we can apply what we have learned arbitrarily many times. In PQ it suffices that the learner be capable of applying what it had learned just once to be considered successful. It, however, follows that if the number of examples is polynomial, PQ learnability also implies that the verification of learning can be successfully executed polynomially many times as well.

<sup>103</sup> As usual, success probability which is polynomially bounded away from  $1/2$  would also do.

<sup>104</sup> This simple formulation of the claim of Servedio and Gortler (2004) was presented in Arunachalam and de Wolf (2017).

<sup>105</sup> These ideas exploit the connections between asymmetric cryptography and learning. In asymmetric cryptography, a message can be decrypted easily using a public key, but the decryption is computationally hard unless one has a private key. To exemplify, the public key could be a Blum integer whereas the private key could be one of the factors. The data points are essentially the encryptions of integers  $k$ ,  $E(k, N)$ , for a public key  $N$ . The concept is defined by the least significant bit of  $k$ , which, provably, is not easier to obtain with bounded error than the decryption itself, which is computationally hard. A successful efficient learner of such a concept could factor Blum integers. The full proposal has further details which we omit for simplicity.

<sup>106</sup> The integer  $n$  is a Blum integer if it is a product of two distinct prime numbers  $p$  and  $q$ , which are congruent to  $3 \pmod{4}$  (i.e. both can be written in the form  $4t + 3$ , for a non-negative integer  $t$ ).

## 6.2. Improvements in learning capacity

**Executive summary:** The observation that a complete description of quantum systems typically requires the specification of exponentially many complex-valued amplitudes has led to the idea that those same amplitudes could be used to store data using only logarithmically few systems. While this idea fails for most applications, it has inspired some of the first proposals to use quantum systems for dramatic improvement in the capacities of associative, or content-addressable, memories. More likely quantum upgrades of content-addressable memories (CAMs), however, may come from a substantially different direction, which explores methods of extracting information from HNs—used as CAMs—and which is inspired by quantum adiabatic computing to realize a recall process which is similar to yet different from standard recall methods. The quantum methods may yield advantages by outputting superpositions of data and it has been suggested that they also utilize the memory more efficiently, leading to increased capacities.

The pioneering investigations in the areas between COLT, NNs and QIP challenged the classical sample complexity bounds. Soon thereafter (and likely independently), the first proposals suggesting quantum improvements in the context of space complexity emerged—specifically the efficiency of associative memories. Recall that associative, or CAM is a storage device which can be loaded with *patterns*, typically a subset of  $n$ -bit bit-strings  $P = \{\mathbf{x}_i\}_i$ ,  $\mathbf{x}_i \in \{0, 1\}^n$ , which are then, unlike in the case of standard RAM-type memories, not recovered by address but by content similarity: given an input string  $\mathbf{y} \in \{0, 1\}^n$ , the memory should return  $\mathbf{y}$  if it is one of the stored patterns (i.e.  $\mathbf{y} \in P$ ) or else a stored pattern which is ‘closest’ to  $\mathbf{y}$ , with respect to some distance, typically the Hamming distance. Deterministic perfect storage of any set of patterns clearly requires  $O(n \times 2^n)$  bits (there are in total  $2^n$  distinct patterns each requiring  $n$  bits), and the interesting aspects of CAMs begin when the requirements are somewhat relaxed. We can identify roughly two basic groups of ideas which were suggested to lead to improved capacities. The first group, sketched next, relies directly on the structure of the Hilbert space, whereas the second group of ideas stems from the quantization of a well-understood architecture for a CAM system: the HN.

**6.2.1. Capacity from amplitude encoding.** In some of the first works (Ventura and Martinez 2000 and Trugenberger 2001) it was suggested that the proverbial ‘exponential-sized’ Hilbert space describing systems of qubits may allow exponential improvements: intuitively even exponentially numerous pattern sets  $P$  can be ‘stored’ in a quantum state of only  $n$  qubits:  $|\psi_P\rangle = |P|^{-\frac{1}{2}} \sum_{\mathbf{x} \in P} |\mathbf{x}\rangle$ . These early works suggested creative ideas on how such a memory could be used to recover patterns (e.g. via modified amplitude amplification), though these often suffered from a lack of scalability and other quite fundamental issues, preventing them from yielding complete proposals<sup>107</sup>, and thus we will not dig into the details. We will, however, point out that these works may be interpreted

to propose some of the first examples of ‘amplitude encoding’ of classical data, which is heavily used in modern approaches to quantum ML. In particular, the stored memory of a CAM can always be represented as a single bit-string  $(b_{(0\dots 0)}, b_{(0\dots 1)}, \dots, b_{(1\dots 1)})$  of length  $2^n$  (each bit in the bit-string is indexed by a pattern, and its value encodes whether it is stored or not). This data vector (in this case binary, but this is not critical) is thus encoded into amplitudes of a quantum state of an exponentially smaller number of qubits:  $\mathbf{b} = (b_{(0\dots 0)}, b_{(0\dots 1)}, \dots, b_{(1\dots 1)}) \rightarrow \sum_{\mathbf{x} \in \{0,1\}^n} b_{\mathbf{x}} |\mathbf{x}\rangle$  (up to normalization).

**6.2.2. Capacity via quantized Hopfield networks.** A different approach to increasing the capacities of CAMs arises from the ‘quantization’ of different aspects of classical HNs, which constitute well-understood classical CAM systems.

**Hopfield networks as content-addressable memories.** Recall that an HN is a recurrent NN characterized by a set of  $n$  neurons, whose connectivity is given by a (typically symmetric) real matrix of weights  $W = (w_{ij})_{ij}$  and a vector of (real) local thresholds  $\{\theta_i\}_{i=1}^n$ . In the context of CAMs, the matrix  $W$  encodes the stored patterns, which are in this setting best represented as sequences of signs, so  $\mathbf{x} \in \{1, -1\}^n$ . The retrieval, given an input pattern  $\mathbf{y} \in \{1, -1\}^n$ , is realized by setting the  $k$ th neuron  $s_k$  to the  $k$ th value of the input pattern  $y_k$  followed by the ‘running of the network’ according to standard perceptron rules: each neuron  $k$  computes its subsequent value by checking if its inbound weighted sum is above the local threshold:  $s_k \leftarrow \text{sign}(\sum_l w_{kl} s_l - \theta_k)$  (assuming  $\text{sign}(0) = +1$ )<sup>108</sup>. As discussed previously, under moderate assumptions the described dynamical system converges to local attractive points, which also correspond to the energy minima of the Ising functional

$$E(\mathbf{s}) = -\frac{1}{2} \sum_{ij} w_{ij} s_i s_j + \sum_i \theta_i s_i. \quad (24)$$

Such a system still allows significant freedom in the rule specifying the matrix  $W$ , given a set of patterns to be stored: intuitively, we need to ‘program’ the minima of  $E$  (choosing the appropriate  $W$  will suffice, as the local thresholds can be set to zero) to be the target patterns, ideally without storing too many unwanted, so-called spurious, patterns. This and other properties of a useful storing rule, that is, a rule which specifies  $W$  given the patterns, are given as follows (Storkey 1997): (a) *locality*: an update of a particular connection should depend only on the information available to the neurons on either side of the connection<sup>109</sup>; (b) *incrementality*: the rule should allow the updating of the matrix  $W$  to store an additional pattern based only on the new pattern and  $W$  itself<sup>110</sup>,

<sup>107</sup> For a discussion on some of the shortcomings see e.g. Brun *et al* (2003) and Trugenberger (2003), and we also refer the reader to more recent reviews (Schuld *et al* 2014b, 2014a) for further details and analysis of the potential application of such memories to pattern recognition problems.

<sup>108</sup> The updates can be synchronous, meaning all neurons update their values at the same time, or asynchronous, in which case usually a random order is assigned. In most analyses, and here, asynchronous updates are assumed.

<sup>109</sup> Locality matters as the lack of it prohibits parallelizable architectures.

<sup>110</sup> In particular, it should not be necessary to have external memory storing e.g. all stored patterns, which would render HN-based CAMs undesirably non-adaptive and inflexible.

(c) *immediateness*: the rule should not require a limiting computational process for the evaluation of the weight matrix (rather, it should be a simple computation of few steps). The most critical property of a useful rule is that it (d) results in a CAM with a non-trivial capacity: it should be capable of storing and retrieving some number of patterns with controllable error (which includes few spurious patterns, for instance).

The first rule, historically speaking, the Hebbian rule, satisfies all the conditions above and is given by a simple recurrence relation: for the set of patterns  $\{\mathbf{x}^k\}_k$  the weight matrix is given by  $w_{ij} = \sum_k \mathbf{x}_i^k \mathbf{x}_j^k / M$  (where  $\mathbf{x}_j^k$  is the  $j$ th sign of the  $k$ th pattern, and  $M$  is the number of patterns). The capacity of HNs under standard recall and Hebbian updates has been investigated from various perspectives and, in the context of absolute capacity (the asymptotic ratio of the number of patterns that can be stored without error to the number of neurons, as the network size tends to infinity), it is known to scale as  $O(\frac{n}{2 \log(n)})$ . A well known result in the field improves on this to the capacity of  $O(\frac{n}{\sqrt{2 \log(n)}})$ , and is achieved by a different rule introduced by Storkey (1997), while maintaining all the desired properties. Here, we should emphasize that, in broad terms, the capacity is typically (sub)-linear in  $n$ . Better results, however, can be achieved in the classical settings if some of the assumptions (a)–(c) are dropped, but this is undesirable.

*Quantization of Hopfield-based content-addressable memories.* In early works by Rigatos and Tzafestas (2006, 2007), the authors have considered fuzzy and probabilistic learning rules, and have broadly argued that (a) such probabilistic rules correspond to a quantum deliberation process and that (b) the resulting CAMs can have significantly larger capacities. However, more rigorous (and fully worked out) results were shown more recently, by combining HNs with ideas from adiabatic QC.

The first idea, presented in Neigovzen *et al* (2009) connects HNs and quantum annealing. Recall that the HN can be characterized by the Ising functional  $E(\mathbf{s}) = -\frac{1}{2} \sum_{ij} w_{ij} s_i s_j$  (see equation (2)), where the stored patterns correspond to local minima and where we have, without the loss of generality, assumed that the local thresholds are zero. The classical recall corresponds to the problem of finding local minima closest to the input pattern  $\mathbf{y}$ . However, an alternative system, with similar features, is obtained if the input pattern is added in place of the local thresholds:  $E(\mathbf{s}, \mathbf{y}) = -\frac{1}{2} \sum_{ij} w_{ij} s_i s_j - \Gamma \sum_i y_i s_i$ . Intuitively, this lowers the energy landscape of the system specifically around the input pattern configuration. But then the stored pattern (previous local minimum) which is closest to the input pattern is the most likely candidate for a global minimum. Further, the problem of finding such configurations can now be tackled via quantum annealing: we define the quantum ‘memory Hamiltonian’ naturally by  $H_{\text{mem}} = -\frac{1}{2} \sum_{ij} w_{ij} \sigma_i^z \sigma_j^z$  and the HN Hamiltonian, given input  $\mathbf{y}$  by  $H_p = H_{\text{mem}} + \Gamma H_{\text{inp}}$ , where the input Hamiltonian is given by  $H_{\text{inp}} = -\sum_i y_i \sigma_i^z$ . The quantum recall is obtained by adiabatic evolution via the Hamiltonian trajectory  $H(t) = \Lambda(t) H_{\text{init}} + H_p$ , where  $\Lambda(0)$  is large enough that  $H_{\text{init}}$  dominates, and  $\Lambda(1) = 0$ . The system is initialized in the ground state of the (arbitrary and simple) Hamiltonian  $H_{\text{init}}$ , and if the evolution in  $t$  is slow enough to

satisfy the criteria of the adiabatic theorem, the system ends in the ground state of  $H_p$ . This proposal exchanged local optimization (classical retrieval) for global optimization. While this is generally a bad idea<sup>111</sup>, what is gained is a quantum formulation of the problem which can be run on adiabatic architectures, and also the fact that this system can return quantum superpositions of recalled patterns, if multiple stored patterns are approximately equally close to the input, which can be an advantage (Neigovzen *et al* 2009). However, the system above does not behave *exactly* the same as the classical recall network, which was further investigated in subsequent work (Seddiqi and Humble 2014) analyzing the sensitivity of the quantum recall under various classical learning rules. Further, in Santra *et al* (2016) the authors have provided an extensive analysis of the capacity of the Hebb-based HN, but under quantum annealing recall as proposed in Neigovzen *et al* (2009), showing, surprisingly, that this model yields *exponential* storage capacity under the assumption of random memories. This result stands in apparent stark contrast to standard classical capacities reported in textbooks<sup>112</sup>.

Regarding near-term implementability, in Santra *et al* (2016) the authors have investigated the suitability of the Chimera graph-based architectures of the D-Wave programmable quantum annealing device for quantum recall HN tasks, showing the potential for demonstrable quantum improvements in near-term devices.

### 6.3. Run-time improvements: computational complexity

**Executive summary:** The theory of quantum algorithms has provided examples of computational speed-ups for decision problems, various functional problems, oracular problems, sampling tasks and optimization problems. This section presents quantum algorithms which provide speed-ups for learning-type problems. The two main classes of approaches differ in the underlying computational architecture—a large class of algorithms relies on quantum annealers, which may not be universal for QC but may natively solve certain sub-tasks important in the context of ML. These approaches then have an increased likelihood of being realizable with near-term devices. In contrast, the second class of approaches assumes universal quantum computers, and often data prepared and accessible in quantum databases, but offers up to exponential improvements. Here we distinguish between quantum amplitude amplification and amplitude encoding approaches, which, with very few exceptions, cover all quantum algorithms for supervised and unsupervised learning.

<sup>111</sup> Generically, local optimization is easier than global, and in the context of the Ising system, global optimization is known to be NP-hard.

<sup>112</sup> At this point it should be mentioned that recently exponential capacities of HNs have been proposed for fully classical systems, by considering different learning rules (Hillar and Tran 2014, Karbasi *et al* 2014), which also tolerate moderate noise. The relationship and potential advantages of the quantum proposals remains to be elucidated.

The most prolific research area within quantum ML in the last few years has focused on identifying ML algorithms, or their computationally intensive subroutines, which may be sped up using quantum computers. While there are multiple natural ways to classify the performed research, an appealing first-order delineation follows the types of quantum computational architectures assumed<sup>113</sup>. Here we can identify research which is focused on using quantum annealing architectures, which are experimentally well justified and even commercially available in recent times (mostly in terms of the D-Wave system set-ups). In most such research, the annealing architecture will be utilized to perform a classically hard optimization problem usually emerging in the training phases of many classical algorithms. An involved part of such approaches will often be a meaningful rephrasing of such ML optimization to a form which an annealing architecture can (likely) handle. While the overall supervised task comprises multiple computational elements, it is only the optimization that will be treated by a quantum system in these proposals.

The second approach to speeding up ML algorithms assumes universal QC capabilities. Here, the obtained algorithms are typically expressed in terms of quantum circuits. For most proposals in this research line, to guarantee actual speed-ups there will be additional assumptions. For instance, most proposals can only guarantee improvements if the data which is to be analyzed is already present in a type of quantum oracle or a quantum memory and, more generally, that certain quantum states, which depend on the data, can be prepared efficiently. The overhead of initializing such a memory in the first place is not counted, but this may not be unreasonable as in practice the same database is most often used for a great number of analyses. Other assumptions may also be placed on the structure of the dataset itself, such as low condition numbers of certain matrices containing the data (Aaronson 2015).

**6.3.1. Speed-up via adiabatic optimization.** Quantum optimization techniques play an increasingly important role in quantum ML. Here, we can roughly distinguish two flavors of approaches, which differ in what computationally difficult aspect of training of a classical model is tackled by adiabatic methods. In the (historically) first approach, we deal with clear-cut optimization in the context of binary classifiers and, more specifically, boosting (see section 2.1.3). Since then, it has been shown that annealers can also help by generating samples from hard-to-simulate distributions. We will mostly focus on the earlier approaches, and only briefly mention the other more recent results.

**Optimization for boosting.** The representative line of research, which also initiated the development of this topic of quantum-enhanced ML based on adiabatic QC, focuses on a particular family of optimization problems called *quadratic unconstrained binary optimization* (QUBO) problems of the form

<sup>113</sup> Other classification criteria could be according to tasks, i.e. supervised versus unsupervised versus generative models etc, or depending on the underlying quantum algorithms used, e.g. amplitude amplification or equation solving.

$$\mathbf{x}^* = (x_1^*, \dots, x_n^*) = \operatorname{argmin}_{(x_1, \dots, x_n)} \sum_{i < j} J_{ij} x_i x_j, \quad x_k \in \{0, 1\} \quad (25)$$

specified by a real matrix  $J$ . QUBO problems are equivalent to the problem of identifying lowest energy states of the Ising functional<sup>114</sup>  $E(\mathbf{s}) = -\frac{1}{2} \sum_{ij} J_{ij} s_i s_j + \sum_i \theta_i s_i$ , provided we make no assumptions on the underlying lattice. Modern annealing architectures provide means for tackling the problem of finding such ground states using adiabatic QC. Typically we are dealing with systems which can implement the tunable Hamiltonian of the form

$$H(t) = -A(t) \underbrace{\sum_i \sigma_i^x}_{H_{\text{initial}}} + B(t) \underbrace{\sum_{ij} J_{ij} \sigma_i^z \sigma_j^z}_{H_{\text{target}}}, \quad (26)$$

where  $A, B$  are smooth positive functions such that  $A(0) \gg B(0)$  and  $B(1) \gg A(1)$ , that is, by tuning  $t$  sufficiently slowly we can perform adiabatic preparation of the ground state of the Ising Hamiltonian  $H_{\text{target}}$ , thereby solving the optimization problem. In practice, the parameters  $J_{ij}$  cannot be chosen fully freely (e.g. the connectivity is restricted to the so-called Chimera graph (Hen *et al* 2015) in D-Wave architectures), and also the realized interaction strength values have a limited precision and accuracy (Neven *et al* 2009b, Bian *et al* 2010), but we will ignore this for the moment. In general, finding ground states of the Ising model is functional NP-hard<sup>115</sup>, which is likely beyond the reach of quantum computers. However, annealing architectures still may have many advantages; for instance it is believed that they may still provide speed-ups in all, or at least average, instances and/or that they may provide good heuristic methods and hopefully near optimal solutions<sup>116</sup>.

In other words, any aspect of optimization occurring in ML algorithms which has an efficient mapping to (non-trivial) instances of QUBO problems, specifically those which can be realized by experimental set-ups, is a valid candidate for quantum improvements. Such optimization problems have been identified in a number of contexts, mostly dealing with training binary classifiers, and so belong to the class of supervised learning problems. The first setting considers the problem of building optimal classifiers from linear combinations of simple hypothesis functions, which minimize empirical error, while controlling the model complexity through a so-called regularization term. This is the common optimization setting of boosting (see section 2.1.3) and, with appropriate

<sup>114</sup> More precisely, an efficient algorithm which solves general QUBO problems can also efficiently solve arbitrary Ising ground state problems. One direction is trivial as QUBO optimization is a special case of ground state finding where the local fields are zero. In the converse, given an Ising ground state problem over  $n$  variables, we can construct a QUBO over  $n + 1$  variables which can be used to encode the local terms.

<sup>115</sup> Finding ground states is not a decision problem, so technically it is not correct to state that it is NP-hard. The class functional NP (FNP) is the extension of the NP class to functional (relational) problems.

<sup>116</sup> Indeed, one of the features of adiabatic models in general is that they provide an elegant means for (generically) providing approximate solutions, by simply performing the annealing process faster than prescribed by the adiabatic theorem.

mathematical gymnastics and few assumptions, it can be reduced to a QUBO problem.

The overarching setting of this line of work can be expressed in the context of training a binary classifier by combining weaker hypotheses. For this setting, consider a dataset  $D = \{\mathbf{x}_i, y_i\}_{i=1}^M$ ,  $\mathbf{x}_i \in \mathbb{R}^n$ ,  $y_i \in \{-1, 1\}$ , and a set of hypotheses  $\{h_j\}_{j=1}^K, h_j : \mathbb{R}^n \rightarrow \{-1, 1\}$ . For a given weight vector  $\mathbf{w} \in \mathbb{R}^n$  we define the composite classifier of the form  $hc_{\mathbf{w}}(\mathbf{x}) = \text{sign}(\sum_k w_k h_k(\mathbf{x}))$ .

The training of the composite classifier is achieved by the optimization of the vector  $\mathbf{w}$  so as to minimize misclassification on the training set, and so as to decrease the risk of overtraining. The misclassification cost is specified via a loss function  $L$ , which depends on the dataset and the hypothesis set in the boosting context. The overtraining risk, which tames the complexity of the model, is controlled by a so-called regularization term  $R$ . Formally we are solving

$$\text{argmin}_{\mathbf{w}} L(\mathbf{w}; D) + R(\mathbf{w}). \quad (27)$$

This constitutes the standard boosting frameworks exactly, but is also closely related to the training of certain SVMs, i.e. hyperplane classifiers<sup>117</sup>. In other words, quantum optimization techniques which work for the boosting setting can also help for hyperplane classification.

There are a few well-justified choices for  $L$  and  $R$ , leading to classifiers with different properties. Often, best choices (the definition of which depends on the context) lead to hard optimization (Long and Servedio 2010), and some of those can be reduced to QUBOs, but not straightforwardly.

In the pioneering paper on the topic (Neven *et al* 2008), Neven and co-authors consider the boosting setting. The regularization term is chosen to be proportional to the 0-norm, which counts the number of non-zero entries, that is,  $R(\mathbf{w}, \lambda) = \lambda \|\mathbf{w}\|_0$ . The parameter  $\lambda$  controls the relative importance of regularization in the overall optimization task. A common choice for the loss function would be the 0–1 loss function  $L_{0-1}$ , optimal in some settings, given by  $L_{0-1}(\mathbf{w}) = \sum_{j=1}^M \Theta(-y_j \sum_k w_k h_k(\mathbf{x}_j))$  (where  $\Theta$  is the step function), which simply counts the number of misclassifications. This choice is reasonably well motivated in terms of performance and is likely to be computationally hard. With appropriate discretization of the weights  $\mathbf{w}$ , which the authors argue likely does not hurt performance, the above forms a solid candidate for a general adiabatic approach. However, it does not fit the QUBO structure (which has only quadratic terms) and hence cannot be tackled using existing architectures. To achieve the desired QUBO structure the authors impose two modifications: they opt for a quadratic loss function  $L_2(\mathbf{w}) = \sum_{j=1}^M |y_j - \sum_k w_k h_k(\mathbf{x}_j)|^2$  and restrict the weights to binary (although this can be circumvented to an extent). Such a system is also tested using numerical experiments. In

<sup>117</sup> If we allow the hypotheses  $h_j$  to attain continuous real values, then by setting  $h_j$  to be the projection on the  $j$ th component of the input vector, so  $h_j(\mathbf{x}) = x_j$ , then the combined classifier attains the inner-product-threshold form  $hc_{\mathbf{w}}(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$  which contains hyperplane classifiers—the only component missing is the hyperplane offset  $b$  which is incorporated into the weight vector by increasing the dimension by 1.

a follow-up paper (Neven *et al* 2009b), the same team has generalized the initial proposal to accommodate another practical issue: problem size. Available architectures allow optimization over a few thousand variables, whereas in practice the number of hypotheses one optimizes over ( $K$ ) may be significantly larger. To resolve this, the authors show how to break a large optimization problem into more manageable chunks while maintaining (experimentally verified) good performance. These ideas were also tested in an actual physical architecture (Neven *et al* 2009a), and combined and refined in a more general, iterative algorithm in Neven *et al* (2012), tested also using actual quantum architectures.

While  $L_{0-1}$  loss functions were known to be good choices, they were not the norm in practice as they lead to non-convex optimization—so convex functions were preferred. However, in 2010 it became increasingly clear that convex functions are provably bad choices. For instance, in the seminal paper (Long and Servedio 2010) Long and Servedio<sup>118</sup> showed that boosting with convex optimization completely fails in noisy settings. Motivated by this, in Denchev *et al* (2012), the authors re-investigate D-Wave type architectures and identify a reduction which allows a non-convex optimization. Expressed in the hyperplane classification setting (as explained, this is equivalent to the boosting setting in structure), they identify a reduction which (indirectly) implements a non-convex function  $l_q(x) = \min\{(1-q)^2, (\max(0, 1-x))^2\}$ . This function is called the  $q$ -loss function, where  $q$  is a real parameter. The implementation of the  $q$ -loss function allows for the realization of optimization relative to the total loss of the form  $L_q(\mathbf{w}, b; D) = \sum_j l_q(y_j(\mathbf{w}^T \mathbf{x} + b))$ . The resulting regularization term is in this case proportional to the 2-norm of  $\mathbf{w}$ , instead of the 0-norm as in the previous examples, which may be sub-optimal. Nonetheless, the above forms a prime example where quantum architectures lead to ML settings which would not have been explored in the classical case (the loss  $L_q$  is unlikely to appear naturally in many settings) yet are well motivated, as (a) the function is non-convex and thus has the potential to circumvent all the no-go results for convex functions and (b) the optimization process can be realized in a physical system. The authors perform a number of numerical experiments demonstrating the advantages of this choice of a non-convex loss function when analyzing noisy data, which is certainly promising. In later work (Denchev *et al* 2015), it was also suggested that combinations of loss-regularization which are realizable in quantum architectures can also be used for so-called totally corrective boosting with cardinality penalization, which is believed to be classically intractable.

The details of this go beyond the scope of this review, but we can at least provide a flavor of the problem. In corrective boosting, the algorithm updates the weights  $\mathbf{w}$  essentially one step at a time. In totally corrective boosting, at the  $t$ th step of the boosting algorithm optimization,  $t$  entries of  $\mathbf{w}$  are updated simultaneously. This is known to lead to better regularized solutions, but the optimization is harder. Cardinality penalization pertains to using explicitly the 0-norm for the

<sup>118</sup> Servedio also, incidentally, provided some of the earliest results in quantum COLT, discussed in previous sections.

regularization (discussed earlier), rather than the more common 1-norm. This, too, leads to harder optimization which may be treated using an annealing architecture. In Babbush *et al* (2014), the authors significantly generalized the scope of loss functions which can be embedded into quantum architectures by observing that any polynomial unconstrained binary optimization can, with small overhead, be mapped onto a (slightly larger) QUBO problem. In particular, this opens up the possibility of implementing odd-degree polynomials which are non-convex and can approximate the 0–1 loss function. This approach introduced new classes of unusual yet promising loss functions.

*Applications of quantum boosting.* Building on the ‘quantum boosting’ architecture described above, in Pudenz and Lidar (2013) the authors explore the possibility of (aside from boosting) realizing anomaly detection, specifically envisioned in the computationally challenging problem of software verification and validation<sup>119</sup>. In the proposed learning step the authors use quantum optimization (boosting) to learn the characteristics of the program being tested. In the novel testing step the authors modify the target Hamiltonian so as to lower the energy of the states which encode input–outputs where the real and ideal software differ. These can then be prepared in superposition (i.e. they can prepare a state which is a superposition over the inputs where  $P$  will produce an erroneous output) similarly to the previously mentioned proposals in the context of adiabatic recall of superpositions in HNs (Neigovzen *et al* 2009).

*Beyond boosting.* Beyond the problems of boosting, annealers have been shown to be useful for the training of so-called Bayesian Network Structure Learning problems (O’Gorman *et al* 2015), as their training can also be reduced to QUBOs.

In recent years, there has been significant focus on utilizing restricted annealing devices to realize distributions which are computationally expensive to generate. One application of this is in the training of certain ML models. A notable example of this is based on the fact that the training of deep networks usually relies on the use of a so-called generative deep belief network, which is a relative of deep Boltzmann machines (restricted BMs with multiple layers; see section 2.1.1 for more details). The training of deep belief networks, in turn, is the computational bottleneck, as it requires the sampling of hard-to-generate distributions, specifically particular Gibbs distributions. Samples from this distributions may be more efficiently prepared using annealing architectures; see e.g. Adachi and Henderson (2015). Another popular idea for the application of devices capable of preparing hard distributions is to utilize them as generative models directly. This is one of the faster growing research areas in recent times, in part due to the recent increase in popularity of generative models in general. It is impossible to provide a comprehensive review of this topic and we provide the interested reader with a few of

the pioneering references (Benedetti *et al* 2016, 2017) and a recent review covering the topic (Perdomo-Ortiz *et al* 2017).

Further novel ideas introducing fully quantum BM-like models have been proposed (Amin *et al* 2016). Further, in recent work by Sieberer and Lechner (2017) which builds on the flexible construction in Lechner *et al* (2015), the authors have shown how to achieve programmable adiabatic architectures, which allows running algorithms where the weights themselves are in superposition. This possibility is also sure to inspire novel QML ideas. Moving on from BMs, in recent work by Wittek and Gogolin (2017) the authors have also shown how suitable annealing architectures may be useful to speed up the performance of probabilistic inference in so-called Markov logic networks<sup>120</sup>. This task involves the estimation of partition functions arising from statistical models, concretely Markov random fields, which include the Ising model as a special case. Quantum annealing may speed up this sub-task.

More generally, the idea that restricted, even simple, quantum systems which may be realizable with current technologies could implement information processing elements useful for supervised learning is beginning to be explored in settings beyond annealers. For instance, in Schuld *et al* (2017) a simple interferometric circuit is used for the efficient evaluation of distances between data vectors, useful for classification and clustering. A more complete account of these recent ideas is beyond the scope of this review.

*6.3.2. Speed-ups in circuit architectures.* One of the most important applications of ML in recent times has been in the context of data mining and analyzing so-called *big data*. The most impressive improvements in this context have been achieved by proposing specialized quantum algorithms which solve particular ML problems. Such algorithms assume the availability of full-blown quantum computers and have been tentatively probed since the early 2000s. In recent times, however, we have witnessed a large influx of ideas. Unlike the situation we have seen in the context of quantum annealing, where an optimization subroutine alone was run on a quantum system, in most of the approaches of this section the entire algorithm, and even the dataset, may be quantized.

The ideas for quantum-enhancements for ML can roughly be classified into two groups: (a) approaches which rely on Grover’s search and amplitude amplification to obtain up-to-quadratic speed-ups and (b) approaches which encode relevant information into quantum amplitudes and which have a potential for even exponential improvements. The second group of approaches forms perhaps the most developed research line in quantum ML, and collects a plethora of quantum tools—most notably quantum linear algebra, utilized in quantum ML proposals.

<sup>119</sup> A software is represented as a map  $P$  from input to output spaces, here specified as a subset of the space of pairs  $(x_{\text{input}}, x_{\text{output}})$ . An implemented map (software)  $P$  is differentiated from the ideal software  $\hat{P}$  by the mismatches in the defining pairs.

<sup>120</sup> Markov logic networks (Richardson and Domingos 2006) combine first-order logic as used for knowledge representation and reasoning with statistical modeling—essentially, the world is described via first-order sentences (a knowledge base), which gives rise to a graphical statistical model (a Markov random field) where correlations stem from the relations in the knowledge base.

*Speed-ups by amplitude amplification.* In Anguita *et al* (2003), it was noticed that the training of SVMs may be a hard optimization task, with no obviously better approaches than brute-force search. In turn, for such cases of optimization with no structure, QIP offers at least a quadratic relief in the guise of variants of Grover (1996) search algorithm or its application to minimum finding (Durr and Hoyer 1999). This idea predates and is, in spirit, similar to some of the early adiabatic-based proposals of the previous subsection, but the methodology is substantially different. The potential of quadratic improvements stemming from Grover-like search mechanisms was explored more extensively in Aïmeur *et al* (2013), in the context of unsupervised learning tasks. There the authors assume access to a black-box oracle which computes a distance measure between any two data points. Using this, combined with amplitude amplification techniques (e.g. minimum finding in Durr and Hoyer (1999)), the authors achieve up-to-quadratic improvements in key subroutines used in clustering (unsupervised learning) tasks. Specifically, improvements are obtained in algorithms performing minimum spanning tree clustering, divisive clustering and  $k$ -medians clustering<sup>121</sup>. Additionally, the authors also show that quantum effects allow for a better parallelization of clustering tasks, by constructing a distributed version of Grover’s search. This construction may be particularly relevant as large databases can often be distributed.

More recently, in Wiebe *et al* (2014c) the author considers the problem of training deep (more than two-layered) BMs. As we mentioned earlier, one of the bottlenecks of exactly training BMs stems from the fact that it requires the estimation of probabilities of certain equilibrium distributions. Computing this analytically is typically not possible (it is as hard as computing partition functions) and sampling approaches are costly as they require attaining the equilibrium distribution and many iterations to reliably estimate small values. This is often circumvented by using proxy solutions (e.g. relying on contrastive divergence) to train approximately, but it is known that these methods are inferior to exact training. In Wiebe *et al* (2014c), a quantum algorithm is devised which prepares coherent encodings of the target distributions, relying on quantum amplitude amplification, often attaining quadratic improvements in the number of training points and even exponential improvements in the number of neurons, in some regimes. Quadratic improvements have also been obtained in pure data mining contexts, specifically in *association rules mining* (Yu *et al* 2016), which, roughly speaking, identifies correlations between objects in large databases<sup>122</sup>. As our final example in the class of quantum algorithms relying on

amplitude amplification we mention the algorithm for the training of perceptrons (Wiebe *et al* 2016). Here, quantum amplitude amplification was used not only to quadratically speed up training but also, interestingly, to quadratically reduce the error probability. Since perceptrons constitute special cases of SVMs, this result is similar in motivation to the much older proposal (Anguita *et al* 2003), but relies on more modern and involved techniques.

*Precursors of amplitude encoding.* The first ideas suggesting the use of amplitudes to store data vectors exponentially efficiently, which might be utilized for classification based on state fidelity (effectively, the inner product of the vectors), were mentioned in very early works (Vlasov 1994, 1997) which escaped broader attention. In another early, and similarly mostly overlooked work by Schützhold (2003), Schützhold proposed an interesting application of QC to pattern recognition problems, which addresses many ideas which have only been investigated, and re-invented, by the community relatively recently. The author considers the problem of identifying ‘patterns’ in images, specified by  $N \times M$  black-and-white bitmaps, characterized by a function  $f : \{1, \dots, N\} \times \{1, \dots, M\} \rightarrow \{0, 1\}$  (which technically coincides with a *concept* in COLT see section 2.2.1), specifying the color-value  $f(x, y) \in \{0, 1\}$  of a pixel at coordinate  $(x, y)$ . The function  $f$  is given as a quantum oracle  $|x\rangle|y\rangle|b\rangle \xrightarrow{U_f} |x\rangle|y\rangle|b \oplus f(x, y)\rangle$ . The oracle is used in quantum parallel (applied to a superposition of all coordinates) and conditioned on the bit-value function being 1 (this process succeeds with constant probability, whenever the density of points is constant,) leading to the state  $|\psi\rangle = \mathcal{N} \sum_{x,y \text{ s.t. } f(x,y)=1} |x\rangle|y\rangle$ , where  $\mathcal{N}$  is a normalization factor. Note that this state is proportional to the vectorized bitmap image itself, when given in the computational basis. Next, the author points out that ‘patterns’—repeating macroscopic features—can often be detected by applying the discrete Fourier transform to the image vector, which has classical complexity  $O(NM \log(NM))$ . However, the quantum Fourier transform (QFT) can be applied to the state  $|\psi\rangle$  utilizing exponentially fewer gates. The author proceeds to show that the measurements of the QFT-transformed state may yield useful information, such as pattern localization. This work is innovative in a few aspects. First, the author utilized the encoding of data points (here strings of binary values) into amplitudes by using a quantum memory, in a manner which is related to the applications in the context of content-addressable memories discussed in section 6.2.1. It should be pointed out, however, that in the present application of amplitude encoding, non-binary amplitudes have clear meaning (in say grayscale images), although this is not explicitly discussed by the author<sup>123</sup>. Second, in contrast to all previous proposals, the author shows the potential for a quantifiable exponential computational complexity improvement for a family of tasks. However, this is all contingent on having

<sup>121</sup> In minimum tree clustering, data is represented as a weighted graph (weight being the distance) and a minimum weight spanning tree is found.  $k$  clusters are identified by simply removing the  $k - 1$ - highest weight edges. Divisive clustering is an iterative method which splits sets into two subsets according to a chosen criterion, and this process is iterated.  $k - \text{median}$  clustering identifies clusters which minimize the cumulative within-cluster distances to the median point of the cluster.

<sup>122</sup> To exemplify the logic behind association rules mining, in the typical context of shopping, if shopping item (list element) B occurs in nearly every shopping list in which shopping item A occurs as well, one concludes that the person buying A is also likely to buy B. This is captured by the rule denoted  $B \Rightarrow A$ .

<sup>123</sup> The earlier works Vlasov (1994) and Vlasov (1997) explicitly utilize amplitude encoding; however, the proposals are not as detailed as Schützhold (2003).

access to the pre-filled database ( $U_f$ ), the loading of which would nullify any advantage. Aside from the fact that this may be considered to be a one-off overhead, Schützhold discusses physical means of loading data from optical images in a quantum-parallel approach, which may be effectively efficient.

**Amplitude encoding: linear algebra tools.** The very basic idea of amplitude encoding is to treat states of  $N$ -level quantum systems as data vectors themselves. More precisely, given a data vector  $\mathbf{x} \in \mathbb{R}^n$ , the amplitude encoding would constitute the normalized quantum state  $|x\rangle = \sum_i x_i |i\rangle / \|\mathbf{x}\|$ , where it is often also assumed that norm of the vector  $\|\mathbf{x}\|$  can always be accessed.

Note that  $N$ -dimensional data points are encoded into amplitudes of  $n \in O(\log(N))$  qubits. Any polynomial circuit applied to the  $n$ -qubit register encoding the data thus constitutes only a polylogarithmic computation relative to the data-vector size, and this is at the basis of all exponential improvements (also in the case of Schützhold (2003), discussed in the previous section)<sup>124</sup>.

These ideas have led to a research area which could be called ‘quantum linear algebra’ (QLA), that is, a collection of algorithms which solve certain linear algebra problems by directly encoding numerical vectors into state vectors. These quantum sub-routines have then been used to speed up numerous ML algorithms, some of which we describe later in this section. QLA includes algorithms for matrix inversion, principal component analysis (Harrow *et al* 2009, Lloyd *et al* 2014) and many others. For didactic purposes, we will first give the simplest example which performs the estimation of inner products in logarithmic time.

**Tool 1: inner product evaluation.** Given access to boxes which prepare quantum states  $|\psi\rangle$  and  $|\phi\rangle$ , the overlap  $|\langle\phi|\psi\rangle|^2$  can be estimated to precision  $\epsilon$  using  $O(1/\epsilon)$  copies, using the so-called the swap test.

The swap test (Buhrman *et al* 2001) applies a controlled-SWAP gate onto the state  $|\psi\rangle|\phi\rangle$ , where the control qubit is set to the uniform superposition  $|+\rangle$ . The probability of ‘succeeding’, i.e. observing  $|+\rangle$  on the control after the circuit, is given by  $(1 + |\langle\phi|\psi\rangle|^2)/2$ , and this can be estimated by iteration (a more efficient option using quantum phase estimation is also possible). If the states  $|\psi\rangle$  and  $|\phi\rangle$  encode unit-length data vectors, the success value encodes their inner product up to sign. Norms and phases can also be estimated by minor tweaks to this basic idea—in particular, actual norms of the amplitude-encoded states will be accessible in a separate oracle and used in algorithms. The sample complexity of this process depends only on precision, whereas the gate complexity is proportional to  $O(\log(N))$  as that many qubits need to be control-swapped and measured.

<sup>124</sup>In a related work by Wiebe and Granade (2015), the authors investigate the learning capacity of ‘small’ quantum systems and identify certain limitations in the context of Bayesian learning, based on Grover optimality bounds. Here, ‘small’ pertains to systems of logarithmic size, encoding information in amplitudes. This work thus probes the potential of space complexity improvements for quantum-enhanced learning, related to early ideas discussed in section 6.2.

The swap test also works as expected if the reduced states are mixed and the overall state is product. This method of computing inner products, relative to classical vector multiplication, offers an exponential improvement with respect to  $N$  (if calls to devices which generate  $|\psi\rangle$  and  $|\phi\rangle$  take  $O(1)$ ), at the cost of significantly worse scaling with respect to errors, as classical algorithms have typical error scaling with the logarithm of inverse error,  $O(\log(1/\epsilon))$ . However, in the context of ML problems, this can constitute an excellent compromise.

**Tool 2: quantum linear system solving.** Perhaps the most influential technique for quantum enhanced algorithms for ML is based on one of the quintessential problems of linear algebra: solving systems of equations. In their seminal paper (Harrow *et al* 2009), the authors proposed the first algorithm for ‘quantum linear system’ (QLS) solving, which performs the following. Consider an  $N \times N$  linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , where  $\kappa$  and  $d$  are the condition number<sup>125</sup> and sparsity of the Hermitian system matrix  $A$ <sup>126</sup>. Given (quantum) oracles giving positions and values of non-zero elements of  $A$  (that is, given standard oracles for  $A$  as encountered in Hamiltonian simulation; see Berry *et al* (2015)) and an oracle which prepares the quantum state  $|\mathbf{b}\rangle$  which is the amplitude encoding of  $\mathbf{b}$  (up to norm), the algorithm in Harrow *et al* (2009) prepares the quantum state  $|\mathbf{x}\rangle$  which is  $\epsilon$ -close to the amplitude encoding of the solution vector  $\mathbf{x}$ . The run-time of the first algorithm is  $\tilde{O}(\kappa^2 d^2 \log(N)/\epsilon)$ . Note that the complexity scales proportionally to the *logarithm* of the system size. Note that any classical algorithm must scale at least with  $N$ , and this offers room for exponential improvements. The original proposal in Harrow *et al* (2009) relies on Hamiltonian simulation (implementing  $\exp(iAt)$ ), upon which phase estimation is applied. Once phases are estimated, inversely proportional amplitudes—that is, the inverses of the eigenvalues of  $A$ —are imprinted via a measurement. It has also been noted that certain standard matrix pre-conditioning techniques can also be applicable in the QLS scheme (Clader *et al* 2013). The linear scaling in the error in these proposals stems from the phase estimation subroutine. In more recent work by Childs *et al* (2015), the authors also rely on best Hamiltonian simulation techniques, but forgo the expensive phase estimation. Roughly speaking, they (probabilistically) implement a linear combination of unitaries of the form  $\sum_k \alpha_k \exp(ikAt)$  upon the input state. This constitutes a polynomial in the unitaries which can be made to approximate the inverse operator  $A^{-1}$  (in a measurement-accessible subspace) more efficiently. This, combined with other numerous optimizations, yields a final algorithm with complexity  $\tilde{O}(\kappa d \text{polylog}(N/\epsilon))$ , which is essentially optimal. It is important to note that the apparently exponentially more efficient schemes above do not trivially imply provable computational improvements, even if we assume free access to all oracles. For instance, one of the issues is that the quantum algorithm outputs a quantum state,

<sup>125</sup>Here, the condition number of the matrix  $A$  is given by the quotient of the largest and smallest singular values of  $A$ .

<sup>126</sup>The assumption that  $A$  is Hermitian is non-restrictive, as an oracle for any sparse matrix  $A$  can be modified to yield an oracle for the symmetrized matrix  $A' = |0\rangle\langle 1| \otimes A^\dagger + |1\rangle\langle 0| \otimes A$ .

from which classical values can only be accessed by sampling. This process for the reconstruction of the complete output vector would kill any improvements. On the other hand, certain functions of the amplitudes can be computed efficiently, the computation of which may still require  $O(N)$  steps classically, yielding the desired exponential improvement. Thus this algorithm will be most useful as a sub-routine, an intermediary step of bigger algorithms, such as those for quantum ML.

**Tool 3: density matrix exponentiation.** Density matrix exponentiation (DME) is a remarkably simple idea, with few subtleties and arguably profound consequences. Consider an  $N$ -dimensional density matrix  $\rho$ . Now, from a mathematical perspective,  $\rho$  is nothing but a semidefinite positive matrix, although it is also commonly used to denote the quantum state of a quantum system—and these two are subtly different concepts. In the first reading, where  $\rho$  is a matrix (we will denote it  $[\rho]$  to avoid confusion),  $[\rho]$  is also a valid description of a physical Hamiltonian, with time-integrated unitary evolution  $\exp(-i[\rho]t)$ . Could one approximate  $\exp(-i[\rho]t)$ , having access to quantum systems prepared in the state  $\rho$ ? Given sufficiently many copies ( $\rho^{\otimes n}$ ), the obvious answer is yes—one could use full state tomography to reconstruct  $[\rho]$ , to arbitrary precision, and then execute the unitary using, say, Hamiltonian simulation (efficiency notwithstanding). In Lloyd *et al* (2014), the authors show a significantly simpler method: given any input state  $\sigma$  and one copy of  $\rho$ , the quantum state

$$\sigma' = \text{Tr}_B[\exp(-i\Delta t S)(\sigma_A \otimes \rho_B) \exp(i\Delta t S)], \quad (28)$$

where  $S$  is the Hermitian operator corresponding to the quantum SWAP gate, approximates the desired time evolution to first order, for small  $\Delta t$ :  $\sigma' = \sigma - i\Delta t[\rho, \sigma] + O(\Delta t^2)$ . If this process is iterated, by using fresh copies of  $\rho$ , we obtain that the target state  $\sigma_\rho = \exp(-i\rho t)\sigma \exp(i\rho t)$  can be approximated to precision  $\epsilon$ , by setting  $\Delta t$  to  $O(\epsilon/t)$  and using  $O(t^2/\epsilon)$  copies of the state  $\rho$ . DME is, in some sense, a generalization of the process of using SWAP tests between two quantum states, to simulate aspects of a measurement specified by one of the quantum states. One immediate consequence of this result is in the context of Hamiltonian simulation, which can now be efficiently realized (with no dependency on the sparsity of the Hamiltonian) whenever one can prepare quantum systems in a state which is represented by the matrix of the Hamiltonian. In particular, this can be realized using qRAM stored descriptions of the Hamiltonian whenever the Hamiltonian itself is of low rank. More generally, this also implies e.g. that QLS algorithms can also be efficiently executed when the system matrix is not sparse, but rather dominated by few principal components, i.e. close to a low rank matrix<sup>127</sup>.

**Remark.** Algorithms for QLS, inner product evaluation, quantum PCA and, consequently, almost all quantum algo-

<sup>127</sup> Since a density operator is normalized, the eigenvalues of data matrices are rescaled by the dimension of the system. If the eigenvalues are close to uniform, they are rendered exponentially small in the qubit number. This then requires exponential precision in DME, which would offset any speed-ups. However, if the spectrum is dominated by a constant number of terms, the precision required and overall complexity are again independent from the dimension, allowing overall efficient algorithms.

gorithms listed in the remainder of this section also assume ‘pre-loaded databases’, which allow accessing of information in quantum parallel, and/or the accessing or efficient preparation of amplitude encoded states. The problem of parallel access, or even the storing of quantum states, has been addressed and mostly resolved using so-called quantum random access memory (qRAM) architectures (Giovannetti *et al* 2008)<sup>128</sup>. The same qRAM structures can also be used to realize oracles utilized in the approaches based on quantum search. However, having access to quantum databases pre-filled with classical data does not *a priori* imply that quantum-amplitude-encoded states can also be generated efficiently, which is, at least implicitly, assumed in most works below. For a separate discussion on the cost of some similar assumptions, we refer the reader to Aaronson (2015).

**Amplitude encoding: algorithms.** With all the quantum tools in place, we can now present a selection of quantum algorithms for various supervised and unsupervised learning tasks, grouped according to the class of problems they solve. The majority of proposals of this section follow a clear paradigm: the authors investigate established ML approaches and identify those where the computationally intensive parts can be reduced to linear algebra problems, most often diagonalization and/or equation solving. In this sense, further improvements in quantum linear algebra approaches are likely to lead to new results in quantum ML.

As a final comment, all the algorithms below pertain to discrete-system implementations. Recently, in Lau *et al* (2017), the authors have also considered continuous variable variants of qRAM, QLS and DME, which immediately lead to continuous variable implementations of all the quantum tools and most quantum-enhanced ML algorithms listed below.

**Regression algorithms.** One of the first proposals for quantum enhancements tackled linear regression problems, specifically, least-squares fitting, and relied on QLS. In least-squares fitting, we are given  $N$   $M$ -dimensional real data points paired with real labels, so  $(\mathbf{x}_i, y_i)_{i=1}^N$ ,  $\mathbf{x}_i = (x_i^j)_j \in \mathbb{R}^M$ ,  $\mathbf{y} = (y_i)_i \in \mathbb{R}^N$ . In regression  $y$  is called the response variable (also regressand or dependent variable), whereas the data points  $\mathbf{x}_i$  are called predictors (or regressors or explanatory variables), and the goal of least-squares linear regression is to establish the best linear model, that is  $\beta = (\beta_j)_j \in \mathbb{R}^M$  given by

$$\text{argmin}_\beta \|\mathbf{X}\beta - \mathbf{y}\|^2, \quad (29)$$

where the data matrix  $\mathbf{X}$  collects the data points  $\mathbf{x}_i$  as rows. In other words, linear regression assumes a linear relationship between the predictors and the response variables. It is well-established that the solution to the above least-squares problem is given by  $\beta = \mathbf{X}^+ \mathbf{y}$ , where  $\mathbf{X}^+$  is the Moore–Penrose

<sup>128</sup> qRAM realizes the following mapping:  $|\text{addr}\rangle |b\rangle \xrightarrow{\text{qRAM}} |\text{addr}\rangle |b \oplus d_{\text{addr}}\rangle$ , where  $d_{\text{addr}}$  represents the data stored at the address  $\text{addr}$  (the  $\oplus$  represents modulo addition, as usual), which is the reversible variant of conventional RAM memories. In Giovannetti *et al* (2008), it was shown that a qRAM can be constructed such that its internal processing scales logarithmically in the number of memory cells.

pseudoinverse of the data matrix, which is, in the case that  $\mathbf{X}^\dagger \mathbf{X}$  is invertible, given by  $\mathbf{X}^+ = (\mathbf{X}^\dagger \mathbf{X})^{-1} \mathbf{X}^\dagger$ . The basic idea in Wiebe *et al* (2012) is to apply  $\mathbf{X}^\dagger$  onto the initial vector  $|\mathbf{y}\rangle$  which amplitude-encodes the response variables, obtaining a state proportional to  $\mathbf{X}^\dagger |\mathbf{y}\rangle$ . This can be done e.g. by modifying the original QLS algorithm (Harrow *et al* 2009) to imprint not the inverses of eigenvalues but the eigenvalues themselves. Following this, the task of applying  $(\mathbf{X}^\dagger \mathbf{X})^{-1}$  (onto the generated state proportional to  $\mathbf{X}^\dagger |\mathbf{y}\rangle$ ) is interpreted as an equation-solving problem for the system  $(\mathbf{X}^\dagger \mathbf{X})\boldsymbol{\beta} = \mathbf{X}^\dagger \mathbf{y}$ .

The end result is a quantum state  $|\boldsymbol{\beta}\rangle$  proportional to the solution vector  $\boldsymbol{\beta}$ , in time  $O(\kappa^4 d^3 \log(N)/\epsilon)$ , where  $\kappa$ ,  $d$  and  $\epsilon$  are the condition number, the sparsity of the ‘symmetrized’ data matrix  $\mathbf{X}^\dagger \mathbf{X}$  and the error, respectively. Again, we have in general few guarantees on the behavior of  $\kappa$  and an obvious restriction on the sparsity  $d$  of the data matrix. However, whenever both are  $O(\text{polylog}(N))$ , we have a potential<sup>129</sup> for exponential improvements. This algorithm is not obviously useful for actually finding the solution vector  $\boldsymbol{\beta}$ , as it is encoded in a quantum state. Nonetheless, it is useful for estimating the quality of fit: essentially by applying  $\mathbf{X}$  onto  $|\boldsymbol{\beta}\rangle$  we obtain the resulting prediction of  $\mathbf{y}$ , which can be efficiently compared to the actual response variable vector via a swap test<sup>130</sup>.

These basic ideas for quantum linear regression have since been extended in a few works. In an extensive and complementary work (Wang 2014), the authors rely on the powerful technique of ‘qubitization’ (Low and Chuang 2016) and optimize the goal of actually producing the best-fit parameters  $\boldsymbol{\beta}$ . By necessity, the complexity of their algorithm is proportional to the number of data points  $M$ , but is logarithmic in the data dimension  $N$ , and quite efficient in other relevant parameters. In Schuld *et al* (2016), the authors follow the ideas of Wiebe *et al* (2012) more closely and achieve the same results as in the original work when the data matrix, rather than being sparse, is low-rank. Further, they improve on the complexities by using other state-of-the-art methods. This latter work critically relies on the technique of DME.

**Clustering algorithms.** In Lloyd *et al* (2013), amplitude encoding and inner product estimation are used to estimate the distance  $\|\mathbf{u} - \bar{\mathbf{v}}\|$  between a given data vector  $\mathbf{u}$  and the average of a collection of data points (centroid)  $\bar{\mathbf{v}} = \sum_i \mathbf{v}_i/M$

<sup>129</sup> In this section we often talk about the ‘potential’ for exponential speed-ups because some of the algorithms as given do not solve classical computational problems for which classical lower bounds are known. Consider the conditions which have to be satisfied for the QLS algorithm to offer exponential speed-ups. First, we need to be dealing with problems where the preparation of the initial state and qRAM memory can be done in  $O(\text{polylog}(N))$ . Next, the problem condition number must be  $O(\text{polylog}(N))$  as well. Assuming all this is satisfied, we are still not done: the algorithm generates a quantum state. As classical algorithms do not output quantum states, we cannot talk about quantum speed-ups. The quantum state can be measured, outputting at most  $O(\text{polylog}(N))$  (more would kill exponential speed-ups due to printout alone) bits which are functions of the quantum state. However, the hardness of computing these output bits, given that all the initial assumptions are clearly not obvious, needs to be proven.

<sup>130</sup> In the paper, the authors take care to appropriately symmetrize all the matrices in a manner we discussed in a previous footnote but, for clarity, we ignore this technical step.

for  $M$  data points  $\{\mathbf{v}_i\}_i$  in time which is logarithmic in both the vector length  $N$  and number of points  $M$ . Using this as a building block, the authors also show an algorithm for  $k$ -means classification/clustering (where the computing of the distances to the centroid is the main cost), achieving an overall complexity  $O(M \log(MN)/\epsilon)$ , which may be even further improved in some cases. Here, it is assumed that amplitude-encoded state vectors and their normalization values are accessible via an oracle, or that they can be efficiently implemented from a qRAM storing all the values. Similar techniques, combined with coherent quantum phase estimation and Grover-based optimization, have also been used for the problem of  $k$ -nearest neighbor algorithms for supervised and unsupervised learning (Wiebe *et al* 2015).

**Quantum principal component analysis.** In the same paper Lloyd *et al* (2014) the ideas of DME were immediately applied to a quantum version of principal component analysis (PCA). PCA constitutes one of the most standard unsupervised learning techniques useful for dimensionality reduction but, naturally, has a large scope of applications beyond ML. In quantum PCA, for a quantum state  $\rho$  one applies quantum phase estimation of the unitary  $\exp(-i[\rho])$  using DME applied onto the state  $\rho$  itself. In the ideal case of absolute precision, given the spectral decomposition  $\rho = \sum_i \lambda_i |\lambda_i\rangle \langle \lambda_i|$ , this process generates the state  $\sum_i \lambda_i |\lambda_i\rangle \langle \lambda_i| \otimes |\tilde{\lambda}_i\rangle \langle \tilde{\lambda}_i|$ , where  $\tilde{\lambda}_i$  denotes the numerical estimation of the eigenvalue  $\lambda_i$  corresponding to the eigenvector  $|\lambda_i\rangle$ . Sampling from this state recovers both the (larger) eigenvalues and the corresponding quantum states which amplitude-encode the eigenvectors, which may be used in further quantum algorithms. The recovery of high-value eigenvalues and eigenvectors constitutes the essence of classical PCA as well.

**Quantum support vector machines.** One of the most influential papers in quantum-enhanced ML relies on QLS and DME for the task of quantizing SVM algorithms. For the basic ideas behind SVMs see section 2.1.2.

We focus our attention to the problem of training SVMs, as given by the optimization task in its dual form, in equation (6), repeated here for convenience:

$$(\alpha_1^*, \dots, \alpha_N^*) = \operatorname{argmin}_{\alpha_1, \dots, \alpha_N} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j,$$

such that  $\alpha_i \geq 0$  and  $\sum_i \alpha_i y_i = 0$ .

The solution of the desired SVM is then easily computed by  $\mathbf{w}^* = \sum_i y_i \alpha_i \mathbf{x}_i$ .

As a warm-up result, in Rebentrost *et al* (2014) the authors point out that using quantum evaluation of inner products, appearing in equation (30), can already lead to exponential speed-ups with respect to the data-vector dimension  $N$ . The quantum algorithm complexity is, however, still polynomial in the number of data points  $M$  and the error dependence is now linear (as the error of the inner product estimation is linear). The authors proceed to show that full exponential

improvements can be possible (with respect to both  $N$  and  $M$ ), however, for the special case of least-squares SVMs. Given the background discussions we have already laid out with respect to DME and QLS, the basic idea is here easy to explain. Recall that the problem of training least-squares SVMs reduces to a linear program, specifically a least-squares minimization. As we have seen previously, such minimization reduces to equation solving, which was given by the system in equation (14) which we repeat here:

$$\begin{bmatrix} 0 & 1^T \\ 1_N & \Omega + \gamma^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix}. \quad (30)$$

Here,  $1$  is an ‘all ones’ vector,  $Y$  is the vector of labels  $y_i$ ,  $\alpha$  is the vector of the Lagrange multipliers yielding the solution,  $b$  is the offset,  $\gamma$  is a parameter depending on the hyperparameter  $C$  and  $\Omega$  is the matrix collecting the (mapped) inner products of the training vectors so  $\Omega_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$ . The key technical aspects of Rebstroff *et al* (2014) demonstrate how the system above is realized in a manner suitable for QLS. To give a flavor of the approach, we will simply point out that the system sub-matrix  $\Omega$  is proportional to the reduced density matrix of the quantum state  $\sum_i |\mathbf{x}_i\rangle\langle i|_1 |\mathbf{x}_i\rangle\langle 2$ , obtained after tracing out the subsystem 2. This state can, under some constraints, be efficiently realized with access to qRAM encoding the data points. Following this, DME enables the application of QLS where the system matrix has a block proportional to  $\Omega$ , up to technical details which we omit for brevity. The overall quantum algorithm generates the quantum state proportional to  $|\psi_{\text{out}}\rangle \propto b|0\rangle + \sum_{i=1}^M \alpha_i |i\rangle$ , encoding the offset and the multipliers. The multipliers need not be extracted from this state by sampling. Instead any new point can be classified by (1) generating an amplitude-encoded state of the input and (2) estimating the inner product between this state and  $|\psi'_{\text{out}}\rangle \propto b|0\rangle|0\rangle + \sum_{i=1}^M \alpha_i |\mathbf{x}_i\rangle|i\rangle$ , which is obtained by calling the quantum data oracle using  $|\psi_{\text{out}}\rangle$ . This process has an overall complexity of  $O(\kappa_{\text{eff}}^3 \epsilon^{-3} \log(MN))$ , where  $\kappa_{\text{eff}}$  depends on the eigenstructure of the data matrix. Whenever this term is polylogarithmic in data size, we have a potential for exponential improvements.

*Gaussian process regression.* In Zhao *et al* (2015) the authors demonstrate how QLS can be used to dramatically improve Gaussian process regression (GPR), a powerful supervised learning method. GPR can be thought of as a stochastic generalization of standard regression: given a training set  $\{\mathbf{x}_i, y_i\}$ , it models the latent function (which assigns labels  $y$  to data points), assuming Gaussian noise on the labels  $f(\mathbf{x}) = y + \epsilon$  where  $\epsilon$  encodes independent and identically distributed. More precisely, GPR is a process in which an initial distribution over possible latent functions is refined by taking into account the training set points, using Bayesian inference. Consequently, the output of GPR is, roughly speaking, a distribution over models  $f$  which are consistent with the observed data (the training set). While the descriptions of such a distribution may be large, in computational terms to predict the value of a new point  $\mathbf{x}_*$  in GPR one needs to compute two numbers: a *linear predictor* (also referred to

as the *predictive mean*, or simply *mean*) and the *variance of the predictor*, which are specific to  $\mathbf{x}_*$ . These numbers characterize the distribution of the predicted value  $y_*$  by the GPR model which is consistent with the training data. Further, it turns out, both values can be computed using modified QLS algorithms. The fact that this final output size is independent from the dataset size, combined with QLS, provides possibilities for exponential speed-ups in terms of data size. This naturally holds, provided the data is available in qRAM, as is the case in most algorithms of this section. It should be mentioned that the authors take meticulous care in listing out all the ‘hidden costs’ (and the working out of intermediary algorithms) in the final tally of the computational complexity.

*Geometric and topological data analysis.* All the algorithms we presented in this subsection thus far critically depend on having access to ‘pre-loaded’ databases—the loading itself would introduce a linear dependence on the database size, whereas the inner-product, QLS and DME algorithms provide potential for just logarithmic dependence. However, this can be circumvented in the cases where the data points in the quantum database can be efficiently computed individually. This is reminiscent of the fact that most applications of Grover’s algorithm have a step in which the Grover oracle is efficiently computed. In ML applications, this can occur if the classical algorithm requires, as a computational step, a combinatorial exploration of the (comparatively small) dataset. Then the quantum algorithm can generate the combinatorially larger space in quantum parallel—thereby efficiently computing the effective quantum database. The first example where this was achieved was presented in Lloyd *et al* (2016), in the context of topological and geometric data analysis. These techniques are very promising in the context of ML, as topological features of data do not depend on the metric of choice, and thus capture the truly robust features of the data. The notion of topological features (in the ML world of discrete data points) are given by those properties which exist when data is observed at different spatial resolutions. Such persistent features are thus robust and less likely to be artefacts of noise or choice of parameters, and are mathematically formalized through so-called persistent homology. A particular family of features of interest are the number of connected components, holes and voids (or cavities). These numbers, which are defined for simplicial complexes (roughly, a closed set of simplices), are called Betti numbers. To extract such features from data, one must therefore construct nested families of simplicial complexes from the data and compute the corresponding features captured by the Betti numbers. However, there are combinatorially many simplices which one should consider and which should be analyzed, and one can roughly think of all possible simplices as data points which need further analysis. However, they are efficiently generated from a small set—essentially the collection of the pair-wise distances between data points. The authors show how to generate quantum states which encode the simplices in logarithmically few qubits and further show that, from this representation, the Betti numbers can be efficiently estimated. Iterating this at various resolutions allows the identification of persistent features. As usual, full

exponential improvements happen under some assumptions on the data, and here they are manifest in the capacity to efficiently construct the simplicial states—in particular, having the total number of simplices in the complex be exponentially large would suffice, although it is not clear when this is the case; see Aaronson (2015). This proposal provides evidence that quantum ML methods based on amplitude encoding may, at least in some cases, yield exponential speed-ups even if data is not pre-stored in a qRAM or an analogous system.

Since the first versions of this review appeared, a number of new algorithms have appeared, which we list without a detailed review, for the convenience of the reader. In Reberntrost *et al* (2017), the authors have utilized amplitude encoding to exponentially compress a variant of an associative memory, related to Hopfield networks.

As mentioned a large component of modern approaches to quantum-enhanced ML relies on quantum linear algebra techniques, and any progress in this area may lead to new quantum ML algorithms. Promising recent examples of this were given in terms of algorithms for quantum gradient descent (Reberntrost *et al* 2016a, Kerenidis and Prakash 2017), which could e.g. lead to novel quantum methods for training NNs. Similarly, recent breakthroughs in quantum algorithms for semi-definite programming are already inspiring a novel and particularly promising route for quantum enhanced ML algorithms (Brandão and Svore 2017, Brandão *et al* 2017, van Apeldoorn *et al* 2017).

## 7. Quantum learning agents and elements of quantum artificial intelligence

The topics discussed thus far in this review, with few exceptions, deal with the relationship between physics, mostly QIP, and traditional ML techniques which allow us to better understand data, or the process which generates them. In this section, we go beyond data analysis and optimization techniques to address the relationship between QIP and more general learning scenarios, or even between QIP and AI. As mentioned, in more general learning or AI discussions, we typically talk about agents interacting with their environments, which may be, or more often fail to be, intelligent. In our view, by far the most important aspect of any intelligent agent is its capacity to learn from its interactions with its environment. However, general intelligent agents learn in environments which are complex and may change. Further, the environments are susceptible to being changed by the agent itself, which is the crux of, e.g., learning by experiments. All this delineates general learning frameworks, which begin with RL, from more restricted settings of data-driven ML.

In this section, we will consider physics-oriented approaches to learning via interaction, specifically the projective simulation (PS) model, and then focus on quantum-enhancements in the context of RL<sup>131</sup>. Following this, we will

<sup>131</sup> While RL is a particularly mathematically clean model for learning by interaction, it is worthwhile to note that it is not fully general—for instance learning in real environments always involves supervised and other learning paradigms to control the size of the exploration space, but also various other techniques which occur when we try to model settings in continuous, or otherwise not turn-based, fashion.

discuss an approach for considering the most general learning scenarios, where the agent, the environment and their interaction, are treated quantum-mechanically: this constitutes a quantum generalization of the broad AE framework, underlying modern AI. We will finish off by briefly discussing other results from QIP, which do not directly deal with learning, but which may still play a role in the future of QAI.

### 7.1. Quantum learning via interaction

**Executive summary:** The first proposal which addressed the specification of learning agents, designed with the possibility of quantum processing of episodic memory in mind, was the model of PS. The results on quantum improvements of agents which learn by interacting with classical environments have mostly been given within this framework. The PS agent deliberates by effectively projecting itself into conceivable situations using its memory, which organizes its episodic experiences in a stochastic network. Such an agent can solve basic RL problems, meta-learn and solve problems with aspects of generalization. The deliberation is a stochastic diffusion process, allowing for a few routes for quantization. Using quantum random walks, quadratic speed-ups can be obtained.

The applications of QIP to reinforcement and other interactive learning problems has been comparatively less studied, when compared to quantum enhancements in supervised and unsupervised problems. One of the first proposals which provided a coherent view on learning agents from a physics perspective was that of PS (Briegel and De las Cuevas 2012). We first give a detailed description of the PS model and review the few other works related to this topic at the end of the section.

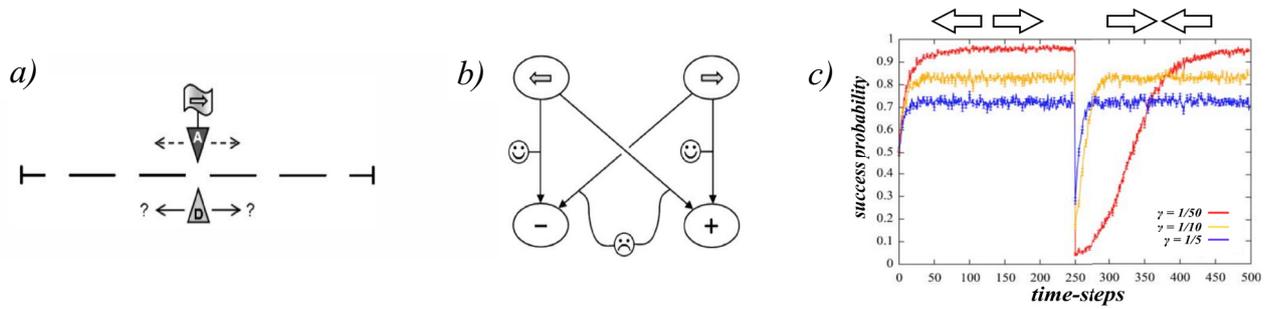
PS is a flexible framework for the design of learning agents, drawing motivation both from agency and physics, and influenced by modern views on robotics, which also provides a natural route to quantization.

The PS agent is an embodied physical entity<sup>132</sup> situated in an environment on which it can act, and which re-acts in the form of certain physical inputs.

A PS agent *learns from experience*, by perceiving percepts from the set  $\mathcal{S} = \{s_i\}_i$  and by performing actions from the set  $\mathcal{A} = \{a_i\}_i$ <sup>133</sup>. The learning agent's behavior—that is, the choice of actions, given certain percepts—is based on its cumulative experience, accumulated in the agent's memory, which is structured. The central concept of the PS framework is this structured memory system of the agent: the *episodic and compositional memory* (ECM). The ECM is a network of *clips*, which are the units of episodic memory. A clip,

<sup>132</sup> The PS perspective is in line with the framework of embodied learning agents; see section 2.3 for more detail.

<sup>133</sup> The sets of percepts and actions are constrained by the physical interfaces of the agent. Further, this basic model assumes discretized time and sensory space, which is consistent with actual realizations, although this could be generalized.



**Figure 12.** (a) An *invasion game* where the agent is facing an attacker, who must be blocked by appropriately moving to the left or right (Briegel and De las Cuevas 2012). These two options form the actions of the agent. The agent (D) learns to associate symbols, presented by the attacker (A), to one of the two movements. The basic scenario here is, in RL terms, a contextual two-armed bandit problem (Langford and Zhang 2008), where the agent gets rewarded when it correctly couples the two percepts to the two actions. Initially, the percepts have no meaning for the agent, and indeed the attacker can alter the meaning over time. (b) The internal network of such a simple PS agent requires only action clips (bottom layer) and percept clips (top layer), arranged in two layers, with connections only from percepts to actions. The ‘smiling’ edges are rewarded in this initial scenario. (c) Basic learning curves for PS with non-zero  $\gamma$  in the invasion game where the rule specifying which actions are rewarded switches at time step 250, i.e. the attacker changes its strategy. Re-learning speed and asymptotic efficiency depend on  $\gamma$ . Adapted from Briegel and De las Cuevas (2012).

denoted  $c_i$ , can represent<sup>134</sup> an individual percept or action, so  $c_i \in \mathcal{S} \cup \mathcal{A}$ —and indeed there is no other external *type* appearing in the PS framework. However, experiences may be more complex (such as an autobiographical episodic memory, similar to short video-clips, where we remember a temporally extended sequence of actions and percepts that we experienced). This brings us to the following recursive definition: a clip is either a percept, an action, or a structure over clips.

Typical examples of structured clips are percept-action sequences  $(s_1, a_1, \dots, s_k, a_k)$ , describing what happened, i.e. a  $k$ -length history of interaction between the agent and environment, or sets of percepts  $(s_1$  or  $s_2 \dots)$ , which can be used to achieve aspects of generalization. The overall ECM is a network of clips (that is, a labeled directed graph, where the vertices are the clips), where the edges encode the agent’s previous experiences and deliberation paths and have a functional purpose explained momentarily. The ECM is the basis of the agent’s deliberation mechanism: it is used to define the agent’s instantaneous policy, i.e. the agent probabilistically decides on (or rather ‘falls into’) the next action and performs it, depending on the state of the current ECM network. There are multiple mechanisms for this which we discuss shortly. Finally, the learning of the agent is manifest in the updates of the ECM network, which occurs in two modes: by (1) changing the weights of the edges and (2) the topology of the network, through the addition or deletion of clips. The above principles describe the basic blueprint behind PS agents. This framework can, for instance, be used to construct various types of RL agents, related to tabular RL. The set of clips here comprises the set of actions and the set of environmental states (as the percept set). The ECM is characterized by an  $h$ -matrix, specifying real-valued weights ( $h$ -values) labeling the connections from percept clips to action clips, with, by convention,  $h_{ij} \geq 1$ . The deliberation, and as a consequence the agent’s policy, is realized by a random walk in the memory space governed by the  $h$ -matrix: that is, the probability of transition from percept

$s$  to action  $a$  is given with  $p(a|s) = \frac{h_{s,a}}{\sum_{a'} h_{s,a'}}$ . In other words, the row-wise normalized  $h$ -matrix specifies the stochastic transition matrix of the PS model, in the Markov chain sense. Finally, the learning is manifest in the tuning of the  $h$ -values, via an update rule, and one of the simplest options is:

$$h^{t+1}(c_i, c_j) = h^t(c_i, c_j) - \gamma(h^t(c_i, c_j) - 1) + \delta_{c_i, c_j} \lambda, \quad (31)$$

where  $t, t + 1$  denote consecutive time steps,  $\lambda$  denotes the reward received in the last step ( $t$ ), the function  $\delta_{c_i, c_j}$  is 1 if and only if the transition from clip  $c_i$  to clip  $c_j$  occurred in the same previous step, and zero otherwise. Finally,  $\gamma \in [0, 1]$  is called a damping, or dissipation, meta-parameter. This is a simple reward-driven rule which, coupled with the deliberation rule, incites the agent to favor percept-action transitions which are rewarded. The  $\gamma$  term prevents the divergence of the  $h$ -values and speeds up the re-learning capacities of the model; see figures 12 for illustration.

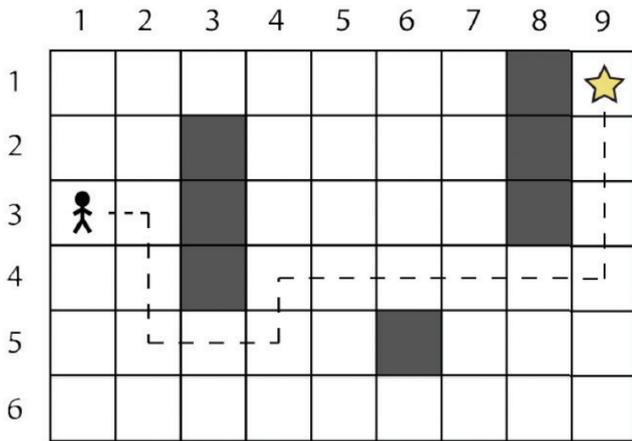
To handle delayed reward settings, for instance in a maze or a so called grid-world setting, illustrated in figure 13, the reward propagation to all relevant percept-action pairs can be realized via a so-called *glow* mechanism. To each edge, a glow value  $g_{ij}$  is assigned in addition to the  $h_{ij}$ -value. It is (re-) set to 1 whenever the edge is used, and decays with the rate  $\eta \in [0, 1]$ , that is,  $g_{ij}^t = (1 - \eta)g_{ij}^{t-1}$ . The  $h$ -value update rule is appended to reward all ‘glowing’ edges, proportional to the glow value, whenever a reward is issued:

$$h^{t+1}(c_i, c_j) = h^t(c_i, c_j) - \gamma(h^t(c_i, c_j) - 1) + g_{ij}^t \lambda. \quad (32)$$

Thus, all the edges which contributed to the final reward get a fraction, in proportion to how recently they were used. This parallels the intuition that the more recent actions relative to the rewarded move played a larger role in the final success.

The expression in equation (32) has functional similarities to the Q-learning action-value update rule in equation (21). However, the learning dynamics is different, and the expressions are conceptually different—Q-learning updates estimate bounded Q-values, whereas the PS is not a state-value estimation method, but rather a purely reward-driven system.

<sup>134</sup> Representation means that we, strictly speaking, distinguish actual percepts from the memorized percepts, and the same for actions. This distinction is however not crucial for the purposes of this exposition.



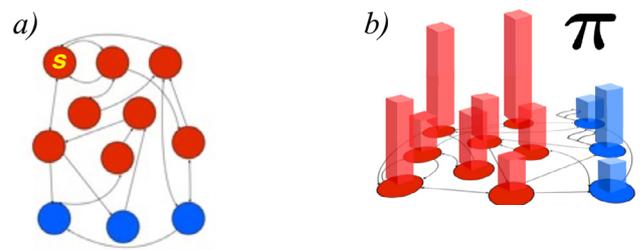
**Figure 13.** The environment is essentially a grid, where each site has an individual percept. The moves dictate the movements of the agent (say, up, down, left or right) and certain sites are blocked off—walls. The agent explores this world looking for the rewarded site. When the rewarded site is found, a reward is given and the agent is reset to the same initial position. Adapted from Melnikov *et al* (2014).

The PS framework allows other constructions as well. In Briegel and De las Cuevas (2012), the authors also introduced emoticons—edge-specific flags which capture aspects of intuition. These can be used to speed-up re-learning via a *reflection mechanism*, where a random walk can be iterated multiple times, increasing the chances that a desired—flagged—set of actions is hit; see Briegel and De las Cuevas (2012) for more detail. Further in this direction, the deliberation of the agent can be based not on a *hitting process*—where the agent performs the first action it hits—but rather on a *mixing process*.

In the latter case, the ECM is a collection of Markov chains and the correct action is sampled from the stationary distribution over the ECM. This model is referred to as the reflective PS (rPS) model; see figure 14. Common to all models, however, is that the deliberation process is governed by a stochastic walk, specified by the ECM.

Regarding performance, the basic PS structure with a two-layered network encoding percepts and actions—which matches standard tabular RL approaches—was extensively analyzed and benchmarked against other models (Melnikov *et al* 2014, Mautner *et al* 2015). However, the questions that are emphasized in PS literature go beyond questions of performance in standard RL tasks, in two directions.

For instance, in Mautner *et al* (2015) it was shown that the action composition aspects of the ECM allow the agent to perform better in some benchmarking scenarios, which has natural applications in, e.g., the context of protecting MBQC from unitary noise (Tiersch *et al* 2015) and the context of finding novel quantum experiments (Melnikov *et al* 2017), elaborated on in section 4.3. Further, by utilizing the capacity of ECM to encode larger and multiple networks, we can not only also address problems which require generalization (Melnikov *et al* 2015)—inferring correct behavior by percept similarity—but also design agents which autonomously optimize their own meta-parameters, such as  $\gamma$  and  $\eta$  in the



**Figure 14.** Quantum rPS representation of the ECM network, and its steady state over non-action (red) and action (blue) clips.

PS model. That is, the agents can meta-learn (Makmal *et al* 2016). These problems go beyond the basic RL framework, and the PS framework is flexible enough to also allow the incorporation of other learning models—e.g. NNs could be used to perform dimensionality reduction (which could allow for broader generalization capabilities) or even to directly optimize the ECM itself. The PS model has been combined with such additional learning machinery in an application to robotics and haptic skill learning (Hangl *et al* 2016, 2017a, 2017b, 2017c).

However, there is an advantage in keeping the underlying PS dynamics homogeneous, that is, essentially solely based on random walks over the PS network, in that it offers a few natural routes to quantization. This is the second direction of foundational research in PS. For instance, in Briegel and De las Cuevas (2012) the authors expressed the entire classical PS deliberation dynamics as an incoherent part of a Liouvillean dynamics (master equation for the quantum density operator) which also includes some coherent part (Hamiltonian-driven unitary dynamics). This approach may yield advantages both in deliberation time and also expands the space of internal policies the agent can realize.

Another perspective on the quantization of the PS model was developed in the framework of discrete-time quantum walks.

In Paparo *et al* (2014), the authors have exploited the paradigm of Szegedy-style quantum walks, to improve quadratically deliberation times of rPS agents. The Szegedy (2004) approach to random walks can be used to specify a unitary random walk operator  $U_P$  for a given transition matrix  $P$ <sup>135</sup> whose spectral properties are intimately related to those of  $P$  itself. We refer the reader to the original references for the exact specification of  $U_P$ , and just point out that  $U_P$  can be efficiently constructed via a simple circuit depending on  $P$ , or given black-box access to entries of  $P$ .

Assume  $P$  corresponds to an irreducible, aperiodic (guaranteeing a unique stationary distribution) and also time-reversible (meaning it satisfies detailed balance conditions) Markov chain. Let  $\boldsymbol{\pi} = (\pi_i)_i$  be the unique stationary distribution of  $P$ ,  $\delta$  the spectral gap of  $P$ <sup>136</sup> and  $|\boldsymbol{\pi}\rangle = \sum_i \sqrt{\pi_i} |i\rangle$  the coherent encoding of the distribution  $\boldsymbol{\pi}$ . Then we have that

<sup>135</sup> By transition matrix, we mean an entry-wise non-negative matrix, with entries in columns adding to unity.

<sup>136</sup> The spectral gap is defined by  $\delta = 1 - |\lambda_2|$ , where  $\lambda_2$  is, in norm, the second largest eigenvalue.

(a)  $U_P |\pi\rangle = |\pi\rangle$  and (b) the eigenstates  $\{\lambda_i\}$  of  $P$  and eigenphases  $\theta_i$  of  $U_P$  are related by  $\lambda_i = \cos(\theta_i)$ <sup>137</sup>.

This is important as the spectral properties, specifically the spectral gap  $\delta$ , more-or-less tightly fix the mixing time—that is the number of applications of  $P$  needed to obtain the stationary distribution—to  $\tilde{O}(1/\delta)$ , by the famous Aldous bounds (Aldous 1982). This quantity will later bound the complexity of classical agents. In contrast, for  $U_P$ , we have that its non-zero eigenphases  $\theta$  are not smaller than  $\tilde{O}(1/\sqrt{\delta})$ . This quadratic difference between the inverse spectral eigenvalue gap in the classical case and the eigenphase gap in the quantum case is at the crux of all speed-ups.

In Magniez *et al* (2011), it was shown how the above properties of  $U_P$  can be used to construct a quantum operator  $R(\pi) \approx \mathbb{1} - 2|\pi\rangle\langle\pi|$ , which exponentially efficiently approximates the reflection over the encoding of the stationary distribution  $|\pi\rangle$ . The basic idea in the construction of  $R(\pi)$  is to apply phase estimation on  $U_P$  with precision high enough to detect non-zero phases, impose a global phase on all states with a non-zero detected phase, and undo the process.

Due to the quadratic relationship between the inverse spectral gap and the smallest eigenphase, this can be achieved in time  $\tilde{O}(1/\sqrt{\delta})$ . That is, we can reflect over the (coherent encoding of the) stationary distribution, whereas obtaining it by classical mixing takes  $\tilde{O}(1/\delta)$  applications of the classical walk operator. In Paparo *et al* (2014) this was used to obtain quadratically accelerated deliberation times for the rPS agent. In the rPS model, the ECM network has a special structure, enforced by the update rules. In particular, for each percept  $s$  we can consider the subnetwork  $\text{ECM}_s$ , which collects all the clips one can reach starting from  $s$ . By construction, it contains all the action clips, but also other intermediary clips. The corresponding Markov chain  $P_s$ , governing the dynamics of  $\text{ECM}_s$ , is irreducible, aperiodic and time-reversible. In the deliberation process, given percept  $s$ , the corresponding Markov chain  $P_s$  mixes, a clip is sampled from the stationary distribution, and then output, provided it is an action clip. If it is not, the process is repeated.

Computationally speaking, we are facing the problem of outputting a single sample, clip  $c$ , drawn according to the conditional probability distribution  $p(c) = \pi_c/\epsilon$  if  $c \in \mathcal{A}$  and  $p(c) = 0$  otherwise. Here  $\epsilon$  is the total weight of all action clips in  $\pi$ . The classical computational complexity of this task is given by the product of  $\tilde{O}(1/\delta)$ , which is the mixing cost, and  $O(1/\epsilon)$  which is the average time needed to actually hit an action clip. Using the Szegedy quantum walk techniques, based on constructing the reflector  $R(\pi)$ , followed by an amplitude amplification algorithm to ‘project’ onto the action space, we obtain a quadratically better complexity of  $\tilde{O}(1/\sqrt{\delta}) \times O(1/\sqrt{\epsilon})$ . In full detail, this is achievable if we can generate one copy of the coherent encoding of the stationary distribution efficiently at each step, and in the context of

<sup>137</sup> In full detail, these relations hold whenever the MC is lazy (all states transition back to themselves with probability at least 1/2), ensuring that all the eigenvalues are non-negative, which can be ensured by adding the identity transition with probability 1/2. This slows down mixing and hitting processes by an irrelevant factor of 2.

the rPS this can be done in many cases as was shown in Paparo *et al* (2014) and further generalized in Dunjko and Briegel (2015a) and Dunjko and Briegel (2015b).

The proposal in Paparo *et al* (2014) was the first example of a provable quantum speed-up in the context of RL<sup>138</sup> and it was followed up by a proposal for an experimental demonstration (Dunjko *et al* 2015a), which identified a possibility of a modular implementation based on coherent controlization—the process of adding control to *almost* unknown unitaries.

It is worthwhile to note that further progress in algorithms for quantum walks and quantum Markov chain theory has the potential to lead to quantum improvements of the PS model. This to an extent mirrors the situation in quantum ML, where new algorithms for quantum linear algebra may lead to quantum speed-ups of other supervised and unsupervised algorithms.

Computational speed-ups of deliberation processes in learning scenarios are certainly important, but in strict RL paradigm such internal processing does not matter and the learning efficiency depends only on the number of interaction steps needed to achieve high quality performance. Since the rPS and its quantum analog, the so-called quantum rPS agent, are, by construction, behaviorally equivalent (i.e. they perform the same action with the same probability, given identical histories), their learning efficiency is the same. The same, however, holds in the context of all the supervised learning algorithms we discussed in previous sections, where the speed-ups were in the context of computational complexity. In contrast, quantum COLT learning results did demonstrate improvements in sample complexity, as discussed in section 6.1.

While formally distinct, computational and sample complexity can become more closely related the moment the learning settings are made more realistic. For instance, if the training of a given SVM requires the solution of a BQP complete problem<sup>139</sup>, classical machines will most likely be able to run only classification instances which are uselessly small. In contrast, a quantum computer could run such a quantum-enhanced learner. The same observation motivates most research into quantum annealers for ML; see section 6.3.1.

In Paparo *et al* (2014), similar ideas were more precisely formalized in the context of active RL, where the interaction is occurring relative to some external real time. This is critical, for instance, in settings where the environment changes relative to this real time, which is always the case in reality. If the deliberation time is slow relative to this change, the agent perceives a ‘blurred’, time-averaged environment where one cannot learn. In contrast, a faster agent will have time to learn before the environment changes—and this makes a qualitative difference between the two agents. In the next section we will show how actual learning efficiency, in the rigid

<sup>138</sup> We point out that the first ideas suggesting that quantum effects could be useful had been previously suggested in Dong *et al* (2005).

<sup>139</sup> BQP stands for *bounded-error quantum polynomial* and collects decision problems which can be solved with bounded error using a quantum computer. Complete problems of a given class are, in a sense, the hardest problems in that class, as all others are reducible to the complete instances using weaker reductions. In particular, it is not believed that BQP complete problems are solvable on a classical computer, whereas all decision problems solvable by classical computers do belong to the class BQP.

metronomic turn-based setting, can also be improved, under stronger assumptions.

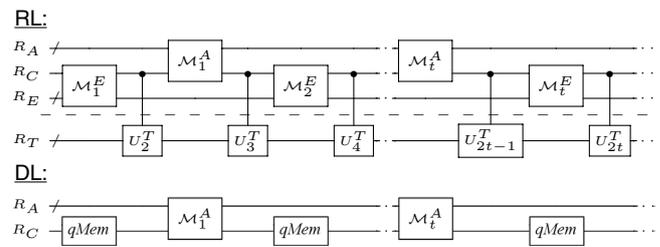
As mentioned, works which directly apply quantum techniques to RL, or other interactive modes of learning, are comparatively few in numbers, despite the ever-growing importance of RL. These results still constitute quite isolated approaches, and we briefly review two recent papers. In Crawford *et al* (2016) the authors design an RL algorithm based on a deep Boltzmann machine and combine this with quantum annealing methods for training such machines to achieve a possible speed-up. In related work by Neukart *et al* (2017), the authors demonstrate how to embed particular policy iteration algorithms (based on Monte Carlo techniques) into the D-Wave architecture. These works combine multiple interesting ideas and may be particularly relevant in the light of recent advances in quantum annealing architectures. Related to quantum RL settings where the interaction is also quantized, in Lamata (2017) the authors demonstrated certain building blocks of larger quantum RL agents in systems of superconducting qubits.

## 7.2. Quantum agent-environment paradigm for reinforcement learning

**Executive summary:** To characterize the ultimate scope and limits of learning agents in quantum environments, one must first establish a framework for quantum agents, quantum environments and their interaction: a quantum AE paradigm. Such a paradigm should maintain the correct classical limit and preserve the critical conceptual components—in particular the history of the agent–environment interaction, which is non-trivial in the quantum case. With such a paradigm in place the potential of quantum enhancements of classical agents is explored, and it is shown that quantum effects, under certain assumptions, can help near-generically improve the learning efficiency of agents. A by-product of the quantum AE paradigm is a classification of learning settings, which is different and complementary to the classification stemming from a supervised learning perspective.

The topics of learning agents acting in quantum environments and the more general questions of the how agent–environment interactions should be defined have to this day only been broached in a few works by the authors of this review and other co-authors. As these topics may form the general principles underlying the upcoming field of quantum AI, we take the liberty of presenting them in substantial detail.

Motivated by the pragmatic question of the potential of quantum enhancements in general learning settings, in Dunjko *et al* (2016) it was suggested that the first step should be the identification of a quantum generalization of the AE paradigm, which underlies both RL and AI. This is comparatively easy to do in finite-sized, discrete space settings.



**Figure 15.** RL: tested agent–environment interaction suitable for RL. In general, each map of the tester  $U_k^T$  acts on a fresh subsystem of the register  $R_T$ , which is neither under the control of the agent nor the environment. The crossed wires represent multiple systems. DL: the simpler setting of standard quantum ML, where the environmental map is without internal memory, presented in the same framework.

*Quantum agent–environment paradigm.* The (abstract) AE paradigm, roughly illustrated in figure 6, can be understood as a two-party communication scenario, the quantum descriptions of which are well-understood in QIP. In particular, the two players—here the agent and the environment—are modeled as (infinite) sequences of unitary maps  $\{\mathcal{E}_A^i\}_i$  and  $\{\mathcal{E}_E^i\}_i$ , respectively. They both have private memory registers  $R_A$  and  $R_E$ , with matching Hilbert spaces  $H_A$  and  $H_E$ , and to enable precise specification of how they communicate (and to cleanly delineate the two players), the register of the communication channel,  $R_C$ , is introduced, and it is the register alone which is accessible to *both* players—that is, the maps of the agent act on  $H_A \otimes H_C$  and of the environment on  $H_E \otimes H_C$ <sup>140</sup>. The two players then interact by sequentially applying their respective maps in turn (see figure 15).

To further tailor this fully general setting for the purposes of the AE paradigm, the percept and action sets are promoted to sets of orthonormal vectors  $\{|s\rangle | s \in \mathbf{S}\}$  and  $\{|a\rangle | a \in \mathbf{A}\}$ , which are also mutually orthogonal. These are referred to as classical states. The Hilbert space of the channel is isomorphic to a space spanned by these two sets, so  $H_C = \text{span}\{|x\rangle | x \in \mathbf{S} \cup \mathbf{A}\}$ .

This also captures the notion that the agent/environment only performs one action, or issues one percept, per turn. Without loss of generality, we assume that the reward status is encoded in the percept space. It should be mentioned that the quantum AE paradigm also includes all other quantum ML settings as a special case. For instance, most quantum-enhanced ML algorithms assume access to quantum database—a quantum memory—and this setting is illustrated in figure 15, part DL. The ‘environment’ of the agent here is the database (or whatever the mechanism generating the data may be), the actions are the queries and percepts the data points. Since the quantum database is, without loss of generality, a unitary map (see e.g. qRAM described in section 6.3.2), it requires no additional memory of its own nor does it change over interaction steps<sup>141</sup>.

<sup>140</sup> Other delineations are possible, where the agent and environment have individually defined interfaces—a part of E accessible to A and a part of A accessible to E—leading to a four-partite system, but we will not be considering this here (Dunjko *et al* 2015b).

<sup>141</sup> To avoid possible confusion, the actual memory stored in the database is formally encoded in the map itself, and requires no additional registers.

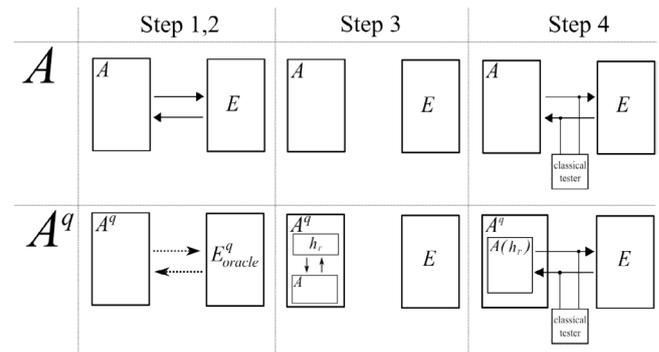
At this point, the classical AE paradigm can be recovered when the maps of the agent and environment are restricted to ‘classical maps’, which, roughly speaking, do not generate superpositions of classical states nor entanglement when applied to classical states.

Further, we now obtain a natural classification of generalized AE settings: CC, CQ, QC and QQ, depending on whether the agent or the environment are classical (C) or quantum (Q). We will come back to this classification in section 7.2.1.

The performance of a learning agent, beyond internal processing time, is a function of the realized *history of interaction*, which is the percept-action sequence (of a given finite length) which has occurred between a given agent and environment<sup>142</sup>. Any *genuine learning-related* figure of merit, for instance, the probability of a reward at a given time-step (efficiency) or number of steps needed before the efficiency is above a threshold (learning speed), is a function of the interaction history. In the classical case, the history can simply be read out by a classical-basis measurement of the register  $H_C$ , as the local state of the communication register is diagonal in this basis and not entangled to the other systems—meaning the measurement does not perturb, i.e. commutes with, the interaction. In the quantum case this is not, in general, the case. To recover a robust notion of a history (needed to gauge the learning), a more detailed description of measurement is used, which captures weaker measurements as well: an additional system, a *tester*, is added, which interchangeably couples to the  $H_C$  register and can copy full or partial information to a separate register. Formally, this is a sequence of controlled maps, relative to the classical basis, controlled by the states on  $H_C$  and acting on a separate register, as illustrated in figure 15. The tester can copy the full information when the maps are generalized controlled-NOT gates—in which case it is called a classical tester—or even do nothing, in which case the interaction is untested. The restriction of the tester to maps which are controlled with respect to the classical basis guarantees that a classical interaction will never be perturbed by its presence. With this basic framework in place, the authors show a couple of basic theorems characterizing when any quantum separations in learning-related figures of merit can be expected at all. The notion of a quantum-classical separation here is essentially the same as in the context of oracular computation, or quantum PAC theory: a separation means no classical agent could achieve the same performance. The authors prove basic expected theorems: quantum improvements (separations) require a genuine quantum interaction and, further, full classical testing prohibits this. Further, they show that, for any specification of a classical environment, there exists a ‘quantum implementation’—a sequence of maps  $\{\mathcal{E}_E^i\}_i$ —which is consistent with the classical specification and prohibits any quantum improvements.

*Provable quantum improvements in reinforcement learning.* However, if the above no-go scenarios are relaxed, much can be achieved. The authors provide a structure of task environments (roughly speaking, maze-type problems), specification

<sup>142</sup> In the case where either the agent or environment are stochastic, different histories of interaction can occur with different probabilities. The possible distribution over histories is sometimes also referred to as the history of interaction.



**Figure 16.** The interactions for the classical ( $A$ ) and quantum-enhanced classical agent ( $A^q$ ). In Steps 1 and 2,  $A^q$  uses quantum access to an oracularized environment  $E_{\text{oracle}}^q$  to obtain a rewarding sequence  $h_r$ . Step 3:  $A^q$  simulates the agent  $A$  and ‘trains’ the simulation to produce the rewarding sequence. In Step 4,  $A^q$  uses the pre-trained agent for interactions with the classical environment  $E$ , which are now classically tested. Adapted from Dunjko *et al* (2016).

of quantum-accessible realizations of these environments and a sporadic tester (which leaves a part of the interaction untested), for which classical learning agents can often be quantum-enhanced.

The idea has a few steps, which we only very briefly sketch out. As a first step, the environments considered are deterministic and strictly episodic—this means the task is reset after some  $M$  steps. Since the environments are deterministic, whether or not rewards are given depends *only* on the sequence of actions, as the interlacing percepts are uniquely specified. Since everything is reset after  $M$  steps there are no correlations in the memory of the environment between the blocks, i.e. episodes. This allows for the specification of a quantum version of the same environment, which can be accessed in superpositions and which takes blocks of actions and returns the same sequence plus a reward status—moreover, it can be realized such that it is self-inverse<sup>143</sup>. With access to such an object, a quantum agent can actually Grover-search for an example of a winning sequence.

To convert this exploration advantage to a learning advantage, the set of agents and environments is restricted to pairs which are ‘luck-favoring’, i.e. those where better performance in the past implies improved performance in the future, relative to a desired figure of merit. Under these conditions, any learning agent which is luck-favoring relative to a given environment can be quantum enhanced by first using quantum access to quadratically faster find the first winning instance, which is then used to ‘pre-train’ the agent in question. The overall quantum-enhanced agent provably outperforms the basic classical agent.

The construction is illustrated in figure 16. These results can be generalized to a broader class of environments. In more recent work, the authors have also applied a similar approach to provide improvements in the context of meta-learning (Dunjko *et al* 2017). Further, by showing how to embed oracular identification problems into Markov decision processes, in Dunjko *et al* (2017) the authors have provided constructions of RL task environments which satisfy certain criteria for genuine RL

<sup>143</sup> This realization is possible under a couple of technical assumptions, for details see Dunjko *et al* Briegel (2015b).

problems, while still allowing for a provable super-polynomial and exponential separation between quantum agents and optimal classical agents. These results showed that even in genuine RL problems, better-than-quadratic improvements are possible.

Although these results form the first examples of quantum improvements in learning figures of merit in RL contexts, the assumptions of having access to ‘quantized’ environments of the type used—in essence, the amount of quantum control the agent is assumed to have—are quite restrictive from a practical perspective. The questions of minimal requirements and the questions of the scope of improvements possible are still unresolved.

At this point, it should be highlighted that the most probable application of results where quantum environments are assumed is in the contexts of model-based learning, learning with off-line pre-training and learning in simulations. In all these settings, the learning interaction is performed with a virtual, rather than a real, environment, which opens doors for the techniques described in this section. In model-based learning, the agent builds up an internal representation of the environment, which allows, e.g., planning. Since this representation is internal to the agent, it certainly can be quantized (provided sufficiently large quantum devices are available). An example of off-line pre-training is the AlphaGo system, in particular, the AlphaGo zero variant (Silver *et al* 2017a, 2017b). The AlphaGo system learns via self-play: an instance of the model plays against itself. Only after an extensive training phase is the system confronted with a human (or software) adversary. Such pre-training is again internal to the agent and can in principle be quantized. Similarly, beyond self-play, training can be performed using a simulation: for instance, in the precursor to the AlphaGo system, a learning machine was trained to play many Atari games at superhuman level (Mnih *et al* 2015). Although, once trained, the machine could in principle be used to build a robot to play on an actual computer, all the training and evaluation is performed relative to an emulation of the Atari system. Again, such emulations or simulations can be quantized, as they can be realized internal to the agent.

**7.2.1. Agent–environment-based classification of quantum machine learning.** The AE paradigm is typically encountered in the contexts of RL, robotics and more general AI settings, while it is less common in ML communities. Nonetheless, conventional ML scenarios can naturally be embedded in this paradigm, since it is, ultimately, mostly unrestrictive. For instance, supervised learning can be thought of as an interaction with an environment which is, for a certain number of steps, an effective database (or the underlying process, generating the data) providing training examples. After a certain number of steps, the environment starts providing unlabeled data points and the agent responds with the labels. If we further assume the environment additionally responds with the correct label to whatever the agent sent, when the data point/percept was from the training set, we can straightforwardly read out the empirical risk (training set error) from the history. Since the quantization of the AE paradigm naturally leads to four settings—CC, CQ, QC and QQ—depending on whether the agent, or environment, or both are fully quantum systems, we can classify all of the results in quantum ML into one of the four groups. Such coarse-grained division places standard ML in CC, results on using ML to control quantum systems in CQ, quantum speed-ups in

ML algorithms (without a quantum database, as is the case in annealing approaches) in QC and quantum ML/RL where the environments, databases or oracles are quantum-accessible are QQ. This classification is closely related to the classification introduced in Aïmeur *et al* (2006), which uses the  $L_{\text{goal}}^{\text{context}}$  notation, where ‘context’ may denote whether we are dealing with classical or quantum data and/or learner and ‘goal’ specifies the learning task (see section 5.1.1 for more details).

The quantum-AE-based separation is not, however, identical to the  $L_{\text{goal}}^{\text{context}}$  systematics. For instance, classical learning tasks may require quantum access or classical access, depending on the details of the setting. Quantum access is required in most quantum ML algorithms which rely on a quantum database. Thus, in the quantum AE paradigm, even when we deal with a classical ML problem, we may be facing a QQ scenario. However, if one considers the bigger picture, the database itself must be pre-filled at some point using classical interfaces. If this is included in the description of the same quantum ML algorithm, we obtain a QC setting (which now may fail to be as efficient). Both viewpoints make sense depending on the scenario which we choose to consider, which may not be obvious in the classification of Aïmeur *et al* (2006). On the other hand, the  $L_{\text{goal}}^{\text{context}}$  systematics does a nice job separating classical ML from quantum generalizations of ML, discussed in section 5.

This mismatch between these two classifications also illustrates the difficulties one encounters if a sufficiently coarse-grained classification of the quantum ML field is required. The classification criteria of this field, and also aspects of QAI, in this review have been inspired by both the AE-induced criteria (perhaps natural from a physics perspective) and the  $L_{\text{goal}}^{\text{context}}$  classification (which is more objective driven, and natural from a computer science perspective).

### 7.3. Towards quantum artificial intelligence

**Executive summary:** Can quantum computers help us build (quantum) AI? The answer to this question cannot be simpler than the answer to the deep, and largely open, question of what intelligence is in the first place. Nonetheless, at least for very pragmatic readings of AI, early research directions into what QAI may be in the future can be identified. We have seen that quantum ML enhancements and generalizations cover data analysis and pattern matching aspects. Quantum RL demonstrates how interactive learning can be quantum-enhanced. General QC can help with various planning, reasoning and similar symbol manipulation tasks which intelligent agents seem to be good at. Finally, the quantum AE paradigm provides a framework for the design and evaluation of whole quantum agents, built also from quantum-enhanced subroutines. These conceptual components, along with many components which are yet to be developed, will form a basis for a behavior-based theory of quantum-enhanced intelligent agents.

AI is quite a loaded concept, in a manner in which ML is not. The question of how genuine AI can be realized is likely to be as difficult as the more basic question of what intelligence is at all, which has been puzzling philosophers and scientists for centuries. Starting a broad discussion of when *quantum* AI will be reached, and what it will be like, is thus clearly ill-advised. We can nonetheless provide a few less controversial observations. The first observation is the fact that the overall concept of quantum AI might be given multiple meanings. First, it may pertain to a generalization of the very notion of intelligence, in the sense section 5 discusses how classical learning concepts generalize to include genuinely quantum extensions. A second, and a perhaps more pragmatic, reading of quantum AI may ask whether quantum effects can be utilized to generate more intelligent agents, where the notion of intelligence itself is not generalized: quantum-enhanced AI. We will focus on this latter reading for the remainder of this review, as quantum generalization of basic learning concepts on its own, just as the notion of intelligence on its own, seems complicated enough.

To comment on the question of quantum-enhanced AI, we first remind the reader that the conceptual debates in AI often have two perspectives. The ultimately pragmatic perspective is concerned only with behavior in relevant situations. This is perhaps best captured by Alan Turing, who suggested that it may be irrelevant what intelligence is if it can be recognized by virtue of similarity to a ‘prototype’ of intelligence—a human (Turing 1950)<sup>144</sup>. Another perspective tends to try to capture cognitive architectures, such as *SOAR* developed by Laird (2012). Cognitive architectures try to identify the components needed to build intelligent agents capable of many tasks and thus also care about *how* intelligence is implemented. They often also serve as models of human cognition and are both theories of what cognition is and of how to implement it. A third perspective comes from the practitioners of AI who often believe that AI will be a complicated combination of various methods and techniques including learning and specialized algorithms, but are also sympathetic to the Turing test as the definitional method. A simple reading of this third perspective is particularly appealing, as it allows us to all but equate computation, ML and AI. Consequently, all quantum ML algorithms and, even broader, even most quantum algorithms already constitute progress in quantum AI. Aspects of such reading can be found in a few works on the topic (Sgarbas 2007, Wichert 2014, Moret-Bonillo 2015)<sup>145</sup>.

The current status of the broad field of quantum ML and related research is showing signs of activity with respect to all of the three aspects mentioned. The substantial activity in the context of ML improvements, in all aspects presented, is certainly filling the toolbox of methods which one day may play a role in the complicated designs of quantum AI practitioners.

<sup>144</sup> Interestingly, the Turing test assumes that humans are good supervised learners of the concept of ‘intelligent agents’, all the while being incapable of specifying the classifier—the definition of intelligence—explicitly.

<sup>145</sup> It should be mentioned that some of the early discussions on quantum AI also consider the possibilities that human brains utilize some form of quantum processing, which may be at the crux of human intelligence. Such claims are still highly hypothetical, and not reviewed in this work.

In this category, a relevant role may also be played by various algorithms which may help in planning, pruning, reasoning via symbol manipulation and other tasks which AI practice and theory encounters. Many possible quantum algorithms which may be relevant come to mind. Examples include the algorithm for performing Bayesian inference (Low *et al* 2014) and algorithms for quadratic and super-polynomial improvements in NAND- and boolean-tree evaluations, which are important in evaluation of optimal strategies in two-player games<sup>146</sup> (Farhi *et al* 2008, Childs *et al* 2009, Zhan *et al* 2012). Furthermore, even more exotic ideas such as quantum game theory (Eisert *et al* 1999) may be relevant. Regarding approaches to quantum AGI and, related, to quantum cognitive architectures, while no proposals exist that explicitly address this possibility, the framework of PS offers sufficient flexibility and structure that it may be considered a good starting point. Further, this framework has so far maintained a homogeneous structure, preparing the grounds for a more straightforward global quantization, in comparison to models which are built out of inhomogeneous blocks—already in classical systems the performance of a system combined out of inhomogeneous units may lead to difficult-to-control behavior and it stands to reason that it may be more difficult to synchronize quantum devices. It should be mentioned that recently there have been works providing a broad framework describing how composite large quantum systems can be precisely treated (Portmann *et al* 2017). Finally, from the ultimate pragmatic perspective, the quantum AE paradigm presented can offer a starting point for a quantum-generalized Turing test for QAI, as the Turing test itself fits in the paradigm: the environment is the administrator of the test and the agent is the machine trying to convince the environment it is intelligent. Although at the moment the only suitable referees for such a test are classical devices—humans—it may be conceivable that they, too, may find quantum gadgets useful to better ascertain the nature of the candidate<sup>147</sup>. However, at this point it is prudent to remind ourselves and the reader that all the above considerations are still highly speculative and that the research into genuine AI has barely broken ground.

## 8. Outlook

In this review, we have presented an overview of various lines of research that connect the fields of quantum information and QC, on the one hand, and ML and AI, on the other hand. Most of the work in this new area of research is still largely theoretical and conceptual and there are, for example, hardly any dedicated experiments demonstrating how quantum mechanics can be exploited for ML and AI. However, there are a number of theoretical proposals (Friis *et al* 2015, Lamata 2017, Dunjko *et al* 2015a) and also first experimental works showing how these ideas can be implemented in the laboratory (Neigovzen *et al* 2009, Li *et al* 2015b, Cai *et al* 2015,

<sup>146</sup> See [www.scottaaronson.com/blog/?p=207](http://www.scottaaronson.com/blog/?p=207) for a simple explanation.

<sup>147</sup> This is reminiscent of the problem of quantum verification, where quantum Turing test is a term used for the test which efficiently decides whether the agent is a genuine quantum device/computer (Kashefi 2013).

Ristè *et al* 2017)<sup>148</sup>. At the same time it is clear that certain quantum technologies, which have been developed in the context of QIP and QC, can be readily applied to quantum learning, to the extent that learning agents or algorithms employ elements of QIP in their very design. Similarly, it is clear, and there are by now several examples, how techniques from classical ML can be fruitfully employed in data analysis and the design of experiments in quantum many-body physics (see section 4.4). One may ask about the long-term impact of the exchange of concepts and techniques between QM and ML/AI. What implications will this exchange have on the development of the individual fields, and what is the broader perspective of these individual activities leading towards a new field of research, with its own questions and promises? Indeed, returning the focus back to the topics of this review, we can highlight one overarching question encapsulating the collective effort of the presented research:

⇒ *What is the potential and what are the limitations of an interaction between quantum physics and ML and AI?*

From a purely theoretical perspective, we can learn from analogies with the fields of communication, computation or sensing. QIP has shown that to understand the limits of such information processing disciplines, both in pragmatic and conceptual senses, one must consider the full extent of quantum theory. Consequently, we should expect that the limits of learning and of intelligence can also only be fully answered in this broader context. In this sense, the topics discussed in section 5 already point to the rich and complex theory describing what learning may be, when even information itself is a quantum object, and aspects of section 7.3 point to how a general theory of quantum learning may be set-up. While the motivation for establishing such a general theory may be fundamental, it may also have more pragmatic consequences. In fact, arguments can be made that the field of quantum ML and the future field of quantum AI may constitute one of the most important research developments to emerge in recent times. A part of the reason behind such a bold claim stems from the obvious potential of both directions of influence between the two constituent sides of quantum learning (and quantum AI). For instance, the potential of quantum enhancements for ML is profound. In a society where data is generated at a geometric rate<sup>149</sup> and where its understanding may help us combat global problems, the potential of faster, better analyses cannot be overestimated. Conversely, ML and AI technologies are becoming indispensable tools in all high technologies, but they are also showing the potential to help us do research in a novel, better way. A more subtle reason supporting optimism lies in positive feedback loops between ML, AI and QIP which are becoming apparent, and which are, moreover, specific to these two disciplines. To begin with, we can claim that QC, once realized, will play an integral part in future AI systems, on general grounds. This can be deduced from even a cursory

overview of the history of AI, which reveals that qualitative improvements in computing and information technologies result in progress in AI tasks, which is also intuitive. In simple terms, state-of-the-art in AI will always rely on state-of-the-art in computing. In contrast, ML and AI technologies are becoming indispensable tools in all high technologies.

The perfect match between ML, AI and QIP, however may have deeper foundations. In particular,

→ *advancements in ML/AI may help with critical steps in the building of quantum computers.*

In recent times, it has become ever more apparent that learning methods may make the difference between a given technology being realizable or being effectively impossible—beyond obvious examples, for instance direct computational approaches to build a human-level Go-playing software had failed, whereas AlphaGo (Silver *et al* 2016, 2017a, 2017b), a fundamentally learning AI technology, achieved this complex goal. QC may in fact end up being such a technology, where exquisite fast and adaptive control—realized by an autonomous smart laboratory perhaps—helps mitigate the hurdles towards quantum computers. However, cutting edge research discussed in sections 4.3 and 4.4 suggests that ML and AI techniques could help at an even deeper level, by helping us discover novel techniques that may provide the missing link for full blown quantum technologies. Thus ML and AI may be what we need to build quantum computers.

Another observation, which is hinted at with increasing frequency in the community, and which fully entwines ML, AI and QIP, is that

→ *AI/ML applications may be the best reasons to build quantum computers.*

Quantum computers have been proven to dramatically outperform their classical counterparts only on a handful of problems. Perhaps the best applications of quantum computers that have enticed investors until recently were quantum simulation and quantum cryptology (i.e. using QC to break encryption), which may have been simply insufficient to stimulate broad-scale public investments. In contrast ML- and AI-type tasks may be regarded as the ‘killer applications’ QC has been waiting for. However, not only are ML and AI applications well motivated, but in recent times arguments have been put forward that ML-type applications may be uniquely suited to be tackled by quantum technologies. For instance, ML-type applications deal with massive parallel processing of high dimensional data—quantum computers seem to be good for this. Further, while most simulation and numerical processing tasks require data stability, which is incompatible with the noise modern days quantum devices undergo, ML applications always work with noisy data. This means that such an analysis makes sense only if it is robust to noise to start with, which is the often unspoken fact of ML: the important features are the robust features. Under such laxer sets of constraints on the desired information processing, various current-day technologies such as quantum annealing methods may become a possible solution.

<sup>148</sup> These complement the experimental work based on superconducting quantum annealers (Neven *et al* 2009a, Adachi and Henderson 2015), which is closely related to one of the approaches to QML.

<sup>149</sup> <https://insidebigdata.com/2017/02/16/the-exponential-growth-of-data/> (accessed July 2017).

The two main flavors, or directions of influence, in quantum ML thus have a natural synergistic effect. This is a further motivation that, despite their quite fundamental differences, both directions of QML should be investigated in close collaboration. Naturally, at the moment, each individual sub-field of quantum ML comes with its own set of open problems, key issues which need to be resolved before any credible verdict on the future of quantum ML can be made. Most fit into one of the two quintessential categories of research into topics considering quantum-enhancements: (a) what are the limits/how much of an edge over best classical solutions can be achieved and (b) could the proposals be implemented in practice in any reasonable term. For most of the topics discussed, both questions above remain widely open. For instance, regarding quantum enhancements using universal computation, only a few models have been beneficially quantized, and the exact problem they solve, even in theory, does not match the best established methods used in practice. Regarding the second facet, the most impressive improvements (barring isolated exceptions) can be achieved only under a significant number of assumptions, such as quantum databases, and certain suitable properties of the structure of the datasets<sup>150</sup>. Beyond particular issues which were occasionally pointed out in various parts of this review, we will forgo providing an extensive list of specific open questions for each of the research lines and refer the interested reader to the more specialized reviews for more detail (Wittek 2014a, Schuld *et al* 2014b, Biamonte *et al* 2016, Arunachalam and de Wolf 2017, Ciliberto *et al* 2017).

This leads us to the final topic of speculation of this outlook section: whether QC will truly be instrumental in the construction of genuine artificial (general) intelligence. On one hand, there is no doubt that quantum computers could help in heavily computational problems one typically encounters in, e.g., ML. In so far as AI reduces to sets of ML tasks, QC may help. However, we have ample evidence that AI is more than a sum of such specific-task-solving parts, and in this sense even radical (quantum) speed-ups in the solving of such tasks may yield only a limited progress into the design of artificially intelligent systems. For instance, human brains are (usually) taken as a reference for systems capable of generating intelligent behavior. Yet there is little, and no non-controversial, reason to believe genuine quantum effects play any critical part in their performance (rather, there are ample reasons to dismiss the relevance of quantum effects), and thus whatever makes us intelligent most likely is not just a consequence of mere computational speed. In other words, quantum computers may not be necessary for general AI. The extent to which quantum mechanics has something to say about general AI will be the subject of research in years to come. Nonetheless, we can already set aside any doubt that quantum computers and AI can help each other, to an extent which will not be disregarded.

<sup>150</sup>In many proposals, the condition number of a matrix depending on the dataset explicitly appears in run-time; see section 6.3.2.

## Acknowledgments

The authors are grateful to Walter Boyajian, Jens Clausen, Joseph Fitzsimons, Nicolai Friis, Alexey A Melnikov, Davide Orsucci, Hendrik Poulsen Nautrup, Patrick Reberntrost, Katja Ried, Maria Schuld, Gael Sentís, Omar Shehab, Sebastian Stabinger, Jordi Tura i Brugués, Petter Wittek, Sabine Wölk and an anonymous referee for helpful comments to various parts of the manuscript. This work was supported in part by the Austrian Science Fund (FWF) through the SFB FoQuS F4012, and by the Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg (AZ: 33-7533.-30-10/41/1).

## ORCID iDs

Vedran Dunjko  <https://orcid.org/0000-0002-2632-7955>

Hans J Briegel  <https://orcid.org/0000-0002-9065-1565>

## References

- Aaronson S 2007 The learnability of quantum states *Proc. R. Soc. A* **463** 3089–114
- Aaronson S 2015 Read the fine print *Nat. Phys.* **11** 291–3
- Aaronson S and Arkhipov A 2011 The computational complexity of linear optics *Proc. of the 43rd Annual ACM Symp. on Theory of Computing* (New York: ACM) pp 333–42
- Aaronson S, Ben-David S and Kothari R 2016 Separations in query complexity using cheat sheets *Proc. of the 48th Annual ACM Symp. on Theory of Computing* (New York: ACM) pp 863–76
- Adachi S H and Henderson M P 2015 Application of quantum annealing to training of deep neural networks (arXiv:1510.06356)
- Aharonov D, Jones V and Landau Z 2006 A polynomial quantum algorithm for approximating the Jones polynomial *Proc. of the 38th Annual ACM Symp. on Theory of Computing* (New York: ACM) pp 427–36
- Aïmeur E, Brassard G and Gambs S 2006 *Machine Learning in a Quantum World* (Berlin: Springer) pp 431–42
- Aïmeur E, Brassard G and Gambs S 2013 Quantum speed-up for unsupervised learning *Mach. Learn.* **90** 261–87
- Aldous D J 1982 Some inequalities for reversible Markov chains *J. Lond. Math. Soc.* **25** 564–76
- Alpaydin E 2010 *Introduction to Machine Learning* 2nd edn (Cambridge, MA: MIT Press)
- Altaisky M, Zolnikova N, Kaputkina N, Krylov V, Lozovik Y E and Dattani N 2017 Entanglement in a quantum neural network based on quantum dots *Photonics Nanostruct.—Fundam. Appl.* **24** 24–8
- Amin M H, Andriyash E, Rolfe J, Kulchitsky B and Melko R 2016 Quantum Boltzmann machine (arXiv:1601.02036)
- Angluin D 1988 Queries and concept learning *Mach. Learn.* **2** 319–42
- Anguita D, Ridella S, Riviello F and Zunino R 2003 Quantum optimization for training support vector machines *Neural Netw.* **16** 763–70 (*Advances in Neural Networks Research: {IJCNN}*)
- Arsenault L-F, Lopez-Bezanilla A, von Lilienfeld O A and Millis A J 2014 Machine learning for many-body physics: The case of the anderson impurity model *Phys. Rev. B* **90** 155136
- Arulkumaran K, Deisenroth M P, Brundage M and Bharath A A 2017 A brief survey of deep reinforcement learning **24** 26–38
- Arunachalam S and de Wolf R 2016 Optimal quantum sample complexity of learning algorithms (arXiv:1607.00932)

- Arunachalam S and de Wolf R 2017 A survey of quantum learning theory *CoRR* (arXiv:1701.06806)
- Atıcı A and Servedio R A 2007 Quantum algorithms for learning and testing juntas *Quantum Inf. Process.* **6** 323–48
- August M and Ni X 2017 Using recurrent neural networks to optimize dynamical decoupling for quantum memory *Phys. Rev. A* **95** 012335
- Babbush R, Denchev V, Ding N, Isakov S and Neven H 2014 Construction of non-convex polynomial loss functions for training a binary classifier with quantum annealing (arXiv:1406.4203)
- Baireuther P, O’Brien T E, Tarasinski B and Beenakker C W J 2018 Machine-learning-assisted correction of correlated qubit errors in a topological code *Quantum* **2** 48
- Banchi L, Pancotti N and Bose S 2016 Quantum gate learning in qubit networks: Toffoli gate without time-dependent control *NPJ Quantum Inf.* **2** 16019
- Bang J, Lim J, Kim M S and Lee J 2008 Quantum learning machine (arXiv:0803.2976)
- Bar-Yossef Z, Jayram T S and Kerenidis I 2008 Exponential separation of quantum and classical one-way communication complexity *SIAM J. Comput.* **38** 366–84
- Barry J, Barry D T and Aaronson S 2014 Quantum partially observable markov decision processes *Phys. Rev. A* **90** 032311
- Bartlett P L and Mendelson S 2003 Rademacher and gaussian complexities: Risk bounds and structural results *J. Mach. Learn. Res.* **3** 463–82
- Bears R, Buhrman H, Cleve R, Mosca M and de Wolf R 2001 Quantum lower bounds by polynomials *J. ACM* **48** 778–97
- Behrman E, Niemel J, Steck J and Skinner S 1996 A quantum dot neural network *Proc. 4th Workshop on Physics of Computation* pp 22–4
- Ben-David S, Srebro N and Urner R 2011 Universal learning versus no free lunch results *Workshop at NIPS 2011*
- Benedetti M, Realpe-Gómez J, Biswas R and Perdomo-Ortiz A 2016 Estimation of effective temperatures in quantum annealers for sampling applications: a case study with possible applications in deep learning *Phys. Rev. A* **94** 022308
- Benedetti M, Realpe-Gómez J, Biswas R and Perdomo-Ortiz A 2017 Quantum-assisted learning of hardware-embedded probabilistic graphical models *Phys. Rev. X* **7** 041052
- Bengio Y and Delalleau O 2009 Justifying and generalizing contrastive divergence *Neural Comput.* **21** 1601–21
- Benjamin S C and Bose S 2003 Quantum computing with an always-on heisenberg interaction *Phys. Rev. Lett.* **90** 247901
- Bergou J A and Hillery M 2005 Universal programmable quantum state discriminator that is optimal for unambiguously distinguishing between unknown states *Phys. Rev. Lett.* **94** 160501
- Bernstein E and Vazirani U 1997 Quantum complexity theory *SIAM J. Comput.* **26** 1411–73
- Berry D W and Wiseman H M 2000 Optimal states and almost optimal adaptive measurements for quantum interferometry *Phys. Rev. Lett.* **85** 5098–101
- Berry D W, Childs A M and Kothari R 2015 Hamiltonian simulation with nearly optimal dependence on all parameters *IEEE 56th Annual Symp. on Foundations of Computer Science*
- Berry D W, Wiseman H M and Breslin J K 2001 Optimal input states and feedback for interferometric phase estimation *Phys. Rev. A* **63** 053804
- Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195–202
- Bian Z, Chudak F, Macready W G and Rose G 2010 The Ising model: teaching an old problem new tricks ([www.dwavesys.com/sites/default/files/weightedmaxsat\\_v2.pdf](http://www.dwavesys.com/sites/default/files/weightedmaxsat_v2.pdf))
- Bisio A, Chiribella G, D’Ariano G M, Facchini S and Perinotti P 2010 Optimal quantum learning of a unitary transformation *Phys. Rev. A* **81** 032324
- Bisio A, D’Ariano G M, Perinotti P and Sedláč M 2011 Quantum learning algorithms for quantum measurements *Phys. Lett. A* **375** 3425–34
- Boixo S, Isakov S V, Smelyanskiy V N, Babbush R, Ding N, Jiang Z, Bremner M J, Martinis J M and Neven H 2016 Characterizing quantum supremacy in near-term devices (arXiv:1608.00263)
- Brandão F G S L, Kalev A, Li T, Lin C Y Y, Svore K M and Wu X 2017 Exponential quantum speed-ups for semidefinite programming with applications to quantum learning (arXiv:1710.02581)
- Brandão F G S L and Svore K M 2017 Quantum speed-ups for solving semidefinite programs *58th IEEE Annual Symp. on Foundations of Computer Science (Berkeley, CA, USA, 15–17 October 2017)*
- Braunstein S L and Caves C M 1994 Statistical distance and the geometry of quantum states *Phys. Rev. Lett.* **72** 3439–43
- Bravyi S, Gosset D and Koenig R 2017 Quantum advantage with shallow circuits (arXiv:1704.00690)
- Bremner M J, Montanaro A and Shepherd D J 2017 Achieving quantum supremacy with sparse and noisy commuting quantum computations *Quantum* **1** 8
- Briegel H J 2012 On creative machines and the physical origins of freedom *Sci. Rep.* **2** 522
- Briegel H J 2013 *Projective Simulation for Classical and Quantum Autonomous Agents. Talk delivered at the KITP Program Control of Complex Quantum Systems, (Santa Barbara)*
- Briegel H J and De las Cuevas G 2012 Projective simulation for artificial intelligence *Sci. Rep.* **2** 400
- Briegel H J and Raussendorf R 2001 Persistent entanglement in arrays of interacting particles *Phys. Rev. Lett.* **86** 910–3
- Briegel H J, Browne D E, Dur W, Raussendorf R and Van den Nest M 2009 Measurement-based quantum computation *Nat. Phys.* pp 19–26
- Broadbent A, Fitzsimons J and Kashefi E 2009 Universal blind quantum computation *50th Annual IEEE Symp. on Foundations of Computer Science*
- Broecker P, Assaad F F and Trebst S 2017 Quantum phase recognition via unsupervised machine learning (arXiv:1707.00663)
- Broecker P, Carrasquilla J, Melko R G and Trebst S 2017 Machine learning quantum phases of matter beyond the fermion sign problem *Sci. Rep.* **7** 8823
- Brooks R A 1990 Elephants do not play chess *Robot. Auton. Syst.* **6** 3–15
- Brun T, Klauck H, Nayak A, Rötteler M and Zalka C 2003 Comment on ‘Probabilistic quantum memories’ *Phys. Rev. Lett.* **91** 209801
- Bshouty N H and Jackson J C 1998 Learning DNF over the uniform distribution using a quantum example oracle *SIAM J. Comput.* **28** 1136–53 (Appeared in *Computational Learning Theory (COLT) Conf. Proc. in 1995*)
- Buhrman H, Cleve R, Watrous J and de Wolf R 2001 Quantum fingerprinting *Phys. Rev. Lett.* **87** 167902
- Bukov M, Day A G R, Sels D, Weinberg P, Polkovnikov A and Mehta P 2017 Machine learning meets quantum state preparation. The phase diagram of quantum control (arXiv:1705.00565)
- Cai X D, Wu D, Su Z E, Chen M C, Wang X L, Li L, Liu N L, Lu C Y and Pan J W 2015 Entanglement-based machine learning on a quantum computer *Phys. Rev. Lett.* **114** 110504
- Carleo G and Troyer M 2017 Solving the quantum many-body problem with artificial neural networks *Science* **355** 602–6
- Carrasquilla J and Melko R G 2017 Machine learning phases of matter *Nat. Phys.* **13** 431–4

- Cartwright H 2007 *Development and Uses of Artificial Intelligence in Chemistry* (New York: Wiley) pp 349–90
- Castelvecchi D 2015 Artificial intelligence called in to tackle LHC data deluge *Nature* **528** 18–9
- Chapelle O, Schölkopf B and Zien A 2010 *Semi-Supervised Learning* 1st edn (Cambridge, MA: MIT Press)
- Chen C, Dong D, Li H X, Chu J and Tarn T J 2014 Fidelity-based probabilistic q-learning for control of quantum systems *IEEE Trans. Neural Netw. Learn. Syst.* **25** 920–33
- Cheng H C, Hsieh M H and Yeh P C 2016 The learnability of unknown quantum measurements *Quantum Inf. Comput.* **16** 615–56
- Childs A M and Goldstone J 2004 Spatial search by quantum walk *Phys. Rev. A* **70** 022314
- Childs A M and van Dam W 2010 Quantum algorithms for algebraic problems *Rev. Mod. Phys.* **82** 1–52
- Childs A M, Cleve R, Deotto E, Farhi E, Gutmann S and Spielman D A 2003 Exponential algorithmic speedup by a quantum walk *Proc. of the 35th Annual ACM Symp. on Theory of Computing* (New York: ACM) pp 59–68
- Childs A M, Cleve R, Jordan S P and Yonge-Mallo D L 2009 Discrete-query quantum algorithm for NAND trees *Theory Comput.* **5** 119–23
- Childs A M, Kothari R and Somma R D 2017 Quantum linear systems algorithm with exponentially improved dependence on precision *SIAM J. Comp.* **46** 1920–50
- Ch'ng K, Carrasquilla J, Melko R G and Khatami E 2017 Machine learning phases of strongly correlated fermions *Phys. Rev. X* **7** 031038
- Ch'ng K, Vazquez N and Khatami E 2018 Unsupervised machine learning account of magnetic transitions in the hubbard model *Phys. Rev. E* **97** 013306
- Chung J, Gülçehre Ç, Cho K and Bengio Y 2014 Empirical evaluation of gated recurrent neural networks on sequence modeling (arXiv:1412.3555) (Presented at the *Deep Learning workshop at NIPS2014*)
- Ciliberto C, Herbster M, Ialongo A D, Pontil M, Rocchetto A, Severini S and Wossnig L 2018 Quantum machine learning: a classical perspective *Proc. R. Soc. A* **474** 20170551
- Clader B D, Jacobs B C and Sprouse C R 2013 Preconditioned quantum linear system algorithm *Phys. Rev. Lett.* **110** 250504
- Clausen J and Briegel H J 2018 Quantum machine learning with glow for episodic tasks and decision games *Phys. Rev. A* **97** 022303
- Cleophas T J and Zwinderman A H 2015 *Machine Learning in Medicine—a Complete Overview* (Berlin: Springer)
- Combes J, Ferrie C, Cesare C, Tiersch M, Milburn G J, Briegel H J and Caves C M 2014 *In situ* characterization of quantum devices with error correction (arXiv:1405.5656)
- Cover T M 1965 Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition *IEEE Trans. Electron. Comput.* **EC-14** 326–34
- Crawford D, Levit A, Ghadermarzy N, Oberoi J S and Ronagh P 2016 Reinforcement learning using quantum Boltzmann machines (arXiv:1612.05695)
- Croke S, Andersson E and Barnett S M 2008 No-signaling bound on quantum state discrimination *Phys. Rev. A* **77** 012113
- Cross A W, Smith G and Smolin J A 2015 Quantum learning robust against noise *Phys. Rev. A* **92** 012327
- Crosson E and Harrow A W 2016 Simulated quantum annealing can be exponentially faster than classical simulated annealing *IEEE 57th Annual Symp. on Foundations of Computer Science*
- Curtarolo S, Morgan D, Persson K, Rodgers J and Ceder G 2003 Predicting crystal structures with data mining of quantum calculations *Phys. Rev. Lett.* **91** 135503
- Cybenko G 1989 Approximation by superpositions of a sigmoidal function *Math. Control Signals Syst.* **2** 303–14
- Dann C and Brunskill E 2015 Sample complexity of episodic fixed-horizon reinforcement learning *Proc. of the 28th Int. Conf. on Neural Information Processing Systems* (Cambridge, MA: MIT Press) pp 2818–26
- de Wolf R 2002 Quantum communication and complexity *Theor. Comput. Sci.* **287** 337–53
- Denchev V S, Ding N, Matsushima S, Vishwanathan S V N and Neven H 2015 Totally corrective boosting with cardinality penalization (arXiv:1504.01446)
- Denchev V S, Ding N, Vishwanathan S V N and Neven H 2012 Robust classification with adiabatic quantum optimization *Proc. 29th Int. Conf. Machine Learning (Edinburgh, Scotland, UK, 2012)*
- Deng D L, Li X and Das Sarma S 2017 Quantum entanglement in neural network states *Phys. Rev. X* **7** 021021
- Deutsch D 1985 Quantum theory, the Church-Turing principle and the universal quantum computer *Proc. R. Soc. A* **400** 97–117
- Devroye L, Györfi L and Lugosi G 1996 *A Probabilistic Theory of Pattern Recognition* (Berlin: Springer)
- Doersch C 2016 Tutorial on variational autoencoders (arXiv:1606.05908)
- Dong D, Chen C and Chen Z 2005 *Quantum Reinforcement Learning* (Berlin: Springer) pp 686–9
- Dunjko V and Briegel H J 2015a Quantum mixing of Markov chains for special distributions *New J. Phys.* **17** 073004
- Dunjko V and Briegel H J 2015b Sequential quantum mixing for slowly evolving sequences of markov chains (arXiv:1503.01334)
- Dunjko V, Friis N and Briegel H J 2015a Quantum-enhanced deliberation of learning agents using trapped ions *New J. Phys.* **17** 023006
- Dunjko V, Taylor J M and Briegel H J 2015b Framework for learning agents in quantum environments (arXiv:1507.08482)
- Dunjko V, Taylor J M and Briegel H J 2016 Quantum-enhanced machine learning *Phys. Rev. Lett.* **117** 130501
- Dunjko V, Taylor J M and Briegel H J 2017 Advances in quantum reinforcement learning *IEEE Int. Conf. on Systems, Man, and Cybernetics* pp 282–7
- Durr C and Hoyer P 1999 A quantum algorithm for finding the minimum (arXiv:quant-ph/9607014)
- Eisert J, Wilkens M and Lewenstein M 1999 Quantum games and quantum strategies 2006 *Phys. Rev. Lett.* **83** 3077–80
- Eliasmith C, Bechtel W *Symbolic Versus Subsymbolic* (New York: Wiley)
- Farhi E, Goldstone J and Gutmann S 2008 A quantum algorithm for the Hamiltonian NAND tree *Theory Comput.* **4** 169–90
- Farhi E, Goldstone J and Gutmann S 2014 A quantum approximate optimization algorithm (arXiv:1411.4028)
- Farhi E, Goldstone J, Gutmann S and Sipser M 2000 Quantum computation by adiabatic evolution (arXiv:quant-ph/0001106)
- Feynman R 1982 Simulating physics with computers *Int. J. Theor. Phys.* **21** 467–88
- Fiurášek J 2001 Maximum-likelihood estimation of quantum measurement *Phys. Rev. A* **64** 024102
- Fiurášek J and Hradil Z 2001 Maximum-likelihood estimation of quantum processes *Phys. Rev. A* **63** 020101
- Freedman M H, Kitaev A, Larsen M J and Wang Z 2002 Topological quantum computation *Bull. Am. Math. Soc.* **40** 31–9
- Freund Y and Schapire R E 1997 A decision-theoretic generalization of on-line learning and an application to boosting *J. Comput. Syst. Sci.* **55** 119–39
- Friis N, Melnikov A A, Kirchmair G and Briegel H J 2015 Coherent controlization using superconducting qubits *Sci. Rep.* **5** 18036
- Gambis S 2008 Quantum classification (arXiv:0809.0444)
- Gammelmark S and Imer K M 2009 Quantum learning by measurement and feedback *New J. Phys.* **11** 033017
- Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M and Lempitsky V 2016 Domain-adversarial training of neural networks *J. Mach. Learn. Res.* **17** 2096–30

- Gao X and Duan L M 2017 Efficient representation of quantum many-body states with deep neural networks *Nat. Comm.* **8** 662
- Garman J A 2011 A heuristic review of quantum neural networks *Master's Thesis* Imperial College, London
- Gavinsky D 2012 Quantum predictive learning and communication complexity with single input *Quantum Info. Comput.* **12** 575–88
- Georgescu I M, Ashhab S and Nori F 2014 Quantum simulation *Rev. Mod. Phys.* **86** 153–85
- Giovannetti V, Lloyd S and Maccone L 2008 Quantum random access memory *Phys. Rev. Lett.* **100** 160501
- Giovannetti V, Lloyd S and Maccone L 2011 Advances in quantum metrology *Nat. Photon.* **5** 222–9
- Glasser I, Pancotti N, August M, Rodriguez I D and Cirac J I 2018 Neural-network quantum states, string-bond states, and chiral topological states *Phys. Rev. X* **8** 011006
- Goodfellow I *et al* (ed) 2014 *Advances in Neural Information Processing Systems* vol 27 (Red Hook, NY: Curran Associates) pp 2672–80
- Granade C E, Ferrie C, Wiebe N and Cory D G 2012 Robust online Hamiltonian learning *New J. Phys.* **14** 103013
- Grilo A B and Kerenidis I 2017 Learning with errors is easy with quantum samples (arXiv:1702.08255)
- Grondman I, Busoni L, Lopes G A D and Babuska R 2012 A survey of actor-critic reinforcement learning: Standard and natural policy gradients *IEEE Trans. Syst. Man Cybern. C* **42** 1291–307
- Grover L K 1996 A fast quantum mechanical algorithm for database search *Proc. of the 28th Annual ACM Symp. on Theory of Computing* (New York: ACM) pp 212–9
- Guță M and Kotłowski W 2010 Quantum learning: asymptotically optimal classification of qubit states *New J. Phys.* **12** 123032
- Hangl S, Dunjko V, Briegel H J and Piater J 2017a Skill learning by autonomous robotic playing using active learning and creativity (arXiv:1706.08560)
- Hangl S, Mennel A and Piater J 2017b A novel skill-based programming paradigm based on autonomous playing and skill-centric testing (arXiv:1709.06049)
- Hangl S, Ugur E and Piater J 2017c Autonomous robots: potential, advances and future direction *e&i Elektrotechnik Informationstechnik* **134** 293–8
- Hangl S, Ugur E, Szedmak S and Piater J 2016 Robotic playing for hierarchical complex skill learning *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* pp 2799–804
- Harrow A W, Hassidim A and Lloyd S 2009 Quantum algorithm for linear systems of equations *Phys. Rev. Lett.* **103** 150502
- Hayashi A, Horibe M and Hashimoto T 2005 Quantum pure-state identification *Phys. Rev. A* **72** 052306
- Hayashi A, Horibe M and Hashimoto T 2006 Unambiguous pure-state identification without classical knowledge *Phys. Rev. A* **73** 012328
- Heim B, Brown E W, Wecker D and Troyer M 2017 Designing adiabatic quantum optimization: a case study for the traveling salesman problem (arXiv:1702.06248)
- Helstrom C W 1969 Quantum detection and estimation theory *J. Stat. Phys.* **1** 231–52
- Hen I, Job J, Albash T, Rønnow T F, Troyer M and Lidar D A 2015 Probing for quantum speedup in spin-glass problems with planted solutions *Phys. Rev. A* **92** 042325
- Hentschel A and Sanders B C 2010 Machine learning for precise quantum measurement *Phys. Rev. Lett.* **104** 063603
- Hentschel A and Sanders B C 2011 Efficient algorithm for optimizing adaptive quantum metrology processes *Phys. Rev. Lett.* **107** 233601
- Hillar C and Tran N M 2018 Robust exponential memory in Hopfield networks *J. Math. Neur.* **8** 1
- Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80
- Hochreiter S, Bengio Y, Frasconi P and Schmidhuber J 2001 Gradient flow in recurrent nets: the difficulty of learning long-term dependencies *A Field Guide to Dynamical Recurrent Neural Networks* ed S C Kremer *et al* (New York: IEEE Press)
- Holevo A 1982 *Probabilistic and Statistical Aspects of Quantum Theory (North-Holland Series in Statistics and Probability)* (Amsterdam: North-Holland)
- Hopfield J J 1982 Neural networks and physical systems with emergent collective computational abilities *Proc. Natl Acad. Sci. USA* **79** 2554–8
- Hopfield J J and Tank D W 1985 ‘Neural’ computation of decisions in optimization problems *Biol. Cybern.* **52** 141–52
- Hornik K 1991 Approximation capabilities of multilayer feedforward networks *Neural Netw.* **4** 251–7
- Hradil Z 1997 Quantum-state estimation *Phys. Rev. A* **55** R1561–R1564
- Hu W, Singh R R P and Scalettar R T 2017 Discovering phases, phase transitions and crossovers through unsupervised machine learning: a critical examination *Phys. Rev. E* **95** 062122
- Huembeli P, Dauphin A and Wittek P 2017 Adversarial domain adaptation for identifying phase transitions (arXiv:1710.08382)
- Hume D 1739 *Treatise on Human Nature* (Oxford: Oxford University Press)
- Hutter M 2005 *Universal Artificial Intelligence* (Berlin: Springer)
- Hutter M 2010 *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability* (Berlin: Springer)
- insideBIGDATA 2017 The exponential growth of data (<https://insidebigdata.com/2017/02/16/the-exponential-growth-of-data/>)
- Jarzyna M and Demkowicz-Dobrzański R 2015 True precision limits in quantum metrology *New J. Phys.* **17** 013010
- Johnson P D, Romero J, Olson J, Cao Y and Aspuru-Guzik A 2017 Qvector: an algorithm for device-tailored quantum error correction (arXiv:1711.02249)
- Kak S 1995 On quantum neural computing *Inf. Sci.* **83** 143–60
- Kakade S 2003 On the sample complexity of reinforcement learning *PhD Thesis* University College London
- Karbasi A, Salavati A H, Shokrollahi A and Varshney L R 2014 Noise facilitation in associative memories of exponential capacity *Neural Comp.* **26** 2493–526
- Kashefi E 2013 *Turing Research Symp. (May 2012)* ([www.youtube.com/watch?v=3y7JCjaNZLY](http://www.youtube.com/watch?v=3y7JCjaNZLY))
- Kashefi E and Pappa A 2016 Multiparty delegated quantum computing (arXiv:1606.09200)
- Kearns M and Valiant L 1994 Cryptographic limitations on learning boolean formulae and finite automata *J. ACM* **41** 67–95
- Kearns M J and Schapire R E 1994 Efficient distribution-free learning of probabilistic concepts *J. Comput. Syst. Sci.* **48** 464–97
- Kempe J 2003 Quantum random walks: an introductory overview *Contemp. Phys.* **44** 307–27
- Kerenidis I and Prakash A 2017 Quantum gradient descent for linear systems and least squares (arXiv:1704.04992)
- Kingma D P and Welling M 2013 Auto-encoding variational bayes *CoRR* (arXiv:1312.6114)
- Kothari R 2014 *Proc. 31st Int. Symp. Theoretical Aspects of Computer Science (STACS2014)*, (*Leibniz Int. Proc. in Informatics*) vol 25 pp 482–93
- Krastanov S and Jiang L 2017 Deep neural network probabilistic decoder for stabilizer codes *Sci. Rep.* **7** 11003
- Krawec W O 2016 A genetic algorithm to analyze the security of quantum cryptographic protocols *IEEE Congress on Evolutionary Computation*
- Krawec W O, Nelson M G and Geiss E P 2017 Automatic generation of optimal quantum key distribution protocols *Proc. of the Genetic and Evolutionary Computation Conf.* (New York: ACM) pp 1153–60

- Krenn M, Malik M, Fickler R, Lapkiewicz R and Zeilinger A 2016 Automated search for new quantum experiments *Phys. Rev. Lett.* **116** 090405
- Laird J E 2012 *The Soar Cognitive Architecture* (Cambridge, MA: MIT Press)
- Lamata L 2017 Basic protocols in quantum reinforcement learning with superconducting circuits *Sci. Rep.* **7** 1609
- Langford J and Zhang T 2008 The epoch-greedy algorithm for multi-armed bandits with side information *Advances in Neural Information Processing Systems* vol 20, ed J C Platt *et al* (Red Hook, NY: Curran Associates) pp 817–24
- Larochelle H, Bengio Y, Louradour J and Lamblin P 2009 Exploring strategies for training deep neural networks *J. Mach. Learn. Res.* **10** 1–40
- Lattimore T and Hutter M 2011 No free lunch versus Occam’s razor in supervised learning *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence—Papers from the Ray Solomonoff 85th Memorial Conf. (Melbourne, VIC, Australia, 30 November–2 December 2011)* ([https://doi.org/10.1007/978-3-642-44958-1\\_17](https://doi.org/10.1007/978-3-642-44958-1_17))
- Lattimore T, Hutter M and Sunehag P 2013 The sample-complexity of general reinforcement learning (arXiv:1308.4828)
- Lau H K, Pooser R, Siopsis G and Weedbrook C 2017 Quantum machine learning over infinite dimensions *Phys. Rev. Lett.* **118** 080501
- Lechner W, Hauke P and Zoller P 2015 A quantum annealing architecture with all-to-all connectivity from local interactions *Sci. Adv.* **1** e1500838
- Levine Y, Yakira D, Cohen N and Shashua A 2017 Deep learning and quantum entanglement: Fundamental connections with implications to network design (arXiv:1704.01552)
- Lewenstein M 1994 Quantum perceptrons *J. Mod. Opt.* **41** 2491–501
- Li Z, Kermode J R and De Vita A 2015a Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces *Phys. Rev. Lett.* **114** 096405
- Li Z, Liu X, Xu N and Du J 2015b Experimental realization of a quantum support vector machine *Phys. Rev. Lett.* **114** 140504
- Libbrecht M W and Noble W S 2015 Machine learning applications in genetics and genomics *Nat. Rev. Genet.* **16** 321–32
- Lipton Z C 2016 The mythos of model interpretability *CoRR* (arXiv:1606.03490)
- Lipton Z C, Berkowitz J and Elkan C 2015 A critical review of recurrent neural networks for sequence learning (arXiv:1506.00019)
- Lloyd S, Garnerone S and Zanardi P 2016 Quantum algorithms for topological and geometric analysis of data *Nat. Commun.* **7** 10138
- Lloyd S, Mohseni M and Rebentrost P 2013 Quantum algorithms for supervised and unsupervised machine learning (arXiv:1307.0411)
- Lloyd S, Mohseni M and Rebentrost P 2014 Quantum principal component analysis *Nat. Phys.* **10** 631–3
- Long P M and Servedio R A 2010 Random classification noise defeats all convex potential boosters *Mach. Learn.* **78** 287–304
- Loredo T J 2004 Bayesian adaptive exploration *AIP Conf. Proc.* **707** 330–46
- Lovett N B, Crosnier C, Perarnau-Llobet M and Sanders B C 2013 Differential evolution for many-particle adaptive quantum metrology *Phys. Rev. Lett.* **110** 220501
- Low G H and Chuang I L 2016 Hamiltonian simulation by qubitization (arXiv:1610.06546)
- Low G H, Yoder T J and Chuang I L 2014 Quantum inference on bayesian networks *Phys. Rev. A* **89** 062315
- Lu S and Braunstein S L 2014 Quantum decision tree classifier *Quantum Inf. Process.* **13** 757–70
- Lu S, Huang S, Li K, Li J, Chen J, Lu D, Ji Z, Shen Y, Zhou D and Zeng B 2017 A separability-entanglement classifier via machine learning (arXiv:1705.01523)
- Lund A P, Bremner M J and Ralph T C 2017 Quantum sampling problems, boson sampling and quantum supremacy *NPJ Quantum Inf.* **3** 15
- Ma Y C and Yung M H 2017 Transforming Bell’s inequalities into state classifiers with machine learning (arXiv:1705.00813)
- Machnes S, Sander U, Glaser S J, de Fouquières P, Gruslys A, Schirmer S and Schulte-Herbrüggen T 2011 Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework *Phys. Rev. A* **84** 022305
- Magniez F, Nayak A, Roland J and Santha M 2011 Search via quantum walk *SIAM J. Comput.* **40** 142–64
- Makmal A, Melnikov A A, Dunjko V and Briegel H J 2016 Meta-learning within projective simulation *IEEE Access* **4** 2110–22
- Manin Y 1980 *Computable and Uncomputable* (Moscow: Sovetskoye Radio)
- Manwani N and Sastry P S 2013 Noise tolerance under risk minimization *IEEE Trans. Cybernetics* **43** 3
- Mautner J, Makmal A, Manzano D, Tiersch M and Briegel H J 2015 Projective simulation for classical learning agents: a comprehensive investigation *New Generation Computing* **33** 69–114
- Mavadia S, Frey V, Sastrawan J, Dona S and Biercuk M J 2017 Prediction and real-time compensation of qubit decoherence via machine learning *Nat. Commun.* **8** 14106
- McCarthy J, Minsky M L, Rochester N and Shannon C E 1955 A proposal for the dartmouth summer research project on artificial intelligence ([www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html](http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html))
- McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 The theory of variational hybrid quantum-classical algorithms *New J. Phys.* **18** 023023
- McCulloch W S and Pitts W 1943 A logical calculus of the ideas immanent in nervous activity *Bull. Math. Biophys.* **5** 115–33
- Medler D A 1998 A brief history of connectionism *Neural Comput. Surv.* **1** 61–101
- Mehta P and Schwab D J 2014 An exact mapping between the variational renormalization group and deep learning (arXiv:1410.3831)
- Melnikov A A, Makmal A and Briegel H J 2014 Projective simulation applied to the grid-world and the mountain-car problem (arXiv:1405.5459)
- Melnikov A A, Makmal A, Dunjko V and Briegel H 2017 Projective simulation with generalization *CoRR Sci. Rep.* **7** 14430
- Melnikov A A, Nautrup H P, Krenn M, Dunjko V, Tiersch M, Zeilinger A and Briegel H J 2018 Active learning machine learns to create new quantum experiments *PNAS* **18** 201714936
- Mnih V *et al* 2015 Human-level control through deep reinforcement learning *Nature* **518** 529–33
- Monràs A, Sentís G and Wittek P 2017 Inductive supervised quantum learning *Phys. Rev. Lett.* **118** 190503
- Montanaro A 2016 Quantum algorithms: an overview *NPJ Quantum Inf.* **2** 15023
- Moret-Bonillo V 2015 Can artificial intelligence benefit from quantum computing? *Prog. Artif. Intell.* **3** 89–105
- Morningstar A and Melko R G 2017 Deep learning the Ising model near criticality (arXiv:1708.04622)
- Neigovzen R, Neves J L, Sollacher R and Glaser S J 2009 Quantum pattern recognition with liquid-state nuclear magnetic resonance *Phys. Rev. A* **79** 042321
- Neukart F, Dollen D V, Seidel C and Compostella G 2017 Quantum-enhanced reinforcement learning for finite-episode games with discrete state spaces (arXiv:1708.09354)
- Neven H, Denchev V, Rose G and Mcready W 2012 Qboost: Large scale classifier training with adiabatic quantum optimization *Proc. of the Asian Conf. on Machine Learning (Proc. of Machine Learning Research vol 25)* ed S C H Hoi and

- W Buntine (Singapore: Singapore Management University) pp 333–48
- Neven H, Denchev V S, Drew-Brook M, Zhang J, Macready W G and Rose G 2009a NIPS 2009 demonstration: binary classification using hardware implementation of quantum annealing
- Neven H, Denchev V S, Rose G and Macready W G 2008 Training a binary classifier with the quantum adiabatic algorithm (arXiv:0811.0416)
- Neven H, Denchev V S, Rose G and Macready W G 2009b Training a large scale classifier with the quantum adiabatic algorithm (arXiv:0912.0779)
- Newell A and Simon H A 1976 Computer science as empirical inquiry: Symbols and search *Commun. ACM* **19** 113–26
- NFL 2018 For a discussion on no free lunch theorems, and for relevant links see e.g. [www.no-free-lunch.org/](http://www.no-free-lunch.org/)
- Nielsen M A and Chuang I L 2011 *Quantum Computation and Quantum Information: 10th Anniversary Edition* 10th edn (New York: Cambridge University Press)
- Nishimori H and Nonomura Y 1996 Quantum effects in neural networks *J. Phys. Soc. Japan* **65** 3780–96
- O’Gorman B, Babbush R, Perdomo-Ortiz A, Aspuru-Guzik A and Smelyanskiy V 2015 Bayesian network structure learning using quantum annealing *Eur. Phys. J. Spec. Top.* **224** 163–88
- Orsucci D, Tiersch M and Briegel H J 2016 Estimation of coherent error sources from stabilizer measurements *Phys. Rev. A* **93** 042303
- Östlund S and Rommer S 1995 Thermodynamic limit of density matrix renormalization *Phys. Rev. Lett.* **75** 3537–40
- Palittapongarnpim P, Wittek P, Zahedinejad E and Sanders B C 2017 Learning in quantum control: high-dimensional global optimization for noisy quantum dynamics *Neurocomputing* **268** 116–26
- Paparo G D, Dunjko V, Makmal A, Martin-Delgado M A and Briegel H J 2014 Quantum speedup for active learning agents *Phys. Rev. X* **4** 031002
- Penrose R 1989 *The Emperor’s New Mind: Concerning Computers, Minds, and the Laws of Physics* (New York: Oxford University Press)
- Perdomo-Ortiz A, Benedetti M, Realpe-Gómez J and Biswas R 2017 Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers (arXiv:1708.09757)
- Peruš M 2000 Neural networks as a basis for quantum associative networks *Neural Netw. World* **10** 1001–13
- Peruzzo A, McClean J, Shadbolt P, Yung M H, Zhou X Q, Love P J, Aspuru-Guzik A and O’Brien J L 2014 A variational eigenvalue solver on a photonic quantum processor *Nat. Commun.* **5** 4213
- Peshkin L 2001 Reinforcement learning by policy search *PhD Thesis* Brown University US
- Poggio T, Mhaskar H, Rosasco L, Miranda B and Liao Q 2017 Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review *Int. J. Autom. Comput.* **14** 503
- Ponte P and Melko R G 2017 Kernel methods for interpretable machine learning of order parameters *Phys. Rev. B* **96** 205146
- Portmann C, Matt C, Maurer U, Renner R and Tackmann B 2017 Causal boxes: quantum information-processing systems closed under composition *IEEE Trans. Inf. Theory* **63** 3277–305
- Poulin D and Wocjan P 2009 Sampling from the thermal quantum Gibbs state and evaluating partition functions with a quantum computer *Phys. Rev. Lett.* **103** 220502
- Preskill J 2012 *25th Solvay Conf.*
- Pudenz K L and Lidar D A 2013 Quantum adiabatic machine learning *Quantum Inf. Process.* **12** 2027–70
- Rabitz H A, Hsieh M M and Rosenthal C M 2004 Quantum optimally controlled transition landscapes *Science* **303** 1998–2001
- Raussendorf R and Briegel H J 2001 A one-way quantum computer *Phys. Rev. Lett.* **86** 5188–91
- Rawat W and Wang Z 2017 Deep convolutional neural networks for image classification: a comprehensive review *Neural Comput.* **29** 2352–449
- Rebentrost P, Bromley T R, Weedbrook C and Lloyd S 2017 A quantum Hopfield neural network (arXiv:1710.03599)
- Rebentrost P, Mohseni M and Lloyd S 2014 Quantum support vector machine for big data classification *Phys. Rev. Lett.* **113** 130503
- Rebentrost P, Schuld M, Wossnig L, Petruccione F and Lloyd S 2016a Quantum gradient descent and newton’s method for constrained polynomial optimization (arXiv:1612.01789)
- Rebentrost P, Steffens A and Lloyd S 2018 Quantum singular value decomposition of non-sparse low-rank matrices *Phys. Rev. A* **97** 012327
- Reitzner D, Nagaj D and Buzek V 2012 Quantum walks *ACTA Phys. Slov.* **61** 603–725
- Review M T 2017 10 breakthrough technologies of 2017 ([www.technologyreview.com/s/603501/10-breakthrough-technologies-2017-reinforcement-learning/](http://www.technologyreview.com/s/603501/10-breakthrough-technologies-2017-reinforcement-learning/))
- Richardson M and Domingos P 2006 Markov logic networks *Mach. Learn.* **62** 107–36
- Rigatos G and Tzafestas S 2006 Quantum learning for neural associative memories *Fuzzy Sets Syst.* **157** 1797–813
- Rigatos G G and Tzafestas S G 2007 Neurodynamics and attractors in quantum associative memories *Integr. Comput.—Aided Eng.* **14** 225–42
- Ristè D, da Silva M P, Ryan C A, Cross A W, Córcoles A D, Smolin J A, Gambetta J M, Chow J M and Johnson B R 2017 Demonstration of quantum advantage in machine learning *NPJ Quantum Inf.* **3** 16
- Rocchetto A 2017 Stabiliser states are efficiently pac-learnable (arXiv:1705.00345)
- Rosenblatt F 1957 *The Perceptron, a Perceiving and Recognizing Automaton (Project Para) (Report: Cornell Aeronautical Laboratory Cornell Aeronautical Laboratory)* (Buffalo, NY: Cornell Aeronautical Laboratory)
- Rupp M, Tkatchenko A, Müller K R and von Lilienfeld O A 2012 Fast and accurate modeling of molecular atomization energies with machine learning *Phys. Rev. Lett.* **108** 058301
- Russell B and Rabitz H 2017 Common foundations of optimal control across the sciences: evidence of a free lunch *Phil. Trans. R. Soc. Lond. A* **375** 20160210
- Russell S and Norvig P 2009 *Artificial Intelligence: a Modern Approach* 3rd edn (Upper Saddle River, NJ: Prentice Hall)
- Salakhutdinov R and Hinton G E 2009 Deep Boltzmann machines *Proc. of the 12th Int. Conf. on Artificial Intelligence and Statistics (Clearwater Beach, FL, USA, 16–18 April 2009)*
- Sanders B C and Milburn G J 1995 Optimal quantum measurements for phase estimation *Phys. Rev. Lett.* **75** 2944–7
- Santra S, Shehab O and Balu R 2017 Exponential capacity of associative memories under quantum annealing recall *Phys. Rev. A* **96** 062330
- Sasaki M and Carlini A 2002 Quantum learning and universal quantum matching machine *Phys. Rev. A* **66** 022303
- Sasaki M, Carlini A and Jozsa R 2001 Quantum template matching *Phys. Rev. A* **64** 022317
- Schapiro R E 1990 The strength of weak learnability *Mach. Learn.* **5** 197–227
- Schuld M, Fingerhuth M and Petruccione F 2017 Quantum machine learning with small-scale devices: implementing a distance-based classifier with a quantum interference circuit *EPL* **119** 6
- Schuld M, Sinayskiy I and Petruccione F 2014a Quantum computing for pattern classification *Trends in Artificial Intelligence (LNAI vol 8862)* (Berlin: Springer) pp 208–20
- Schuld M, Sinayskiy I and Petruccione F 2014b The quest for a quantum neural network *Quantum Inf. Process.* **13** 2567–86
- Schuld M, Sinayskiy I and Petruccione F 2016 Prediction by linear regression on a quantum computer *Phys. Rev. A* **94** 022342

- Schützhold R 2003 Pattern recognition on a quantum computer *Phys. Rev. A* **67** 062311
- Seddiqi H and Humble T S 2014 Adiabatic quantum optimization for associative memory recall *Front. Phys.* **2** 79
- Sedlák M and Ziman M 2014 Optimal single-shot strategies for discrimination of quantum measurements *Phys. Rev. A* **90** 052312
- Sedlák M, Bisio A and Ziman M 2017 Perfect probabilistic storing and retrieving of unitary channels *Featured in QPL/QSA 2017*
- Seldin Y and Tishby N 2010 PAC-Bayesian analysis of co-clustering and beyond *J. Mach. Learn. Res.* **11** 3595–646
- Sentís G 2017 personal communication
- Sentís G, Bagan E, Calsamiglia J and Muñoz Tapia R 2013 Programmable discrimination with an error margin *Phys. Rev. A* **88** 052304
- Sentís G, Calsamiglia J, Muñoz-Tapia R and Bagan E 2012 Quantum learning without quantum memory *Sci. Rep.* **2** 708
- Sentís G, Guță M and Adesso G 2015 Quantum learning of coherent states *EPJ Quantum Technol.* **2**
- Sergeevich A and Bartlett S D 2012 Optimizing qubit Hamiltonian parameter estimation algorithm using PSO *IEEE Congress on Evolutionary Computation (June 2012)* (IEEE)
- Servedio R A and Gortler S J 2004 Equivalences and separations between quantum and classical learnability *SIAM J. Comput.* **33** 1067–92
- Sgarbas K N 2007 The road to quantum artificial intelligence *Current Trends in Informatics* (Athens: New Technologies Publications) pp 469–77
- Shalev-Shwartz S and Ben-David S 2014 *Understanding Machine Learning: From Theory to Algorithms* (New York: Cambridge University Press)
- Shepherd D and Bremner M J 2009 Temporally unstructured quantum computation *Proc. R. Soc. A* **465** 1413–39
- Shi Y 2002 Both Toffoli and controlled-NOT need little help to do universal quantum computation (arXiv:quant-ph/0205115)
- Shir O M, Roslund J, Leghtas Z and Rabitz H 2012 Quantum control experiments as a testbed for evolutionary multi-objective algorithms *Genet. Program. Evolvable Mach.* **13** 445–91
- Shor P W 1997 Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer *SIAM J. Comput.* **26** 1484–509
- Sieberer L M and Lechner W 2017 Programmable superpositions of Ising configurations (arXiv:1708.02533)
- Silver D *et al* 2016 Mastering the game of go with deep neural networks and tree search *Nature* **529** 484–9
- Silver D *et al* 2017a Mastering chess and shogi by self-play with a general reinforcement learning algorithm
- Silver D *et al* 2017b Mastering the game of go without human knowledge *Nature* **550** 354
- Slussarenko S, Weston M M, Li J G, Campbell N, Wiseman H M and Pryde G J 2017 Quantum state discrimination using the minimum average number of copies *Phys. Rev. Lett.* **118** 030502
- Snyder J C, Rupp M, Hansen K, Müller K R and Burke K 2012 Finding density functionals with machine learning *Phys. Rev. Lett.* **108** 253002
- Steane A 1998 Quantum computing *Rep. Prog. Phys.* **61** 117
- Stenberg M P V, Köhn O and Wilhelm F K 2016 Characterization of decohering quantum systems: machine learning approach *Phys. Rev. A* **93** 012122
- Storkey A 1997 Increasing the capacity of a Hopfield network without sacrificing functionality *Proc. of the 7th Int. Conf. on Artificial Neural Networks* (London: Springer) pp 451–6
- Sutton R S and Barto A G 1998 *Introduction to Reinforcement Learning* 1st edn (Cambridge, MA: MIT Press)
- Suykens J and Vandewalle J 1999 Least squares support vector machine classifiers *Neural Process. Lett.* **9** 293–300
- Szegedy M 2004 Quantum speed-up of Markov chain based algorithms *45th Annual IEEE Symp. on Foundations of Computer Science*
- Tegmark M 2000 Importance of quantum decoherence in brain processes *Phys. Rev. E* **61** 4194–206
- Temme K, Osborne T J, Vollbrecht K G, Poulin D and Verstraete F 2011 Quantum metropolis sampling *Nature* **471** 87–90
- Tiersch M, Ganahl E J and Briegel H J 2015 Adaptive quantum computation in changing environments using projective simulation *Sci. Rep.* **5** 12874
- Torlai G and Melko R G 2017 Neural decoder for topological codes *Phys. Rev. Lett.* **119** 030501
- Torlai G, Mazzola G, Carrasquilla J, Troyer M, Melko R and Carleo G 2018 Neural-network quantum state tomography *Nat. Phys.* **14** 447–50
- Trugenberger C A 2001 Probabilistic quantum memories *Phys. Rev. Lett.* **87**
- Trugenberger C A 2003 Trugenberger replies *Phys. Rev. Lett.* **91** 20
- Turing A M 1950 Computing machinery and intelligence (<http://cogsci.umn.edu/millennium/final.html>)
- Valiant L G 1984 A theory of the learnable *Commun. ACM* **27** 1134–42
- van Apeldoorn J, Gilyén A, Gribling S and de Wolf R 2017 Quantum sdp-solvers: better upper and lower bounds *Proc. 58th IEEE Symp. Foundations of Computer Science (FOCS 2017)* pp 403–14
- van Nieuwenburg E P L, Liu Y H and Huber S D 2017 Learning phase transitions by confusion *Nat. Phys.* **13** 435–9
- Vapnik V N 1995 *The Nature of Statistical Learning Theory* (New York: Springer)
- Varsamopoulos S, Criger B and Bertels K 2018 Decoding small surface codes with feedforward neural networks *Quantum Sci. Technol.* **3** 015004
- Ventura D and Martinez T 2000 Quantum associative memory *Inf. Sci.* **124** 273–96
- Verstraete F and Cirac J I 2004 Renormalization algorithms for quantum-many body systems in two and higher dimensions (arXiv:cond-mat/0407066)
- Verstraete F, Murg V and Cirac J 2008 Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems *Adv. Phys.* **57** 143–224
- Vickers J 2016 The problem of induction *The Stanford Encyclopedia of Philosophy* ed E N Zalta (Stanford, CT: Stanford University)
- Vlasov A Y 1994 Representation and processing of information in quantum computers *Tech. Phys. Lett.* **20** 992–3
- Vlasov A Y 1997 Quantum computations and images recognition (arXiv:quant-ph/9703010)
- Wang G 2017 New quantum algorithm for linear regression *Phys. Rev. A* **96** 012335
- Wang J *et al* 2017 Experimental quantum Hamiltonian learning *Nat. Phys.* **13** 551–5
- Wang L 2016 Discovering phase transitions with unsupervised learning *Phys. Rev. B* **94** 195105
- Watkins C J and Dayan P 1992 Technical note: Q-learning *Mach. Learn.* **8** 279–92
- Wichert A 2014 *Principles of quantum artificial intelligence* (Hackensack, NJ: World Scientific)
- Wiebe N and Granade C 2017 Can small quantum systems learn? *Quant. Inf. Comp.* **17** 568–94
- Wiebe N, Braun D and Lloyd S 2012 Quantum algorithm for data fitting *Phys. Rev. Lett.* **109** 050505
- Wiebe N, Granade C, Ferrie C and Cory D 2014a Quantum Hamiltonian learning using imperfect quantum resources *Phys. Rev. A* **89** 042314

- Wiebe N, Granade C, Ferrie C and Cory D G 2014b Hamiltonian learning and certification using quantum resources *Phys. Rev. Lett.* **112** 190501
- Wiebe N, Kapoor A and Svore K M 2014c Quantum deep learning (arXiv:1412.3489)
- Wiebe N, Kapoor A and Svore K M 2015 Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning *Quantum Inf. Comput.* **15** 0318–58
- Wiebe N, Kapoor A and Svore K M 2016 Quantum perceptron models (arXiv:1602.04799)
- Wigley P B *et al* 2016 Fast machine-learning online optimization of ultra-cold-atom experiments *Sci. Rep.* **6** 25890
- Wired 2017 Apple is bringing the AI revolution to your iPhone ([www.wired.com/2016/06/apple-bringing-ai-revolution-iphone/](http://www.wired.com/2016/06/apple-bringing-ai-revolution-iphone/))
- Wiseman H and Milburn G 2010 *Quantum Measurement and Control* (Cambridge: Cambridge University Press)
- Wittek P 2014a *Seminar at University of Innsbruck*
- Wittek P 2014b *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Amsterdam: Elsevier)
- Wittek P and Gogolin C 2017 Quantum enhanced inference in Markov logic networks *Sci. Rep.* **7** 45672
- Wolpert D H 1996 The lack of *a priori* distinctions between learning algorithms *Neural Comput.* **8** 1341–90
- Wootters W K and Zurek W H 1982 A single quantum cannot be cloned *Nature* **299** 802–3
- Wu Y *et al* 2016 Google's neural machine translation system: Bridging the gap between human and machine translation (arXiv:1609.08144)
- Ye J and Xiong T 2007 SVM versus least squares SVM *Proc. of the 11th Int. Conf. on Artificial Intelligence and Statistics (San Juan, Puerto Rico 2007)* (*Proc. of Machine Learning Research* vol 2) ed M Meila and X Shen pp 644–51
- Yu C H, Gao F, Wang Q L and Wen Q Y 2016 Quantum algorithm for association rules mining *Phys. Rev. A* **94** 042311
- Yung M H and Aspuru-Guzik A 2012 A quantum-quantum metropolis algorithm *Proc. Natl Acad. Sci.* **109** 754–9
- Zahedinejad E, Ghosh J and Sanders B C 2015 High-fidelity single-shot Toffoli gate via quantum control *Phys. Rev. Lett.* **114** 200502
- Zahedinejad E, Ghosh J and Sanders B C 2016 Designing high-fidelity single-shot three-qubit gates: a machine-learning approach *Phys. Rev. Appl.* **6** 054005
- Zahedinejad E, Schirmer S and Sanders B C 2014 Evolutionary algorithms for hard quantum control *Phys. Rev. A* **90** 032310
- Zhan B, Kimmel S and Hassidim A 2012 Super-polynomial quantum speed-ups for boolean evaluation trees with hidden structure *Innovations in Theoretical Computer Science (Cambridge, MA, USA, 8–10 January 2012)*
- Zhang C, Bengio S, Hardt M, Recht B and Vinyals O 2017a Understanding deep learning requires rethinking generalization (arXiv:1611.03530)
- Zhang Q, Wu Y N and Zhu S C 2017b Interpretable convolutional neural networks (arXiv:1710.00935)
- Zhao L, Pérez-Delgado C A and Fitzsimons J F 2016 Fast graph operations in quantum computation *Phys. Rev. A* **93** 032314
- Zhao Z, Fitzsimons J K and Fitzsimons J F 2015 Quantum assisted Gaussian process regression (arXiv:1512.03929)
- Ziman M, Plesch M, Bužek V and Štelmachovič P 2005 Process reconstruction: from unphysical to physical maps via maximum likelihood *Phys. Rev. A* **72** 022106



**Hans J. Briegel** received his doctorate (1994) and habilitation (2002) in physics from the Ludwig-Maximilians-University of Munich. He held postdoctoral positions at Texas A&M, Innsbruck, and Harvard University. He has been a Full Professor of Theoretical Physics with the University of Innsbruck since 2003. His main field of research is quantum information and computation where he has authored and co-authored papers on a wide range of topics, including work on microscopic lasers, quantum repeaters for long-distance quantum communication, quantum entanglement and cluster states, and measurement-based quantum computers. His recent research has focused on physical models for classical and quantum machine learning, artificial intelligence, and the philosophical problem of learning and agency.



**Vedran Dunjko** obtained his PhD at the Heriot-Watt University, Edinburgh UK, with research focusing on various quantum cryptographic protocols. Since, he has been a post-doctoral researcher at the School of Informatics, University of Edinburgh, UK and at the Theoretical Physics Department of University of Innsbruck, Austria. Currently he is a post-doc at the Max Planck Institute of Quantum Optics in Garching, Germany. His recent works concentrate on the exploration of the intersection between quantum information processing, machine learning and artificial intelligence. This includes the potential of quantum enhancements, perspectives of learning devices in the contexts of quantum experiments, and theoretical limits of learning agents in fully quantum domains.