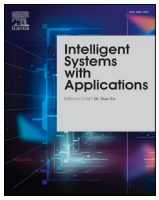
Intelligent Systems with Applications 23 (2024) 200346

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications



Review

Quantum computing and neuroscience for 6G/7G networks: Survey

Savo Glisic [a,*], Beatriz Lorenzo [a,b]

[a] Senior Member IEEE, Worcester Polytechnic Institute, Department of Physics, MA, United States
[b] Senior Member IEEE, UMASS, MA, United States

ARTICLE INFO

Keywords:
7G networks
New paradigms in modern communication systems
QC
ML
Complex networks
n-Sci
q- biology and q-chemistry for brain modeling
Tensor networks
Synchronization

ABSTRACT

Recently significant effort has been invested in studying commonalities of human brain operation and advanced algorithms for machine learning to answer the question: Can the learning mechanisms, identified in the operation of the brain, be mimicked in artificial neural networks to enhance the learning efficiency with simultaneous reduction in complexity and power consumption.

At the same time, machine learning algorithms, on their own, become increasingly complex, resulting in complex neural networks. To speed up the machine learning algorithms, research on 7G networks will be looking for new computing technologies, like quantum (q-) computing (QC), and new models for complex networks that will enable us to efficiently control/optimize the processes run on them.

In this paper, under the umbrella of well-established complex networks theory, we provide a unified presentation of how quantum computing, implemented on near-future computers, can enable solving various problems in the above disciplines, otherwise difficult to solve by using classical (c-) approaches. The emphasis is on the commonalities in QC applications and modeling for the different systems listed above. For 7G network designers, the survey is expected to provide an insight into how much the research results in natural, QC based sciences can be integrated into new network paradigms to support above initiatives.

Abbreviations: aL, active Learning; aA, adaptive aperiodicity (aCtr) adaptive control; afbC, adaptive feed-back control; $\mathscr{A}$, ancilla; aN, artificial neurons; a^{mplt}, amplitude (s); ap-, aperiodical; bp, backpropagation; bD, big Data; bL, bio-learning; bioP, biological processes; bpTT, bp through time; bp, barren plateau; biI, brain-inspired intelligence; bI, brain intelligence; ChSim, chemistry simulation; cir-, circadian; c-, classical; cS, coherent states; cSync, collective synchronization; CP, completely positive; cbN, complex brain networks; cN, complex network; cDN, complex dynamic networks; cdN, complex dynamic N; cvV, complex valued vectors; c-arch, computing architecture; c$\mathscr{B}$, computational basis; cCh, computational chemistry; compSci, computers science; Ctr, control; CV, continuous-variable; corS, cortical sheet; corL, cortical learning; corN, cortical networks; Co, Coulomb operator; cr-ass, credit assignment; c-dim, curse of dimensionality; DFC, Deep Feedback Control; DNN, deep neural network; DL, deep learning; dC, dendritic computation; dP, differential privacy; dG, discrete groups; DQNN, dqNN, dissipative quantum NN; distrS, distributed systems; dN, dynamic networks; dM, dynamic models; dS, dynamic systems; eigS, eigenstate; eigV, eigenvalues; ent-, entanglement; eT, evolution time; e-sync, exponential synchronization; $\mathscr{EP}$, expressive power; FL, fL, federated learning; FLS, fLS, federated learning system; fF, fidelity function; hL, human learning; G, gradient; gL, graphical language; gS, ground state; gI, group-invariant; H, Hamiltonian; hD, high dimensionality; h$\mathscr{E}$, homomorphic encryption; hS, hybrid systems; iE, information extraction; iF, integrate-and-fire; iC, intermittent control; lsD, large-scale (ls-) data; lsN, large-scale networks; $\mathscr{L}$, learning; LML, lifelong machine learning; LCU, linear combination of unitaries; lT, long-term; lr$\mathscr{T}$ f, low rate tensor factorization; ls, large scale; Ly, Lyapunov; MI, machine intelligence; m$\mathscr{V}$, $\mathscr{M}^{om}$ -vector(s); ms, multiscale; Mjs, Markov jump systems; Msc, Markovian stochastic coupling; MPS, matrix product states; $\mathscr{M}$, measurement; $\mathscr{M}^{om}$, momentum; $\mathscr{MD}$, Multidimensional; $\mathscr{ML}$, multilayer; MTL, multi-Task Learning; nD, network dynamics; nS, network science; N-sync, network synchronization; nM, neuromodulators; n-Sci, neuroscience; nS, neural systems; niI, n-Sci Inspired AI; nD, neutral (n-) delay; oF, objective function; o^{prtr}, operator; oSync, onset of synchronization; ooF, optimization of objective function; P^{er}, perceptron; phε, phase estimate; phM, phenomenological models; phy-, p^{hys}) physical; pW, plane wave; $^{+}$sy, postsynaptic; $\mathscr{P}^{os}$, position; p$\mathscr{V}$, position vectors; $^{-}$sy, presynaptic; Q, q-, quantum; QC, Quantum Computing; QML, Quantum ML; qN, quantum neuron; qNN, quantum NN; qP, quantum perceptrons; QS, quantum simulation; qCh, quantum chemistry (rvV) real valued vector; r^{al}, real; $\mathscr{R}$), r^{gst}, registers; r-con, restrictive conditions; rEng-, reverse engineering; rL, reinforcement learning; SF, scale-free; 2ndQ, second quantization; SMC, secure Multi-party Computation; s^{pace}, space; sT, short-term; SW, "small-world"; sT, spike-timing; sp-, spin; sS, squeezed states; STDP, spike-timing dependent plasticity; sN, spiking neuron; SNN, spiking neural networks; sI, State Initialization; st-, statistical; s-, stochastic; s-pert, stochastic perturbation; σS, superposition state; sL, supervised learning; sG, symmetry group; sy, synaptic; syP, synaptic plasticity; sync, synchronization; s^{yst}-, system; t-crass, temporal credit assignment; $\mathscr{T}$, $\mathscr{T}^{nsor}$, Tensor; TN, Tensor Networks; TD, $\mathscr{T}$-decomposition; tD, time delay; tvD, time varying delay; tM, topic modeling; t-map, transition map; U-, unitary; UCC, unitary coupled cluster; uL, unsupervised Learning; wF, wavefunction; v^{lu}, value.

* Corresponding author.
E-mail addresses: sglisic@wpi.edu (S. Glisic), blorenzo@umass.edu (B. Lorenzo).

https://doi.org/10.1016/j.iswa.2024.200346
Received 11 November 2023; Received in revised form 11 February 2024; Accepted 20 February 2024
Available online 14 May 2024

1. Introduction

It is anticipated that 6G/7G networks will continue growing and extending their demands with respect to data rates, number of users and network access points, energy efficiency, network intelligence, decision latency in network dynamic reconfiguration and control protocols and efficiency of spectrum and overall resource sharing among multiple network operators.

In large scale (*ls*-) networks, social, economic, information, technological, biological, quantum chemistry, *n-Sci* and brain study, here collectively referred to as complex networks, there is a trend of controlling the complex systems in real time by more and more relaying on the help of artificial intelligence.

At the same time, machine learning (ML) algorithms, on their own, become increasingly more complex resulting in complex neural networks. To speed up the ML algorithms new computing technologies have been investigated, like quantum computing QC, and new models for complex networks that will enable efficient control/optimization of the processes run on them. This includes applying results of lessons learned in neuroscience to enhance ML algorithms. In this paper, under the umbrella of well-established complex networks (*cN*) theory, a unified presentation is provided on how *QC*, can enable solving various problems in above disciplines that are difficult to solve otherwise.

The emphasis is on the commonalities in QC applications and modelling for different systems listed above. In addition to discussing the algorithms the pertaining implementation problems are reviewed like the network synchronization (*N-sync*) and analytical and simulation tools for the system analysis. A comprehensive survey of the work in these fields is provided resulting in a long list of references. For this reason, in the form of tables, a selected list of references is provided, for the first iteration of additional readings.

Tutorial and survey type papers are rather efficient ways to get an insight into the latest advances in a specific technology. Usually, these papers focus on a specific technology and often, even more narrowly, on a specific application of a given technology. For example, references (Cui et al., 2018; Buczak & Guven, 2016; Fatima & Pasha, 2017) present surveys on application of ML for Internet of things, cyber security intrusion detection and disease diagnostic respectively. Similarly (Herman et al., 2022) presents a survey of QC for finance. On the other hand, when designing complex systems/networks consisting of variety of different technologies there is a need for an integral view of the optimum solution and especially on the interdependence of the optimal choices in different segments of the system.

Such an example is a recent initiative to integrate results from n-Sci and brain research into new paradigms of modern communication systems (Moioli et al., 2020; Moioli et al., 2021). In support to these initiatives here a survey of enabling technologies for such an approach is provided with emphasis on how much the research results in natural, QC based sciences can be integrated into such systems.

Structure of the paper: To achieve the above objective the paper is organized in the following way: As the very first step application of the results from neuroscience in 6G wireless networks is discussed in Section 1.1 and extension to QC in 7G in Section 1.2. Advances in classical NN are surveyed in Section 2. Advantages of using spiking NN for power savings and QC based solution for speeding up algorithms are covered in Sections 3 and 4 respectively. The prospects of using experience in complex quantum models form computational chemistry in future networks are discussed in Section 5. The key implementation problem of these algorithms, network synchronization, is discussed in Section 6. At this point the impact of processing and propagation delays on the pulse position coding in spiking networks and resulting limitations in leveraging fully the potential in energy savings are discussed in detail. The work on tools for network analysis, namely tensor networks and quantum simulations respectively are surveyed in Sections 7 and 8 respectively. The network optimization frame work, a specific contribution of this paper, id presented in Section 9. The structure/flow of the presentation is shown graphically in Fig. 1.

1.1. n-Sci and 6G wireless networks

As already indicated above, the starting point for this work are already recognized potentials in convergence of neuroscience and modern paradigms in communications presented in Moioli et al. (2020); Moioli et al. (2021).

The papers discuss how brain signals will be incorporated in future wireless systems. The brain is modelled as, densely connected set of neurons, with small-world properties, which is a category explicitly studied by complex networks theory (Glisic, 2016). For this reason, the presentation in this paper is organized under the umbrela of complex networks theory.

Energy consumption limits the optimization of neural connectivity. Spiking activity, the way neurons communicate (Laughlin & Sejnowski, 2003), contributes significantly to the brain energy consumption. Consequently, the paper presents a detailed survey on spiking neural networks offering several orders of magnitude energy savings compared to the classical (*c*-) solutions with continuous signaling within the network.

Discussion on how the research results in *n-Sci* are used to improve

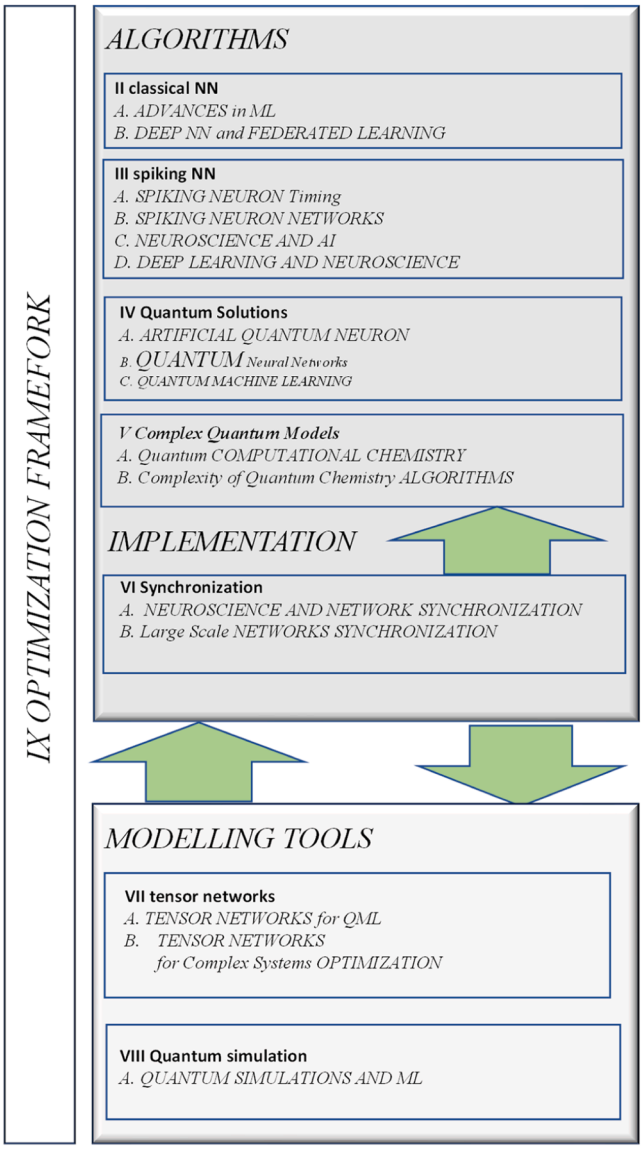


Fig. 1. Overview of the paper content.

development of the 6G/7G systems through brain type communications BTC, is presented also in Moioli et al. (2020); Moioli et al. (2021) providing additional details and challenges of wireless brain implants.

On the other hand, advantages that wireless networks may provide to *n-Sci*'s are also discussed in detail, in particular new generation of brain machine interfaces BMIs based on wireless connectivity for BTC (Lebedev & Nicolelis, 2006; Lebedev, 2014; Moxon & Foffani, 2015; Andersen et al., 2004; Lebedev & Nicolelis, 2017) and even the Internet of Bio-Nano Things (IoBNT) (Akyildiz et al., 2015; Akyildiz et al., 2008; Veletić et al., 2019). This also includes the theory of chaotic neuronal communications.

The references also include discussion on data security and privacy as well as ethical issues recommendations to guide both wireless communications and *n-Sci's* in the near future. Future solutions for network security will be heavily based on QC and due to the complexity of the subject we will address the problem separately in our future publications (Glisic & Lorenzo, 2022, Glisic, 2023a, 2023b).

1.2. QC n-Sci and 6G/7G networks

1.2.1. Motivation

As already indicated, the main motivation here is to support the initiatives like those in Moioli et al. (2020); Moioli et al. (2021) to leverage the convergence of neuroscience and modern paradigms in communications by providing a comprehensive survey of the work on QC as enabling technology for above initiatives. The survey is extended to include additional relevant fields as quantum biology and quantum chemistry which are also participating in the overall modelling and analysis of the operation of the brain and its relation to the design principles of modern communications.

As already indicated in Glisic and Lorenzo (2022); Chen and Liu (2016) complex networks theory enables us to use a high level of abstraction to model basic principles supporting the operation of the human brain.

In their own way, these networks evolve into large-scale networks (*lsN*) controlled by algorithms based on artificial intelligence. Here, the advances are reviewed in modeling, analysis, design, and operation of these networks with focus on optimum energy and computational efficiency. In addition, advanced solutions for network synchronization (*N-sync*) are reviewed along with the available tools for their analysis, and simulations under the umbrella of complex networks theory. This "*cross technology*" coverage of the survey should help network designers to get a comprehensive insight into the interrelation/interdependency of the different technologies used for the overall future communication networks design.

1.2.2. Energy efficiency

Since ML is used for the network control, after a survey of classical solutions (Section 2) the survey will focus on ML based on spiking neural networks (SNN) as a possible solution in 7G, which is supposed to provide several orders of magnitude in energy savings (Section 3).

Even when the paper talks about classical ML, the paper will focus on the latest advances in the field like Lifelong ML, and review work on related problems.

Under the umbrella of classical ML, the paper will also survey the work on Deep NN (DNN) and Federated Learning (FL).

Most of the advanced signal processing algorithms often require considerable complexity, making them less attractive for practical applications. Here, the paper will review the work demonstrating how ML can help to solve this issue. The solutions based on ML use a DNN to find an acceptable approximation of the input/output nonlinear mapping of an SP algorithm.

The survey will include work on: Optimization algorithm approximation by DNN, generic optimization problem, algorithm approximation background, spatial scheduling by DNN, wireless link scheduling, scheduling by DNN, training process, link deactivation, spatial scheduling by DNN with proportional fairness, DNN in vehicular networks, system model, channel model, modeling age of information problem (AoI), link clustering, network optimization, AoI-aware RRM objectives and Bellman's equation and DRL algorithm.

This segment will also include federated learning system (FLS) (Mohassel & Rindal, 2018; Araki et al., 2016; Furukawa et al., 2016; Mohassel et al., 2015). Here ML models process data sets coming from the sources from different locations. In this scenario the objective is to prevent data leakage. The survey will include works on algorithms, classification, FLS architecture, and block chains.

After a survey of classical solutions (Section 2) the paper will focus on ML based on spiking neural networks (SNN) as a possible solution, which is supposed to provide several orders of magnitude in energy savings (Section 3). This will include work on a. Spiking neuron timing. b. Spiking neuron networks c. *n-Sci* and AI and d. Deep learning and *n-Sci*.

The basics of bio-learning ($b\mathscr{L}$) and memory is synaptic plasticity (*sP*). Here, the paper will survey works modeling short- and long-term *sP*, with focus on spike-timing (*sT*) dependent plasticity (STDP). This approach requires detailed analysis of the synchronization in these networks which the paper covers in Section 6. The objective is to build up a framework for modeling different types of plasticity. All models discussed here can be used for large-scale network simulations.

In summary, in this segment the survey will include the works on models of *sP* based on *sT*, works on plasticity, experimental results, theoretical concepts, short-term plasticity, Markram-Tsodyks model, Abbott model, long-term plasticity (STDP), pair-based STDP rules, weight dependence of STDP, beyond pair effects, voltage dependence, induction versus maintenance, supervised and reinforcement learning ($r\mathscr{L}$), $r\mathscr{L}$ and STDP, $r\mathscr{L}$ algorithm for SNN, mathematical foundation of the basic $\mathscr{L}$-rule, bidirectional associative plasticity, intrinsic plasticity, modulation of STDP by reward with experimental examples and adaptive $\mathscr{L}$ in brain.

In the sequel, the paper will review the work on spiking neuron (*sN*) networks (*sNN*'s). They are based on computing principles in the brain and latest research results in *n-Sci* requiring precise timing of spike firing and detailed description of synaptic (*sy*) role in the interactions between neurons. When it comes to computational power, sNN's outperform the classical NN's. They provide a basis for building up models with high capacity for memorizing and an ability to adapt fast. The objective here is to design learning rules that combine both good features of *sNN*'s and useful properties of traditional connectionist models.

Here the paper starts by summarizing currently available models of neurons and *sP*, the capabilities of sNN's, different approaches for designing algorithms for learning in sNN and finally by discussing applications, implementations and several simulation frameworks. The work reviewed in this segment will include: Artificial neural networks, biological inspiration, information coding by spikes timing, sNN, network science (*nS*) and *sP*, integrate&fire (*iF*) model, model of the spike response, *sP* and STDP, learning algorithms in sNN, mimicking classical models, the neural code, input encoding, output decoding, objective functions, activity regularization, training sNN, shadow training, backpropagation (*bp*) using spike times, *bp* using spikes, long-term temporal dependencies, online learning, temporal locality, spatial locality and comments on further research in sNN.

Before going to the review of artificial quantum neurons (*qN*) the paper will survey the work on *n-Sci* and AI dealing with physical processes in the brain and discuss in depth the latest research results in *nS, n-Sci* and dynamic systems (*dS*), and their interrelations with focus on the latest trend in the filed referred to as brain-inspired intelligence. A practical way to imitate brain intelligence is to mimic cortical networks (*corN*) with its dynamics that support the brain functions, rather than use classical artificial NN. Here the paper reviews the work providing a complex network (*cN*) and space/time dynamics (referred to as network dynamics *-nD*) models for analyzing the brain and *corN* and develop *n-Sci* and *nD* based integrated approaches for designing intelligence

including learning and resilience functions mimicking the brain behavior. For this, the paper needs to cover issues such as fundamentals of *cN, n-Sci*, and hybrid *dS*, and the work about the brain and intelligence.

The survey includes in particular works on *n-Sci* and *nD*, modeling dynamic neurons and networks, hybrid network control applications, optimization for learning and resilience, sNN with computing dynamics, different options for encoding spiking data, and electrophysiological connectivity patterns in cortex.

In the sequel, the paper surveys the work studying commonalities between Deep Learning and *n-Sci*. The interest of *n-Sci* is in the implementation of computation, constructing neural codes, studying and modeling dynamics, and designing circuits. In ML, however, ANNs tend to intentionally avoid exactly designed codes, dynamics or circuits but rather use optimization objective function (*ooF*), with simple architectures. These seemingly divergent perspectives are lately converging based on two recent results within ML.

1) structured architectures are used for memory storage. 2) *ooF* and training algorithms evolve into more complex procedures varying across layers and over time. In this paper the work exploring the brain with focus on these ideas is surveyed. It was hypothesized in Marblestone (2016b) that: (1) the brain also uses *ooF*, (2) There is a variety of such *ooF* and they are different at different brain segments and time instances, and (3) Objective function (*oF*) minimization/maximization works within a preset structure adjusted to the problems defined by behavior. The system combines a number of interdependent *oF*, makes $\mathscr{L}$-data--efficient and adjusted to the human individual.

Here the paper reviews work on possible options *n-Sci* has available to improve and test these hypotheses. Intending to answer the questions *Can the brain optimize cost functions*? and *Can the brain learn differently*? the paper focuses in particular on work including: Local organization/optimization, biological optimization, impact of training data on supervised and reinforcement learning, analytical models for credit assignment in NN, modeling Deep Feedback Control (DFC), learning theory, stability of DFC, learning the feedback weights, derivation of the key theorems, learning theory revisited, linearized dynamics and fixed points, DFC approximates Gauss-Newton optimization, DFC uses minimum norm updates, Gauss-Newton optimization with a mini-batch size of 1, effects of the nonlinearity φ in the weight update, continuous DFC vs steady-state DFC weight updates, compatibility of DFC with various controller types, stability of DFC revisited, stability analysis with instantaneous system dynamics, stability of the full system, design examples, learning the feedback weights: revisited, learning the feedback weights in a sleep phase, learning the forward and feedback weights simultaneously, influence of noisy dynamics on learning the forward weights, simulations and algorithms of DFC, simulating DFC and DFC-SS for training the forward weights and simulating DFC with noisy dynamics for training the feedback weights.

1.2.3. Computing efficiency and complexity reduction

For reducing required computational resources and speeding up the execution of the algorithms, the paper will survey solutions for ML based on QC, referred to as Quantum ML (QML) (Section 4).

The starting point in studying ML algorithms and AI protocols in ANN is the modeling of an artificial neuron (*aN*). The first work of an *aN* is the classical Rosenblatt's "perceptron" (P^{er}), with the main drawback being complexity, preventing the use for training of multilayered P^{er}-networks. QC-based perceptron, with significant reduction in hardware resources over classical (*c*-) options was presented in Tacchino et al. (2018) together with experimental test.

The survey of the work in this segment includes modeling quantum P^{er}, unitary transformations, P^{er} -model complexity, online quantum P^{er}, quantum version space P^{er}, hybrid quantum-classical P^{er} algorithm, quantum computation, Grover's search algorithm, quantum activation functions for QNN, computation of the polynomial series, approximation of analytical activation functions, quantum neuron (*qN*), feedforward neural network, and Hopfield network.

Moving from *qN* to the quantum NN (*qNN*): The q-analog of an *aN*, presented above, leads to a q-feed forward NN capable of universal QC. Here the paper surveys work on qNN using graph-structured data and those using ansatz for a Q generative adversarial network.

Quantum machine learning (QML) can be used as c-ML enhancing q-tasks, q-algorithms speeding up c-ML, or using QC circuits for tasks with q-data. The dissipative qNN (DQNN) uses QC hardware for tasks with q-data. It consists of several layers of qubits and a pair of q- states is used for its training. An input state and a desired output reflecting the training objective represent a training data pair.

The DQNN is built of q-perceptrons (qP^{er}) like those reviewed in the previous section. A qP^{er} interconnects two adjacent layers of qubits and is modeled as a completely positive (CP) transition map (*tM*). The term dissipative specifies the operation of *tM*'s which defines both tensoring the states of the two layer's qubits and executing unitary (U) operations and tracing out the qubits from the first of the two layers. So, through the interaction between the two adjacent layers, tM's propagates input states forward throughout the DQNN. At the end the resulting output state is compared with the targeted result. For this step the fidelity of two q-states is used to determine how the perceptron U's should be updated to perform the training efficiently. The work that will be reviewed in this section will include Network architecture, optimization of objective functions, training QNN, implementation, performance limits of QNN, performance limits of classical and quantum optimization algorithm, continuous-variable (CV) QNN, the CV model, embedding classical neural networks, convolutional, recurrent, and residual CV QNN.

Given the full understanding of the QNN the paper then reviews the work on Quantum Machine Learning (QML) algorithms. Here the paper surveys the work on: Methods of ML, ANN and deep learning, support vector machines, Learning Theory, Computational learning theory, VC theory, reinforcement learning theory, ML in q-physics, estimating Hamiltonian, phase estimation (phε) settings, group-theoretic approach to QML, gradient (G) - invariance and Lie group-invariant models.

1.2.4. Complex quantum models

The presentation of the use of multiple-body q-systems will be illustrated on different models of molecules in chemistry (Section 5). Familiarity with this work should provide solid base for further work, within 7G umbrella, on advanced modelling on BTC and especially on brain to network interface (BNI).

Quantum computing is used nowadays more and more for solving otherwise difficult chemistry problems. These solutions may also help us to improve algorithms used in other fields, especially in ML learning the paper is focused on in this survey. Research results in solving these problems with existing q-resources are important since building a sufficiently large QC may take time. Based on these expectations, q-computational chemistry (*cCh*) has become a research field using knowledge from both QC and *cCh*. Here the paper surveys the work in both *cCh* and QC, reducing the current knowledge gap. Here the paper discusses the main results in the field, with a focus on potential provided with the existing results in QC. The reviewed work shows how to model chemical problems in such a way that can be solved using QC.

Works to be reviewed in Section 4 will already indicate the results significantly reducing the implementation complexity needed to simulate problems in q-computing chemistry. The solutions are based on using linear combinations of U's and plane wave (*pW*) basis Coulomb (Co) operator. In the sequel the paper will further detail these techniques and review the work achieving approximately same results even by using arbitrary basis sets by leveraging structure in the *Co* operator. This is based either on leveraging sparseness, or a low rank tensor ($\mathscr{T}$) factorization (*lr*$\mathscr{T}$*f*). As an example, authors in Dominic (2019) provide circuits with $\overline{\mathscr{O}}(N^{3/2}\lambda)T$ (Toffoli gates) complexity, with N representing a number of orbitals and λ being the 1-norm of the chemical

Hamiltonian (*che-H*).

Along these lines the paper will review the work in this field including arbitrary basis q-chemistry, $lr\mathscr{T}f$ of the *Co*, linear combination of unitaries (LCU) based simulation, the Hamiltonian as a LCU, state preparation, controlled U's, complexity exploiting sparsity in the *Co*, QC molecular energies, classical approaches to q-chemistry, unitary coupled cluster (UCC) and variational q-eigensolver for (UCC).

1.2.5. Large scale system level implementation

Synchronization, as a pertaining problem in the implementation of large-scale networks (LSN) will be reviewed in Section 6.

Complex dynamic networks (*cDN*) can simulate practical models such as LSN of sensors and Internet objects or neural nodes in SNN (Pecora, 1990; Abbasi et al., 2013; Stanoev, 2013). Since *cDN* exhibits more sophisticated and uncertain behaviors than a single NN (Stanoev, 2013), its synchronization is a challenging problem. In this paper works studying relation between *n-Sci* and *N-sync* are reviewed with focus on: Synchronization of NN with stochastic perturbation, synchronization analysis, synchronization with aperiodically (adaptive) intermittent control, stability of spiking NN synchronization under stochastic perturbations, feedback control of NN synchronization, exponential synchronization of NN under time-varying sampling, synchronizing cort-oscillations in human brain, analytical description of a single oscillator, phase reduction methods for a single oscillator and oscillator network and complex networks synchronization.

In the next iteration the paper will revisit the problem of synchronization by generalizing the problem and reviewing the work on the *cN*'s with extremely large number of nodes. Such networks include physical, biological, chemical, and technological networks, as well as in the economic and social systems.

The topics covered include oscillator models on *cN*, phase oscillators, the onset of synchronization in *cN,* the evolution of synchronization process in *cN*, stability of synchronized *cN*, graph theoretical bounds to synchronizability, other stability function formalisms, relevance for bio-systems and *n-Sci*, and computer science and engineering.

1.2.6. Analytical tools

Tensor Networks (TN) as a useful tool in the modeling and design of LSN will be reviewed in Section 7.

TNs use an intuitive graphical language (*gL*) enabling efficient reasoning about them. The methods have been adapted to studying problems in physics, mathematics, and computer science.

In this segment the paper reviews the work covering: TN models, wire tensors, graphical singular value decomposition, matrix product states, TN based ML, the presence/absence of barren plateaus (bp) for global *ooF*, ML by q-TN and q-entanglement based learning architecture.

As a further extension of the survey, the objective will be to survey the work showing how TN's can be used for big data optimization problems by using relatively small size matrices and tensors.

These works will include topic like: low-rank tensor approximations via TN, TN models, reconfiguration of TN, distributed (concatenated) representation of tensors, tensorization, analytical representation of tensor trains (TT), matrix TT decomposition, operations in TT representation, (TT/MPS) splitting, large-scale optimization problems, generalized eigenvalue problems in TT formats, canonical correlation analysis in TT format and solving large-scale systems of linear equations.

1.2.7. Quantum simulations

Quantum (q) Simulations (qS), as an unavoidable part of the overall network design, will be reviewed in Section 8.

In digital *qS*, the evolution of the physical process on time scale is mapped, using the mathematical formulation of *q*-mechanics, onto the effective algebra of q-registers ($\mathscr{R}$'s) made of qubits. The q-time propagator, modelled by U operation, can then be implemented in digital steps as a sequence of q- logic gates (i.e. U transformations on the qubits) defining a q- circuit (Nielsen, 2000). In this segment the paper will concentrate on the contributions using the class of quantum spin models, which besides being extremely interesting on their own right, usually constitute the ideal formal conjunction between general q-mechanical models and their corresponding representation in terms of qubits. Spin models are in this sense the key to the *qS* of many-body q-models (Troyer & Wiese, 2005). Examples are the Hubbard model discussed in Casanova et al. (2012); Barends et al. (2014), or the Schwinger model in lattice theory (Hauke et al., 2013; Martinez et al., 2016; Klco et al., 2018). Here, the paper will emphasize the role of quantum correlations (Roggero et al., 2018) in many-body systems. Despite being in principle much more powerful, universal quantum simulators UQS are typically difficult to realize in practice compared to analog simulators, mainly due to the well-known stringent requirements for general purpose quantum computation (DiVincenzo, 2000). Here, it should be mentioned that hybrid qS have also been analyzed (Mezzacapo et al., 2015). The revied work covers the following topics: Preliminaries on qS, spin mapping, universal sets of quantum gates on NISQ processors, library of quantum gates sequences for quantum simulations, approximation and digital error, extracting physical observables, experimental results for tunneling of S = 1 total magnetization and Spin-1/2 chains, simulation of QML, quantum classification and quantum neurons, implementing quantum algorithm for binary-valued artificial neurons, implementation on NISQ processors, a quantum feed-forward neural network and pattern classification on a real quantum processor.

1.2.8. Complex networks theory framework

Throughout the survey, all these components of network design will be unified under the umbrella of the complex network theory framework. The relations between different sections of the paper are illustrated in Fig. 1.

1.2.9. Contributions

The paper provides comprehensive survey of sources of information to 7G network designer on:

- How to choose the ML algorithms to minimize the energy consumption in the network
- Possible speed ups by using lifelong learning.
- Achievable speed up in computing rates when using quantum neurons.
- How to choose U's in Q algorithms for a given objective
- How to create complex quantum models: lessons learned from quantum chemistry.
- How to estimate complexity of the algorithms
- How to estimate probability of wrong decision in quantum neural network
- How to design network connectivity to achieve network onset and full synchronization
- How to ensure network stability
- How to use TN theory in the system analysis
- How to use QS for the system analysis
- How to leverage experience from *n-Sci* and brain studies to enhance efficiency of complex networks: lessons used form assigning spatially and temporally varying waiting coefficients in neural networks, referred to as credit assignments.
- How to design and optimize Multi-Task Learning [MTL]:
- How to optimize the network by considering the complex interdependency between the different components of the system. This is a specific contribution of the paper which is valid for existing algorithms and the new algorithms that might be developed in the future.

NOTE: At this point it should be repeated that 6/7G networks will be, as all previous generations, open standards enabling competition between different technical solutions and standardizing a minimum of the system parameters that will be required to secure the compatibility of these solutions. For these reasons here the paper does not propose

specific solutions for different problems that 6/7G networks will face, but rather present a variety of technology enablers for a designer to choose from when building its own specific solution. However, in addition to identifying these solutions the paper presents an original complex network optimization framework considering the complicated interdependency between the different components of the system, that can be used for both, the existing algorithms and those that might be developed in the future. By using this framework, the network designers should be able to compare the performance of the compound systems using different components.

2. Classical NN

2.1. Advances in ML

Lifelong machine learning (LML): In real life, every learning experience or decision made increase the human's knowledge (experience ε), so that when next time faced with a similar question human can decide more efficiently. On the other hand, classical ML algorithms reset the learning process back to the beginning once they face a new problem to learn. For classical ML algorithms see Glisic and Lorenzo (2022).

Lifelong machine learning (*lifelong ML or LML*) is an advanced ML concept that learns continuously and uses accumulated experience ε from the past to improve the $\mathscr{L}$-process in the future. In other words, up to a certain moment, the $\mathscr{L}$-object (*lo*) has completed an ordered set/sequence of learning assignments, $\mathscr{T}_1, \mathscr{T}_2, \ldots, \mathscr{T}_N$ (*earlier assignments*), by using the respective datasets $\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_N$. The assignments are not of the same type nor from the same *domains*. For the (*N*+*1*)th assignment $\mathscr{T}_{N+1}$ using data $\mathscr{D}_{N+1}$, the learner can leverage the memorized *experience* ε to enhance the decision about $\mathscr{T}_{N+1}$. LML is supposed to optimize the decision about the new assignment. It can also optimize the decision of any $\mathscr{L}$-assignment by considering the rest of the assignments as the previous assignments. The memory keeps ε and after the completion of learning $\mathscr{T}_{N+1}$, the memory will be updated with the knowledge gained from learning $\mathscr{T}_{N+1}$.

Multi-Task Learning (MTL): By using the information of common interest shared by multiple $\mathscr{L}$-assignments (Caruana, 1997; Chen et al., 2009; Li et al., 2009) here the algorithm learns multiple related $\mathscr{L}$- assignments in parallel, to improve the performance. The algorithm introduces inductive bias in the joint hypothesis space of all assignments by leveraging the $\mathscr{L}$-assignment correlations.

MTL referred to as *batch MTL*, learns multiple learning assignments $\mathscr{T} = \{1, 2, \ldots, N\}$ at the same time. Each learning assignment $t \in \mathscr{T}$ with pertaining training data $\mathscr{D}^t$ has objective to maximize the improvements for *all* assignments. The work here mainly involves supervised learning (*sL*). For each $\mathscr{L}$-assignments t with corresponding training data $\mathscr{D}^t = \{(\mathbf{x}_i^t, y_i^t) : i = 1, \ldots, n_t\}$, and n_t the number of training/reference instances in $\mathscr{D}^t$, $\mathscr{D}^t$ is given by an unknown true mapping $\hat{f}^t(\mathbf{x})$ from an instance space $\mathscr{X}^t \subseteq \mathbb{R}^d$ to a set of labels $\mathscr{Y}^t(y_i^t \in \mathscr{Y}^t)$ (or $\mathscr{Y}^t = \mathbb{R}$ for regression). Parameter d is dimension of the feature /pattern. The system is supposed to learn $f^t(\mathbf{x})$ for each $\mathscr{L}$ assignment t so that $f^t(\mathbf{x}) \approx \hat{f}^t(\mathbf{x})$. Therefore, for a given *ooF* $\mathscr{L}$, *MTL* minimizes $\sum_{t=1}^{N}\sum_{i=1}^{n_t}\mathscr{L}(f(\mathbf{x}_i^t), y_i^t)$. Different from this batch *MTL, online MTL* (*OMTL*) learns the assignments one by one and memorizes decisions made earlier and uses the memorized experience ε to improve future $\mathscr{L}$-tasks (or to help some previous ones). So, *OMTL* is lifelong learning. *MTL* expects that $\mathscr{L}$-assignments are closely *related (high correlated)*. For different assumptions about $\mathscr{L}$*-assignment relatedness/correlation*, different modeling solutions are used (Evgeniou & Pontil, 2004; Baxter, 2000; Ben-David & Schuller, 2003; Daumé, 2009; Argyriou et al., 2008; Jacob et al., 2009; Ruvolo & Eaton, 2013).

Lifelong sL is a sequential $\mathscr{L}$-process where the $\mathscr{L}$-object has completed a sequence of *sL* $\mathscr{L}$-assignments, $\mathscr{T}_1, \mathscr{T}_2, \ldots, \mathscr{T}_N$ and saved the $\mathscr{L}$-results (experience ε). For $\mathscr{L}$- assignment $\mathscr{T}_{N+1}$, the learner leverages the memorized ε to improve learning f_{N+1} from $\mathscr{T}_{N+1}$'s reference D_{N+1}. After $\mathscr{L}$- $\mathscr{T}_{N+1}$, the memory is updated as well with the $\mathscr{L}$-results from $\mathscr{T}_{N+1}$. In (Thrun, 1996), a lifelong *sL* procedure was presented for two ε-based $\mathscr{L}$-methods: *k*-nearest neighbors (Altman, 1992) and Shepard's method (Shepard, 1968).

Lifelong Neural Networks: MTL net (MTL with neural network) (Caruana, 1997) is presented as a lifelong learning procedure in Thrun (1996). It operates as a batch MTL method. In MTL net, instead of creating a NN for each $\mathscr{L}$-assignment separately, a joint /universal NN for all the learning assignment is constructs. Several upgrades of *MTL net* are presented in Silver and Mercer (2005), Silver and Poirier (2004), Silver and Poirier (2007). These upgrades either use virtual training examples to generate the training data of all earlier $\mathscr{L}$-assignments or add contexts.

Lifelong Unsupervised Learning (uL): Most of the research here is focused on *topic or subject modeling (tM)* and *information extraction (iE)*. In *tM*, the knowledge acquired about the topic or subject in the past (experience on a topic- ε_t) in related domains or fields ($\varepsilon_{t,d}$) is impacting the modeling in the other or current field (Chen & Liu, 2014; Liu et al., 2016). Memory stores the ε_t. When it comes to *iE*, LML is also a natural choice since the objective of *iE* is to maximize the amount of extracted and stored useful data or knowledge. The *iE* is thus cumulative. The information extracted in the past helps to extract more data later with higher quality (Liu et al., 2016).

Lifelong Reinforcement Learning (LrL): In each iteration of reinforcement learning (*rL*), a $\mathscr{L}$-object learns decision steps through action/correction relation with a time varying environment (Kaelbling et al., 1996; Sutton and Barto, 1998a). In each iteration, the $\mathscr{L}$-object observes the environment in a given instant and selects a move from the available options. The move changes the state of the environment. Depending on the gain of the state transition, the agent selects, it receives a reward or fine. The agent learns a sequence of movements, through the repetition of the process, so maximizing the long run sum of rewards. The objective of *rL* is to learn an optimal policy that maps states to movements.

A *rL* algorithm has no input/output pair as in *sL*. For details, and more examples, see sections on Q-Learning in Glisic and Lorenzo (2022). For good results, a large number of iterations is needed, especially in high-dimensional (*hd*) control problems, resulting in high computational complexity. To mitigate the problem of complexity, *LrL* has been proposed. The expectation is that using the accumulated experience ε from other $\mathscr{L}$-assignments should improve the $\mathscr{L}$-object's decision making in the new $\mathscr{L}$-assignments. *LrL* was introduced in Thrun and Mitchell (1995) and further elaborated by several other authors (Ammar et al., 2015).

2.2. Deep NN and federated learning (fL)

Optimization algorithms for signal processing (SP) applications in advanced wireless networks nowadays often require high complexity. This diminishes feasibility for real-time processing. A deep neural network (DNN) is used to approximate any unknown input/output nonlinear mapping of SP algorithm. If an acceptable accuracy in the approximation can be achieved by a DNN this may be a solution, since a DNN may require a smaller number of operations while providing acceptable performance.

In the open literature, similar logic has been used for replacing an iterative optimization algorithm by its DNN approximation. As an example, reference (Gregor & LeCun, 2010) proposes using a multilayer network to approximate by a separate layer each iteration of the iterative soft-thresholding algorithm (ISTA) for sparse optimization (Beck & Teboulle, 2009).

Similar ideas are used in Hershey et al. (2014), Sprechmann et al. (2013) for nonnegative matrix factorization. Authors in Andrychowicz et al. (2016), Li and Malik (2017) propose learning per-iteration behavior of *G*-based algorithms.

Regarding the communication tasks, the recent works in O'Shea

et al. (2016) reported using DNN in anomaly detection and decoding. Different from the above, these papers have not been interested in algorithm approximation but rather in signal modeling. Following the unfolding concept from Gregor and LeCun (2010) authors in Samuel (2017) proposed a DNN to approximate MIMO detection problem while using DNN, for links scheduling is discussed in Cui et al. (2019).

For *autonomous driving* algorithms the freshness (age) of information (AoI) about the vehicular network state is of paramount importance and proper network resource allocation aware of the AoI is the major technical issue in this field. The problem modelling and possible solutions based on DNN have been also considered (Chen et al., 2020; Zhuang, 2012).

Federated Learning (fL): The concept of *fL* system (*fLS)* is proposed in Yang et al. (2019), Konecný et al. (2016a, 2016b), Brendan et al. (n.d.). These ML models are designed to prevent data leakage for data distributed on separate locations.

Privacy of fL: This is the main feature of *fL*. Secure Multi-party Computation (SMC) involves a group of users and provides security based on the so called zero knowledge (*zN*) concept, where each user knows only its own input and output. Unfortunately, practical implementation of the *zN*, is not efficient. So, it could be considered to allow disclosure of a part of knowledge if security guarantees are provided. Accepting controlled reduction of security requirement in SMC in exchange for efficiency in implementation, is presented in Du et al. (2004). Authors in Mohassel and Zhang (2017) used SMC for training ML with semi-honest assumptions and in Kilbertus (2018) MPC protocols for model training while preserving user's data privacy. A solution for SMC is Sharemind's framework presented in Bogdanov et al. (2008). Authors in Mohassel and Rindal (2018) proposed a model assuming that the majority of the participants are honest and discussed the data protection in such environments. For the implementation of these solutions users' data need to be confidentially distributed among non-colluding servers (Araki et al., 2016; Furukawa et al., 2016; Mohassel et al., 2015).

Differential Privacy (*dP*) or -Anonymity (Sweeney, 2002) for data privacy protection (Abadi et al., 2016a) is another line of work here. The methods of *dP* (Agrawal & Srikant, 2000) are based on the idea of intentionally disturbing data by adding noise or obscuring certain sensitive features so that the adversary cannot identify the individual. For this the data needs to be transmitted somewhere else, and a part of the algorithm is a trade-of between accuracy and privacy. Reference (Geyer et al., 2017), introduces a *dP* approach to *fL* to additionally protect user's data by not discovering user's role during training.

Homomorphic Encryption (h&), presented in Rivest et al. (1978) is also included for protecting data privacy by encrypted parameter exchange during ML (Giacomelli et al., 2017). Here, no information is exchanged, and it cannot be estimated by the other party's data. So, the leakage of data is significantly reduced. The *h&* has been also adopted for training data on cloud (Yuan & Yu, 2014; Zhang et al., 2016). Additive *h&* (Acar et al., 2018) are implemented in practice, using polynomial approximations of non-linear functions in ML algorithms. This enables accuracy-privacy trading-offs (Aono et al., 2016; Kim et al., 2018).

Block Chain protection of FL has been considered as well in references (Pokhrel, 2020). In Table 1 Cross-Technology Coverage (focus-NN) of the reviewed papers focused on neural Networks is presented. One can see that these papers besides focusing on NN cover very little if any of the other topics needed for getting a fair insight into complex network design. In the sequel, papers with different focus will be reviewed. Their *coverage* will be higher and higher but still significantly lower than one achieved in this paper.

3. Spiking NN

The interest of this paper in spiking neural networks is twofold. First, using spikes instead of continuous presence of signals enables several orders of magnitude better energy efficiency. Second, deep understanding of the neurological processes enables better insights in the operation of the human brain which is expected to help in further developing better modelling and design of the algorithms for control of artificial neural networks.

3.1. Spiking neuron

Learning and memory in biology are based on synaptic (*sy*) plasticity (*syP*) (see Fig. 2). This section surveys models of short-term (*sT*) and long-term (*lT*) *syP*, including *spike-timing dependent plasticity (STDP)*. The algorithms depending on timing, require precise *N-sync*, topic covered in Section 6. This paper focuses on simple models based on integrate-and-fire type neurons. Here *sy* update rules for *sT* or *lT syP* depend on spike timing, membrane potential, and the value of the *sy* weight. The paper also reviews the literature discussing the relations of these rules to *sL* and *rL*.

It is believed that learning, memory, and *cort*-plasticity (*coP*) are based on *syP* changes, although the relation between *syP* features and

Table 1
CrossTechology *coverage (focus-NN)*.

1	2	3	4	5	6	7	8	9	10
(Glisic and Lorenzo, 2022)	ML	√	√			√	*Details on ML and QC modelling and analysis*		
(Chen & Liu, 2016)	ML	√					*Detailed Modelling and analysis of Life Long ML*		
(Caruana, 1997; Chen et al., 2009; Li et al., 2009)	ML	√					*Multi-Task Learning [MTL]*		
(Evgeniou & Pontil, 2004; Baxter, 2000; Ben-David & Schuller, 2003; Daumé, 2009; Argyriou et al., 2008; Jacob et al., 2009; Ruvolo & Eaton, 2013)	ML	√					*online multi-task learning (OMTL)*		
(Thrun, 1996; Caruana, 1997; Silver & Mercer, 2005, Silver & Poirier, 2004, Silver & Poirier, 2007)	ML	√					*Lifelong Neural Networks: modelling and analysis*		
(Chen & Liu, 2014, Liu et al., 2016)	ML	√					*Lifelong Unsupervised Learning*		
(Kaelbling et al., 1996, Sutton and Barto, 1998a)	ML	√					Lifelong *Reinforcement Learning (LRL): modelling and analysis*		
(Gregor & LeCun, 2010, Beck & Teboulle, 2009, Hershey et al., 2014, Sprechmann et al., 2013, Andrychowicz et al., 2016, Li & Malik, 2017, O'Shea et al., 2016, Samuel, 2017, Cui et al., 2019, Chen et al., 2020, Zhuang, 2012, Yang et al., 2019, Konecný et al., 2016a, 2016b, Brendan et al., n.d.; Du et al., 2004, Mohassel & Zhang, 2017, Kilbertus, 2018, Bogdanov et al., 2008, Mohassel & Rindal, 2018, Araki et al., 2016, Furukawa et al., 2016, Mohassel et al., 2015, Sweeney, 2002, Abadi et al., 2016a, Agrawal & Srikant, 2000, Geyer et al., 2017)	ML	√					*Deep NN and Federated Learning*		
(Rivest et al., 1978, Giacomelli et al., 2017, Yuan & Yu, 2014, Zhang et al., 2016, Acar et al., 2018, Aono et al., 2016, Kim et al., 2018)	ML	√					*Homomorphic Encryption*		

1 reference, 2 focus, 3 classic, 4 quantum, 5 complex networks, 6 tensors, 7 q-simulations, 8 contribution, 9 energy efficiency, 10 computational efficiency.

functional consequences is still rather unclear. In experimental initiation, *syP* changes can be a consequence of stimulations defined by pre-sy ($^{-}$sy) firing rates (Bliss & Lomo, 1973), postsynaptic ($^{+}$sy) membrane potential (Artola et al., 1990), calcium entry (Lisman, 1989), or spike timing (Markram & Sakmann, 1995).

While comprehensive biophysical models are of paramount importance to understand *syP* mechanisms, phenomenological models (*phM*) describing the *syP* changes with no need to specify the mechanism of these changes, are simpler. Therefore, the latter are widely used in analytical and simulation analysis. Experimental and theoretical results have been compared for several *phM*. Here, a *syP* changes from a $^{-}$*sy* to a $^{+}$*sy* neuron are considered. The intensity of a connection *from-to* is measured by the height of the $^{+}$*sy* potential at onset. The directions and magnitudes for *syP* changes can be formulated as *sy* $\mathscr{L}$*-rules*. These rules can be defined either theoretically or experimentally. The rules can be developed for experiments in which *syP* is measured as a result of $^{-}$*sy* and $^{+}$*sy* spikes (Cooper et al., 2004).

Example 1. *For the classification of the syP rules, the time required to induce a change and the duration of persistence of the change are used. It takes about* 1 s *or less to induce the changes for both sT and lT syP. In sT syP, it takes a sequence of 8* $^{-}$*sy spikes at* 20 Hz *to get decreasing (depression dep) or increasing (facilitation fac) responses in the* $^{+}$*sy cell. For sT syP this change persists less than a few 100's ms: the time required for the amplitude of the* $^{+}$*sy response to comes back to near-normal value is less than a second* (Markram et al., 1998).

Different from sT syP, lT potentiation (pot) and dep (lTP and lTD) characterize the persistent changes of syP reactions. The time needed for inducing the changes is relatively short. In STDP (Morrison et al., 2008) *for instance, a change of the syP can be caused by 60 pairs of* $^{-}$*sy and* $^{+}$*sy spikes with a frequency of 20 Hz, ending the stimulation after* 3 s. *On the other hand, this change can persist for more than 1 h. The final stabilization of a post-syP can be reached only thereafter, referred to as the late phase of lTP* (Frey, 1997). *In this process, despite the changes induced by lTP and lTD, the neurons in the brain must remain within a sustainable activity regime* (Turrigiano & Nelson, 2004).

The models discussed here operate by using *uL* rules. Learning is a continuous process of adjustment of the *syP* to the statistics of the activity of $^{-}$*sy* and $^{+}$*sy* neurons. In *rL* (Sutton & Barto, 1998b), on the other hand, the change depends on a decision outcome, that reflects the current reward (Schultz & Montague, 1997). The *rL* rules are different from *sL* since the success signal is considered as an *uL* rule (Bohte et al., 2002). The *sL, uL*, or *rL* rules are widely studied in open literature.

The importance of molecular mechanisms (Lisman, 1989) for models of *syP* (Lisman & Zhabotinsky, 2001; Shouval et al., 2002; Florian, 2007; Graupner, 2007; Zou, 2007; Badoual et al., 2006) as well as the importance of the $^{+}$*sy* voltage (Kelso et al., 1986; Sjostrom & Turrigiano, 2001), is also explored in the literature.

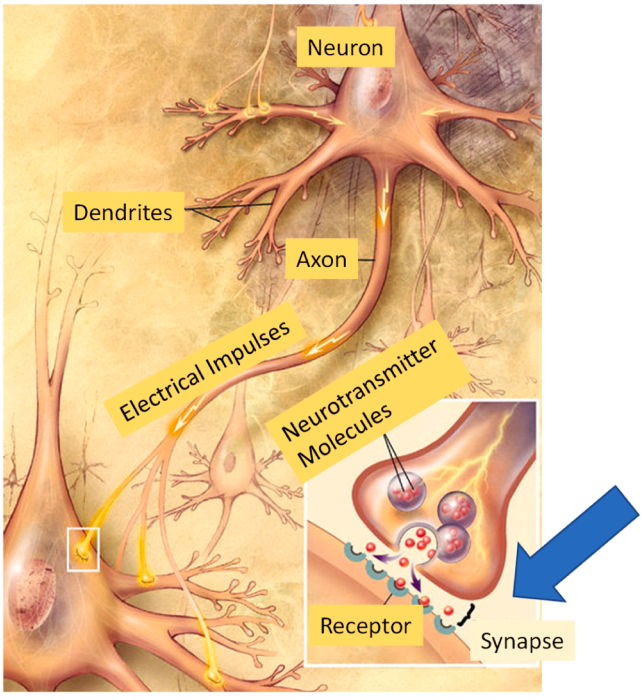


Fig. 2. Artistic interpretation of the major elements in chemical synaptic transmission (http, n.d.a).

3.1.1. *Implementing various types of syP rules in VLSI*

Reference (Azghadi et al., 2014) reviews the most common and useful electronic building blocks required for implementing various types of *syP* rules in VLSI. In addition, authors describe analog very large-scale integration (VLSI) circuit implementations of multiple *syP* rules, ranging from phenomenological ones (e.g., based on spike timing, mean firing rates, or both) to biophysically realistic ones (e.g., calcium-dependent models). They discuss the application domains, weaknesses, and strengths of various representative approaches proposed in the literature, and provide insight into the challenges that engineers face when designing and implementing synaptic plasticity rules in VLSI technology for utilizing them in practical applications.

In (Azghadi et al., 2017) a high-performance nano-scale Complementary Metal Oxide Semiconductor (CMOS)-memristive circuit, was presented, which mimics a number of essential learning properties of biological *sy*'s. The proposed *sy* circuit that is composed of memristors and CMOS transistors, alters its memristance in response to timing differences among its $^{-}$sy and $^{+}$sy action potentials, giving rise to a family of STDP. The presented design advances preceding memristive synapse designs with regards to the ability to replicate essential behaviors characterized in a number of electrophysiological experiments performed in the animal brain, which involve higher order spike interactions. Furthermore, the proposed hybrid device CMOS area is estimated as 600 μm^2 in a 0.35 μm process—this represents a factor of ten reduction in area with respect to prior CMOS art. The new design is integrated with silicon neurons in a crossbar array structure amenable to large-scale neuromorphic architectures and may pave the way for future neuromorphic systems with spike timing-dependent learning features. These systems are emerging for deployment in various applications ranging from basic neuroscience research to pattern recognition, to Brain-Machine-Interfaces.

3.2. *sN networks (SNN)*

Design principles of *SNN's*, sometimes called the 3rd generation of NN, use widely accumulated knowledge on brain operation and research results in *n-Sci*'s. They are based on modeling of cross-neuronal *sy* interactions, related to the time of spike firing. When it comes to computational power SNN's outperform the classical NN using threshold or sigmoidal units. They enable design of the models with high capacity of memorizing and adaptability. Here, an adequate $\mathscr{L}$-algorithms is needed that at the same time exploit specific characteristics of SNN's and easy-to-use, traditional connectionist models for which sophisticated simulators are already available.

The original work (McCulloch & Pitts, 1943) presented a NN model with cross-neuronal connections in the form of ($N_i \subset N_j$) links with weight (w_{ij}). If the weighted sum of the states of all the neurons N_i, representing the input to a neuron N_j reaches level above a certain threshold of N_j, the state of N_j is set to active, otherwise it stays inactive as shown in Fig. 3.

Even, such simple networks can realize several analytical functions mapping input/output states. By optimizing the cross-neuronal weights, these ANN can "learn" such mapping. A number of $\mathscr{L}$-rules are used, for both families *sL*, and *uL*. Gradient descent algorithm with error

backpropagation (*bp*) (Glisic & Lorenzo, 2022; Rumelhart et al., 1986) that enforces the NN behavior to some target function, is an example of the *sL* algorithms.

The origins of uL in NN are associated with the work on syP presented in Hebb (1949), stating: "When an axon of cell is near enough to excite cell or repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that its efficiency, as one of the cells firing, is increased."

Later this work was followed by extensive work on:

Calculability: Discussion on NNs processing power (Huh & Sejnowski, 2017).
Complexity: The *loading problem* being NP-complete (Valiant, 1984; Mehta et al., 2002)
Capacity: MLP (Multi-Layer P^{er}), RBF (Radial Basis Function networks) and WNN (Wavelet Neural Networks) being universal approximators (Cybenko, 1988)
Regularization theory (Poggio & Girosi, 1989); PAC (Probably Approximately Correct)-learning (Valiant, 1984); St-learning theory, VC (Vapnik - Chervonenkis)-dimension, SVM (Support Vector Machines) (Vapnik, 1998)

Information coding: In general, data can be encoded into the number of spikes (spike rate) at the output of the neuron or by a position (timing) of a spike at the output.

Example 2. *The relevance of the two options has been intensively studied. Different options for timing coding are shown in* Fig. 4. *Arguments against rate coding are discussed in* Thorpe et al. (1996). *Poisson distributing rate code for describing how neurons transmit information has a lot of supporters among physiologists. The opponents of this theory argue that the high volume of transmitted information needed to support human vision, is hard to reconcile with Poisson rate codes. To differentiate selectively between complex visual stimuli a human needs* 100–150 ms. *On the other hand, in the feedforward (ff-) architecture of visual system, with multiple layers of neurons, in practice at most one spike could be fired by each neuron during round trip adaptation process. A group of neurons with st-firing as a function of the stimulus, could realize a time varying rate code: a spike density code. Having such a group to encode a single variable is expensive energy wise* (Olshausen, 1996). *This suggests that the timing of individual spikes can encode data, and not just the rate of spikes.*

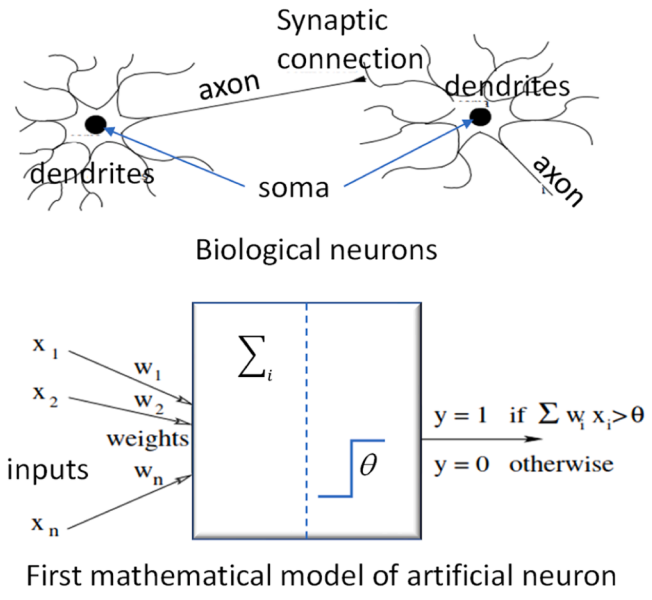


Fig. 3. The first model of neuron incorporated the main characteristics of a natural neuron: All-or-none output resulting from a non-linear transfer function applied to a weighted sum of inputs.

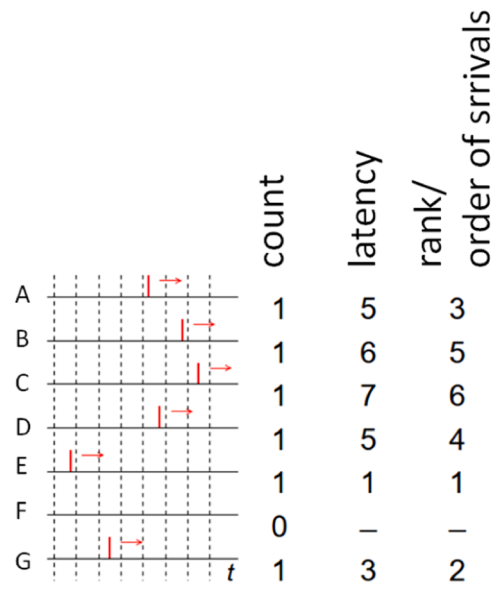


Fig. 4. Comparing different coding alternatives for spiking neurons.

High resolution time positioned spikes achieve better encoding, given a small set of *sN*. The pros and cons of other coding algorithms have been discussed in Recce (1999) and analyzed in Thorpe et al. (2001). For sNN with backpropagation learning for brain visual dynamics decoding see Stauffer and Zhang (2023).

3.3. *n-Sci and AI*

In the sequel, some physical processes in the brain are described and their relationship with research in *n-Sci*, research challenges covered by *nS* and *dS* theory, with focus on the incoming interest in research of brain-inspired intelligence (*biI*). A practical way to mimic brain intelligence (*bI*) is to build up dynamic cortical networks (*corN's*) that implement the brain functions, instead of using only ANN (Hu, 2021).

Here this paper provides a complex network (*cN*) and space/time dynamics (network dynamics *nD*) approach to modeling the brain and *corN's* and develop integrated concept of *n-Sci* and *nD* for building *biI* with $\mathscr{L}$ and resilience functions. For this the paper covers issues as fundamental concepts and principles of *cN* (to be revisited in few additional iterations in the following sections), major challenges and solutions in *n-Sci*, and theory and implementation issues in *hS*, including corresponding experimental and theoretical studies about the brain and intelligence. Topics like theory and practice of brain science, data science with emphasis on big data mining, q- information science including discrete and continuous variables (to be addressed in the sequel in more details), and machine behavior covering dynamics and stability problems are also briefly discussed toward future applications.

Here this paper first reviews the main problems in modeling, optimization, and implementation of *cN* including analytical tools for performance analysis. Then it surveys the relevant work in *n-Sci* and hybrid space/time dynamic systems, highlighting problems and solutions related to the brain and intelligence. Commonalities in *n-Sci* and *nD* are further discussed to explain the reverse-engineered steps toward *biI*.

Complex Network-cN: A *cN* theory is used to model a multitude of elements interconnected in such a way that the network characteristics cannot be understood from the current state of a few of its individual components (Glisic, 2016). Internet, mobile communication networks, the brain and neural systems are examples of such networks. *cN* are represented by a graph $\mathscr{G}$ having a set of nodes $\mathscr{V}$ and a set of edges $\mathscr{E}$, in which $\mathscr{V}$, $\mathscr{E}$ may change in time and space. The connections among

nodes are specified by the adjacency matrix $W = [w_{ij}](i, j \in \mathscr{V})$. In practice, variations in W may enable the development of the brain or in extreme case cause a breakdown of the Internet.

Based on observation of real networks, fundamental properties in *cN* include modeling of both topology and collective dynamics. Main features used to characterize the network include clustering coefficient, distribution of node degree, modularity, path length, community, and hierarchy structures. On the other hand, collective dynamics, such as synchronization (to be discussed in detail later in Section 6), intelligence, emergence, and resilience contribute to maintaining network functions. The Erdös–Rényi (ER) network, discussed in Glisic (2016), examines how a graph, representing the network, is modified with changes in density of edges, indicating the importance of studying the random networks. Small-world networks (SW), presented in Glisic (2016) by Watts-Strogatz (WS), have a short average path length and large clustering coefficient, whereas the Barabási–Albert (BA) scale-free (SF) network, described in Glisic (2016), has node degree characterized by power-law distribution. These models are used to represent or design systems in practice, with features (e.g., development, $\mathscr{L}$, and recovery) that can be improved by optimizing network control (Arenas et al., 2008b).

Works on the *cN* also include investigation of the structural effects on overall network behaviors, induced by failures, attacks, or viruses in network elements (Majdandzic et al., 2013). Quantitative analyses of structures and functions in complex brain networks (*cbN*) has also attracted considerable attention (Bullmore & Sporns, 2009). In social networks, people with common interests interact among themselves and then make decisions. This type of overall network behavior can be modeled via a *cN* spreading and regulated with dynamic control (Arenas et al., 2008b). Similarly, in a *corN*, each neuron has about 10,000 *sy* connections. Here this paper is interested in finding out how the neurons interact with nearby and distant neighbors and how the network dynamics impacts cognitive behaviors. *n-Sci*, especially the cross disciplinary studies with *cN*, has offered some plausible answers (Bullmore & Sporns, 2009).

n-Sci Inspired AI (niI): The research in *n-Sci* including neurobiology, physiology, and cognitive science form a fundamental basis to the work on intelligence (Marblestone et al., 2016a).

With advances in brain research, the field of *n-Sci* has been broadened to include different methods for investigating *corN* at different scales in time and space. By studying the structure-function relations in *bN* (Bullmore & Sporns, 2009) empirical research of *cN* have attracted significant attention. The quality and resolution in visualizing *bN* have been significantly improved by advances in image processing. Among the other activities, computational *n-Sci* study also analytical representation of the brain and neural operations, for full understanding of the relations, neural activities vs. behavior (Dayan & Abbott, 2001).

Work presented in Hodgkin and Huxley (1952) models the inhibition- excitation time variation of an elementary neuron for producing action potential (spike or nerve impulse) by using ion channel mechanisms. Given an excitation, neurons generate spikes and then pass the spikes to $^{+}$*sy* neurons. Here recall that synapse (*sy*) is the unit that connects two neurons. It can be modeled by a coupling weight, referred to as *sy* efficacy. Hundreds of milliseconds long changes of a *sy* are called *sT* syP, that can be detected after a $^{-}$*sy* neuron generates spikes. In a *lT* lasting few seconds or hours, the syP depends on the joint activity of $^{-}$*sy* and $^{+}$*sy* neurons (Dayan & Abbott, 2001). There is more and more evidence of integration of *n-Sci* and ML, in an effort to construct computing and intelligence based on the principles they operate in human body (Marblestone et al., 2016a).

Neurons in the brain act collectively to adjust themselves to the environment (Graves et al., 2014) by interconnecting with each other. Neural coding defines how spiking patterns depend on excitation and how reactions are interpreted from spike trains. The mechanisms of *nD* are used for solving coding design, along with *st*-analysis (Dayan & Abbott, 2001; Clopath et al., 2010). Since mesoscopic dynamics may relate micro-spiking activities and system-level functions (Curto, 2019; Whiteway, 2019), there is also a need for interpretable dynamic models (*dM*) of neuronal nodes activities. An important segment of network *n-Sci* research models the evolution in time of the *bN*, facilitating anticipation of irregular impacts on the brain (Bassett et al., 2017). Because of the multitude of neurons, *sy*'s and spike patterns, multiscale (*ms-*) modeling is needed to further elaborate the study of *n-Sci*, focused on the brain and intelligence.

ms-Space/Time Dynamics: An analytical model of a *ds* can be defined either by a continuous-time (*ct-*) differential or a discrete-time (*dt-*) algebraic equation. The model facilitates prediction of the system's long-term behavior, together with stability and synchronization analyses. Hybrid systems (*hS*) involving evolution in time across multiple space/time scales, are recently attracting significant attention (Cassandras, 2014). Such a system is defined by a mixture of differential and algebraic equations (Goebel, 2009), jointly characterizing both *ct-* and *dt*-dynamic behaviors.

To simulate excitatory *sy*, a pulse-coupled model is incorporated into ct-systems and is studied on integrate-and-fire neural oscillators (Arenas et al., 2008a; Guan & Chen, 1999). The main characteristic of a *hS* is its space/time correlated behaviour, bringing adaptability in control actions (Branicky et al., 1998). The *hS-s* are appropriate to model neural units in *corN*, considering the spiking stimulus and the stimulus-response pattern (Dayan & Abbott, 2001; Whiteway, 2019). Incorporating *ms*-space/time dynamics enhances ML by introducing a causality mechanism (David et al., 2006).

Authors in Deneve (2017) describe the brain as a learner with adaptability. When adapting to surroundings, different neurons emit spike patterns with varying inter spike intervals and demonstrate different dynamics at a mean firing rate (Breakspear, 2017). Observations show that neural systems (*nS*) may demonstrate collective dynamics, like frequency synchronization (to be discussed later in Section 6) and asynchronous oscillations with certain frequencies (e.g., α -band and β-band) (Fultz et al., 2019). Such collective behaviors of *nS* enable the establishment of memory, $\mathscr{L}$ and motion actions (Connors, 2007; Gerstner, 2014). Although having the deep hierarchy structure, ANN does not utilize advanced dynamic mechanisms of NN (Abbott et al., 2016b). They use reverse- engineered (*rEng-*) methods trying to design ms-*nS* for mimicking brain intelligence.

A multitude of recent works are focused on the interdisciplinary research of *n-Sci* and *nD*, indicating such *rEng*-intelligence steps (Hu, 2019).

Brain-Inspired Intelligence (biI): Authors in Guan and Chen (1999) consider that a machine has "intelligence" if the behaviour of an intelligent object and a human cannot be distinguish. The word intelligence assumes objective directed behaviors, like prediction and optimization (Werbo, 2009). When it comes to structure and function AI use biological lessons from cortical networks (*corN*) (Werbo, 2009). In *corN*, both hierarchical structures and heterogeneous neuro-*sy* node dynamics can be seen. Dynamics of neural circuits and nodes supports cognitive capabilities, like those related to perceptual decision making and learning (Marblestone et al., 2016a; Gerstner, 2014).

By broad $\mathscr{L}$-networks network types using versatile neural nodes instead the deep hierarchy configuration for image recognition assignments are considered (Chen et al., 2019). There are proposals for using an integrated approach to *nD* with nonlinearity (O'Reilly et al., 2014b) for characterizing macro behaviors, such as opinion propagations. Network dynamics models are being used to specify and reconstruct neural activities supporting possible brain intelligence (Breakspear, 2017). It is believed that *nD* with *n-Sci* could be used to design brain--mimicking computer and intelligence, based on the spike-based machine intelligence (*MI*) principles (Ekman, 2012).

In this context the question is: *How much MI is sensitive to different types of adversities?* In general, this depends on the type of *MI* being used. Nowadays ML algorithms are trained on preset data. Perturbed inputs,

like damaged images or noise corrupted signals could impact the optimization process and so reduce the overall network performance (Frady & Sommer, 2019). Authors in Goodfellow et al. (2014) use a small--sample learning method. *St*-model-based $\mathscr{L}$-schemes have a discrimination capability to operate with inadequate sample data, but still are vulnerable to tampered structure or data. Besides the $\mathscr{L}$-capabilities, intelligent networks need more brain mimicking features, especially when operating with perturbed (or damaged) data or structures. Here, a resilience function like immunity is needed, to enhance the $\mathscr{L}$-process when operating with perturbed or insufficient data and unpredictable failures.

The brain and neural systems (Avena-Koenigsberger et al., 2018) have a resilience function incorporated in the overall system. Certain immunity is included in the natural organisms, making them insensitive to certain diseases and damage. This idea is also built into artificially designed systems, leading to resilience control with fault tolerance (Dolk, 2017). Using *nD* solutions to immunize real systems and make them insensitive to structural vulnerabilities and changes in environment, has also been studied (Cassandras, 2014; Dolk, 2017). More work is still needed to better understand how dynamic resilience could be included in intelligence. When it comes to *cN* and hybrid systems (Guan et al., 2019; Gao et al., 2016; Yan et al., 2017; Yu & Kaynak, 2017), *n-Sci* and *nD*, can integrated by using high dimensional (*hD*) NN's, for building *biI*.

The latest works in this field include use of neuroscience and network dynamics paradigms in designing brain-inspired intelligence (Hu et al., 2022). Brain-like intelligent data mining mechanism based on convolutional neural network is presented in Wen et al. (2023). A discussion on how brain organoids are revolutionizing neuroscience is presented in Abrams (2023). Higher-order interactions in functional brain networks in state of disorder is discussed in Kurkin (2023). An EEG based study of scientific problem solving and brain symmetry index is presented in Wang et al. (2023), Romanchuk et al. (2023) and potential of brain-computer interfaces in Rostami et al. (2023). An insights into electrophysiological brain states dynamics is presented in Tabbal et al. (2023). Experiments using R code, MRI measurements and SpikerNet are discussed in Panayotova et al. (2023), Udayakumar and Subhashini (2023), Coventry and Bartlett (2023). Assessments of variability in cortical and subcortical measurements and within-network connectivity of the brain using test-retest data is presented in Tavakoli et al. (2023). The latest work on devices used in these measurements are presented in Wu et al. (2023), Aimone and Misra (2023), Jia and Zhao (2023).

3.4. Deep learning and n-Sci

Until recently, ML and *n-Sci* have been carrying on their research without any insight into each other's results. Brain science has become aware of several brain segments, cell types with different functions, molecules, cellular states, and methods for computation and data storage. Opposite to this, most of the time ML implements simple objective function optimization (Glisic, 2023a, 2023b). Such an approach enables comprehensive internal modelling and algorithmic processing in multilayer ($\mathscr{ML}$) and *rN* (LeCun et al., n.d.). Here the commonalities of these two research areas are of interest and the possibility to connect these prospects.

Authors in Marblestone et al. (2016b) suggest that research in *n-Sci* and ML is moving towards convergence, which is the argument of this survey as well. There are three aspects of ML supporting this argument.

1) ML optimizes predefined objective functions.
2) Recently the research in ML has been introducing complex *ooF*'s, varying across layers and time, resulting from network segments interactions. To improve $\mathscr{L}$-process ML algorithms can incorporate in *ooF* request for temporal coherence in lower layers (spatially non-uniform *ooF*) (Sermanet et al., 2013). The *ooF* schedules (temporally non-uniform *ooF*) improve generalization (Andrew et al., 2013; Goodfellow et al., 2014b; Gülcehre et al., 2016) and adversarial networks allow *G*-based training of generative models (Goodfellow, 2014a). Solutions of using simpler networks, to provide initial guesses "hints" to improve the training of complex networks are presented in Romero et al. (2014).
3) In the ongoing process of diversification/optimization of the $\mathscr{L}$-architectures, ML nowadays uses memory cells with multiple persistent states (Hochreiter et al., n.d.), more advanced elementary units such as "capsules" and other structures (Hinton et al., 2011) content addressable (Weston et al., 2014; Graves et al., 2014) and location addressable memories (Graves et al., 2014), as well as pointers (Kurach et al., 2015) and hard-coded arithmetic operations (Neelakantan et al., 2015).

To attract more attention to these three ideas in *n-Sci*, authors in Marblestone et al. (2016b) summarize them in the form of three hypotheses about the brain, that can be restated as:

H1 The brain finds the best optimization objective functions (ooF). The main premise for relating the two fields is that the brain, like ML algorithms, can find the best *ooF*. The idea is that neurons in a brain can modify the characteristics of their *sy*, to perform better at executing the assignment the *ooF* defines for them. Humans occasionally get close to the best performance in a given segment, e.g., during movement (Körding, 2007), suggesting that the brain can learn optimal strategies. They manage to maximize movement gains with minimum energy (Taylor et al., n.d.). From the computation point of view, nowadays *ooF* enables solutions for very complex motor tasks (Mordatch et al., 2012). Marblestone et al. (2016b) suggests that the best *ooF* is found by optimizing the models used by the brain. It also suggests that for this the brain needs to have algorithms for credit assignment *(cr-ass)* in $\mathscr{ML}$ and *rN*.
H2 The ooF are space/time diverse: A second fact is that *ooF* does not need to be global. Neurons in different brain segments may focus on optimizing different *ooF*, minimizing the error of movements, managing reactions to visual stimulus, or paying attention. For locally generating such an *ooF*, neurons should locally asses the fitness of their *st*-model of their inputs. An *ooF* may be different in time, e.g., enabling a human when young first to comprehend simple visual data early on, and more comprehensive $\mathscr{L}$'s in the following steps later allowing the evolving brain to establish more complex knowledge based on simpler experience. The *ooF* in the brain are complex and space/time dependent.
H3 Problem Specialized Systems: The types of information flow appear to be distinct across different brain segments, specialized to solve different processing problems. Some segments are recurrent, adjusted for short-term memory storage (Wang, n.d.). Others cell types can switch between different states of activation, persistent/-transient firing mode, in response to given neurotransmitters (Hasselmo, n.d.). The thalamus is for example a segment having information from other segments flowing through it, suggesting being responsible for information routing (Sherman, n.d.). Basal ganglia are a segment participating in *rL* and gating of discrete decisions (Sejnowski et al., n.d.). Specialized algorithms enhance efficiency of solutions to data processing problems, and the brain is using them when appropriate.

These concepts are mimicked by recent advances in ML but (Marblestone et al., 2016b) argues that the brain operation is still much different from any of today's ML techniques. For example, biology gives little information that could be used for *sL* (Fodor et al., 2002). On the other hand, there is a lot of information available for *uL*. For making *uL* solve the "right" problems a sequence of deterministic *ooF*'s is deeded to design circuits and reactions according to preset evolutional stages, enabling a relatively small amount of information to produce the right behavior.

Based on this and other studies (Ullman et al., n.d.), authors in Marblestone et al. (2016b) suggest that many of *ooF's* used by the brain come from such an internal bootstrapping process. The potential bootstrapping mechanisms, operate with *uL* and *rL* and go well beyond the types of $\mathscr{L}$-algorithms used in today's ML (Bengio et al., 2009; Werbos, 1974; George et al., 2009a; Kappel et al., n.d.; Baldi et al., 2015).

Bio-implementation of gradient (G-) descent: To develop bio-$\mathscr{L}$ (*bL*) as good as ML, better *G*- propagation methods are needed. Different from earlier used assumptions, nowadays there are many *bio-plausible* methods enabling a neural circuit to realize optimization processes that, like *bp*, can use the *G*. Examples of these methods are generalized recirculation (O'Reilly, n.d.), contrastive Hebbian $\mathscr{L}$ (Xie et al., n.d.), random feedback (*fb*-) weights together with *sy* homeostasis (Lillicrap et al., 2014; Liao et al., 2015), STDP with iterative guessing and target propagation (Bengio et al., 2015a), complex neurons with *bp* action--potentials (Körding et al., n.d.), and others (Balduzzi et al., 2014).

Temporal credit assignment (t-cra): The bio-realizations of *bp* explained above, can be used in feedforward networks, but it does not provide a natural realization of bp *through time* (*bpTT*) (Werbos, 1990) for *rN*, used in ML for training *rN* on sequential processing assignments. The *bpTT unfolds* a *rN* over a number of discrete time steps and then uses *bp* over such structure to assign credit to different units at a corresponding time steps. Authors in Ollivier and et al., n.d.) argue that it is unclear how much such assignment is truly needed for $\mathscr{L}$ given temporally extended assignments.

If memorized data and representations (Gershman et al., 2012; Weston et al., 2014) of temporal context would be available, this could eliminate the need for *t-cra* since memory could "spatialize" the problem of *t-cra*. As an example, certain deep network architectures possessing specific types of *skip connections*, represent actually not so deep networks (Veit et al., 2016) applied in the time domain eliminating the need to pass errors far backwards in time.

The next question would be whether generic *rN* use *t-cra* which is more bio-plausible than *bp*TT? Authors in Werbos (1974) propose an algorithm approximating *bpTT* obtained by predicting the backward-through-time *G*- signal (costate) in the same way as the prediction of value functions in *rL* (Sussillo et al., 2009; Si, 2004). All this is still only starting point in comprehending how neural activity itself can represent the time variable (Finnerty et al., 2015), and how *rN* can learn evolutions of nodes' activity over time (Liu et al., n.d.). Several *corM* suggests ways, different from *bp*TT, for networks training on sequential prediction assignments (Cui et al., 2015; O'Reilly et al., 2014b). A number of different methods are available to mimic *bp*TT more realistically.

Spiking networks (SN): It is not straightforward to apply *G*-descent algorithm to *SN*. Several optimization algorithms are available to create *SN* which can execute complex assignments, by optimizing an *ooF* of the *nD* and incorporating varying parameters into high dimensional (*hD*) spaces with many *sN* representing each parameter (Abbott et al., 2016a). The *bp* can be replaced in the direct training of spiking *rN* by using recurrent connections with multiple timescales (Bourdoukan et al., 2015).

Despite the existing work reviewed above better understanding of the relations between the temporal dynamics of bio-realistic networks and methods for time/space credit assignment *(cr-ass)* is still needed. Even so, authors in Marblestone et al. (2016b) suggest that existing work already supports the argument that bio- plausible *NN* are solutions to these problems. In other words, SN can optimize complex functions of temporal history using bio-realistic neurons and realistic population coding.

The next question is whether the brain can learn differently? The answer is yes, the brain does have mechanisms and structures supporting $\mathscr{L}$-algorithms different from typical *G*-based optimization methods. *How exactly*? *It may exploit bio-neural methods:* The physiology of bio-neurons (*bN*) explains how *G*-descent could be used within the brain and enable learning different from *bp*. This indicates that the brain may be using methods of *cr-ass* quite different from those used by ML.

Dendritic computation (dC) is an example of such mechanisms, impacting $\mathscr{L}$-algorithms in several ways:

1) The dendrites of each neuron are implemented like a three-layer NN (Mel, 1992). 2) Propagation back, from the soma into the dendritic tree, of action potential generated by neuron's spikes, penetrates more intense the parts of the dendritic tree that have been active (Williams et al., n.d.), which simplifies the problem of *cr-ass* (Körding et al., n.d.a). 3) Neurons possess multiple, partly independent dendritic and somatic sections, suggesting that the neuron is storing more than one variable.

So, it could store both its activation itself, and the error derivative of an *ooF*, needed for *bp,* and such bio-realization of *bp* have been proposed in Körding et al. (n.d.b). In summary, more research is needed for a full understanding of the implications in *dC* in credit assignment in deep networks.

Beyond *dC*, diverse mechanisms (Marblestone et al., n.d.) like retrograde (^{+}ps to ^{-}ps) signals using cannabinoids (Wilson and Nicoll, n.d., Arancio et al., n.d.), go beyond *bp*. It was proposed in Harris (2008) how slow, retro axonal transport of molecules could enable NN to provide data to downstream neurons that are trained via faster error signals.

Another bio-mechanism is neuromodulation (*nM*). The same neuron can exhibit different *sP* as a function of a concentrations of various *nM* like opioids (Bargmann et al., 2013; Perea et al., 2009). The *nM* could have many roles in learning.

1) *nM* can gate *sP* on/off selectively in different space/time points, enabling orchestration of space/time *ooF's* application. There are suggestions that a neural circuit can be seen as a group of circuits with *nM* switching between them (Bargmann et al., 2013) potentially enabling sharing of *sy* weight data. Authors in Dayan (n.d.) discuss further possibilities *nM* could add for creating algorithms for optimization.

Learning in the cortical sheet (corS): Several models use specifics of the 6-layered *corS* to explain *cort*-learning (*corL*). The cortex uses uL via prediction (O'Reilly et al., 2014b). Some *corL* models map *cort*-structure onto the message-passing algorithms for Bayesian estimation (George et al., 2009b).

One-Shot Learning: Human learning (*hL*) can remember even a single stimulus and leverage it in new examples. $\mathscr{L}$-invariant modeling for object recognition (Anselmi et al., n.d.; Serre et al., 2007) is one example. It can be shown that objects can be represented invariantly and discriminatively using a single sample, even of a new class (Anselmi et al., n d.).

Active Learning (aL): hL is most often active, conscious, and intentional act chosen to generate training examples of interest, and for testing specific hypotheses. Such ideas of *aL,* initiated by Piaget, have been revisited more recently in Gopnik et al. (2000). The most effective $\mathscr{L}$ should use maximally informative samples.

Obviously, it would help here to have explicit representations of the sample uncertainty available and use it to instruct the system how to reduce it most quickly. The use of population coding algorithms could enable explicit probabilistic computations (Ma et al., 2006). At this point it is not fully understood how much and in what segment the brain uses an explicitly probabilistic (*prob*) framework (Emin et al., 2016).

Classical *G*-descent algorithm does not consciously and intentionally choose data in such a way to reduce its uncertainty. On the other hand, stochastic *G*-descent algorithm can be used to support a system that samples adaptively (Bouchard et al., 2015).

4. Quantum solutions

Implementation of the algorithms discussed so far by using Quantum Computing (QC) is anticipated to be more effective in the sense that complexity should be lower and the speed of the algorithms' execution should be higher. For the basics of QC see Glisic and Lorenzo (2022).

NOTE: ***Qubit:*** Similar to the *bit*, used in *c- c^{ompt}, q- c^{ompt}* is conceived upon a similar concept, the *q- bit*, called *qubit (qb)*. Just as a classical bit has a *state* - either 0 or 1- a qubit also has a state. For example a *qb* can be

Table 2
CrossTechology coverage (focus-Spiking NN).

1	2	3	4	5	6	7	8	9	10
(http, n.d.a, Bliss & Lomo, 1973, Artola et al., 1990, Lisman, 1989, Markram & Sakmann, 1995, Morrison et al., 2008, Cooper et al., 2004, Markram et al., 1998, Frey, 1997, Turrigiano & Nelson, 2004, Sutton & Barto, 1998b, Schultz & Montague, 1997, Lisman, 1989, Lisman & Zhabotinsky, 2001, Shouval et al., 2002, Florian, 2007, Graupner, 2007, Zou, 2007, Badoual et al., 2006)	SNN	√					*Details on spiking neuron modelling and analysis*	√	
(Badoual et al., 2006, McCulloch & Pitts, 1943, Sasaki & Carlini, 2002, Rumelhart et al., 1986, Hebb, 1949, Bohte et al., 2002, Huh & Sejnowski, 2017, Valiant, 1984, Mehta et al., 2002, Cybenko, 1988, Poggio & Girosi, 1989, Valiant, 1984, Vapnik, 1998, Thorpe et al., 1996, Olshausen, 1996, Recce, 1999, Thorpe et al., 2001)	SNN	√					*Details on spiking neuron networks modelling and analysis*	√	
(Hu, 2021)	SNN	√					*Details on Neuroscience and AI networks modelling and analysis*	√	
(Glisic, 2016, Arenas et al., 2008b, Majdandzic et al., 2013, Bullmore & Sporns, 2009, Marblestone et al, 2016a, Dayan & Abbott, 2001, Hodgkin & Huxley, 1952, Deneve, 2017, Clopath, 2010, Curto, 2019, Whiteway, 2019, Bassett et al., 2017)	SNN	√		√			*Modelling and analysis of* SNN *as Complex Network*	√	
(Cassandras, 2014, Goebel, 2009, Arenas et al., 2008a, Guan & Chen, 1999, Branicky et al., 1998, David, 2006, Deneve, 2017, Breakspear, 2017, Fultz et al., 2019, Connors, 2007, Gerstner, 2014, Rabinovich, 2006, Hu, 2019)	SNN	√		√			*Hybrid Models: Multiscale Spatiotemporal Dynamics*	√	
(Werbo, 2009, LeCun, 2015, Chen et al., 2019, Poter, 2019, Ekman, 2012, Goodfellow et al., 2014, Avena-Koenigsberger et al., 2018, Dolk, 2017, Abbott et al., 2016b, Frady & Sommer, 2019)	SNN	√					*Brain-Inspired Intelligence*	√	
(Glisic, 2023a, 2023b, Johansson et al., 2013, LeCun et al., Sermanet et al., 2013, Andrew et al., 2013, Goodfellow et al., 2014b, Gülcehre et al., 2016, Goodfellow, 2014a, Romero et al., 2014, Hochreiter et al., n.d.; Hinton et al., 2011, Weston et al., 2014, Graves et al., 2014, Kurach et al., 2015, Neelakantan et al., 2015, Körding, 2007, Taylor et al., Mordatch et al., 2012, Wang, Hasselmo, Sherman, Sejnowski, Fodor et al., 2002, Ullman et al., Bengio et al., 2009, Werbos, 1974, George et al., 2009a, Kappel et al., n.d.; Baldi et al., 2015)	SNN	√					*Deep learning and neuroscience*	√	
(O'Reilly, n.d.; Xie et al., n.d.; Lillicrap et al., 2014, Liao et al., 2015, Bengio et al., 2015a, Körding et al., n.d.a; Balduzzi et al., 2014, Werbos, 1990, Ollivier et al., n.d.; Gershman et al., 2012, Weston et al., 2014, Veit et al., 2016, Sussillo et al., 2009, Si, 2004, Finnerty et al., 2015, Liu et al., n.d.a; Cui et al., 2015, O'Reilly et al., 2014b)	SNN	√					*Biologically plausible approximations of gradient descent*	√	
(Abbott et al., 2016a, Bourdoukan et al., 2015, Mel, 1992, Williams et al., n.d.; Körding et al., n.d.a; Körding et al., n.d.b; Marblestone et al., n.d.; Wilson & Nicoll, n.d.; Arancio et al., n. d.; Harris, 2008, Bargmann et al., 2013, Perea et al., 2009, Dayan, n.d.)	SNN	√					*Spiking networks*	√	
(O'Reilly et al., 2014b, George et al., 2009b)	SNN	√					*Learning in the cortical sheet*	√	
(Anselmi et al., n.d.; Serre et al., 2007)	SNN	√					*One-Shot Learning*	√	
(Gopnik et al., 2000, Ma et al., 2006, Emin et al., 2016, Bouchard et al., 2015)	SNN	√					*Active Learning*	√	

1 reference, 2 focus, 3 classic, 4 quantum, 5 complex networks, 6 tensors, 7 q-simulations, 8 contribution, 9 energy efficiency, 10 computational efficiency.

in $\mathscr{S}^{tat}$'s $|0\rangle$ and $|1\rangle$, corresponding to the $\mathscr{S}^{tat}$'s 0 and 1 for a *c*- bit. Here we use so called the *Dirac notation* '$|\ \rangle$', adopted from *q*- mechanics. A *qb* can be in a $\mathscr{S}^{tat}$ *other* than $|0\rangle$ or $|1\rangle$. A l^{ine} *combinations* of $\mathscr{S}^{tat}$'s, referred to as *superpositions* (s^{pos}'*s*): $|q\rangle = \alpha|0\rangle + \beta|1\rangle$ is also used.

Assuming only r^{al} - v^{lu} -a^{mplt}'s for a q- $\mathscr{S}^{tat}$ $\alpha, \beta \in \mathbb{R}$, the resultant 2-D graphical model of a *qb*'s -$\mathscr{S}^{tat}$ is shown in Fig. i1.

Multi- qb -q- r^{gst} *'s*: In a two- *qb* -r^{gst}, there are four legitimate $\mathscr{S}^{tat}$'s that the composite *q*- s^{yst} can be superimposed in. If the $\mathscr{S}^{tat}$*'s* are $|q_1\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|q_2\rangle = \gamma|0\rangle + \delta|1\rangle$, the $\mathscr{S}^{tat}$ of the s^{yst} is

$$|q\rangle = |q_1\rangle \otimes |q_2\rangle = |q_1 q_2\rangle =$$

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle)$$

$$= \alpha{\cdot}\gamma|00\rangle + \alpha{\cdot}\delta|01\rangle + \beta{\cdot}\gamma|10\rangle + \beta{\cdot}\delta|11\rangle$$

where $\otimes$ is the $\mathscr{T}^{nsor}$ product o^{prtr} and the s^{yst}'s $\mathscr{S}^{tat}$ - $\mathscr{V}$ entries are the a^{mplt}'s of the four *q*- $\mathscr{S}^{tat}$'s $|00\rangle$,$|01\rangle$, $|10\rangle$ and $|11\rangle$. In *G*, in an *n*- *qb* -r^{gst}, the $\mathscr{S}^{tat}$ - $\mathscr{V}$ will include 2^n entries, each corresponding to the a^{mplt} of the r^{spct}'ive o^{rth} -$\mathscr{S}^{tat}$.

4.1. Artificial quantum neuron (qN)

The starting point in studying ML algorithms and AI protocols in artificial neural networks is modeling of artificial neurons (*aN*) (Tacchino, 2018). The simplest implementation of an *aN* is Rosenblatt's *perceptron* (P^{er}), although its realization is not straightforward due to complexity, especially of interest for the training of multilayered ($\mathscr{ML}$) P^{er}-networks. A q-computer version of a P^{er}, showing exponential savings in computational components over classical (c-) options is presented first. Test of this model on a q-processor, is presented in Tacchino (2018) showing good agreement with expected theoretical results. This q- model of a P^{er} can be used as an initial step towards training of artificial q-NN, discussed in the next section, to be practically realized on near-term q-processing hardware.

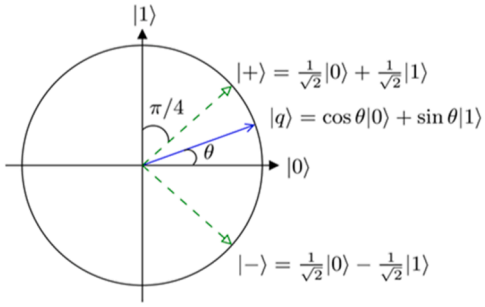


Fig. i1. The 2D graphical model of a qb, when the a^{mplt}'s of its q- S^{tat}'s are r^{al} - v^{lu}.

In practice, ANN are usualy run as classical (*c*-) algorithms on *c*-computers, although there is an interest in NN implemented on dedicated hardware (Merolla, 2014).

Prospective q-computers are well suited for implementing ANN. The capabilities of q- mechanics to store large complex valued vectors (*cvV*) and matrices and process such vectors by a variety of linear operations, provides an exponential savings either in memory storage or processing power for NN built on q-processors. A model of an *aN*, the so-called P^{er}, is outlined in Fig. 3 and redrawn in Fig. 5(a). Real valued *n* dimensional

input vector (*rvV*) **x**, is combined with a *rvV* of weight **w**. The P^{er} output y is a *yes/no* response to the input. In the simplest realization, **x** and **w** are binary valued vectors themselves (Schmidhuber, 2015; Zurada, 1992; McCulloch, 1943).

Perceptrons can perform a limited range of operations, although they are the basis of ML in more complex ANN in ($\mathscr{ML}$) P^{er}*'s* architectures. Their implementation is not feasible due to complexity, even though different solutions can be used to improve the efficiency of *c*-algorithms (Mocanu, 2018). So, several new solutions have been published recently to implement P^{er}*'s* on q-computers. In *qubit neuron* concept, each qubit operates as a separate neuron in the network and nonlinearity of the measurement ($\mathscr{M}$) implements the threshold function (Schuld, 2014).

Example 3. A solution, for mimicking a Rosenblatt perceptron on a q-computer, has been presented in Tacchino (2018). As a first step, an m -dimensional *c*-input is encoded on the q-hardware by using N qubits, where $m = 2^N$. Here (Tacchino, 2018) implements a new method to create multipartite entangled (*ent*-) states based on q-data principles by significantly reducing the q-computational resources needed. Reference (Tacchino, 2018) shows by experiment the efficacy of such a solution by realizing the algorithm on the IBM q-processor.

For illustration purposes here the scheme of the q-algorithm is presented in Fig. 5(b). The binary input and weight coefficients have the form $i_j, w_j \in \{-1, 1\}$.

By using N qubits, where $m = 2^N$, an input vector is encoded to define a general wavefunction (*wF*) $|\psi_i\rangle$. For an input $\left(\vec{i}\right) = (i_0, i_1, \ldots, i_{m-1})^T$ and weight $(\vec{w}) = (w_0, w_1, .., w_{m-1})^T$ vectors with i_j, $w_j \in \{-1, 1\}$, two q- states: $|\psi_i\rangle = \sum_{j=0}^{m-1} i_j|j\rangle/\sqrt{m}$ and $|\psi_w\rangle = \sum_{j=0}^{m-1} w_j|j\rangle/\sqrt{m}$. are defined.

The states $|j\rangle \in \{|00\ldots00\rangle, |00\ldots01\rangle, \ldots, 11\ldots11\rangle\}$ creating the basis in the Hilbert space of N qubits, are labeled with integers $j \in \{0, \ldots, m-1\}$ obtained by the decimal equivalent of the respective binary string. Now factors ±1 to encode the m-dimensional c- vectors into a uniformly weighted superposition of the full computational basis

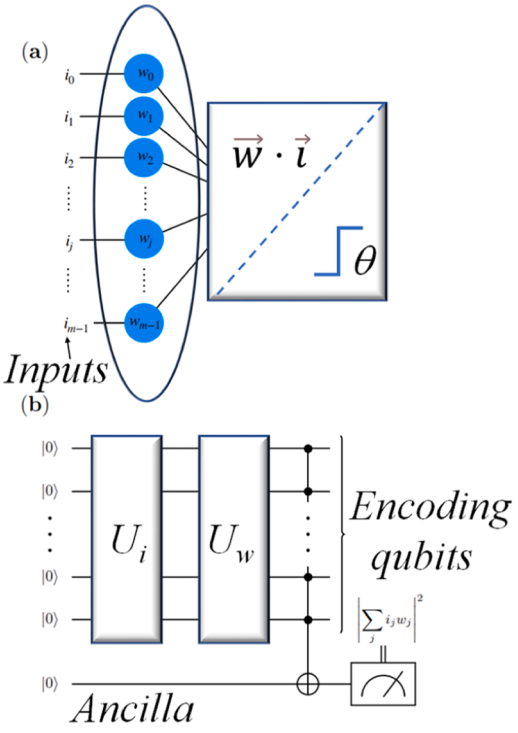


Fig. 5. P^{er} models. (a) c- P^{er} as a model of aN: (b) Scheme of the q-algorithm for the realization of an model on a q-processor.

are used.

Then, $\vec{w} \cdot \vec{i}$ using the q-$\mathscr{R}$. For this U-transformation, U_w is used, rotating q-state as $U_w|\psi_w\rangle = |1\rangle^{\otimes N} = |m-1\rangle$ is computed. Again, any $m \times m$ U-matrix having $\vec{w}^T$ in the last row satisfies this condition. If now U_w is applied overall N-qubits q- state becomes $U_w|\psi_i\rangle = \sum_{j=0}^{m-1} c_j|j\rangle \equiv |\varphi_{i,w}\rangle$. Using this $\langle\psi_w|\psi_i\rangle = \langle\psi_w|U_w^{\dagger}U_w|\psi_i\rangle = \langle m\text{-}1|\varphi_{i,w}\rangle = c_{m-1}$ is obtained and from the above definitions of $|\psi_i\rangle$ and $|\psi_w\rangle$ follows $\vec{w} \cdot \vec{i} = m\langle\psi_w|\psi_i\rangle$. Here the useful information is in the c_{m-1} of the final state $|\varphi_{i,w}\rangle$.

Authors in Tacchino (2018) use an $\mathscr{A}$-qubit initiated in the state $|0\rangle$ to extract such an information. A multi-controlled NOT gate between the encoding qubits and the target $\mathscr{A}$ gives: $|\varphi_{i,w}\rangle|0\rangle_a \rightarrow \sum_{j=0}^{m-2} c_j|j\rangle|0\rangle_a + c_{m-1}|m-1\rangle|1\rangle_a$ (Hale, 1993).

Performing a q-$\mathscr{M}$ produces immediately the nonlinearity at the output of the P^{er}. By measuring the state of the $\mathscr{A}$- qubit in the computational basis (c.$\mathscr{B}$) produces the output $|1\rangle_a$ (i.e., an activated P^{er}) with probability $|c_{m-1}|^2$. This choice produces the correct result effectively. In addition, here a refined threshold functions can be applied once the inner product data are stored on the $\mathscr{A}$ (Hu, 2018; Cao et al., 2017; Torrontegui, 2018).

4.2. Quantum NN

The *q*-analogue of a *c*-neuron, presented above, is a *q*-feed-forward NN enabling universal *q*-computation with nice generalization behavior. The scheme is robust to noisy training data and an initial stage for further studies. One can categorize *quantum ML* (qML) techniques into *c*-ML (Glisic, 2023b) improving *q*- assignments, *q*-algorithms speeding up *c*-ML and using *q*- computing devices for tasks with *q*-data. The *dissipative q-NN* (*dqNNs*) belonging to the latter category, comprises layers of qubits and can be trained with pairs of q-states. A training data pair (input state- targeted output), depends on the training objective.

The *dqNN* is built of q-P^{er}*'s* like those discussed in the previous section. This block, connecting two successive layers of qubits, is a completely positive (CP) transition map (*t-map*) (Arunachalam et al., 2017). Such a map tensors the state of the current layer to the state of the next layer's qubits including applying U-operations. In addition, *t-map* is responsible to trace out the qubits from the first of the two layers when forwarding input states through the *dqNN*. The resulting output state is then compared with the targeted output. For this the fidelity function (*fF*) of two q-states is used allowing conclusions about how the P^{er} U must be updated to further improve the training objective.

Example 4. Similarly, to a classical NN (Glisic, 2023b), the *dq*NN is built of *quantum perceptrons* (P^{er}) (see Fig. 6) acting on qubits arranged in layers. The P^{er}*'s* are designed as general U- operators. Such a P^{er} U acts on $m+n$ qubits and depends on $(2^{m+n})^2 - 1$ parameters, where m of the qubits is defined as input qubits and remaining n as output qubits.

Here the input and the output qubits need to be initialized in states ρ^{in} and $|0\ldots0\rangle$, respectively. After applying the P^{er} U, the m input qubits are traced out and leaving the n -qubit state ρ^{out}. Network representation of a dqNN is shown in Fig. 7. One single P^{er} can be considered as a small *dq*NN having only two layers of qubits and one U operation as shown in Fig. 6.

By choosing $n = 1$, (i.e. the P^{er}*'s* are $m+1$-qubit U's) as an example, the model will still remain universal. Although here 2-level qubits are used, the P^{er} defined in this way can be easily generalized for qudits.

As indicated earlier, the entire network can be considered as a collection of layer-to-layer *t-maps*. This notation will be used throughout the entire section. Based on the definition of a single P^{er}, the output of the *dqNN* becomes

$$\rho^{out} = \mathscr{E}(\rho^{in}) = \mathscr{E}^{L+1}\left(\mathscr{E}^{L}\left(\ldots\mathscr{E}^{2}\left(\mathscr{E}^{1}(\rho^{in})\right)\ldots\right)\right),$$

using the CP (Completely Positive) maps defined via

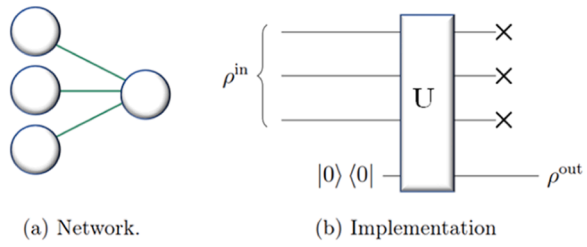


Fig. 6. Quantum P^{er}.

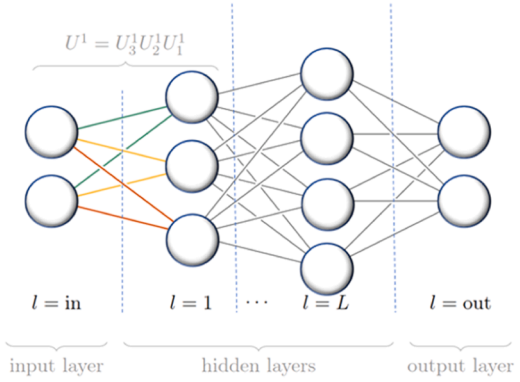


Fig. 7. Network representation of a dqNN.

$$\mathscr{E}^l(X^{l-1}) \equiv \text{tr}_{l-1}\left(\prod_{j=m_l}^{1} U_j^l (X^{l-1} \otimes |0\ldots0\rangle_l\langle 0\ldots0|) \prod_{j=1}^{m_l} U_{j_l}^{l\dagger}\right),$$

where U_j^l corresponds to the jth P^{er} operating on the qubit layers $l-1$ and l, and m_l is the total number of P^{er}s operating on these layers. With this notation, it is evident that the data propagates from the input to the output layer and a q-feed forward NN is obtained. This represents the basis for the *bp* algorithm. The q-circuits of the network can be treated as a single unitary $\mathscr{U} = U^{\text{out}} U^L U^{L-1} \ldots U^1$, where $U^l = U_{m_l}^l \ldots U_1^l$ are the layer U's, consisting of a product of q-P^{er}s operating on the qubits in layers $l-1$ and l, see Fig. 7. To get the correct dimensions when constructing the architecture, every U_k^l must be extended by identities for the remaining qubits. Here these are left off for simplicity. Having this in mind the formula for the output state becomes

$$\mathscr{E}(\rho^{\text{in}}) \equiv \text{tr}_{\text{in,hid}}\left(\mathscr{U}(\rho^{\text{in}} \otimes |0\ldots0\rangle_{\text{hid,out}}\langle 0\ldots0|)\,\mathscr{U}^\dagger\right).$$

For the choices of *ooF* $\mathscr{L}$ and the training of dqNN algorithm see Abbasi et al. (2013), Bouchard et al. (2015), Stanoev (2013). QNNs can be realized on a q-computer using *parametrized q-circuits* (Glisic, 2023b; Du, 2020; Bu et al., 2021) comprising parameterized q-gates. To do so, two aspects are important: the realization should be universal, and the complexity should be kept low. In (Beer et al., 2021), a proper compromise between these objectives enables good training results on NISQ devise. A discussion of over parameterization can be found at (Larocca, 2021). Implementation of P^{er} U's with two-qubit gates (Peterson, 2020) using a two-qubit canonical gate and twelve single qubit gates is presented in Beer (2022).

Performance Limits of QNN: To characterize these limits, here a lower bound on the probability of QNN giving an incorrect output for an arbitrary input is used. These limits are studied in Arunachalam et al. (2017), Sentís et al. (2012), Sasaki (2001), Gammelmark (2009).

Continuous-variable (CV) QNN: Most natural q-computing architecture (*c-arch*) is CV model. Q-data are encoded in the q-states of fields, like the electromagnetic field. The standard variables in the CV picture, e.g., position or momentum are continuous (Du, 2020). Qubit operations can be embedded into the q-field picture (Gottesman, 2001). Authors in Killoran (2019), use the CV model for ML, showing how a number of basic ML primitives can be built in the CV setting. A kernel-based classifier implemented as CV q-circuit was trained in Mitarai (2018).

Convolutional, Recurrent and Residual CV QNN: Nowadays, deep $\mathscr{L}$ (DL) methods are not limited only to the basic architecture. Diverse DL software tools (Bergstra et al., 2010; Jia, 2014; Maclaurin, 2015; Paszke, 2017; LeCun, 1989; Goodfellow, 2016; Bloch, 1929; Noether, 1918) enable us to study more sophisticated and complex topologies. For the q-case, an effort should made to also try to go beyond restrictions imposed by the basic network structure discussed so far.

In this respect, by using CV model the problems can be encoded in a variety of representations (Arunachalam et al., 2017) like the phase space, the wave-function and the Hilbert space picture, or some combination of these. The information can be encoded in coherent states (*cS*), squeezed states (*sS*), Fock states (*fS*), or superpositions of these states (σS).

In addition, the network can be tuned to match a specific class of problems by selecting the gates and their parameters with a specific structure. This should improve efficiency of parameters' use and better overall performance. In (Killoran, 2019; Giovannetti, 2008), authors discuss potential q-versions of various special NN architectures as visualized in Fig. 8.

4.3. Quantum Ml

Having roots in the traditional pattern recognition, such as identifying handwriting, and st-$\mathscr{L}$ theory (using analytical tools for ML modeling), ML studies the design of methods that can learn from data and make predictions about them. With respect to data analysis and data mining-type assignments, *c- ML* can be organized as supervised (*sL*) and unsupervised learning (*uL*) (Shalev-Shwartz, 2014). More generally, reference (Alpaydin, 2010) also includes here reinforcement learning (*rL*) (Sutton et al., 1998), which is closely related to $\mathscr{L}$ as is implemented by bio-intelligent objects. For *c-ML* algorithms see Glisic and Lorenzo (2022), Arunachalam et al. (2017).

ML in Quantum Physics: In recent times, ML is used as a buzzword, for a number of techniques like $\mathscr{L}$-algorithms, but also techniques used for indirectly related problems. From such a broad perspective, ML also includes st-$\mathscr{L}$, the black-box optimization problems and solving hard optimization problems in general.

***NOTE*: Position and momentum space:**

Position ($\mathscr{P}^{os}$) space (s^{pace}) is the collection of all *position vectors* ($p\,\mathscr{V}$) **r** defining points in s^{pace}. Similarly, the collection of all $\mathscr{M}^{om}$ *-vectors*

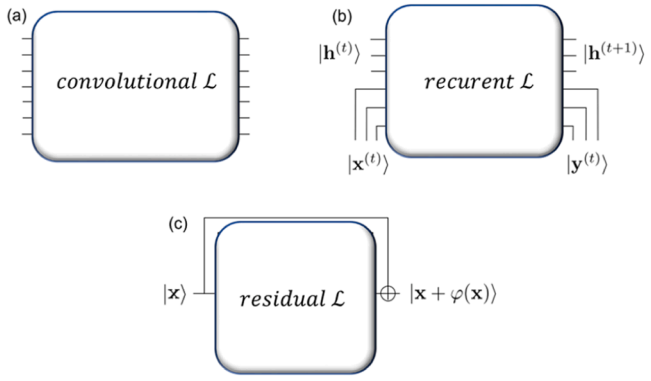


Fig. 8. Q-adaptations of the convolutional (conv), rL, and residual layer. The conv layer is enacted using a Gaussian U with translationally invariant Hamiltonian, resulting in a corresponding sy matrix that has a block Toeplitz structure. The r- layer combines an internal signal from previous layers with an external source, while the residual layer combines its input and output signals using a controlled-X gate.

(m $\mathscr{V}$'s) **p** a physical (p^{hys})-s^{yst} can have is referred to as *momentum* ($\mathscr{M}^{om}$)- s^{pace}. The *m* $\mathscr{V}$ of a particle (p^{icle}) describes its movement . In physics, $\mathscr{M}^{om}$ is usually defined as mass times velocity $\boldsymbol{p} = m\boldsymbol{v}$.

For a function (*F*) $f(\mathbf{r})$ defined in $\mathscr{P}^{os}$-s^{pace}, the Fourier transform ($\mathscr{FT}$) gives the *F*-$\phi(\mathbf{p})$in $\mathscr{M}^{om}$-s^{pace}, and a $\mathscr{P}^{os}$-s^{pace}-*F* is obtained by the inverse-$\mathscr{FT}$ of a $\mathscr{M}^{om}$-s^{pace}-*F (Pontryagin duality). So, a* p^{hys} - s^{yst} *can be represented using either the* $\mathscr{P}^{os}$*'s of its components, or their* $\mathscr{M}^{om}$ *'a, in both cases the information (*$\mathscr{I}$*) about the* s^{yst} *is same. The wave vector (w* $\mathscr{V}$*)* **k** *can be also defined having dimension of 1/length (note the similarities with angular frequency* ω *having dimensions of 1/time). The collection of all w* $\mathscr{V}$*'s is* **k**-s^{pace}.

$\mathscr{P}^{os}$ and $\mathscr{M}^{om}$-s^{pace}- s in classic mechanics:

Lagrangian ($\mathscr{D}$)- m^{ech}: The $\mathscr{L}$- $L(\boldsymbol{q},\ d\boldsymbol{q}/dt, \mathrm{t})$, in configuration space (c$\mathscr{S}$), with vector $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ of the generalized (g^{nrlz}) coordinates (g $\mathscr{C}$). The Euler-$\mathscr{L}$ eq's of motion are

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \frac{\partial L}{\partial q_i},\quad \dot{q}_i \equiv \frac{dq_i}{dt} \tag{i1}$$

By defining the canonical (c^{an}) $\mathscr{M}^{om}$ for each g $\mathscr{C}$,

$$p_i = \partial L/\partial \dot{q}_i$$

one has

$$\dot{p}_i = \partial L/\partial q_i$$

In $\mathscr{M}^{om}$ -s^{pace} $\mathscr{L}$, $L'(\mathbf{p},\ d\mathbf{p}/dt, \mathrm{t})$, $\mathbf{p} = (p_1,\ p_2, \ldots,\ p_n)$ is a vector of the g^{nrlz} -$\mathscr{M}^{om}$ 'a. One can show

$$\begin{aligned} &L' = L - \sum\nolimits_{i=1}^{n}(q_i\dot{p}_i + \dot{q}_i p_i),\\ &-\dot{q}_i = \frac{\partial L'}{\partial p_i},\quad -q_i = \frac{\partial L'}{\partial \dot{p}_i}, \end{aligned} \tag{i2}$$

Combining (1) and (2) gives the $\mathscr{M}^{om}$ -s^{pace} Euler-$\mathscr{L}$ equations

$$\frac{d}{dt}\frac{\partial L'}{\partial \dot{p}_i} = \frac{\partial L'}{\partial p_i} \tag{i3}$$

Hamiltonian ($\mathscr{H}$) mechanics: While $\mathscr{L}$-mechanics uses either $\mathscr{P}^{os}$'s or the $\mathscr{M}^{om}$ 'a, the $\mathscr{H}$ equations of motion place $\mathscr{P}^{os}$'s and $\mathscr{M}^{om}$ 'a together. For a s^{yst} with $\mathscr{H}$- $H(\mathbf{q},\ \mathbf{p},\ t)$, the equations are

$$\begin{aligned} &\partial H/\partial t = -\partial L/\partial t,\ \ \dot{q}_i = \partial H/\partial p_i\\ &\textit{and}\ \ \dot{p}_i = -\partial H/\partial q_i \end{aligned} \tag{i4}$$

A *q*- mechanical (m^{ech}) s^{yst} is governed by the time dependent Schrödinger equation, $\widehat{H}|\psi(t)\rangle = i\hbar\partial|\psi(t)\rangle/\partial t$, where $|\psi(t)\rangle$ is the s^{ta} of the *q*- s^{yst} at time *t*, $\hbar$ is the reduced Planck's constant $\hbar = 2\pi$, and $\widehat{H}$ is the *Hamiltonian* ($\mathscr{H}$) that describes the total energy of the s^{yst}. The "hat" is used to indicate that *H* is a *q*- o^{per} 'or. As the Schrödinger equation is a first-order linear (l^{ine}) differential equation, the temporal dynamics of the *q*- s^{yst} may be viewed as a straightforward example of a l^{ine} dynamical s^{yst} with formal solution, $|\psi(t)\rangle = e^{-i\widehat{H}t/\hbar}|\psi(0)\rangle$.The time- I^{dep} -$\mathscr{H}$- $\widehat{H}$ governs the time evolution of the s^{yst} through the o^{per} 'or $e^{-i\widehat{H}t/\hbar}$. Thus, just as with classical (*c*-) s^{yst}'s, determining the $\mathscr{H}$ of a s^{yst} - whether the *c*-$\mathscr{H}$- *H* or its *q*- counterpart $\widehat{H}-$ is the first step to deriving its dynamical behavior.

$\mathscr{P}^{os}$ and $\mathscr{M}^{om}$-s^{pace} 's in q- mechanics

In q- m^{ech}, a p^{icle} is represented ($\mathscr{R}^{epr}$) by a q- s^{ta} that can be described as a superposition of basis s^{ta} 's. If the eigenfunctions (*eF*'s) of the $\mathscr{P}^{os}$ -operator (o^{per}) are used as a set of b^{as} -*F* 's (*bF*), then a s^{ta} as a wave *F* (*wF*) $\psi(\mathbf{r})$ in $\mathscr{P}^{os}$ - s^{pace} is defined. The Schrödinger ($S^{chrö}$) eq in terms of the $\mathscr{P}^{os}$- **r** is an example of q- m^{ech} in the $\mathscr{P}^{os}$-$\mathscr{R}^{epr}$. Using the *eF* of a different o^{per} as a set of *bF* 's gives different $\mathscr{R}^{epr}$'s of the same s^{ta}. If the *eF* -s of the $\mathscr{M}^{om}$ - o^{per} are used, the resulting *wF* -$\varphi(\mathbf{k})$ is in $\mathscr{M}^{om}$ -s^{pace}. A library of Hamiltonians for benchmarking q- algorithms and hardware can be found in Sawaya et al. (2023).

Estimating Hamiltonian: Hamiltonian (H) estimation, observes a q-system operated by Hamiltonian, unknown within a given family $H(\boldsymbol{\theta})$, with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$. H- estimation identifies the best algorithm for estimating the H- parameters.

This includes selecting the best option for initial state (*iS*), to be processed by operator H, and the selection of the subsequent $\mathscr{M}$'s, which reveal the effect the operator H had, and so, indirectly, the parameter values. This study incorporates a number of limitations, modifications, and generalizations of this assignment. As an example, a situation may be considered where either evolution time *t* of operator H is controlled, or it is fixed so that $t = t_0$, options referred to as frequency and phase estimation respectively.

The quality of the estimation is expressed in different ways. In a frequency estimation method, the focus is on estimation strategies giving the best precision / number of $\mathscr{M}$'s scaling. Here so-called quantum Fisher information is used, which quantifies and bounds the scaling. In this setting, referred to as the local regime, typically having many repetitions of $\mathscr{M}$'s is assumed. On the other hand, the main objects in the single-shot regime (Bayesian) (Jarzyna, 2015) are the prior data, defined by the parameter to be estimated distribution, and its update to the posterior (*p*-) distribution given a $\mathscr{M}$ methods and outcome. Here the aim is to identify the initiation/$\mathscr{M}$ methods which minimize the average variance of the *p*- distribution, computed here via Bayes' theorem.

The interest here is the utilization of specific q-features, such as entanglement, squeezing etc. in the structure of the probe states and $\mathscr{M}$'s may result in a provably better estimation than by so-called c-strategies for many natural estimation problems. Such q-improvements are of important practical relevance (Giovannetti, 2011). Finding the optimal solutions has been achieved in certain *clean* theoretical scenarios, although often impractical. It is in this context that ML-flavored mechanisms, and ML methods can help.

phε settings: A relevant estimation problem from a ML point of view, is already simple example of a phase shift in an optical interferometer, where one of the arms has a phase shift of θ. It is known from earlier discussions, that for an optimal probe state, with mean photon number N, and an optimal (so-called *canonical*) $\mathscr{M}$, the asymptotic phase uncertainty can decay as N^{-1} (Sanders, 1995), known as the Heisenberg limit.

On the other hand, if limited to *simple* $\mathscr{M}$ *strategies* (as defined in Sanders (1995)), involving only photon number $\mathscr{M}$ a scaling of $\sqrt{N^{-1}}$ achieved, called the standard q-limit. Authors in Berry et al. (n.d.) define more general proof: *the optimal measurement* ($\mathscr{M}^{opt}$) *constitutes a complex, experimentally unfeasible Positive-Operator Valued Measure (POVM) and cannot be achieved by the c-post-processing of photon number* $\mathscr{M}$ *of the output arms*. Reference (Berry et al., n.d.) also shows how to overcome this by using *simple* $\mathscr{M}$, provided they can be altered in run-time.

The $\mathscr{M}^{opt}$ *process is an adaptive strategy*: In the process an *ent*-N -photon state is initiated (Berry et al., 2001), the photons are sequentially inserted into the interferometer, and photon numbers are measured. A differing phase shift φ modifies the $\mathscr{M}$ at each step, depending on previous $\mathscr{M}$ outcomes. In (Berry et al., n.d.; Berry et al., 2001), a method achieving the Heisenberg scaling of the optimal order $O(1/N)$ was given. The implementation of these methods was first suggested in Hentschel (2011), and later in Sergeevich et al. (n.d.). In follow up works, differential evolution has been shown to be better and more practical (Lovett et al., n.d.).

Group-theoretic approach to qML: qML models are designed for learning by using data encoded in q-states. To facilitate training, some assumptions about the problem embedded in the model are necessary. So, encoding that includes as much data as possible about the problem at hand is needed. Authors in Larocca (2022), present *group-invariant* (*gI*) models whose decisions is invariant under the action of any element of the *symmetry group* (*sG*) associated to the set of data. Their results support the construction of invariant models and present several *qML* examples, including cases using Lie group and a discrete *sG*. The presented

framework enables efficiently recovering of several well-known algorithms and discovering new ones. In summary, the results encourage further research on geometric and group-theoretic methods for qML model design.

Discrete gI Models: The previous options dealt with models where the *sG* modeling of dataset was based on a U-representation of Lie group. This approach can be used also for representations with discrete groups (*dG*) (http, n.d.b; Frigerio, 2016). As an example, *dG* are the appropriate analytical structures when the q-data is invariant under permutations. This includes cases of structural invariances in states of molecular systems.

Graph isomorphism dataset: Here, a dataset in the context of graph isomorphism problem (Izquierdo, 2020; http, n.d.c) is discussed, with objective to find out if two graphs are isomorphic. This classification problem (Kobler, 2012) is NP hard. Several c-algorithms (with quasi polynomial complexity in the graph size (Babai, 2016)), and q-heuristics (Gaitan, 2014; Izquierdo, 2020) are available to solve this problem. If a q-model is used for graph classification, the first step is to encode graphs onto q-states. For the discussion on q-computing for the brain see Swan et al. (2022) and quantum-like modeling in biology with open quantum systems and instruments (Basieva et al., 2021).

For discussion on parameterized q- circuit approximation, optimizing parameterized q- circuits and variational quantum eigensolver see Ibrahim et al. (2023), Watanabe et al. (2023), Bertels et al. (2023).

5. Complex *q*- models

In the evolution of communication networks from 6G to 7G further growth should be expected of our ambitions to model more sophisticated optimization processes requiring more powerful analytical tools. For these reasons, it is useful to review the work done in the field of quantum computational chemistry where some advances in building complex quantum models have been already achieved.

5.1. *q-Computational chemistry (cCh)*

Q-computing is being used more and more for solving *c*-intractable chemistry problems. This should enable us to solve otherwise unresolved problems related to chemistry specific phenomena.

Among these phenomena this paper is interested in biochemical reactions and their relevancy to brain operation and new advanced solutions to ML.

Since having large QC resources may take time, solutions enabling analysis of these problems with fewer q-resources are very important. Motivated by this need, *q-cCh* is an evolving field requiring knowledge of both QC and *cCh*. This section presents a survey of both *cCh* and *q*-computing works, reducing the existing knowledge gap. Here the progress in this field is discussed.

As a part of integrative research, full understanding of the problems and solutions in this field is expected to help in generating new models and algorithms in the field of *n-Sci* and ANN which are expected to be massively used in 6G/7G networks.

NOTE: ***Functions and o^{per}'s in $\mathscr{P}^{os}$ -space***

For a three-dimensional ($\mathscr{D}^m=\mathscr{D}^3$) *wF* in $\mathscr{P}^{os}$ - s^{pace} -$\psi(\mathbf{r})$, these *F's* can be represented as a weighted sum of orthogonal *bF* 's $\psi_j(\mathbf{r})$: or,

$$\psi(\mathbf{r}) = \int_{\mathbf{k}-space} \varphi(\mathbf{k})\psi_k(\mathbf{r})d^3\mathbf{k} \tag{i5}$$

If the set of *F*'s- $\psi_k(\mathbf{r})$, is the set of *eF* 's of the $\mathscr{M}^{om}$-o^{per}-*s*, the *F*- $\varphi(\mathbf{k})$ contains all the $\mathscr{J}$ needed to regenerate $\psi(\mathbf{r})$ and so is yet another option to describe the s^{tate} -ψ. In q- m^{ech}, the $\mathscr{M}^{om}$-o^{per} is represented by

$$\widehat{\mathbf{p}} = -i\hbar\partial/\partial\mathbf{r} \tag{i6}$$

with appropriate domain. The *eF* 's are

$$\psi_{\mathbf{k}}(\mathbf{r}) = exp(i\mathbf{kr}) \Big/ \left(\sqrt{2\pi}\right)^3 \tag{i7}$$

and eigenvalues (*eV*) $\hbar\mathbf{k}$. So

$$\psi(\mathbf{r}) = \frac{1}{\left(\sqrt{2\pi}\right)^3} \int_{\mathbf{k}-space} \varphi(\mathbf{k})exp(i\mathbf{kr})d^3\mathbf{k} \tag{i8}$$

so, the $\mathscr{M}^{om}$-$\mathscr{R}^{epr}$ is related to the $\mathscr{P}^{os}$ -$\mathscr{R}^{epr}$ by a $\mathscr{FT}$.

$$\varphi(\mathbf{k}) = \frac{1}{\left(\sqrt{2\pi}\right)^3} \int_{\mathbf{k}-space} \psi(\mathbf{r})exp(-i\mathbf{kr})d^3\mathbf{r} \tag{i9}$$

Momentum Operator: Let us now go back to Eq. (*i6*). In q-mechanics, *the* $\mathscr{M}^{om}$ -o^{per} is, in the $\mathscr{P}^{os}$ -$\mathscr{R}^{epr}$, a differential operator. For the case of one p^{icle} in spatial $\mathscr{D}^1$ ($\mathbf{r}$= *x*), the definition is $\widehat{\mathbf{p}} = -i\hbar\partial/\partial x$ with ħ being Planck's constant, $i = \sqrt{-1}$, *x* represents spatial coordinate, and a p^{rti} derivative (denoted by $\partial/\partial x$) is used instead of a total derivative (d/dx) since the *wF* is also a *F* of time. The 'hat' indicates an o^{per}. The a^{pp} of the o^{per} on a *wF* is as $\widehat{\mathbf{p}}\psi = -i\hbar\partial\psi/\partial x$. In a b^{as} of Hilbert s^{pace} -($\mathscr{HS}$)

Table 3
CrossTechology coverage (focus q-NN).

1	2	3	4	5	6	7	8	9	10	11
(Glisic & Lorenzo, 2022)	QC		√			√	Fundamentals of Q-computing (QC)		√	
(Tacchino, 2018, Schmidhuber, 2015, Zurada, 1992, Merolla, 2014, Biamonte et al., 2017b, Neukart, 2013, Schuld, 2014, Schuld, 2015, Kapoor, 2016, Lloyd, 2013, Schuld, 2017, Lamata, 2017, Alvarez-Rodriguez et al., 2017, Otterbach, 2017, Rebentrost, 2018, Rosenblatt, 1957, McCulloch, 1943, Mocanu, 2018, Rossi, 2013, Nielsen, 2004, Hu, 2018, Cao et al., 2017, Torrontegui, 2018)	QC		√			√	*Artificial Quantum Neuron*		√	
(Torrontegui, 2018, Du, 2020, Arunachalam, 2017, Beer, 2022)	QC		√			√	*Quantum Neural Networks*		√	
(Arunachalam, 2017, Sentís et al., 2012, Sasaki, 2001, Gammelmark, 2009, Du, 2020, Killoran, 2019, Gottesman, 2001, Mitarai, 2018)	QC		√			√	*Performance Limits of QNN and CV QNN*		√	
(Bergstra, 2010, Jia, 2014, Maclaurin, 2015, Paszke, 2017, LeCun, 1989, Goodfellow, 2016, Bloch, 1929, Noether, 1918)	QC		√			√	*Convolutional, Recurrent and Residual CV QNN*		√	
(Shalev-Shwartz, 2014, Alpaydin, 2010, Sutton et al., 1998, Arunachalam, 2017)	QC		√			√	*Quantum Machine Learning*		√	
(Jarzyna, 2015, Giovannetti, 2011)	QC		√			√	*Estimating Hamiltonian*		√	
(Larocca, 2022)	QC		√			√	*Group-theoretic approach to QML*		√	
(Larocca, 2022, http, n.d.b, Frigerio, 2016, Kobler, 2012, Babai, 2016, Gaitan, 2014, Izquierdo, 2020)	QC		√			√	*Discrete Group-Invariant Models:*		√	
This paper	NET	√	√	√	√	√	Cross-Technology survey	√	√	√

1 reference, 2 focus, 3 classic, 4 quantum, 5 complex networks, 6 tensors, 7 q-simulations, 8 contribution, 9 energy efficiency, 10 computational efficiency, 11 synchronization.

consisting of $\mathscr{M}^{om}$-eigenstates (*eSt*) expressed in the $\mathscr{M}^{om}$-$\mathscr{R}^{epr}$, the action of the o^{per} is multiplication (m^{tipl}) by p, i.e. it is a m^{tipl} -o^{per}, just as the $\mathscr{P}^{os}$-o^{per} is a m^{tipl} -o^{per} in the $\mathscr{P}^{os}$-$\mathscr{R}^{epr}$.

Hamiltonian Operator: In q- mechanics, the $\mathscr{H}$ of a system (s^{yst}) is an o^{per} describing to the total energy (e^{nrgy}) of that s^{yst}, consisting of kinetic (k^{tic})- e^{nrgy} and potential (p^{tent})- e^{nrgy}. Its s^{ctr}, or its set of e^{nrgy} -*eV* 's is the collection of potential outcomes of a measurement ($\mathscr{M}$) of the s^{yst}'s total- e^{nrgy}. The $\mathscr{H}$ of a s^{yst} is the sum of the k^{tic} - e^{nrgy} of all the p^{icle} 's, plus the p^{tent} - e^{nrgy} of the p^{icle} 's associated with the s^{yst}.

$S^{chrö}$ -$\mathscr{H}$: For *one* p^{icle}, like in c- m^{ech}, the $\mathscr{H}$ is described as the sum of o^{per} 's representing the k^{tic} and p^{tent} -e^{nrgy} 's of a s^{yst} in the form

$$\widehat{H} = \widehat{T} + \widehat{V}$$

where

$$\widehat{V} = V = V(\mathbf{r}, t)$$

is the p^{tent} -e^{nrgy} - o^{per} and

$$\widehat{T} = \frac{\widehat{\mathbf{p}}\cdot\widehat{\mathbf{p}}}{2m} = \frac{\widehat{p}^2}{2m} = -\frac{\hbar^2}{2m}\nabla^2 \tag{i10}$$

is the k^{tic} -e^{nrgy} - o^{per} with m being the mass of the p^{icle}, the $(\cdot)$ defines the dot product of $\mathscr{V}$'s, and

$$\widehat{p} = -i\hbar\nabla$$

is the $\mathscr{M}^{om}$ - o^{per} with a ∇ being the del - o^{per}. The dot product of ∇ with itself is the Laplacian ∇^2. In $\mathscr{S}^3$ using Cartesian coordinates the Laplace - o^{per} is $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2$.

This is the form $\mathscr{H}$ most commonly takes, although this is not the its technical definition in c- m^{ech}. Putting all of these together gives familiar form used in the $S^{chrö}$ -eq:

$$\widehat{H} = \widehat{T} + \widehat{V} = \frac{\widehat{\mathbf{p}}\cdot\widehat{\mathbf{p}}}{2m} + V(\mathbf{r}, t) = -\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r}, t) \tag{i11}$$

allowing us to apply the $\mathscr{H}$ to s^{yst}'s described by a *wF*-$\Psi(\mathbf{r},\ t)$. This is the approach used in introductory discussions of q- m^{ech}, using the formalism of $S^{chrö}$ -wave m^{ech}.

The electronic structure problem: For specifying the energy of the molecule's components, atomic, molecular, and optical physics and q-cCh use molecular H operator. The analytical representation of multiple observables (H's specifying observable quantities) is established by using a set of precise rules (http, n.d.d): a) Use the c-representation of the observable in H form (as a function of momenta **p** and positions **q**) (http, n.d.d; Christiansen, 2012). Both **p** and **q** are represented within a space fixed frame. b) Substitute **p** by $-i\hbar\nabla$ and use **q** as a multiplicative operator.

Here ∇ is a vector of first derivatives as its components. It follows from the differentiation rules that **p** and **q** operators commute.

Kinetic energy (*kE*) in a c-model of the electrons and nuclei in a molecule, has the form $p^2/(2\ m)$ (Glisic, 2023a). They interact via Coulomb (*Co*-) interactions, inversely proportional to the distance r_{ij} between particle i and j.

$$r_{ij} = |\boldsymbol{r}_i\boldsymbol{r}_j| = \sqrt{(\mathbf{r}_i - \mathbf{r}_j)(\boldsymbol{r}_i - \mathbf{r}_j)^T} = \sqrt{(x_i - x_j)^2 + \left(y_i - y_j\right)^2 + (z_i - z_j)^2} \tag{1}$$

Here $\mathbf{r}_i$ is the general coordinate vector of either electron or nucleus. To make the equations self-explanatory, here **R** is used for the nuclear, and **r** for the electrons coordinates of the system. By using quantized c-energy in H-form, the molecular H-operator called the Co-H consisting of five terms is obtained:

1. For each nucleus in the system the *kE* operator is represented as; $\widehat{T}_n = -\sum_i \mathrm{h}^2 {\nabla_{\mathbf{R}_i}}^2/2M_i$
2. The same operator for each electron are given by; $\widehat{T}_e = -\Sigma_i \hbar^2 \nabla^2_{\mathbf{r}_i}/2m_e$
3. The potential energy (*pE*) between the electrons and nuclei is;

$$\widehat{U}_{en} = \sum_i \sum_j \mathrm{Z}_i e^2 \,/\, 4\pi\varepsilon_0 |\mathbf{R}_i - \mathbf{r}_j| \tag{2}$$

4. The *pE* due to Co- electron-electron repulsions

$$\widehat{U}_{ee} = \frac{1}{2}\sum_i \sum_{j\neq i} \frac{e^2}{4\pi\varepsilon_0|\mathbf{r}_i - \mathbf{r}_j|} = \frac{1}{2}\sum_i \sum_{j>i} \frac{e^2}{4\pi\varepsilon_0|\mathbf{r}_i - \mathbf{r}_j|} \tag{3}$$

5. The *pE* due to Co-nuclei-nuclei repulsions is

$$\widehat{U}_{nn} = \frac{1}{2}\sum_i \sum_{j\neq i} \frac{Z_i Z_j e^2}{4\pi\varepsilon_0|\mathbf{R}_i - \mathbf{R}_j|} = \frac{1}{2}\sum_i \sum_{j>i} \frac{Z_i Z_j e^2}{4\pi\varepsilon_0|\mathbf{R}_i - \mathbf{R}_j|} \tag{4}$$

Here M_i stands for the mass of nucleus i, Z_i is the atomic number of nuclei i and m_e is the mass of the electron. The Laplace operator of particle i is:

$$\nabla^2_{\mathbf{r}_i} = \partial^2/\partial x_i^2 + \partial^2/\partial y_i^2 + \partial^2/\partial z_i^2. \tag{5}$$

As indicated above, the *kE* operator is invariant under rotation of the Cartesian frame used to express x_i, y_i, and z_i.

With slight abuse of notations (double sum replaced by one sum with two indices), the H of a molecule comprising N electrons and K nuclei is obtained by summing up the above component resulting into

$$\begin{aligned} H \;=\; & -\sum_i \overset{(1)}{\frac{\hbar^2}{2m_e}}\nabla_i^2 - \sum_I \overset{(2)}{\frac{\hbar^2}{2M_I}}\nabla_I^2 - \sum_{i,I} \overset{(3)}{\frac{e^2}{4\pi\varepsilon_0}} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} \\ & + \underset{(4)}{\frac{1}{2}\sum_{i\neq j} \frac{e^2}{4\pi\varepsilon_0}} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \underset{(5)}{\frac{1}{2}\sum_{I\neq J} \frac{e^2}{4\pi\varepsilon_0}} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_I|}, \end{aligned} \tag{6}$$

where Z_I, M_I, and $\mathbf{R}_I$, stand for atomic number, the mass, and position of the Ith nucleus, and $\mathbf{r}_i$ is vector of coordinates of the ith electron. Terms (1) and (2) in the expression for H are the *kE* components of the electrons and nuclei, respectively. The terms (3), (4) and (5) represent the Co-repulsion between the indicated components of a molecule. For simplicity, atomic units are used, for length $a_0 = 1$ Bohr $(0.529 \times 10^{-10}\mathrm{m})$, for mass electron mass m_e, and for energy 1Hartree (1 Hartree $e^2/4\pi\varepsilon_0 a_0 = 27.211$ eV). Using $M'_I = M_I/m_e$, the H in atomic units becomes

$$\begin{aligned} H = & -\sum_i \frac{\nabla_i^2}{2} - \sum_I \frac{\nabla_I^2}{2M'_I} - \sum_{i,I} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} \\ & + \frac{1}{2}\sum_{i\neq j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \frac{1}{2}\sum_{I\neq J} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} \end{aligned} . \tag{7}$$

The focus here is on the *electronic* component of the molecule. Since a nucleon is considerably heavier than an electron, the Born-Oppenheimer expression is used, considering the nuclei as c-point charges so that for a given nuclear configuration it is only needed to solve the electronic H

$$H_e = -\sum_i \frac{\nabla_i^2}{2} - \sum_{i,I} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} + \frac{1}{2}\sum_{i\neq j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}. \tag{8}$$

The goal is to find energy eigenstate (*eigS*) $|E_i\rangle$ and the corresponding energy eigenvalues (*eigV*) E_i of H_e .

Quantum phε: phε (Kitaev, 1995b) is used to identify the lowest energy *eigS*, $|E_0\rangle$, and excited states, $|E_{i>0}\rangle$, of a physical H (Abrams et al., 1999). The canonical *phε* algorithm is described in Nielsen and Chuang

(2002) and presented in Fig. 9.

1. The qubit $\mathscr{R}$ is prepared in state $|\Psi\rangle$, having partial overlap with the *eigS* of the system. Here an extra $\mathscr{R}$ of ω ancilla ($\mathscr{A}$-) qubits is needed. The $|\Psi\rangle$ can be expanded in terms of energy *eigS*'s of the H, as $|\Psi\rangle = \Sigma_i c_i |E_i\rangle$, where c_i are complex coefficients.
2. Then the $\mathscr{A}$-$\mathscr{R}$ is set in the superposition $\sum_x |x\rangle/\sqrt{2^\omega}$, by applying a Hadamard gate to each $\mathscr{A}$-qubit, where x are all possible bit-strings that can be obtained from ω bits. Then the controlled gates shown in Fig. 9 is used.
3. In the next step the q-IFT is applied to the $\mathscr{A}$-qubits to identify the phase, which encodes the data about the energy *eigV*.
4. Measuring the $\mathscr{A}$-qubits in the Z basis, returns an estimate of *eigV* (E_i), with probability $|c_i|^2$ This $\mathscr{M}$ collapses the main $\mathscr{R}$ into the corresponding energy *eigS*, $|E_i\rangle$

The number of $\mathscr{A}$- qubits ω, used for the mechanism of *phε* presented above, defines the achievable success probability and precision in the energy estimate. Authors in Nielsen and Chuang (2002) show that for a precision of a binary estimate of the energy over n bits, with success probability p, $\omega = n + \lceil \log_2(2+1/2p) \rceil$ $\mathscr{A}$-qubits are needed. The *phε* has been experimentally studied in a variety of q- systems (Du, 2010, Lanyon et al., 2010, Li, 2011, O'Malley, 2016, Paesani, 2017, Santagati, 2018, Wang et al., 2015).

To execute this algorithm, it is needed to sequentially evolve in time the main $\mathscr{R}$ using the H for times $t_0 = 2\pi$, $t_1 = 4\pi$, …, $t_{\omega-1} = 2^\omega \pi$. The overall coherent time evolution, T, is $T = 2^{\omega+1}\pi$. With this, for a $p_{success}$ =0.5, $\omega = n + 2$ $\mathscr{A}$-qubits are needed. The relation between the binary precision $\varepsilon_{\rm PE} = 1/2^n$ and T is $T = 8\pi/\varepsilon_{\rm PE}$. For $p_{success}$ =0.5, the number of the procedure repetitions n_r =2 is needed to get a fair estimate of the *eigS*, $|E_0\rangle$. This leads to an overall number of $16\pi/\varepsilon_{\rm PE}$ uses of the U e^{-iH} (Reiher, 2017). Now, since $c_0 < 1$, the n_r must be multiplied by $1/|c_0|^2$, to get the ground state.

In addition to $\varepsilon_{\rm PE}$, obtained in E_0, the errors $\varepsilon_{\rm U}$ caused by imperfect design of the controlled U evolutions applied to the main $\mathscr{R}$, must be also accounted for. This error is caused by disassembling e^{-iH} into arbitrary single and two qubit gates, in the process of a Trotter disassembling. Circuit synthesis errors, $\varepsilon_{\rm CS}$, arise from building up gates from a discrete set (library) of gates. An example is approximating single qubit rotations from multiple T (Tiffoli) and Hadamard gates. They can be calculated using the Solovay-Kitaev theorem (Dawson, 2005). For a Trotter disassembling of e^{-iH}, (Reiher, 2017) shows that the upper bound on error in the energy *eigV* obtained from *phε* is $\varepsilon_{\rm PE} + \varepsilon_{\rm U} + \varepsilon_{\rm CS}$. In practice, it is not easy to optimize these error budgets in order to minimize the overall error (Reiher, 2017, Kivlichan, 2019a).

For any version of *phε*, there are two general features.

1) The $\mathscr{R}$ needs to be initially in a state partially overlapping with the target *eigS*. 2) There must be a solution for coherent realization of a U-operator defined by an invertible function of the H. The common choice for U-operator is e^{-iH} used above. Methods to satisfy both requirements will be surveyed in the following sections.

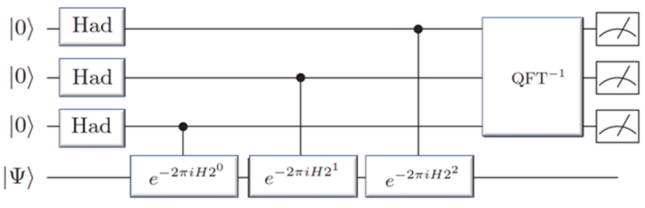


Fig. 9. The canonical q- phε circuit with three $\mathscr{A}$- qubits. When the $\mathscr{A}$- qubits are in state $|x\rangle$, a control rotation $e^{-2\pi i H x}$ is applied to the target state $|\Psi\rangle$. QFT denotes the q- Fourier transform (Caruana, 1997; Glisic, 2023a). By measuring the $\mathscr{A}$- qubits in the computational basis, they collapse to an eigenvalue of H and the $\mathscr{R}$- qubits collapse to an estimate of the corresponding energy eigS (McArdle, 2020).

State Initialization (sI): It is not trivial to initialize the qubit $\mathscr{R}$ in a state with a needed overlap with the target *eigS* (typically the ground state-*gS*). If chosen at random, the state would collapse to the desired *gS* with an exponentially vanishing probability, as the system size increases.

In addition, authors in McClean (2014) show that the complexity of *phε* exponentially increases, by taking into account the imperfect initiation of *eigS*'s of subsystems that do not interact. This supports the need for *sI* procedures with at least a polynomially decreasing overlap with the full configuration interaction (FCI) *gS*, as the system size increases. Different such techniques are available for *sI*. One solution is to initialize reference states obtained from c-tractable calculations, like configuration interaction states (Babbush et al., 2015), open-shell spin symmetry-adapted states (Sugisaki, 2016), multireference states (Sugisaki, 2019), or states generated by adaptive sampling configuration interaction methods (Tubman et al., 2018a). Additional options of interest are also: the variational methods discussed in Yung (2014), q-algorithms for imaginary time evolution (Motta, 2019), or adiabatic *sI* (Aspuru-Guzik et al., 2005). Here the paper focuses on adiabatic *sI*, inspired by the adiabatic model of q-computation (Farhi et al., 2000).

For given Hamiltonian H_s, a state $|\Psi\rangle$ that is near its gS can be initiated by using adiabatic *sI* (Albash et al., 2018). In this process, the starting point is a simple Hamiltonian H_0 initiated in its gS. Then the system is evolved in time by using a H that evolves step by step from H_0 to H_s, thus initiating a state that is near the gS of H_s. Such approach to *sI* will be more efficient if the gap between the gS and the next state on the path between H_0 and H_s is smaller. For *cCh*, adiabatic state preparation (ASP) may be reached by starting the system in the gS of the Hartree-Fock H (H_0), and interpolating in time t between the starting and final H_s as $H(t) = (1 - t/T)H_0 + (t/T)H_s$, where T is the maximum simulation time (Aspuru-Guzik et al., 2005). For other options potentially more suitable for problems of *cCh* see (Veis, 2014, Wecker, 2015). The maximum annealing time, T is given by $T \approx \mathscr{O}\left(M^4/\min_s \Delta(t)\right)$, where $\Delta(t) = E_1(t) - E_0(t)$ and M is the number of spinorbitals in the molecule. Authors in Reiher (2017) suggest that the scaling may be closer to $\mathscr{O}\left(M^2/\min_t \Delta(t)\right)$.

Difficulties in knowing a priori how big the gap along the entire adiabatic path is, limits the possibility to perform ASP in the limited time window. One option for reducing the annealing time needed is to use additional *driving* H's, like what was used in Matsuura (2018), Veis (2014). Although the above expression $T \approx \mathscr{O}\left(M^4/\min_s \Delta(t)\right)$ does not explicitly depend on the *iS* used, by starting in a state that is near the target gS should make the anneal path shorter.

If an *iS* overlapping sufficiently with the *gS* is available, it may be possible to surpass adiabatic *sI* completely, and instead do *phε* directly on that *iS*. As seen from the above discussion, *phℰ* only needs an overlap with the target *gS*. There are several methods for evolving the system under this time-dependent H, as well. H and *phℰ* for chemistry simulation *(ChSim)* will be discussed in Section 8.

5.2. Complexity of q- Chemistry (qCh) algorithms

Most work on *q*-simulation in chemistry provides answers to the electronic structure estimation by using *phℰ* to estimate *eigV*'s by sampling molecular *eigS*'s (Kitaev, 1995a). Even with limited resources (≅ 100 qubits), this could provide meaningful results in different research areas and the development of technologies.

By using a basis of single-particle functions called orbitals to discretize the multiple-body *wF*, molecular systems on a q-computer have been modeled. Most qCh processing uses either plane wave (*pW*) orbitals, or the orbitals obtained by linear combinations of Gaussians. Using *pW* enables highly structured H's. Authors in Babbush et al. (2018a) show that this approach enables asymptotic advantages for q-algorithms. Nowadays, the best-scaling *qCh*-algorithms in second

quantization (2ndQ) use *pW*; achieving $\mathscr{O}(N^3)$ (Babbush et al., 2018b; Kivlichan, 2019) or $\mathscr{O}(N^2 \log N)$ (Low & Wiebe, 2018) gate complexity, with small or large constant factors and more spatial complexity, respectively.

A need for rather large number of spin-orbitals to represent many molecular systems to chemical accuracy represents the main constraint to using *pW* in 2ndQ. For resolving this problem, the work of Babbush et al. (2019) suggests simulating the *pW* H in the first quantization (*1stQ*) offering $\overline{\mathscr{O}}(N^{1/3}\eta^{8/3})$ gate complexity, with η being the number of electrons. *(Note: 1stQ uses quantized variables like position and momentum, as operators. 2ndQ uses quantized fields, rather than variables.)* Due to low scaling in *N*, one might need an extremely large *pW* basis. The feasibility of practical implementation of that approach has not been studied enough. It has not been compiled to specific circuits, and it is not known how large the basis should be (Low & Wiebe, 2018).

The low resolution of *pW* can be improved by using a more compact basis. Nowadays, most approaches for the q- *ChSim* suggest using very compact molecular orbitals. Unfortunately, this results into complex H with coefficients expressed by integrals and $\mathscr{O}(N^4)$ distinct terms. The first such algorithm had gate complexity $\mathscr{O}(N^{11})$ (Whiteld,2011). Later, researchers have reduced the complexity by using tighter bounds (Poulin, 2015), better mappings between fermions and qubits (Jiang, 2018), improved *sI* techniques (Tubman et al., 2018b), application of new time-evolution strategies (Low, 2019, Babbush et al., 2016, Campbell, 2019), considerations of fault-tolerant overheads (Litinski, 2019) and other solutions in representation and algorithmic structures (Motta, 2018).

The minimum implementation cost of earlier work on 2ndQ arbitrary basis *ChSim* is either the $\overline{\mathscr{O}}(N^5)$ scaling of Babbush et al. (2016), or the $\overline{\mathscr{O}}(\lambda^2)$ scaling of Campbell (2019), with λ being the 1-norm of the H. While the solution from Babbush et al. (2016) has large constant factors in the scaling, the method of Campbell (2019) scales quadratically worse than post-Trotter solutions regarding the evolution time (*eT*). For practical implementation, the best prior solution is Lie-Trotter proosal (Motta, 2018), although the step size for that method is unknown. Authors in Dominic (2019) present a solution with $\overline{\mathscr{O}}(N^{3/2}\lambda)$ T (Toffoli) cost, which looks better than any prior work as long as $\lambda = \Omega(N^{3/2})$, which is most common case.

Works presented in Babbush et al. (2018b), Kivlichan (2019) enable compilation of *qCh* algorithms in terms of Clifford +T (Toffoli) gates and calculate the resources needed within an error-correcting code are. These works minimize T-complexity since these gates cannot be transversely realized within practical codes (Litinski, 2019). The gates are implemented by distilling magic states or Toffoli states, which needs orders of magnitude more space/time volume (qubit seconds) than executing Clifford gates, together with a large consumption of physical qubits (Fowler, 2018).

Authors in Reiher et al. (2017), Beinert et al. (1997) focus on the simulation of an active space of the FeMo cofactor of the Nitrogenase enzyme. Due to the complex electronic structure, the process is not fully understood although the reaction is important. Nitrogen fixation is much more efficient than the option used in industry. Authors in Reiher et al. (2017), Litinski (2019) used a 108-qubit active space requiring about 10^{14}T gates.

Authors in Babbush et al. (2018b), Kivlichan (2019) demonstrate feasibility to perform *ChSim* of similar size with approximately 10^8 T-gates, using a *pW* rather than Gaussian basis. By using methods published in Gidney (2019) such calculations could be realized in the surface code (Glisic and Lorenzo, 2022) at 10^{-3} physical error rates with less than a million physical qubits in just hours.

The solution presented in McArdle et al. (2020) suggests performing *ph*$\mathscr{E}$ directly on a *q*- walk (Dominic, 2019, Szegedy, 2004, Low, 2019), used to simulate H's in the linear combinations of U's query model (Childs, 2012). The analysis of the *ph*$\mathscr{E}$ -algorithm presented in McArdle et al. (2020) is similar to that in Babbush et al. (2018b), implementing a solution proposed in Poulin (2017), Berry et al. (2018) based on qubitization (Low, 2019). Reference (McArdle et al., 2020) uses the U-iteration technique from Babbush et al. (2018b), Childs (2018) and the QROM based *sI* and coherent alias sampling methods suggested in Babbush et al. (2018b) and then simplified to reduce the complexity in Low (2018). The algorithm leverages the sparse nature of the Co-operator, using a low rank representation from Motta (2018).

If a limited number of $\mathscr{A}$-qubits is used (McArdle et al., 2020), using the system qubits as "dirty" $\mathscr{A}$, the algorithm can reach *che-* accuracy for FeMoco with about 2×10^{13} Toffoli gates, using the active spaces of Reiher et al. (2017) or Li (2019).

If many $\mathscr{A}$ were used then the number of Toffoli gates needed with the best option of McArdle et al. (2020) would be about 2×10^{11} for the (Reiher and al, 2017) orbitals, or 8×10^{10} for the (Li, 2019) orbitals. The required number of gates is four times larger but since Toffolis are critical, Toffoli states can be distilled directly. The cost for this would be approximately the same as for distilling two magic states for gates (Gidney, 2019).

Although (McArdle et al., 2020) improves upon the distillation space/time volume required by Reiher et al. (2017), at 10^{-3} error rates, about 3 times 10^6 qubitweeks of state distillation are still required, which improves over previous results by a factor of seven hundred but is still unacceptable.

For the latest results on Chemistry Application to Quantum Error Correction Primitives and Benchmarking Adaptive Quantum Circuit Optimization Algorithms for Quantum Chemistry see Blunt et al. (2023) and Saib et al. (2023) respectively.

6. Synchronization

6.1. n-Sci and N-sync

Over the past several decades, an extensive study of stability and periodic oscillation for NNs has been witnessed. Time delay (*tD*) is often present in bio- and ANNs, causing oscillation and instability for a NN. Nowadays research results are available, defining conditions guaranteeing the stability of NNs with *tD* (Zhang, 2008). Sometimes, in *dS, tD* is present in both the system state and in its derivative (Hale, 1993). To model such systems, differential neutral (*n-*) delay (*nD*) equation is used, where the delay consists of the *nD* and the retarded delay (http, n.d.e; http, n.d.f). This model is used in several applications. Since the *nD* exists in NNs, the stability analysis of *n-* NNs has attracted a lot of attention (Gao, 2018; Wen, 2012).

Complex dynamic networks (*dN*) are of interest for this paper, since among the number of important practical models they can also simulate neural networks. Since its introduction in Pecora (1990), synchronizing complex *dN (cdN)* has become an important topic applicable in many fields including *ch*-reaction, bio-systems, and info-science (Abbasi et al., 2013). In addition, *cdN's* exhibit more sophisticated behaviors than a single NN (Stanoev, 2013) making sync of *cdN* an appealing problem. The sync problem has been modeled by using the c-methods such as *feedback control (Ctr)* (*Karimi, 2010*), *sampled-data Ctr* (*Wu et al., 2013a*), *backstepping Ctr* (*Xia, 2009*), *pinning Ctr* (*Yang, 2013*), *impulsive Ctr* (*Lu, 2011*), *adaptive Ctr (aCtr)* (*Yang, 2010*) *and st-perturbation* (Zhang).

In (Qin, 2015), a unified method was used to study the exponential sync (*e-sync*) of *cdN*. Authors in Wang (2015) studied the global *sync* of *cdN* with a limited data rate. Investigating the *sync*-problem in NNs of *n*-type in reliable network of robots (Yamamoto, 2004) and chaotic secure communication (Yang, 2004) has shown promising results. By using the linear matrix inequality (LMI), the global-*sync* of *cdN* with *n*-type delayed nodes is presented in Ji (2011).

Markov jump system (*Mjs*) (Boukas, 2006) is used to model dynamics with abrupt changes. Here parameters describing the state are continuous and the jumping parameters are discrete, therefore such a system is

Table 4
CrossTechology coverage (focus-Complex q-models).

1	2	3	4	5	6	7	8	9	10	11
(http, n.d.d, Christiansen, 2012, Abrams et al., 1999, Nielsen & Chuang, 2002, McArdle, 2020, Du, 2010, Lanyon et al., 2010, Li, 2011, O'Malley, 2016, Paesani, 2017, Santagati, 2018, *Wang* et al., *2015*, *Reiher, 2017*, *Glisic, 2023a*, *Kivlichan, 2019a*)	*QCh*		√			√	*The electronic structure problem in Quantum Computational Chemistry*		√	
(Kitaev, 1995a-3013, Reiher et al, 2017, Gidney, 2019, Poulin, 2017, Li, 2019)	*QCh*		√			√	*Complexity of Quantum Chemistry Algorithms*		√	
(*McClean, 2014*, Babbush et al., 2015, Sugisaki, 2016, Sugisaki, 2019, Tubman et al., 2018a, Yung, 2014, Motta, 2019, Aspuru-Guzik et al., 2005, Farhi et al., 2000, Albash, 2018, Veis, 2014, Wecker, 2015, Reiher, 2017, Matsuura, 2018, *Veis, 2014*)	*QCh*		√			√	*State preparation: Initializing the qubit register*		√	
This paper	*NET*	√	√	√	√	√	*CrossTecnology survey*	√	√	√

1 reference, 2 focus, 3 classic, 4 quantum, 5 complex networks, 6 tensors, 7 q-simulations, 8 contribution, 9 energy efficiency, 10 computational efficiency, 11 synchronization.

regarded as a hybrid system. Modeling using *Mjs* was presented in Bo et al. (2014). *Sync* for *cdN* with *Mjs* has been studied in Liu et al. (2009a), Wu et al. (2013b), Yang (2012), Zhang (2015).

Authors in Yang (2012), derive sufficient *sync* conditions for NN's with the Markovian coupling and different node-delays and random coupling strengths. For the same network conditions and time-varying delay (*tvD*), *st-sync* was investigated in Zhang (2015). The analysis is based on a new mode-dependent augmented Lyapunov (*Ly*-) *sync*-criteria. Unfortunately, sync for the *cdN* of *n*-type with *Mj* parameters did not attract much research interest. By using *Ly-sync*-theory, the problem of robust mode-dependent delayed state feedback control (*fb-Ctr*) was solved for a class of uncertain *tD* systems with *Mj* parameters and mixed discrete, distributed and *n*-delays in Karimi (2011).

The sync of coupled NN's of *n*-type with *Mj* -mode dependent discrete and unbounded distributed delays has been considered in Liu (2013), by constructing a *Ly*-functional. Sync for Markovian *st*-coupled (*Msc*) NNs has also been considered in Wang (2010), Zhu (2012), since external *st*-disturbances, modeled by Brownian motion (Mao, 2011), exist in practice. The *e-sync* for *Msc* NNs of *n*-type with *tvD* has not been discussed due to the difficulty lying in the complexity caused by the simultaneous existence of the n- item, the st-perturbation, and the Markov switching.

Adaptive control (*aCtr*) is useful in designing the sync of *cdN* (Sastry, 1989) since the control variables can adapt via corresponding updating rules in reacting to the variation in the behaviour of the system (Pariño, 2000). Sufficient conditions to guarantee the adaptive *e-Sync* in *st*-coupled NNs of n-type under an adaptive feedback control rule were obtained in Zhang et al. (2013b), by using the *Ly*-method and some properties of the Kronecker product. By using mechanisms presented in Mao (2008) and Kolmanovskii (2003), the adaptive *e-Sync* for *st*-NNs of n-type with *Mjs* was presented in Gu (2003). The algorithms presented in Karimi (2011) Liu (2013), and Zhou (2014) are valid for constant delay case.

As discussed in Gu (2003), in practice, the *tvD* model is more adequate than the model with constant delay. The extensions of those methods to the *tvD* are not simple, especially when the restrictive conditions (*r-Con*), that the derivative of the *tD* function is less than one, are not imposed. If both, n-item and the *st*-perturbation are present at the same time the problem becomes difficult. As an example, in Zhou (2013), although the adaptive *e-Sync* for *n- st*-NNs with *tvD* and Markovian switching is studied by using the methods from Mao (2008) and Kolmanovskii (2003), this *r-Con* on the *tvD* must be imposed. The case without the r-*Con* has not been studied yet. Thus, the global e-*Sync* for the *Msc*- NNs of n-type with *tvD* under an adequate adaptive feedback control (*afbC*) law needs to be further examined.

In the sequel, some of the above-mentioned solutions will be explicitly discussed in more detail.

Synchronization of NN with stochastic perturbation (st-pert): Here the sync problem for NN with *st-pert* with intermittent control (*iC*) via adaptive aperiodicity (*aA*) is discussed. By using *st*-theory and *Ly* stability methodology, methods of *iC* with *aA* are used to achieve the sync of *st*-NN. Sufficient conditions for achieving a sync of the underlying network are established in Zhang et al. (n.d.).

NOTE: ***Preliminaries on Synchronization***

Here a neural network system is considered consisting of *N* identical nodes with nonlinear coupling φ with vector-form stochastic perturbations, described by

$$\dot{x}_i(t)=\left[-Cx_i(t)+Bf(x_i(t))+\sum_{j=1,i\neq j}^{N}a_{ij}\left[\varphi_j\left(x_j(t)\right)-\varphi_i(x_i(t))\right]\right]dt +\sigma(x_i(t))d\omega(t) \tag{i12}$$

where $x_i(t)=(x_{i1}(t),\ x_{i2}(t),\ \ldots,\ x_{in}(t))^T\in\mathbb{R}^n$ represents the state vector of the *i*- th node; $C=diag\ (c_1,\ c_2,\ \ldots,\ c_n)$ with $c_k>0, k=1,2,\ldots,n$, denotes the rate with which the *k*th cell rests its potential to the resting state when isolated from other cells and inputs; $B=\left[b_{ij}\right]_{n\times n}\in\mathbb{R}^{n\times n}$ represent the connection weight matrix; $A=\left[a_{ij}\right]_{n\times n}\in\mathbb{R}^{N\times N}$; $f(x_i(t))=\left[f_1(x_i(t)),f_2(x_i(t)),\ \ldots,\ f_n(x_i(t))\right]^T$ is a continuous vector; $\sigma(x_i(t))=\sigma(x_1,\ x_2,\ \ldots,\ x_n)\in\mathbb{R}^{n\times n}$ is the noise intensity matrix and $\omega(t)=(\omega_1(t),\ \omega_2(t),\ \ldots,\ \omega_n(t))^T\in\mathbb{R}^n$ is bounded vector-form Weiner process, satisfying $E\omega_j(t)=0,\ E\omega_i^2=1,\ E\omega_j(t)\omega_j(s)=0(s\neq t)$.

In the case that system (8.1) reaches synchronization,

$x_1(t)=x_2(t)=\cdots=x_N(t)=s(t)$, by introducing a controller into each individual node, where $s(t)\in\mathbb{R}$ is defined as

$$\dot{s}\ (t)=[-Cs(t)+Bf(s(t))]dt+\sigma(s(t))d\omega(t) \tag{i13}$$

where $s(t)$ can be set to be any desired state: either equilibrium point, or a nontrivial periodic orbit, or even a chaotic orbit.

To achieve the synchronization objective, the aperiodically intermittent controllers will be applied to some of its nodes. For convenience, notation

$$\varphi\left(x_j(t),\ x_i(t)\right)=\varphi_j\left(x_j(t)\right)-\varphi_i(x_i(t))$$

is used. Thus, the intermittent controlled network can be formulated as

$$\dot{x}_i(t)=\left[-Cx_i(t)+Bf(x_i(t))+\sum_{j=1,i\neq j}^{N}a_{ij}\varphi\left(x_j(t),x_i(t)\right)+u_i(t)\right]dt +\sigma(x_i(t))d\omega(t) \tag{i14}$$

where $u_i(t)(i=1,2,\ \ldots,\ n)$ are the intermittent linear state feedback controller and are constructed as following:

$$u_i(t)=\begin{cases}-\varepsilon_i\varphi(x_i(t),\ s(t)), & t\in[t_i,\ s_i),\\ 0, & t\in[s_i,\ t_{i+1}),\ i=0,1,2,\ \ldots\end{cases} \tag{i15}$$

where $\varepsilon_i>0$ represents control gain and $\Xi=diag\ (\varepsilon_1,\ \varepsilon_2,\ \ldots,$

$\varepsilon_N) \in \mathbb{R}^{n\times n}$. The synchronization error is defined to be $e_i(t) = x_i(t) - s(t)$. By the controller expression (8.4), the error dynamics is governed by

$$\dot{e}_i(t) = \left[-Ce_i(t) + Bg(e_i(t)) + \sum_{j=1, i\neq j}^{N} a_{ij}\varphi\left(x_j(t), x_i(t)\right) + u_i(t) \right] dt + \widetilde{\sigma}(e_i(t))d\omega(t) \tag{i16}$$

where $g(e_i(t)) = f(x_i(t)) - f(s(t))$ and $\widetilde{\sigma}_i(t) = \sigma(x_i(t)) - \sigma(s(t))$.

6.1.1. Design example 5

Here, along the lines presented in Zhang et al. (n.d.), numerical simulations of two examples are presented to demonstrate the main results obtained in the previous section.

$$\dot{x}_i(t) = \left[-Cx_i(t) + Bf(x_i(t)) + \sum_{j=1, i\neq j}^{N} a_{ij}\left[\varphi_j\left(x_j(t)\right) - \varphi_i(x_i(t))\right] \right] dt + \sigma(x_i(t))d\omega(t) \tag{i17}$$

where $N = 100, f(\cdot) = (|x+1| - |x-1|)/2$, $\varphi(x(t)) = x(t) + tanh(x(t))$, $C = diag(1.5, 1.5)$.

In this case, the coupling configuration matrix A and the connection weight matrix B are given by

$$A = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 \\ 1 & 0 & \cdots & 0 & -1 \end{pmatrix}_{100\times 100} \qquad B = \begin{pmatrix} 2 & -0.1 \\ -5 & 4.5 \end{pmatrix},$$

Fig. i2a depicts the trajectories of error states of (*i17*) without aperiodically intermittent, which indicates that the network (*i17*) without aperiodically intermittent cannot synchronize itself.

Fig. i2b shows the trajectories of error states of the system with aperiodically intermittent, which approach to zero as time increases. Hence the network is synchronized under the synchronizing aperiodically intermittent control.

6.1.2. Design example 6

In the following example, neural network is modelled as:

$$\dot{x}_i(t) = \left[-Cx_i(t) + Bf(x_i(t)) + \sum_{j=1, i\neq j}^{N} a_{ij}\varphi\left(x_j(t),\ x_i(t)\right) + h_i(t)u_i(t) \right] dt + \sigma(x_i(t))d\omega(t) \tag{i18}$$

where $h_i(t) = \pi_i e^{-\rho_1 t}|||\varphi(e_i(t))|||_2^2$, $\varphi = 3$, $C = diag$ (2.5, 2.5) and $N = 100$. The other parameters are the same as those given in example #2.

Fig. i3a and b show the synchronization errors of $x_{11} - x_{i1}$ and $x_{12} - x_{i2}$, respectively. These figures indicate that synchronization can be achieved.

Complex networks (*cN*) are nowadays becoming more and more present in society. A *cN* is modeled by a large collection of nodes communicating and interacting via a collection of links. Phenomena, appearing in communication network, bio formation, chemical reaction, NN, social organization, the WWW, etc., can be characterized by *cN*. Since the first discussion on the *small-world* (SW) and *scale-free* (SF) properties (Glisic, 2016; Watts & Strogatz, 1998; Barabasi & Albert, 1999), the interest in studying *cN* has been increased and had an impact on the work of researchers in other fields. The theory of *cdN*, is used to study dynamic behaviors, such as sync, consensus, self-organization, and combinatorial optimization (Kitaev, 1995b; Guan, 2010; Li, 2014; He et al., 2014a; Wen, 2015; Lu, 2009; Wen et al., n.d.; Liu et al., 2009b, He et al., 2014b; Wen, 2013; Tang, 2012). In the past, *sync* of large-scale (*ls*-) *cN* consisting of coupled *dS* has been studied in Tang (2012), Wu (2007).

Coupled NN, as a special class of *cN*, have been in the focus of research interest including NN-sync with all dynamical nodes, general NN, secured communication and network updating. As a part of the effort, many control algorithms are available for *sync* of NN governed by nonlinear systems, such as *aCtr* (Zhang, 2009), fuzzy control (Gao, 2014), impulsive control (Zhang et al., 2013b) and *iC* (Hu et al., n.d.).

The *iC*, used to control the nonlinear *dS* in Zochowski (2000), among the other applications has been also used in communication.

Earlier, the *iC* was mainly used as periodical *iC* (Wang, 2013). In (Cai et al., 2009), periodical *iC* is used for the NN with *tvD* in a desired orbit. Ref. (Yu, 2012), discusses the *e-sync* for delayed fuzzy cellular NN using periodically *iC*. Imperfections always exist in practice, such as *st*-forces on the physical systems and noisy $\mathscr{M}$'s. Communication between nodes of NN is always subject to *st*-perturbations from environment, which may cause loss of data. Therefore, *st*-perturbations must be considered as well (Wang, 2013; Lu, 2008; Yang, 2009; Pototsky, 2009). In (Wang, 2013), the *e-sync* of s-perturbed *cN* with *tvD* via periodically intermittent pinning was studied. In (Yang, 2009), st-synchronization of coupled NN with *iC* was also analyzed.

The need for periodicity of *iC* methods may not be convenient in real applications. The electricity production by wind power, for example, depends on the weather, which is aperiodically (*ap*-) intermittent. So, for the analytical study of real systems, it is better to analyze the sync using *ap-iC* model. The analysis of system dynamical behavior by using *ap-iC*

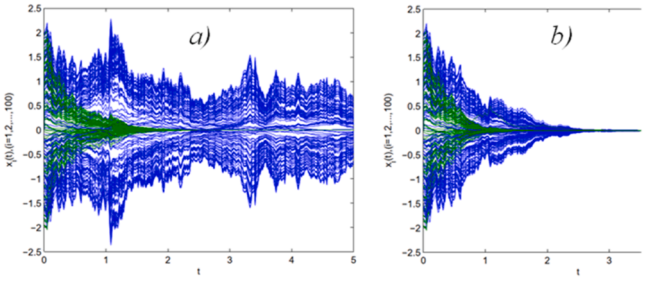


Fig. i2. a) Time response of the error states of (i17) without aperiodically intermittent b) Time response of the error states (i17) with aperiodically intermittent (Zhang et al., n.d.).

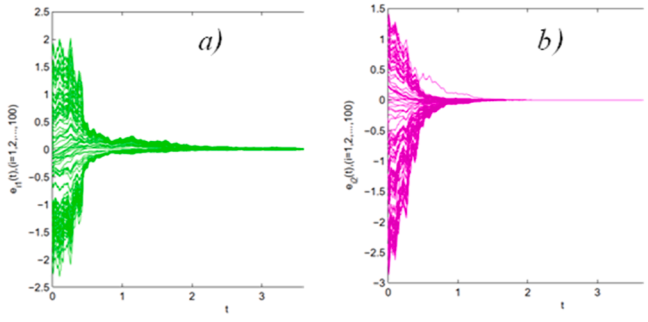


Fig. i3. a) Time response of the error states of (i18) with adaptive aperiodically intermittent b) Time response of the error states (i18) with adaptive aperiodically intermittent (Zhang et al., n.d.).

methods was presented in Liu (2014), investigating sync of nonlinear coupled networks via a-intermittent pinning control.

Inspired by Liu (2014), reference (Zhang et al., n.d.) investigates the problem of sync of NN with st-perturbation via ap-*iC*. The authors establish sufficient conditions to achieve sync for nonlinear coupled networks under ap-*iC*. By using Weiner estimation techniques, suitable *a*-intermittent and adaptive *ap- iC* are developed to ensure *st*-sync for the coupled *cN* with *st-* perturbations. Sync criteria obtained are verifiable, and practically useful.

6.2. Large scale (ls-) networks synchronization

Here the problem of sync is revisited by generalizing the problem to the *cN* with extremely large number of nodes. *Sync* processes of interacting nodes are the focus of study in the networks discussed in the survey so far, as well as in the economic and social systems. The work in *synch* leverages the recent theory of *cN*. Here, the sync phenomena are discussed when the oscillating objects interact in a *cN* topology. The interdependency between the structure and the function of the given type of connections is also discussed. Applications of *sync* in *cN* modeling are also surveyed: bio-systems and *n-Sci*, and other types of networks, discussed earlier in the paper, as well as in economy and social sciences.

Complex Networks: Analytical abstraction of *cN* is a graph $\mathscr{G}$ consisting of a set $\mathscr{N}$ of N nodes interconnected by a set $\mathscr{L}^*$ of L links, with degree k_i of node. Here $\mathscr{G}$ is specified by matrix A, with elements $a_{ij} = 1$ if there is a directed link from j to i, and 0 otherwise. For a weighted network (*wN*), G is defined by a matrix W, with elements w_{ij}, reflecting some parameter (cost, delay, capacity…) of the link between j and i. The study of the *st-* characteristics of many *cN* shows that, even for very different systems, some categorization/ classification of these networks is possible. Among these properties, the most characteristic one refers to the *degree distribution* $P(k)$, representing the likelihood that a node has a degree k. This parameter of *cN* is considered as its most differentiating factor. A number of other parameters are used to additionally elaborate the categorization. Among those the most often used are the *average shortest path (shp) length* $\ell = \langle d_{ij} \rangle$, with d_{ij} being the length of the *shp* between node i and node j, and the *clustering coefficient* C accounting for the fraction of actual triangles (three vertices forming a loop) over possible triangles in the graph.

The first characterization of *cN* uses the degree distribution $P(k)$ where the degree is related to the tail of the distribution. In homogeneous networks, like the Erdös–Rényi (ER) *random graph* (Glisic, 2016; Erdös & Rényi, 1959), the distribution decays exponentially with the degree. The network is heterogeneous if the distribution has a heavy tail. As an example, *scale-free* (SF) networks (Glisic, 2016) have a power-law distribution, $P(k) \sim k^{-\gamma}$, the Barabási-Albert (BA) model (Glisic, 2016; Barabási & Albert, 1999) being the typical model of this type of graph. In this network new incoming nodes are linked preferentially to the existing nodes (Glisic, 2016) with highest degree. (*By the rule, the likelihood of connecting a new edge to a given node is higher if he has higher degree k*).

On the other hand, in lattices networks, all nodes have the same degree. The *average shp length (aspl)* ℓ can be also included into this categorization. For a lattice with N vertices, obviously, $\ell \sim N^{1/d}$. An approximate estimate of ℓ for a random network, is also possible. For the network with the average number of nearest neighbors of a node $\bar{k}$, approximately $\bar{k}$ nodes of the network are at a distance ℓ from the node or closer. Hence, $N \sim \bar{k}^{\ell}$ and then $\ell \sim \ln(N)/\ln\left(\bar{k}\right)$, i.e., the *aspl* is small even for large networks. This is known as the SW (Small World) property (Glisic, 2016). When it comes to distances, there are a number of parameters providing information about *centrality* of nodes. A node centrality is expresed in terms of the relative distance to the rest of the network. The *betweenness* is a parameter that represents the number of *shp's* between any pair of nodes in the network that go through a given node or link.

The clustering coefficient C is also useful for differentiating types of networks. It is expressed as: $C = \sum_{i=1}^{N} C_i/N = \sum_{i=1}^{N} n_i/(Nk_i(k_i - 1)/2)$, with n_i being the number of connections between nearest neighbors of node i, and k_i is its degree. A high C indicates the existence of many transitive connections, and a low C indicates the opposite.

Community structure tells us if the nodes are mutually connected in densely linked groups with sparser connections between the groups. The optimum partitioning of a network into groups is a hard problem. The most accurate and computationally efficient approaches, Danon et al. (2005), use the optimization of a parameter referred to as modularity (Newman & Girvan, 2004), defined as $Q = \sum_{ij}(a_{ij} - k_i k_j/2M)\delta_{c_i,c_j}/2M$ $\delta_{c_i,c_j}/2M$, $M = \sum_i k_i/2$, where c_i is the group to which node is assigned and the Kronecker delta function δ_{c_i,c_j} takes the value 1 if nodes i and j are in the same group, and 0 otherwise. The larger the Q the more modular the network is. This feature is especially adequate to unveil structure- function relationships in *cN* (Girvan, 2002).

Oscillator models on cN: Proper synchronization is of paramount importance for functioning of bio-NN discussed in this paper. The first studies of *sync* consider a network of coupled oscillators and interactions between them.

NOTE:

Coupled oscillators: For illustration, a network of five weakly coupled oscillators is sketched in Fig. i4. Every node shows oscillatory behavior in the two-dimensional state variables $\boldsymbol{x}_k = (x_k, y_k) \in \mathbb{R}^2$, $k = 1, \ldots, 5$. For each of the five nodes, a closed orbit is found, the *limit cycle* (depicted in blue), which describes the nodal dynamics in the absence of coupling. The weak coupling between the oscillators will 'kick' the dynamics away from the closed orbit, but only so far that the convergence toward it is sufficiently fast (see the two exemplary trajectories in black within the limit cycle). This allows for identifying the state of each oscillator $\boldsymbol{x}_k$ with a circular variable, the phase $\theta_k \in \mathbb{S}^1$. Deriving the dynamics $\dot{\theta}_k$ of the phase variables from the network dynamics $\dot{\boldsymbol{x}}_k$ is central to this section. Given the phase dynamics $\dot{\theta}_k$, the collective behavior of the full network can be inferred by means of the nodes' phase synchronization.

Throughout this section, the framework of weakly coupled oscillators is considered. A brief sketch of why this is beneficial for a concise presentation of the theory is given along with proper introduction of all required definitions later at due time. The governing dynamics of a network of N>> 1 interacting oscillators shall be of the form

$$\dot{\boldsymbol{x}}_k = \boldsymbol{f}_k(\boldsymbol{x}_k; \mu_k) + \kappa \boldsymbol{g}_k(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N), \quad k = 1, \ldots, N. \tag{i19}$$

The functions $\boldsymbol{f}_k$ determine the node-specific and uncoupled dynamics, whereas $\boldsymbol{g}_k$ comprises all coupling effects on oscillator $\boldsymbol{x}_k$ through the other nodes $\boldsymbol{x}_{j \neq k}$. The coupling strength is denoted by $\kappa \in \mathbb{R}$ and μ_k are bifurcation parameters. The weak coupling is guaranteed by

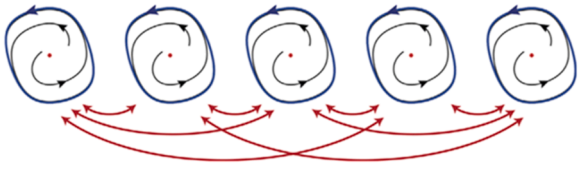


Fig. i4. A network of weakly coupled planar limit cycle oscillators. Each oscillator = 1, $N = 5$, is described in the two-dimensional state variables $\boldsymbol{x}_k = (x_k, y_k)$. The coupling between oscillators is indicated by red arrows. Without coupling, each oscillator follows the blue limit-cycle trajectory. Upon perturbation, the oscillator will be kicked away from the limit cycle and follows a trajectory that leads exponentially fast towards the globally attracting limit cycle. Globally attracting implies that the basin of attraction spans the whole $x - y$ plane except for the unstable origin (red). Two trajectories from within the basin of attraction are shown in black (Pietras & Daffertshofer, 2019).

assuming the coupling strength to be sufficiently small, $\kappa \ll 1$. Furthermore, it is assumed that the oscillators are nearly identical and that the coupling structure is pairwise, i.e. the coupling function $\boldsymbol{g}_k$ can be decomposed into the sum of pairwise interactions. Hence, one can rewrite (*i*19) as

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}_k;\mu) \ + \kappa \sum_{j=1}^{N} \boldsymbol{g}_{kj}(\boldsymbol{x}_k,\ \boldsymbol{x}_j) \tag{i20}$$

with μ being the only bifurcation parameter. Phase reduction implies transforming (*i20*) into the phase model

$$\dot{\theta} = \omega + \kappa \sum_{j=1}^{N} H_{kj}(\theta_k - \theta_j), = 1,\ \ldots,\ N \tag{i21}$$

In particular, the state $\boldsymbol{x}_k$ of every oscillatory node will be characterized by a phase variable θ_k. The corresponding phase dynamics comprises a *natural frequency* term ω and contributions from the other oscillators. The latter add up by means of *phase interaction functions* H_{kj} that depend on the pairwise phase differences $\theta_k - \theta_j$ of oscillators *k* and *j*.

Phase reductions and weakly coupled oscillators: There already exists abundant literature covering approaches to phase reduction. In the following, the paper will review the details of the most commonly used reduction techniques in a unified language focusing on coupled neural oscillators (Ermentrout, 1981, Ermentrout & Kopell, 1984, Ermentrout & Kopell, 1991, Ermentrout, 1996, Hansel et al., 1993, Hansel et al., 1995, Van Vreeswijk et al., 1994, Bressloff & Coombes, 2000, Izhikevich, 2000, Ermentrout & Kleinfeld, 2001, Lewis & Rinzel, 2003, Brown et al., 2004, Ermentrout & Kopell, 1990).

Works presented in Strogatz (1988) and later in Niebur (1991) analyzed the collective *ph-Sync* of non-linear oscillators with random intrinsic frequencies under several different coupling models in 2D lattices. Even with models different from the actual conception of a *cN*, the examples studied in Niebur (1991) are considered as an initial step to study how network complexity affects sync. In the paper a square lattice is used as a referent connectivity scheme to design three different topologies: 1) four nearest neighbors, 2) Gaussian connectivity truncated at 2σ, and finally a 3) random sparse connectivity. The study reveals that type 3) topology leads to a more rapid sync between oscillators than types 1) and 2). This is one of the most important results about sync in *cN* of oscillating network elements. Here the paper surveys the work on three different types of collections of oscillating objects: limit cycle oscillators (Kuramoto), pulse-coupled models and coupled map systems.

The Kuramoto (K-) model: The work in Winfree (1967) initiated the study on collective synchronization (*cSync)* and emphasized the need for analytical methods to investigate the problem. One of these methods, as already discussed, examines a large collection of loosely coupled, almost identical, interacting limit-cycle oscillating objects, where each object impacts a phase of the others and changes its rhythm in accordance with its sensitivity function. Even if these approximations oversimplify the problem, its nature can be captured.

When the natural frequencies of the oscillating objects are too different compared to the strength of the mutual coupling, they are unable to synchronize, and the collection of the objects remains unsynchronized. But, if the coupling is strong enough, all elements oscillate in synchronism. The transition from one regime to the other happens at a certain coupling threshold. At this point some elements lock their relative phase developing a cluster of synchronized nodes. This is referred to as the *onset of synchronization (oSync)*. By further increasing the coupling, more and more elements join the synchronized cluster, and the system finally settles in the completely *sync* state (Kuramoto, 1975)

K-model on cN: For the *K-model* on complex topologies, the paper reformulates $\dot{\theta}$ to include the connectivity $\dot{\theta} = \omega_i + \sum_j \sigma_{ij} a_{ij} \sin\left(\theta_j - \theta_i\right)$ for $(i = 1,\ \ldots,\ N)$, where σ_{ij} is the strength of the coupling between elements *i* and *j* and a_{ij} are the entries of the connectivity matrix. These relations are referred to as equations of motions.

The oSync in cN: Studies on *sync* in *cN* where each node is modeled as a K- oscillating element, are presented for Watts–Strogatz networks in Glisic (2016) and Barabasi-Albert graphs in Glisic (2016). These works study the *oSync*, with objective to model the coupling point when groups of nodes start oscillating coherently. In (Hong, 2002), oscillating elements with Gaussian distributed ω_i in a WS network with varying probability of rewiring *p* where used to study how the order parameters, defined above, change when long-range links are added. In the analysis a normalized coupling strength $\sigma_{ij} = K/\langle k \rangle$ was used, where $\langle k \rangle$ is the average degree of the graph. It was shown that *cSync* emerges even for very small values of *p*.

The results demonstrate that topologies generated by rewiring even a small fraction of links in a regular ring, can be synchronized with a finite *K*. Analysis of the same model in Watts (1999) demonstrates that the K-limit is reached when the average connectivity grows.

In (Dorogovtsev et al., n.d.) the same problem in Barabasi-Albert networks is discussed with ω_i and the initial values of θ_i uniformly distribution in the range (-1/2, 1/2) and $(-\pi,\ \pi)$, respectively. The presence of a critical point for the K-model on SF networks was not expected. Here a dynamical process shows a critical behavior when the network is described by a power- law connectivity distribution (Boccaletti et al., 2006; Dorogovtsev et al., n.d.; Newman, 2003). To find the exact value of σ_c, one can use standard finite-size scaling analysis.

Graph theoretical bounds to synchronizability: In the sync-context, graph theoretical analyses of the Laplacian matrix, of the connectivity matrix A, provides the bounds of its extreme *eigV's*. The impact of these bounds on different types of *cN* was discussed in Chung (2003), Fiedler (1973).

Relevance for this Paper: The objective of the section up to now was to survey the main works relevant to our understanding of theoretical and practical aspects of sync-processes in *cN*. In what follows, the survey will be narrowdown to the applications to specific problems relevant to this paper, such as biology and *n-Sci*, engineering, and computer science, with some comments on economy and social sciences, which are not in the focus of this survey.

Bio-systems and n-Sci: In biology, *cN* are present at different space/time scales: from the molecular level up to the population level, including many scales of bio-systems in between. In the former case the evolution of genetic networks and in the latter case the dynamics of populations of species can be analyzed.

On a different context, *n-Sci* offers applications for the *sync* of individual *sN* and for the coupling between *cort*-areas in the brain, both discussed in the previous sections of this survey. Application of the above concepts in *genetic networks* is considered in García-Ojalvo et al. (2004); Wagemakers (2006); Koseska (2007), Cir-rhythms in Strogatz (2003), neuronal networks in Binzegger et al. (2004) and *corN* of the brain in Bassett et al. (2006). For superconducting-oscillatory NN for image recognition see Cheng et al. (2023).

Computer science and engineering: cN and *sync* are important in many computers science (*compSci*) and engineering areas. In *compSci, sync* is needed for a proper functioning of distributed systems (*distrS*). The objective of the *distrS* is to establish a global common state (consensus). These systems are growing in size and their topologies are becoming increasingly complex. At the same time, some engineering problems also need coordination at the level of large-scale *cN* (distribution of information or energy).

Parallel Distributed Computation is considered in Nicol (1994), *consensus problems* in Olfati-Saber (2004), *Wireless communication networks* in Hekmat (2006) and power grid (Crucitti et al., 2004).

7. Tensor networks

7.1. Tensor networks (TN) for QML

TN's are a powerful tool for studying q-multiple-body systems (Orus,

Table 5
CrossTechology coverage (focus-Synchronization).

1	2	3	4	5	6	7	8	9	10	11
(Zhang, 2008, Hale, 1993, Lu, 2011, http, n.d.e, http, n.d.f, Gao, 2018, Wen, 2012)	sync						*A Neuroscience and Network Synchronization*			√
(Pecora, 1990, Abbasi et al., 2013, Stanoev, 2013, Karimi, 2010, Wu et al, 2013a, Xia, 2009, Yang, 2013, Lu, 2011, Yang, 2010, Zhang, Qin, 2015, Wang, 2015, Yamamoto, 2004, Yang, 2004, Ji, 2011, Boukas, 2006, Bo, 2014, Liu et al., 2009a, Wu et al., 2013b, Yang, 2012, Zhang, 2015, Karimi, 2011, Liu, 2013, Mao, 2011, Wang, 2010, Zhu, 2012, Sastry, 1989, Pariño, 2000, Zhou, 2014, Gu, 2003, Zhou, 2013, Zhang et al., 2013b, Mao, 2008, Kolmanovskii, 2003)	sync						*Complex dynamic networks*			√
(Zhang et al., n.d.)	sync						*Synchronization of neural networks with stochastic perturbation*			√
(Glisic, 2016, Watts & Strogatz, 1998, Barabasi & Albert, 1999, Guan, 2010, Li, 2014, He et al., 2014a, Wen, 2015, Lu, 2009, Wen et al., n.d.; Liu et al., 2009b, He et al., 2014b, Wen, 2013, Tang, 2012, Cai et al., 2009, Yu, 2012)	sync						*Complex networks*			√
(Zhang, 2009, McClean, 2014)	sync						*Coupled neural networks*			√
(Glisic, 2016, Danon et al., 2005, Newman & Girvan, 2004, Girvan, 2002, Strogatz, 1988, Niebur, 1991)	sync						*Large Scale Networks Synchronization*			√
(Strogatz, 1988, Niebur, 1991, Winfree, 1967, Kuramoto, 1975)	sync						*Oscillator models on complex networks*			√
(Glisic, 2016, Hong, 2002, Watts, 1999, Dorogovtsev, et al., n.d.)	sync						*Onset of synchronization in complex networks*			√
(Chung, 2003, Fiedler, 1973, Strogatz, 2003, Binzegger et al., 2004, Bassett et al., 2006, Nicol, 1994, Olfati-Saber, 2004, Hekmat, 2006, Crucitti et al., 2004)	sync						*Graph theoretical bounds to synchronizability, Biological systems and neuroscience*			√
This paper	NET	√	√	√	√	√	Cross-Tecnology survey	√	√	√

1 reference, 2 focus, 3 classic, 4 quantum, 5 complex networks, 6 tensors, 7 q-simulations, 8 contribution, 9 energy efficiency, 10 computational efficiency, 11 synchronization.

2019). With its roots in q-physics, nowadays an increased interest in adapting them to *ML* (Levine, 2019) can be seen. *TN's* have been used in various ML problems, such as dimensionality reduction [622], image recognition (Stoudenmire et al., 2016a), generative models (Han, 2018), natural language processing (Guo, 2018), anomaly detection (Wang et al., 2020a), etc. *ML* models based on *TN* have several useful features from both theoretical and practical perspectives. When used for analytical representations, their expressive power ($\mathscr{EP}$) can be modeled by the *ent*-structure of the underlying TN q-states.

This enables us to understand their applicability to a given learning assignment by analyzing the entanglement properties (Convy, 2021). TN's also provide a foundation to analyze exponential improvements certain *qL* models exhibit over their c-analogues (Gao, 2021; Gao, 2018; Levine, 2019). By using TN's, recently a separation in $\mathscr{EP}$ between Bayesian networks and their q- version has been shown to originate from q-nonlocality (Gao, 2021). In practice, numerical techniques used for TN's, Stoudenmire et al. (2016a), are also handy for optimizing and training of ML models. Several open-source libraries have been released, which have supported and will continue to help the work on TN based ML. Nowadays, this research field is being extended, with significant advances. Even so, a number of important questions remain largely unanswered.

In addition, in classical ML, a serious obstacle for training ANN is so called barren plateau (*bp*) problem, causing that *G* of the *ooF* reduces exponentially with the problem size (Cichocki, 2017). The *bp* also exists for many *qL* models based on variational q-circuits and nowadays the topics are still being studied (McClean, 2018). The problem of *bp* for TN based ML models is crucial but still not enough explored topic. Here the matrix product states (MPS) based circuits are studied, which is a special case of TN in one dimension.

TN Based ML: As already indicated earlier, in classical ML, an obstacle for training ANN is the *bp problem* (Zhao, 2021). The problem also exists for many *qL models* and is still under active study (McClean, 2018; Wang et al., 2020b; Cerezo et al., 2021). By exploring different *ooF*, authors in Liu (2021) proved rigorously that *bp* arise generally for MPS-based $\mathscr{L}$ with *ooF*, making their training by *G*-based methods ineffective and the related circuits unscalable. In contrast, for *local ooF, bp* is not present and these models can be efficiently trained. Ref. (Liu, 2021) also proves that for local *ooF* the *G*-decays exponentially with the distance between the region where the local observable acts and the site that hosts the derivative parameter. This demonstrates the locality property of TN which is useful in reducing the complexity in training corresponding models. In addition, Liu (2021) shows by numerical simulations that these results hold as well for MPS-based learning models which are not too large.

ML by qTN: Authors in Liu (2018) describe an experiment with MPS to show how q-entanglement (*ent*-) can be used for $\mathscr{L}$-architecture. In the experiment, a set of images with a certain shade of grey is encoded onto the many-qubit states in a Hilbert space (Stoudenmire et al., 2016b). The classifiers of the encoded images are implemented as matrix product states (MPS). *Multiscale ent-Renormalization Ansatz (MERA)* (Liu et al., n.d.b) training algorithm is used for optimizing the MPS.

7.2. Tensor networks for complex systems optimization

Tensor ($\mathscr{T}$-) decompositions (TD) and TN's are used as an efficient tool in data analysis and data mining. Here models and pertaining algorithms for *ls*-TN are surveyed, especially decompositions of $\mathscr{T}$-Train (TT) using novel analytical and graphical representations. The survey covers the methods for creating very high-order's from lower-order original data, referred to as $\mathscr{T}$-ization, and data compression by using *quantized TT* networks. The goal here is to show how TN's are used for solving a wide class of big data (*bD*) optimization problems (not tractable otherwise) by applying $\mathscr{T}$-ization using relatively small size matrices and 's and applying iteratively optimized $\mathscr{T}$- contractions.

Big Data (*bD*) of large volume and high complexity cannot be handled by existing standard methods. *bD* is characterized not only by its large volume but also by veracity, variety, velocity, and value. High volume needs scalable algorithms; high velocity needs real-time processing of stream of data; high veracity requires techniques for noisy, incomplete and/or inconsistent data, high variety assumes the use of different types of data, e.g., binary, continuous data, images, time series, etc., and finally the *value* refers to more informative data that provide meaningful and interpretable results.

Multidimensional ($\mathscr{MD}$-) data, like multimedia signals (speech, video), and medical/biological data are present across the sciences and engineering. The analysis of *bD* requires new methods to process large datasets within tolerable time without excessive complexity. Tensors

$\mathcal{MD}$-representation of matrices, provide a natural sparse and distributed models for such data.

$\mathcal{T}$'s are used in different types of data analysis, e.g in signal and image processing, psychometric, chemometrics, biometric, q-physics/ information, q- chemistry (*Che*) and brain science (Cichocki, 2014b; Cichocki et al., 2014c; Cichocki, n.d.; Cichocki et al., 2009). $\mathcal{T}$'s are especially useful for data having simultaneously large volumes and high variety.

TD's enable some upgrades of blind source separation (BSS) (Cichocki et al., 2014c). TN's/TD's are convenient for dimensionality reduction, they can operate with missing and noisy data (Kressner, 2016). They are used for analysis of coupled blocks of big

$\mathcal{T}$'s with non-zero entries, using the map-reduce methods, and out-of-core approaches (Cichocki et al., 2014c; Wang, 2005; Suter, 2013; Phan, 2011; Lee & Cichocki, n.d.). Moreover, multiblock's can be decomposed to correlated and uncorrelated or *st*-independent components. New models and associated methods that can identify the core relations among the different's, and scale to large datasets are needed for the analysis of coupled's.

*EXAMPLE: 7*_____

Complex interactions between $\mathcal{T}$'s can be visualized by TN graphs in which $\mathcal{T}$'s are represented by nodes in the form of circles, spheres, triangular, squares, ellipses and outgoing edges (lines) emerging from a node representing a mode, way, a dimension, indices, as shown in Fig. 10 (Orus, 2012; Orus, 2013). TN diagrams are useful in visualizing TD and expressing complex analytical (multilinear) operations of contractions of $\mathcal{T}$'s.

TN's are used in q-physics, q-chemistry, and q-information, which study the ways to build a q-computer and to program it (Orus, 2012; Orus, 2013).

The benefits of multiway $\mathcal{T}$ -analysis for *bD* include:

1) Compression of big $\mathcal{MD}$-data via $\mathcal{T}$-ization and TD's of a high-order $\mathcal{T}$ into factor matrices and/or core $\mathcal{T}$'s of low-rank and low-order; 2) Executing all calculations in feasible $\mathcal{T}$ -formats (Orus, 2012; Orus, 2013); 3) Flexible distributed representations of structurally rich data; 4) Possibility to process noisy and missing data by using low-rank $\mathcal{T}$ /matrix approximations and by leveraging robustness and stability of TD algorithms; 5) A framework to incorporate various diversities or constraints in different modes or different factors (core $\mathcal{T}$'s) and thus naturally extend the standard (2-way) component analysis and blind source separation methods (Cichocki et al., 2014c) to *ls*- $\mathcal{MD}$- data;
6) TN's provide graphically represented large distributed networks and perform complex $\mathcal{T}$ -operations (i.e., $\mathcal{T}$ -contractions and reshaping) in an intuitive way and without using explicitly analytical expressions.

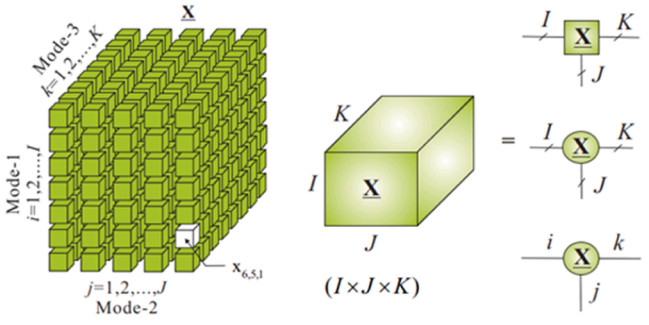


Fig. 10. A 3rd order $\mathcal{T}$ $\underline{\mathbf{X}} \in \mathbb{R}^{I\times J\times K}$with entries $x_{ijk} = \underline{\mathbf{X}}(i, j, k)$ and exemplary symbols used in tensor network diagrams. Each node in the diagram represents a $\mathcal{T}$ and each edge represents a mode or dimension. Maximum size in each mode I, J, K or running indices: i = 1, 2, …, I; j = 1, 2, …, J and k = 1, 2, …, K, is indicated (Orus, 2012).

A wide range of review and tutorial papers (Cichocki et al., 2014c; Comon et al., 2009; Lu et al., 2011; Mørup, 2011; Sidiropoulos, n.d.) and books (Cichocki, 2009) covering TD's and TN's have been available. However, they either only focus exclusively on *s*-models and/or are not explicitly linked to *bD* processing problems and/or do not explore connections to wide class of optimization problems.

This section adds to it and extends beyond the *s*-TD models like the Tucker and CPD models, with objective to demonstrate flexibilities of TN's in the optimization problems of $\mathcal{MD}$, multi-modal data, together with their role as an analytical tool for the discovery of hidden structures in large-scale (*ls*-) data (*lsD*) (Cichocki, 2009).

The objective is to survey the work on TN for *bD*, and to present methods for *ls*-TN's/TD's, together with pointing out to practical applications. In addition to the *optimization framework*, many other problems in *bD* related to anomaly detection, visualization, clustering, feature extraction and classification can also be solved by using TD and low-rank $\mathcal{T}$-approximations.

Low-Rank $\mathcal{T}$-Approximations via TN: TN's can be considered as a new *language* for *bD* TD for mimicking large complex systems (*lcS*) even with using *s*-computers (Cichocki, n.d.; Orus, 2013; Sachdev, 2009; Espig, 2011). In this context, TN can be seen as a methodology for analyzing the internal structure of high-order TD. Discussions on connections between TN's and graphical models used in ML can be found in Vidal (2003).

Reconfiguration of TN: Transformation of TN from one form to another one can be easily performed by using $\mathcal{T}$-synthesis, reshaping and basic matrix factorizations, typically using SVD (Vidal, 2003, Oseledets, 2011). The basic steps in modifying $\mathcal{T}$-structure are: 1) sequential core synthesis, 2) unfolding synthesized's into matrices, 3) performing matrix factorizations (typically, SVD) and then 4) reshaping matrices back to new core $\mathcal{T}$'s.

$\mathcal{T}$-ization: The mechanism of transforming lower-order (*l-ord*) original data into a higher-order (*h-ord*) $\mathcal{T}$ is called *tensorization*. Such $\mathcal{T}$-ization is needed in order to obtain a low-rank approximation with high level of compression. For example, big vectors, matrices even *l-ord* 's can be $\mathcal{T}$-ized to extremely *h-ord* $\mathcal{T}$'s, then compressed by using a suitable TD which is the underlying principle for *bD* analysis (Cichocki et al., 2014c; Oseledets, 2010; Khoromskij, 2011b).

Curse of Dimensionality (c-dim): The term *c-dim*, expresses the fact that in a $\mathcal{T}$ of an *N*-th- order $(I \times I \times \cdots \times I)$, the number of elements, I^N, grows exponentially with the $\mathcal{T}$ order. By using low-rank $\mathcal{T}$ structured approximations the complexity of computation is reduced and the *c-dim* relaxed or avoided (*Oseledets, 2011*; *Khoromskij, 2011a*).

Quantized TN: The c-dim can be overcome by using quantized TN, representing a $\mathcal{T}$ of high order as a set of sparsely interconnected, low dimensions (typically, 3rd-order) cores (Oseledets, 2010; Khoromskij, 2011a). The quantized TN was first proposed in Oseledets (2010) and Khoromskij (2011b).

8. *q*-Simulation

8.1. *q-Simulations and ML*

In a digital *q*- simulation (QS), the evolution of the physical (*phy*-) model is mapped, after the usual mathematical formulation of q-mechanics, onto the effective algebra of q-$\mathscr{R}$'s made of qubits. The q-time propagator, a unitary (U) operation, is then applied in steps through a sequence of q-logic gates (i.e. U operations on the qubits) defining a q-circuit (Nielsen, 2000). In this section, the survey will cover specific examples and general options for the mapping of the target system and for the translation of evolution operators into q-$\mathscr{R}$ operations. Focus will be on the class of q-spin (*sp*-) models, which besides being extremely interesting, usually constitute the ideal formal conjunction between general q-mechanical models and their corresponding representation in terms of qubits.

Sp-models are the key towards the QS of a large class of many-body q-

Table 6
CrossTechology Coverage (focus-TN).

1	2	3	4	5	6	7	8	9	10	11
(Orus, 2019, Biamonte, 2019, Cirac, 2020, Levine, 2019, Cichocki, 2014a, Cichocki, 2016, Stoudenmire et al., 2016a, Han, 2018, Wang et al., 2020a, Guo, 2018, Meichanetzidis, 2020, Convy, 2021, Lu, 2021, Gao, 2021, Levine, 2019, Abadi et al., 2016b, Cichocki, 2017, McClean, 2018)	*TN*		√		√		*Tensor Networks (TN) for QML*			
(McClean, 2018, Wang et al., 2020a, Cerezo, 2021, Liu, 2021, Zhao, 2021)	*TN*	√			√		*Tensor-network Based Machine Learning*			
(Liu, 2018, Stoudenmire et al., 2016b, Liu et al., n.d.b)	*TN*	√	√		√		*ML by Quantum Tensor Network*			
(Cichocki, 2014b, Cichocki et al., 2014c, Cichocki, Cichocki, 2009, Kressner, 2016, Wang, 2005, Suter, 2013, Phan, 2011, Lee & Cichocki, n.d.; Orus, 2012, Orus, 2013, Comon, 2009, Lu et al., 2011, Mørup, 2011, Sidiropoulos, n.d.)	*TN*			√	√		*Tensor Networks for Complex Systems Optimization*			
(Cichocki, Orus, 2013, Sachdev, 2009, Espig, 2011)	*TN*				√		*Low-Rank Tensor Approximations via Tensor Networks*			
(Vidal, 2003, Oseledets, 2011, Oseledets, 2010, Khoromskij, 2011a, 2011b)	*TN*				√		*Reconfiguration of Tensor Networks*			
(Oseledets, 2011, Khoromskij, 2011a)	*TN*				√		*Curse of Dimensionality*			
(Oseledets, 2010, Khoromskij, 2011a, 2011b)	*TN*				√		*Quantized Tensor Networks*			
This paper	NET	√	√	√	√	√	CrossTecnology survey	√	√	√

1 reference, 2 focus, 3 classic, 4 quantum, 5 complex networks, 6 tensors, 7 q-simulations, 8 contribution, 9 energy efficiency, 10 computational efficiency, 11 synchronization.

models, which are known to be c-intractable (Troyer & Wiese, 2005; Casanova et al., 2012; Barends et al., 2014; Hauke et al., 2013; Martinez et al., 2016; Klco et al., 2018). The role is emphasized of quantities that are known to be difficult to compute but extremely important in the characterization of the dynamical response (Roggero, 2018) of many body systems, such as q-correlations.

Despite being in principle much more powerful, universal qS are typically difficult to realize in practice compared to analog simulators, mainly due to the stringent requirements for general purpose q-computation (DiVincenzo, 2000). Here it should be mentioned that hybrid digital-analog qS's, combining analog approaches (easier scalability) with digital QS (intrinsic universality) (Mezzacapo et al., 2015) have also been proposed. This method might lead to universal digital--analog q-computation.

Several general purpose (Preskill, 2018) and more category- or hardware-specific (Lamata et al., 2018) accounts of the development in the field of QS can be found in the literature. The following sections will start by a summary of the theoretical foundations of digital QS, commenting on the related mathematical techniques and with a clear focus on the tools that are most often required in practical cases. Then the recent and future development of the field will be described and commented on in terms of both algorithmic procedures and experimental results. Special attention will be paid to near-term realizations of digital q- computers and QS, and particularly to those technological platforms which are currently leading the way, namely coherently manipulated *trapped ions* (Monroe & Kim, 2013; Schindler et al., 2013) and superconducting circuits working at *microwave frequencies* (Clarke & Wilhelm, 2008; Schoelkopf & Girvin, 2008, Devoret & Schoelkopf, 2013).

The fast pace of advancement will be closely followed which, in recent years, has made programmable devices available even outside research laboratories, attracting widespread interest. Indeed, established technological companies such as IBM and Google, startup ventures (Rigetti Computing and IonQ to name a few) and public institutions (Acín et al., 2018) all have designed and deployed consistent strategies in search of the long-sought goal of *q- advantage*. Such efforts promise to break the barriers currently limiting the simulations of complex many-body physics with c-computing. Without entering rigorous definitions here (Pednault et al., 2017; Boixo et al., 2018; Villalonga et al., 2019; Yamamoto, 2004), such threshold close to the size of 50 to 60 fully operational qubits can be identified. Indeed, by reasoning in orders of magnitude and by assuming that bytes are needed to store a complex number in single-precision, a 50-qubit q-$\mathscr{R}$ would in general be able to manipulate around $8 \times 2^N \sim 9{\cdot}10^{15}$ bytes of information, corresponding to roughly 9 Pb. This in turn approximates the typical amount of random-access memory in state-of-the-art supercomputers (Pednault et al., 2017; Boixo et al., 2018).

Q-advantage for scientific applications is already reached in the existing Noisy Intermediate Scale (NISQ) q-devices (Preskill, 2018), and it represents the landmark which would unequivocally certify the maturity of the field and probably its potential commercial value. Towards this direction, relevant claims of a 53-qubit superconducting *q*-hardware outperforming even the most powerful supercomputer currently available in the completion of a specific algorithmic task have recently been reported (Arute et al., 2019). While being an important result, such a device has not been used so far for the demonstration of practical use cases. A meaningful impact may be achieved if fully fault tolerant and scalable q- circuits will effectively become available (Barends et al., 2014; Schindler et al., 2011; You, 2013; Córcoles et al., 2015), where in addition to logical qubits a much larger number of auxiliary q-bits aimed at correcting noise induced errors are also used. However, it is difficult to say when the practical realization of this paradigm will be available.

Although the object here is restricted to QS of *phy-* models, it should be mentioned already at this stage that the same techniques could in principle be applied to more general computational tasks. Some complex problems in fields outside the physical sciences, such as optimization, stock market pricing (Woerner, 2019; Martin et al., 2019) and ML (Biamonte et al., 2017a) are known to have close relationships with many mathematical models in physics or engineering and might therefore benefit from speedup advantages over classical computers.

Preliminaries on QS: Mathematical description of a system evolution in time is typically formulated by using differential (d-) equations. Solving these equations is the essence of most simulation protocols nowadays. A typical example is a set of differential equations such as $d\vec{x}/dt = M\vec{x}$ where M is a matrix and $\vec{x}$ a vector of variables. Given initial values $\vec{x}(0)$, the solution is $\vec{x}(t) = e^{Mt}\vec{x}(0)$.

In q-mechanics, the equivalent example is the Schrödinger equation (Glisic & Lorenzo, 2022) (assuming $\hbar = 1$) $d|\Psi\rangle/dt = -i\mathscr{H}|\Psi\rangle$ where $\mathscr{H}$ is known as the H- operator and the associated complex linear space is the well know Hilbert space of *wF*'s. The Schrödinger equation can in principle be fully solved by using the U time-evolution operator $\mathrm{U}(t) = e^{-i\mathscr{H}t}$. Once $\mathrm{U}(t)$ is known, any starting condition can be evolved linearly as $|\Psi(t)\rangle = \mathrm{U}(t)|\Psi(0)\rangle$.

From the few examples above, it can be already seen that matrix exponentiation (*m-exp*) is a ubiquitous numerical task in simulation scenarios, and crucially in the field of q- mechanical systems. On c-computers, *m-exp* is not an easy problem whose computational

complexity is believed to scale at least polynomially with the size of the matrix (Moler, 2003). When combined with the exponential growth of the linear dimensions associated to a composite q-mechanical system with the number of sub-systems, this in turn results in general to an exponential demand of resources for manybody QS. Q-computing devices might be able to overcome such limitations in many cases of practical interest.

Here the survey focuses on system $H's$ of the form $\mathrm{H} = \sum_l^L \mathrm{H}_l$ where H_l acts only on an isolated portion of the total system. Computing the corresponding time evolution operator $\mathrm{U}(t) = \exp(-i\mathrm{H}t)$ is equivalent to the task of realizing a well-defined U transformation. As is well known (Nielsen, 2000), a q-computing device composed of a universal set of q-gates can perform any arbitrary U transformation. It was proved that $\mathrm{U}(t)$ can be efficiently obtained whenever H is a sum of local terms.

Any U-operation acting on N qubits can be realized with $O(2^{2N})$ elementary operations (Nielsen, 2000; Barenco et al., 1995). Since the Hilbert space associated with N qubits is $d = 2^N$ for given H as a sum of local terms, with say $L \propto p \cdot N$, where p measures some degree of locality and N is the total number of qubits required to encode the computation, according to the rules above, computing directly $\mathrm{U}(t) = \exp(-i\mathrm{H}t)$ in general requires $O(2^{2N})$ operations, and is therefore exponentially (*exp*-) inefficient. Now, if m_l is the dimension of the subsystem over which the action of each H_l is limited, for $m_l \ll 2^N$, since typical local terms need few- body interactions, the unitary $\mathrm{U}_l(t) = \exp(-i\mathrm{H}_l t)$ can be computed with $O(m_l^2)$ operations. Now the overall product $\widetilde{\mathrm{U}} = \prod_l \mathrm{U}_l(t)$ can be obtained on a universal q-computer by juxtaposing the circuit realizations of the single $\mathrm{U}_l(t)$-matrices with at most $O(Lm^2_{\max})$ elementary operations, where $m_{\max} = \max_l m_l$. The concluding reasoning lies in the Suzuki–Trotter (ST) decomposition: $e^{-i\sum_l \mathrm{H}_l t} = \lim_{n\to\infty}\left(\prod_l e^{-i\mathrm{H}_l t/n}\right)^n$

If all the $\mathrm{H}_l's$ commute, i.e. $[\mathrm{H}_l,\ \mathrm{H}_{l'}] = 0\ \forall l,\ l'$ the ST identity holds already for $n = 1$. In general the product of local time evolution operators will not be exactly equal to the total target $\mathrm{U}(t) = \exp(-i\mathrm{H}t)$, but $\forall n$, $\mathrm{U}(t) = e^{-i\sum_l \mathrm{H}_l t} = \left(\prod_l e^{-i\mathrm{H}_l t/n}\right)^n + O(t^2/n)$. Still, here the initial problem could be broken into smaller segments, $e^{-i\mathrm{H}_l t/n}$, which can now be realized using only a limited set of elementary gates with arbitrarily small digital error $O(t^2/n)$. For any $\varepsilon > 0$ and t, there exists an n_ε such that $\mathrm{U}(t)$ can be applied within an approximation ε in at most $n_\varepsilon L m^2_{\max}$ steps. This is polynomial in N whenever $L = \mathrm{poly}(N)$, as in the case of nearest neighbors' interactions.

A universal q-computer will be described, from now on, as a qubit-based digital q-device operating within the algebra of Pauli matrices and using a universal set of q-gates (Glisic & Lorenzo, 2022; Nielsen, 2000). Given the results presented above, the problem of QS can then be defined and run on such a machine in a few simple steps.

1) A model H of interest from a set $\mathscr{H}$ must be defined. As for any sensible physical description, $\mathscr{H}$ contains all the dynamical data characterizing the *phh-q*-system under study. The best set of variables and operators will be included in the analytical structure of the H, such as for example frequencies and couplings, spin matrices, fermionic/bosonic creation and annihilation operators or lattice-discretized q-fields.
2) The target $\mathrm{H} \subseteq \mathscr{H}$ must be mapped into its equivalent representation on the qubit Pauli algebra $\mathscr{H} \subset \mathrm{H}\ \{\sigma_\alpha\}$. This step requires a suitable encoding of the degrees of freedom of the target system into a number N of qubits. All the relevant q- mechanical states and operators must be rephrased in terms of computational basis states and Pauli matrices $\{\sigma_\alpha\}$ acting on them, thus resulting in mapped Hamiltonian H. It should be recalled here that the Pauli algebra of qubits is characterized by the following well known set of SU(2) matrices (Glisic & Lorenzo, 2022)

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{9}$$

satisfying the following commutation and anti-commutation rules $[\sigma_\alpha, \sigma_\beta] = 2i\varepsilon_{\alpha\beta\gamma}\sigma_\gamma$, $\{\sigma_\alpha, \sigma_\beta\} = 2\delta_{\alpha\beta}\mathbb{I}$, $\alpha, \beta, \gamma \in \{x, y, z\}$, $\varepsilon_{\alpha\beta\gamma}$is the Levi-Civita tensor, $\delta_{\alpha\beta}$ is the Kronecker delta and $\mathbb{I}$ is the identity matrix. The Pauli algebra is at the heart of the physical description of spin - 1/2 q- systems and therefore the mapped H will in general correspond to a model of interacting spin - 1/2 operators, such as for example the Heisenberg or Ising chain. It is easy to understand at this stage the reason why spin systems represent the ideal contact point between physical and computational q-problems. Despite being intrinsically more involved, effective mappings are also known for a large variety of interesting cases, ranging from spin $S > 1/2$ (Santini et al., 2011) to fermionic and fermionic-bosonic systems (Casanova et al., 2012; Mezzacapo et al., 2015; Santini et al., 2011) usually through the famous Jordan-Wigner transformation (Santini et al., 2011; Somma et al., 2002; Jordan, 1928; Bari, 1973). Other recent examples include lattice models related to gauge theories (Martinez et al., 2016; Klco et al., 2018) and even q- gravity studies (García-Álvarez et al., 2017). Regardless of the details of the original $\mathscr{H}$, the q- simulation will be efficient, in Lloyd's sense, whenever the corresponding H is the sum of local terms.
3) For the target H mapped onto a sum of local increments $\mathrm{H} = \sum_l \mathrm{H}_l$ one must check whether $[\mathrm{H}_l,\ \mathrm{H}_{l'}] = 0\ \forall l, l'$. If that is the case, then $e^{-i\mathrm{H}t} = \prod_l e^{-i\mathrm{H}_l t}$ with no error. It not, an appropriate number of ST steps (Trotter steps) must be chosen according to the preset precision, in such a way that, by applying above equations $e^{-i\mathrm{H}t} \simeq \left(\prod_l e^{-i\mathrm{H}_l t/n}\right)^n$. This application of the ST formula is sometimes called *Trotterization* in QS jargon, and the error which arises from the above approximation formula is also named *digitalization* or *digital* error. In pictorial terms, one might somehow compare this situation with the analog-to-digital conversion tasks performed on classical computers, where a finite set of discrete operations accessible to the machine must be used to approximate a continuum of possible signals.
4) Each local unitary $e^{-i\mathrm{H}_l t}$ (or $e^{-i\mathrm{H}_l t/n}$) must be translated into a sequence of q-gates. This is always possible with less than $O(m_l^2)$ operations and with any universal set of single- and two-qubit operations available on a general-purpose q-computer. However, no unique scheme or library exists in general to carry out such translation, as different universal set of operations are in principle all equivalent and specific choices are usually dictated by the processor architecture or by technical efficiency considerations. Once the individual sequences corresponding to the segments in the ST decomposition are known, the total q- circuit encoding the time evolution will simply be the juxtaposition of all of them, repeated in general n times.
5) Finally, the state initiation must be included at the beginning of the circuit, and an appropriate set of $\mathscr{M}$'s must be appended at the end to recover expectation values of the relevant observable quantities on the evolved q-state. Notice that these are by no means trivial tasks, since general q- states and $\mathscr{M}$'s require the ability of preparing and observing highly correlated states and properties.

While the points outlined above represent quite a general set of instructions towards the design of a QS algorithm, moving to practical implementations of such scheme usually requires a number of algorithmic and q- computational tools, which must also be adapted to the specific hardware platform on which the computation is to be performed. Once more, it should be stressed that the class of computational tasks which can be treated with the methods of q- simulations is not at all limited to actual phy-systems but extends towards any problem that is expressible in the form of Hamiltonian q- dynamics.

9. Network optimization framework

9.1. Network specification

Parameters relevant for the network optimization are

τ-optimization of processing cycle
$\tau_{a,c}$-processing time for the algorithm (a=1,2,…,A), assuming there are A different options, mainly compromising between serial and parallel processing. This includes known and new algorithms that might be discovered in the future and evaluated using this optimization framework.

At the moment, in addition to the significant speed up in the computation due to the parallelism in the operation (Google has announced a Q-computer that can perform computing 10^8 times faster than the classical one) quantum information theory offers additional advantages:

QSA algorithm like Grover's algorithm can find the maximum/minimum value of the component in the set of N entries in $\sim N^{1/2}$ iterations while the classical approach with exhaustive search would require ~N iterations. So, if for example $N=10^6$, Grover's algorithm would find the maximum (optimum value) 10^3 time faster than the classical approach.

In general, quantum approximative optimization algorithms (QAOA) can find the maximum of the combinatorial optimization problem in polynomial times so turning the NP hard problems with exponential times into faster problems with the price that the optimum value is an approximation. The compromise between the accuracy and speed up in the execution of the optimization algorithm is the design parameter.

c-computer type (c = 1,2,…, C), assuming C different computers type are available ranging from simple processor laptop/ desktop computer to near future and full size quantum computer.
τ_p-propagation time over network diameter (p = 1,2,…, P). P different options are considered. This time might be negligible when locally optimizing NN design and relevant when optimizing distributed network like routing in large size network or federated ML.
τ_e- size of the encoding slots (e = 1,2,…, E). The larger size of the slot enables encoding the amplitudes in SNN with high precision.
$\tau_{e,a}$- aggregate size of the encoding slots (e_a = 1,2,…, E_a).
τ_s-sync acquisition time (s = 1,2,…, S)
$\mathscr{J}^{A\times P\times Ea\times S\times C}$={a, p, e, s, c}-set of indices, includes all combination of the indices a, p, e, s, c
i=(a, p, e, s) given combination of indices a, p, e, s
$p_{c,p}$-network coherency probability for a given p. The larger p increases the probability that something might go wrong in the network reducing the probability of network coherency. For the faster computer the processing time is reduced reducing the probability of incoherency.

By using above notation, the optimization processing cycle can be expressed as

$$\tau(\mathrm{i}) = (\tau_a + \tau_p + \tau_{e,a})p_{c,p} + (\tau_a + \tau_p + \tau_{e,a} + \tau_s)\left(1 - p_{c,p}\right) \tag{10}$$

9.1.1. Processing cost

$N_{a,c}$-number of operations for algorithm a on a computer c
C_c -cost per operation on computer type c
$\$^c = N_{a,c}\ C_c$ - Processing cost for computer c

9.1.2. Optimization problem

$$\mathscr{P}_1 = \min_{i\in\mathscr{J}} \$^c(i) \times \tau(i)$$

$$\mathscr{P}_2 = \min_{i\in\mathscr{J}} \$^c(i);\ w.c.\ \tau\max\ \geq\ \tau$$

$$\mathscr{P}_3 = \min_{i\in\mathscr{J}} \$^c(i);\ w.c.\ \tau\max\ \geq\ \tau\ and\ \tau_e\ \geq\ \tau_e\min \tag{11}$$

10. Conclusions

It is anticipated that 6G/7G networks will continue growing and extending their demands with respect to data rates, number of users and network access points, energy efficiency, network intelligence, decision latency in network dynamic reconfiguration and control protocols, efficiency of spectrum and overall resource sharing among multiple network operators….

As already indicated in *ls-* networks, social, economic, information, technological, biological, *q-* chemistry, *n-Sci* and the brain study …, here collectively referred to as complex networks, the problem of controlling the complex systems in real time is more and more relaying on the help of artificial intelligence.

The ML algorithms, on their own, become increasingly more complex resulting in complex neural networks. In order to speed up the ML algorithms new computing technologies are considered, like *q-* computing QC, and new models for complex networks that will enable us to efficiently control/optimize the processes run on them. This paper, under the umbrella of well-established complex networks theory, provides a unified presentation of how *q-* computing, implemented on near future computers, can enable solving various problems in above disciplines that cannot be solved efficiently by using classical approaches.

The emphasis is on the commonalities in QC applications and modelling for different systems listed above. In addition to discussing the algorithms the paper reviews the pertaining implementation problems like the *N-sync* and analytical and simulation tools for the system analysis. A comprehensive survey of the work in these fields is provided resulting in a long list of references. For this reason, the paper also encloses a selected list of references in Table 8, for the first iteration of additional readings.

In the segment of classical NN the paper draws reader's attention to

Table 7
Cross-technology coverage (focus-QS).

1	2	3	4	5	6	7	8	9	10	11
(Nielsen, 2000)	*QS*		√			√	*Quantum Simulations and Machine Learning*		√	
(Santini et al., 2011; Somma et al., 2002; Jordan et al., 1928; Bari, 1973; García-Álvarez et al., 2017)	*QS*	√	√			√	*universal quantum computer*		√	
(Troyer & Wiese, 2005, Casanova et al., 2012, Barends et al., 2014, Hauke et al., 2013, Martinez et al., 2016, Klco et al., 2018, Roggero, 2018, DiVincenzo, 2000, Mezzacapo et al., 2015, Preskill, 2018, Lamata et al., 2018, Monroe & Kim, 2013, Schindler et al., 2013, Clarke & Wilhelm, 2008, Schoelkopf & Girvin, 2008, Devoret & Schoelkopf, 2013, Acín et al., 2018, Pednault et al., 2017, Boixo et al., 2018, Villalonga et al., 2019, Schindler et al., 2011, You, 2013, Córcoles et al., 2015, Woerner, 2019, Martin et al., 2019, Biamonte et al., 2017a)	*QS*	√	√			√	Spin models		√	
This paper	NET	√	√	√	√	√	CrossTecnology survey	√	√	√

1 reference, 2 focus, 3 classic, 4 quantum, 5 complex networks, 6 tensors, 7 q-simulations, 8 contribution, 9 energy efficiency, 10 computational efficiency, 11 synchronization.

Table 8
Selected readings.

Research topic	Selected references
Advances in ML	(Chen & Liu, Nov 2016, Caruana, 1997, Li et al., 2009, Argyriou et al., 2008, Altman, 1992, Caruana, 1997, Kaelbling et al., 1996)
Deep NN and federated learning	(Hershey et al., 2014, Sprechmann et al., 2013, Andrychowicz et al., 2016, O'Shea et al., 2016, Cui et al., 2019, Chen et al., 2020, Yang et al., 2019, Konecný et al., 2016a, 2016b, Furukawa et al., 2016, Agrawal & Srikant, 2000, Geyer et al., 2017, Yuan & Yu, 2014, Zhang et al., 2016, Acar et al., 2018, Aono et al., 2016, Kim et al., 2018)
Spiking neuron timing	(Lisman, 1989, Morrison et al., 2008, M. & Wang, 1998, Turrigiano & Nelson, 2004, Florian, 2007, Badoual et al., 2006)
Spiking neuron networks	(Cybenko, 1988, Poggio & Girosi, 1989, Vapnik, 1998, Thorpe et al., 2001)
Neuroscience and AI	(Hu, 2021, Arenas et al., 2008b, Bullmore & Sporns, 2009, Marblestone et al., 2016a, Dayan & Abbott, 2001, Hodgkin & Huxley, 1952, Deneve, 2017, Clopath, 2010, Curto, 2019, Whiteway, 2019, Bassett et al., 2017, Hu, 2019)
Deep learning and neuroscience	(Marblestone et al., 2016b, Andrew et al., 2013, Goodfellow et al., 2014b, Kurach et al., 2015, Baldi et al., 2015, Lillicrap et al., 2014, Liao et al., 2015, Bengio et al., 2015a, Balduzzi et al., 2014, Cui et al., 2015, Marblestone et al., Harris, 2008, Bargmann et al., 2013, Perea et al., 2009, Dayan, n.d.; George et al., 2009b, Emin et al., 2016, Bouchard et al., 2015)
Artificial quantum neuron	(Sentís et al., 2012, Tacchino, 2018, Schmidhuber, 2015, Merolla, 2014, Biamonte et al., 2017b, Neukart, 2013, Schuld, 2014, Schuld, 2015, Kapoor, 2016, Lloyd, 2013, Schuld, 2017, Lamata, 2017, Alvarez-Rodriguez et al., 2017, Otterbach, 2017, Rebentrost, 2018, Mocanu, 2018, Hu, 2018, Cao et al., 2017, Torrontegui, 2018)
Quantum Neural Networks	(Beer, 2022, Bu et al., 2021, Larocca, 2021)
Quantum Machine Learning	(Shalev-Shwartz, 2014, Hentschel, 2011, Larocca, 2022, Izquierdo, 2020)
Quantum Computational Chemistry	(Christiansen, 2012, McArdle, 2020, Lanyon et al., 2010, McClean, 2014, Babbush et al., 2015, Sugisaki, 2016)
Complexity of Quantum Chemistry Algorithms	(Babbush et al., 2018b, Kivlichan, 2019, Dominic, 2019; Szegedy, 2004; Childs, 2012)
Neuroscience and Network Synchronization	(Zhang, 2008, Lu, 2011, Wen, 2012, Stanoev, 2013, Karimi, 2010, Wu et al., 2013a, Xia, 2009, Yang, 2013, Lu, 2011, Yang, 2010, Zhang, Qin, 2015, Wang, 2015, Ji, 2011, Bo, 2014, Liu et al., 2009a, Wu et al., 2013b, Yang, 2012, Zhang, 2015, Karimi, 2011, Liu, 2013, Wang, 2010, Pariño, 2000)
Large Scale Networks Synchronization	(Strogatz, 2001, Albert & Barabasi, 2002, Watts, 1999, Fiedler, 1973)
Tensor Networks for QML	(Orus, 2019, Biamonte, 2019, Cirac, 2020, Cichocki, 2014a, Wang et al., 2020a, Lu, 2021)
Tensor Networks for Complex Systems Optimization	(Cichocki, 2014b, Cichocki et al., 2014c, Cichocki, n.d.; Cichocki, 2009, Kressner, 2016, Wang, 2005, Suter, 2013, Lee & Cichocki, n.d.; Orus, 2012, Orus, 2013, Comon, 2009, Lu et al., 2011, Sachdev, 2009, Oseledets, 2011, Khoromskij, 2011a, 2011b)
Quantum Simulations and Machine Learning	(Hauke et al., 2013, Klco et al., 2018, Roggero, 2018, Preskill, 2018, Pednault et al., 2017, Santini et al., 2011)

advances in ML. One example is *lifelong machine learning which improves performance by using previous decisions when making a decision on a new set of data.* Here the paper also surveys the work on its variants, *lifelong supervised, unsupervised and reinforcement learning including lifelong neural networks.*

Here the paper also discusses an important application of ML to simplify complex signal processing algorithms used in wireless networks. Most of these algorithms require rather high complexity, preventing real-time processing. A learning- based solution has been developed to address this issue. Here an SP algorithm is modeled as an unknown nonlinear mapping and a DNN is used to approximate its operation.

For distributed ML the survey covers works on federated learning including its privacy. *Block Chain* protection of FL has been considered as well.

Then the paper presents the work on spiking neural networks. The interest in spiking neural networks is twofold. First using spikes instead of continuous presence of signals enables several orders of magnitude better energy efficiency. Second, deep understanding of the neurological processes enables us to have better insights in the operation of the human brain which is expected to help us further develop better modelling and design of the algorithms for control of artificial neural networks.

Here the paper surveys the work on spiking neurons and spiking neural networks. The focus in this segment is on relation between the *n-Sci* and AI. The paper surveys the work on ML solutions based on the results of brain science. The latest advancements in modern AI borrow biological results from *corN,* in terms of both structure and function.

Here the authors argue that *n-Sci* and ML are developing towards convergence, since ML focuses nowadays on the optimization of objective functions, is introducing complex cost functions, not uniform across layers and time and begins to diversify the architectures that are subject to optimization.

Implementation of the above algorithms by using q- Computing (QC) is expected to minimize the complexity and increase the execution speed of the algorithms. In this segment the paper surveys the work on artificial q- neuron and q- neural networks. Here deep learning techniques do not use only fully connected architecture. The q- CV models encode problems in a number of different representations. When it comes to q- ML the paper surveys the work on Hamiltonian estimation, which is the most common operator used in q- information processing. The survey also covers the work on group-theoretic approach to QML dealing with group-invariant models (produce outputs that remain invariant under the action of any element of the symmetry group associated to the dataset. The methods can be equally applied in the case of representations of discrete groups (mathematical structures, for instance, when the q-data is invariant under a finite set of permutations).

In the evolution of communication networks from 6G to 7G further growth of our ambitions to model more sophisticated optimization processes requiring more powerful analytical tools should be expected. For these reasons, it is useful to review the work done in the field of *q-* computational chemistry where some advances in building complex *q-* models have been achieved already.

Applying *q-* computing to solve classically intractable chemistry problems generated a number of sophisticated models that can be considered to be used directly or modified in future 7G network optimization as well. This is particularly relevant for direct brain/network communication. Here the paper surveys the work on *molecular Hamiltonian,* again as a *q-* operator commonly used in *q-* chemistry.

The existence of *tD* in biological and artificial NNs, is a source of oscillations and instability. For this reason, synchronizing complex dynamic networks has become a mainstream topic in secure communications (including 6/7G), chemical reactions and biological systems. In this section the problem of synchronization is revisited by generalizing the problem to the complex networks with extremely large number of nodes. Here the paper surveys the work based on different models of complex networks, Erdös–Rényi (ER) random graph network model, scale-free (SF) networks, the Barabási-Albert (BA) model, network with community and modular structure. A survey is provided for application of the above concepts in *genetic networks, Circ-rhythms, neuronal networks* and *cor-networks* of brain. This also includes *parallel distributed computation, consensus problems*, large scale *wireless communication networks*

and power grid.

At this point the reader should become aware of the complexity of the modeling and analysis/optimization of the complex networks the paper is focusing on here. For that reason, it is argued that future 6/7G network designers should become aware of what tensor networks as a tool offer in facilitating the analysis of such networks. A work has been surveyed on how tensor networks could be used in the analysis and design of QML algorithms, the optimization of these algorithms and reconfiguration of tensor networks. Finally, as an unavoidable tool in this process, *q*- simulation tools are reviewed by surveying several papers in this field.

i) Contribution: A list of the contributions, exploiting the results from research in neuroscience and QC, is enclosed at the end of Section 1. *As a specific contribution*, paper lays out a complex network optimization framework enabling the analysis of the interrelations between different segments of the network: network diameter, type of the algorithms, quality of network synchronization, precision in signal encoding in SNN, type of the computer used for running the protocols and network coherency time.

Due to the complexity of the optimization problem, QC offers the ultimate solution to cope with the network dynamics. In addition to the significant speed up in the computation due to the parallelism in the operation (Google has announced a Q-computer that can perform computing 10^8 times faster than the classical one) quantum information theory offers additional advantages:

a) QSA algorithm like Grover's algorithm can find the maximum/minimum value of the component in the set of N entries in $\sim N^{1/2}$ iterations while the classical approach with exhaustive search would require ~N iterations. So, if for example $N = 10^6$, Grover's algorithm would find the maximum (optimum value) 10^3 times faster than the classical approach.
b) In general, quantum approximative optimization algorithms (QAOA) can find the maximum of the combinatorial optimization problem in polynomial times so turning the NP hard problems with exponential times into faster problems with the price that the optimum value is an approximation. The compromise between the accuracy and speed up in the execution of the optimization algorithm is the design parameter.

ii) Limitations of this work: The main limitation of using QC technology in future networks is the need to use centralized rather than distributed information processing making the impact of propagation delays τ_p-relevant. With the advances in distributed QC these limitations might be relaxed to some extent.

Due to the limited space, the second limitation of this paper is that it surveys the work on limited set of potential technologies to be used in 7G. It does not cover the possible advances in legacy technologies dealing with increasing data rates, improving energy efficiency, enhance connectivity, reduce data transmission latency etc.

iii) Improvements and directions for future work: Constant monitoring of the research results in neurosciences and brain studies and the work on their use to enhance AI, should continue in the future as well. Further work on the complex network optimization framework, presented in Section 9, should extend the number of network parameters included in the utility function. Joint optimization always offers better results than optimization that includes only a limited set of parameters. This is especially encouraged since the progress in the QC will continue.

When it comes to the technology coverage of the survey one can see the following: Empty columns in Tables 1 and 2 mean that none of the references in column 1 covers the given technical problem specified at the top of the column. From Tables 3–7 one can see that only our paper covers all technical problems specified at the top of the columns.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

M. Abadi, et al. 2016a. Deep learning with differential privacy. https://doi.org/10.1145/2976749.2978318.

Abadi, M., et al. (2016b). Tensor-flow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX conference on operating systems design and implementation, OSDI'16* (pp. 265–283). USA: USENIX Association.

Abbasi, A., et al. (2013). Hybrid centrality measures for binary and weighted networks. In R. Menezes, A. Evsukoff, & M. C. González (Eds.), *Complex networks*. Berlin, Germany: Springer-Verlag.

L.F. Abbott, et al. Building functional networks of spiking model neurons. 2016a. http://www.neurotheory.columbia.edu/Larry/SpikingNetworkReview.pdf.

Abbott, L. F., et al. (2016b). Building functional networks of spiking model neurons. *Nature Neuroscience, 19*(3), 350–355.

Abrams, D. S., et al. (1999). *Physical Review Letters, 83*, 5162.

Abrams, Z. (2023). How Brain Organoids Are Revolutionizing Neuroscience. *IEEE Pulse, 14*(3), 2–6.

Acín, A., et al. (2018). The quantum technologies roadmap: A European community view. *New Journal of Physics, 20*, Article 080201.

A. Acar, et al. 2018. A survey on homomorphic encryption schemes: Theory and implementation. https://doi.org/10.1145/3214303.

R. Agrawal and R. Srikant. 2000. Privacy-preserving data mining. https://doi.org/10.1145/342009.335438.

Aimone, J. B., & Misra, S. (2023). Will stochastic devices play nice with others in neuromorphic hardware?: There's more to a probabilistic system than noisy devices. *IEEE Electron Devices Magazine, 1*(2).

Akyildiz, I. F., Brunetti, F., & Blázquez, C. (2008). Nanonetworks: A new communication paradigm. *Computer Networks, 52*(12), 2260–2279.

Akyildiz, I. F., Pierobon, M., Balasubramaniam, S., & Koucheryavy, Y. (2015). The Internet of bio-nano things. *IEEE Communications Magazine, 53*(3), 32–40.

Albash, T., et al. (2018). *Reviews of Modern Physics, 90*, Article 015002.

Albert, R., & Barabasi, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics, 74*, 47–97.

Alpaydin, E. (2010). *Introduction to machine learning* (2nd ed.). The MIT Press. ISBN 026201243X,9780262012430.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*(3), 175–185.

Alvarez-Rodriguez, U., et al. (2017). Supervised quantum learning without measurements. *Scientific Reports, 7*, 13645.

Ammar, H. B., Tutunov, R., & Eaton, E. (2015). Safe policy search for lifelong reinforcement learning with sublinear regret. In *ICML*.

Andersen, R. A., Musallam, S., & Pesaran, B. (2004). Selecting the signals for a brain–machine interface. *Current Opinion in Neurobiology, 14*(6), 720–726.

Andrew M. et al. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. 2013. http://arxiv.org/abs/1312.6120.

Andrychowicz, M., et al. (2016). Learning to learn by gradient descent by gradient descent. In *Proc. adv. neural inf. process. syst.* (pp. 3981–3989).

F. Anselmi, et al., Unsupervised learning of invariant representations. http://www.sciencedirect.com/science/article/pii/S0304397515005587.

Y. Aono, et al. 2016. Scalable and secure logistic regression via homomorphic encryption. https://doi.org/10.1145/2857705.2857731.

T. Araki, et al 2016. High-throughput semi-honest secure three-party computation with an honest majority. https://doi.org/10.1145/2976749.2978331.

O. Arancio, et al. Nitric oxide acts directly in the presynaptic neuron to produce long-term potentiation in cultured hippocampal neurons. Cell, http://www.ncbi.nlm.nih.gov/pubmed/8978607.

Arenas, A., et al. (2008a). Synchronization in complex networks. *Physics Reports, 469*(3), 93–153.

Arenas, A., Diaz-Guilera, A., Kurths, J., Moreno, Y., & Zhou, C. (2008b). Synchronization in complex networks. *Physics Reports, 469*(3), 93–153.

Argyriou, A., Evgeniou, T., & Pontil, M. (2008). Convex multi-task feature learning. *Machine Learning, 73*(3), 243–272.

Artola, A., Bröcher, S., & Singer, W. (1990). Different voltage dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature, 347*, 69–72.

S. Arunachalam, et al., A survey of quantum learning theory. 2017.

Arute, F., et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature, 574*, 505–510.

Aspuru-Guzik, et al. (2005). *Science, 309*(5741), 1704.

Avena-Koenigsberger, A., et al. (2018). Communication dynamics in complex brain networks. *Nature Reviews Neuroscience, 19*(1), 17.

Azghadi, M. R., Iannella, N., Al-Sarawi, S. F., Indiveri, G., & Abbott, D. (2014). Spike-based synaptic plasticity in silicon: Design, implementation. *Application, and Challenges Proceedings of the IEEE, 102*(5), 717–737.
Azghadi, M. R., L-Barranco, B., Abbott, D., & Leong, P. H. W. (2017). A hybrid CMOS-memristor neuromorphic synapse. *IEEE Transactions on Biomedical Circuits and Systems, 11*(2), 434–445.
Babai, L. (2016). Graph isomorphism in quasi polynomial time. In *Proceedings of the forty-eighth annual ACM symposium on theory of computing* (pp. 684–697).
Babbush, et al. (2015). *Physical Review A, 91*, Article 022311.
Babbush, R., et al. (2016). *New Journal of Physics, 18*, 33032.
Babbush, R., et al. (2018a). *Physical Review X, 8*, Article 011044.
Babbush, R., et al. (2018b). *Physical Review X, 8*, Article 041015.
Babbush, R., et al. (2019). *npj Quantum Information, 5*, 92.
Badoual, M., et al. (2006). Biophysical and phenomenological models of multiple spike interactions in spike-timing dependent plasticity. *International Journal of Neural Systems, 16*, 79–97.
P. Baldi et al. The ebb and flow of deep learning: A theory of local learning. 2015, http://arxiv.org/abs/1506.06472.
D. Balduzzi, et al. Kickback cuts Backprop's redtape: Biologically plausible credit assignment in neural networks. 7, 2014. http://arxiv.org/abs/1411.6191.
Barabási, A. L., & Albert, R. (1999). Emergenge of scaling in random networks. *Science, 286*, 509–512.
Barabasi, A., & Albert, R. (1999). Emergence of scaling in random networks. *Science, 286* (5439), 509–512.
Barenco, A., et al. (1995). Elementary gates for quantum computation. *Physical Review A, 52*, 3457.
Barends, R., et al. (2014). Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature, 508*, 500.
Bargmann, C. I., et al. (2013). From the connectome to brain function. *Nature Methods, 10*(6), 483–490. https://doi.org/10.1038/nmeth.2451
Bari, R. A. (1973). Classical linear-chain hubbard model: Metal-insulator transition. *Physical Review B, 7*, 4318.
Basieva, I., Khrennikov, A., & Ozawa, M. (2021). Quantum-like modeling in biology with open quantum systems and instruments. *Bio Systems, 201*, Article 104328.
Bassett, D. S., et al. (2006). Small-world brain networks. *The Neuroscientist, 12*, 512–523.
Bassett, D. S., et al. (2017). Network neuroscience. *Nature Neuroscience, 20*, 353–364.
Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research, 12*, 149–198.
Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences, 2*(1), 183–202.
K. Beer, et al., Training quantum neural networks on NISQ devices. 2021.
K. Beer, Quantum neural networks, arXiv:2205.08154v1 [quant-ph] 17 May 2022.
Beinert, H., et al. (1997). *Science, 277*, 653.
Ben-David, S., & Schuller, R. (2003). Exploiting task relatedness for multiple task learning. In *COLT*.
Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (pp. 41–48). ACM.
Y. Bengio, et al. Towards biologically plausible deep learning. 2015a. http://arxiv.org/abs/1502.04156.
Bergstra, J., et al. (2010). A CPU and GPU math compiler in Python. In *, 1. Proc. 9th Python in science conference*.
D.W. Berry, et al. Optimal input states and feedback for interferometric phase estimation. 2001. https://link.aps.org/doi/10.1103/PhysRevA.63.053804.
Berry, D. W., et al. (2018). *npj Quantum Information, 4*, 22.
D.W. Berry et al. Optimal states and almost optimal adaptive measurements for quantum interferometry. https://link.aps.org/doi/10.1103/PhysRevLett.85.5098.
Bertels, L. W., Grimsley, H. R., Economou, S. E., Barnes, E., & Mayhall, N. J. (2023). Symmetry breaking slows convergence of the ADAPT variational quantum eigensolver. In *2023 IEEE international conference on quantum computing and engineering (QCE)*. IEEE. Conference Paper, Publisher.
Biamonte, J., et al. (2017a). Quantum machine learning. *Nature, 549*, 195.
Biamonte, J., et al. (2017b). Quantum machine learning. *Nature, 549*, 195–202.
J. Biamonte, Lectures on quantum tensor networks, arXiv:1912.10049 (2019).
Binzegger, T., et al. (2004). A quantitative map of the circuit of cat primary visual cortex. *The Journal of Neuroscience, 24*, 8441–8453.
Bliss, T. V. P., & Lomo, T. (1973). Long-lasting potentation of synaptic transmission in the dendate area of anaesthetized rabbit following stimulation of the perforant path. *The Journal of Physiology, 232*, 331–356.
Bloch, F. (1929). Über die quantenmechanik der elektronen in kristallgittern. *Zeitschrift für Physik, 52*, 555.
Blunt, N. S., Gehér, G. P., & Moylett, A. E. (2023). Compiling a simple chemistry application to quantum error correction primitives. In *2023 IEEE international conference on quantum computing and engineering (QCE)*. Conference Paper, Publisher: IEEE, arXiv:2307.03233v1 [quant-ph] 6 Jul 2023.
Bo, H., et al. (2014). General observer-based controller design for singular Markovian jump systems. *International Journal of Innovative Computing, Information and Control, 10*(5), 1897–1913.
Boccaletti, S., et al. (2006). Complex networks: Structure and dynamics. *Physics Reports, 424*, 175–308.
D. Bogdanov, et al. 2008. Sharemind: A framework for fast privacy-preserving computations. https://doi.org/10.1007/978-3-540-88313-5_13.
Bohte, S. M., Kok, J. N., & La Poutre, H. (2002). Spike-prop: Error backpropagation in multi-layer networks of spiking neurons. *Neurocomputing, 48*, 17–37.
Boixo, S., et al. (2018). Characterizing quantum supremacy in near-term devices. *Nature Physics, 14*, 595–600.
G. Bouchard, et al. Accelerating stochastic gradient descent via online learning to sample. arXiv preprint arXiv:1506.09016, 2015.
Boukas, E.-K. (2006). *Stochastic switching systems: Analysis and design*. Berlin, Germany: Springer-Verlag.
Bourdoukan, R., et al. (2015). Enforcing balance allows local supervised learning in spiking recurrent networks. *Advances in neural information processing systems* (pp. 982–990).
Branicky, M. S., et al. (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control, 43*(1), 31–45.
Breakspear, M. (2017). Dynamic models of large-scale brain activity. *Nature Neuroscience, 20*(3), 340–352.
H. Brendan et al. Federated learning of deep networks using model averaging. http://arxiv.org/abs/1602.05629.
Bressloff, P. C., & Coombes, S. (2000). Dynamics of strongly coupled spiking neurons. *Neural Computation, 12*(1), 91–129.
Brown, E., Moehlis, J., & Holmes, P. (2004). On the phase reduction and response dynamics of neural oscillator populations. *Neural Computation, 16*(4), 673–715.
K. Bu, et al., On the statistical complexity of quantum circuits. 2021.
Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys and Tutorials, 18*(2), 1153. Second Quarter.
Bullmore, E., & Sporns, O. (2009). Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience, 10*(3), 86–98.
Córcoles, A., et al. (2015). Demonstration of a quantum error detection code using a square lattice of four superconducting qubits. *Nature Communications, 6*, 6979.
Cai, S., et al. (2009). Periodically intermittent controlling complex dynamical networks with time-varying delays to a desired orbit. *Physics Letter A, 373*(42), 3846–3854.
Campbell, E. (2019). *Physical Review Letters, 123*, Article 070503.
Y. Cao, et al., Quantum neuron: An elementary building block for machine learning on quantum computers, arXiv:17110.11240 (2017).
Caruana, R. (1997). Multitask learning. *Machine Learning, 28*(1), 41–75.
Casanova, J., et al. (2012). Quantum simulation of interacting fermion lattice models in trapped ions. *Physical Review Letters, 108*, Article 190502.
Cassandras, C. G. (2014). The event-driven paradigm for control, communication and optimization. *Journal of Control and Decision, 1*(1), 3–17.
Cerezo, M., et al. (2021). Higher order derivatives of quantum neural networks with barren plateaus. *Quantum Science and Technology, 6*, Article 035006.
Chen, Z., & Liu, B. (2014). Mining topics in documents: Standing on the shoulders of big data. In *In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '14)* (pp. 1116–1125). New York, NY, USA: Association for Computing Machinery.
Chen, Z., & Liu, B. (2016). *Lifelong machine learning*. Morgan & Claypool Publishers.
Chen, J., Tang, L., Liu, J., & Ye, J. (2009). A convex formulation for learning shared structures from multiple tasks. In *ICML* (pp. 137–144).
Chen, C. L. P., et al. (Apr. 2019). Universal approximation capability of broad learning system and its structural variations. *IEEE Transactions on Neural Networks and Learning Systems, 30*(4), 1191–1204.
Chen, X., et al. (2020). Age of information aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective. *IEEE Transactions on Wireless Communications, 19*(4).
Cheng, R., Kirst, C., & Vasudevan, D. (2023). Superconducting-oscillatory neural network with pixel error detection for image recognition. *IEEE Transactions on Applied Superconductivity, 33*(5), 1–8.
Childs, A. M., et al. (2012). *Quantum Information and Computation, 12*, 901.
Childs, A. M., et al. (2018). *Proceedings of the National Academy of Sciences, 115*, 9456.
Christiansen, O. (2012). Physical chemistry. *Chemical Physics, 14*(19), 6672.
Chung, F., et al. (2003). Spectra of random graphs with given expected degrees. *Proceedings of the National Academy of Sciences of the United States of America, 100*, 6313–6318.
Cichocki, A., et al. (2009). *Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation*. Chichester: Wiley.
A. Cichocki, et al., Tensor decompositions for signal processing applications from two-way to multiway component analysis, arXiv:1403.4462v1 [cs.NA] 17 Mar 2014.
A. Cichocki, Tensor networks for big data analytics and largescale optimization problems, arXiv:1407.3124 (2014a).
A. Cichocki, Tensor Networks for Big Data Analytics and Large-Scale Optimization Problems, arXiv:1407.3124v2 [cs.NA] 22 Aug 2014.
Cichocki, A., et al. (2016). Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends in Machine Learning, 9*, 249.
Cichocki, A., et al. (2017). Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Foundations and Trends in Machine Learning, 9*, 431–673.
A. Cichocki, "Era of big data processing: A new approach via tensor networks and tensor decompositions," http://arxiv.org/abs/1403.2048.
I. Cirac, et al., Matrix product states and projected entangled pair states: Concepts, symmetries, and theorems, arXiv:2011.12127 (2020).
Clarke, J., & Wilhelm, F. K. (2008). Superconducting quantum bits. *Nature, 453*, 1031–1042.
Clopath, C., et al. (2010). Connectivity reflects coding: A model of voltage-based spike-timing-dependent plasticity with homeostasis. *Nature Neuroscience, 13*, 344–352.
Comon, P., et al. (2009). Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics, 23*, 393–405.
Connors, B. W., et al. (2007). *Neuroscience: Exploring the brain* (3rd ed.). Baltimore, MD, USA: Lippincott Williams and Wilkins.

I. Convy, et al., Mutual information scaling for tensor network machine learning, arXiv: 2103.00105 (2021).
Cooper, L., Intrator, N., Blais, B., & Shouval, H. Z. (2004). *Theory of cortical plasticity*. Singapore: World Scientific.
Coventry, B. S., & Bartlett, E. L. (2023). Closed-loop reinforcement learning based deep brain stimulation using SpikerNet: A computational model. In *2023 11th international IEEE/EMBS conference on neural engineering (NER)*. IEEE. Conference Paper, Publisher.
Crucitti, P., Latora, V., & Marchiori, M. (2004). A topological analysis of the Italian electric power grid. *Physica A, 338*, 92–97.
Y. Cui, et al. Continuous online sequence learning with an unsupervised neural network model. 2015. http://arxiv.org/abs/1512.05463.
Cui, L., Yang, S., Chen, F., et al. (2018). A survey on application of machine learning for Internet of Things. *International Journal of Machine Learning and Cybernetics, 9*, 1399–1417. https://doi.org/10.1007/s13042-018-0834-5
Cui, W., Shen, K., & Yu, W. (2019). Spatial deep learning for wireless scheduling. *IEEE Journal on Selected Areas in Communications, 37*(6), 1248.
Curto, C., et al. (2019). Relating network connectivity to dynamics: Opportunities and challenges for theoretical neuroscience. *Current Opinion in Neurobiology, 58*, 11–20.
Cybenko, G. (1988). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems, 2*, 303–314.
Danon, L., Díaz-Guilera, A., Duch, J., & Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics, 9*, 219–228.
Daumé, H., III (2009). Bayesian multitask learning with latent hierarchies. In *UAI* (pp. 135–142).
David, O., et al. (2006). Dynamic causal modeling of evoked responses in EEG and MEG. *NeuroImage, 30*(4), 1255–1272.
Dawson, C.M., et al. (2005), arXiv:quantph/0505030.
Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience: Computational and mathematical modeling of neural systems*. Cambridge, MA, USA: MIT Press.
P. Dayan. Twenty-five lessons from computational neuromodulation. http://www.sciencedirect.com/science/article/pii/S0896627312008628.
Deneve, S., et al. (2017). The brain as an efficient and robust adaptive learner. *Neuron, 94*(5), 969–977.
Devoret, M. H., & Schoelkopf, R. J. (2013). Superconducting circuits for quantum information: An outlook. *Science, 339*, 1169.
DiVincenzo, D. P. (2000). The physical implementation of quantum computation. *Fortschritte der Physik, 48*, 771.
Dolgov, A., & Kurkin, S. (2023). In *2023 7th scientific school dynamics of complex networks and their applications (DCNA)*. IEEE.
Dolk, V. S., et al. (2017). Event triggered control systems under denial-of-service attacks. *IEEE Transactions on Control of Network Systems, 4*(1), 93–105.
Dominic W. et al., Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization, arXiv:1902.02134v4 [quant-ph] 27 Nov 2019.
S.N. Dorogovtsev, et al., Critical phenomena in complex networks, Reviews of Modern Physics.
Du, W., et al. (2004). Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *SDM*.
Du, Y., et al. (2020). Expressive power of parametrized quantum circuits. *Physical Review Research*.
Du, J., et al. (2010). *Physical Review Letters, 104*, Article 030502.
Ekman, M., et al. (2012). Predicting errors from reconfiguration patterns in human brain networks. *Proceedings of the National Academy of Sciences of the United States of America, 109*(41), 6714–16719.
A. Emin et al. The inevitability of probability: Probabilistic inference in generic neural networks trained with non-probabilistic feedback. 2016. http://arxiv.org/abs/1601.03060.
Erdöos, P., & Rényi, A. (1959). On random graphs. *Publicationes Mathematicae Debrecen, 6*, 290–297.
Ermentrout, G., & Kleinfeld, D. (2001). Traveling electrical waves in cortex: Insights from phase dynamics and speculation on a computational role. *Neuron, 29*(1), 33–44.
Ermentrout, G. B., & Kopell, N. (1984). Frequency plateaus in a chain of weakly coupled oscillators, I. *SIAM Journal on Mathematical Analysis, 15*(2), 215–237.
Ermentrout, G., & Kopell, N. (1990). Oscillator death in systems of coupled neural oscillators. *SIAM Journal on Applied Mathematics, 50*(1), 125–146.
Ermentrout, G., & Kopell, N. (1991). Multiple pulse interactions and averaging in systems of coupled neural oscillators. *Journal of Mathematical Biology, 29*(3), 195–217.
Ermentrout, G. B. (1981). n:m Phase-locking of weakly coupled oscillators. *Journal of Mathematical Biology, 12*(3), 327–342.
Ermentrout, B. (1996). Type I membranes, phase resetting curves, and synchrony. *Neural Computation, 8*(5), 979–1001.
Espig, M. (2011). Optimization problems in contracted tensor networks. *Computing and Visualization in Science, 14*(6), 271–285.
Evgeniou, T., & Pontil, M. (2004). Regularized Multi–task Learning. In *KDD* (pp. 109–117).
Farhi, E., J. Goldstone, S. Gutmann, and M. Sipser (2000), arXiv:quant-ph/0001106.
Fatima, M., & Pasha, M. (2017). Survey of machine learning algorithms for disease diagnostic. *Journal of Intelligent Learning Systems and Applications, 9*, 1–16. http://www.scirp.org/journal/jilsa.
Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal, 23*, 298–305.
G.T. Finnerty et al. Time in cortical circuits, 2015. https://www.jneurosci.org/content/35/41/13912.full.
Florian, R. V. (2007). Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural Computation, 19*(6), 1468–1502. https://doi.org/10.1162/neco.2007.19.6.1468
Fodor, J. D., et al. (2002). Understanding stimulus poverty arguments. *The Linguistic Review, 18*(1-2), 105–145.
A.G. Fowler, et al., arXiv:1808.06709 (2018).
Frady, E. P., & Sommer, F. T. (2019). Robust computation with rhythmic spike patterns. *Proceedings of the National Academy of Sciences of the United States of America, 116*(36), 18050–18059.
Frey, M. (1997). Synaptic tagging and long-term potentiation. *Nature, 385*, 533–536.
R. Frigerio, Bounded cohomology of discrete groups, arXiv:1610.08339v2 [math.AT] 3 Nov 2016.
Fultz, N. E., et al. (2019). Coupled electrophysiological, hemodynamic, and cerebrospinal fluid oscillations in human sleep. *Science, 366*(6465), 628–631.
J. Furukawa, et al. 2016. High-throughput secure three-party computation for malicious adversaries and an honest majority. 2016. https://eprint.iacr.org/2016/944.
Gülcehre, C., et al. (2016). Knowledge matters: Importance of prior information for optimization. *Journal of Machine Learning Research, 17*(8), 1–32. http://jmlr.org/papers/v17/gulchere16a.html.
Gaitan, F., et al. (2014). Graph isomorphism and adiabatic quantum computing. *Physical Review A, 89*, Article 022342.
Gammelmark, S., et al. (2009). Quantum learning by measurement and feedback. *New Journal of Physics, 113*, Article 033017.
Gao, Q., et al. (2014). A new design of robust h sliding mode control for uncertain stochasti T–S fuzzy time-delay systems. *IEEE Transactions on Cybernetics, 44*(9), 1556–1566.
Gao, J., et al. (2016). Universal resilience patterns in complex networks. *Nature, 530*(7590), 307–312.
Gao, X., et al. (2018). A quantum machine learning algorithm based on generative models. *Science Advances, 4*, eaat9004.
X. Gao, et al., Enhancing generative models via quantum correlations, arXiv:2101.08354 (2021).
García-Álvarez, L., et al. (2017). Digital quantum simulation of minimal AdS /CFT. *Physical Review Letters, 119*, Article 040501.
García-Ojalvo, J., Elowitz, M., & Strogatz, S. (2004). Modeling a synthetic multicellular clock: Repressilators coupled by quorum sensing. *Proceedings of the National Academy of Sciences of the United States of America, 101*, 10955–10960.
George, D., et al. (2009a). Towards a mathematical theory of cortical micro-circuits. *PLoS Computational Biology, 5*(10), Article e1000532. https://doi.org/10.1371/journal.pcbi.1000532. ISSN 1553-7358.
D. George et al. Towards a mathematical theory of cortical micro-circuits. 2009b. doi: 10.1371/journal.pcbi. 1000532.
Gershman, S. J., et al. (2012). The successor representation and temporal context. *Neural Computation, 24*(6), 1553–1568.
Gerstner, W., et al. (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge, U.K.: Cambridge Univ. Press.
R.C. Geyer, et al. 2017. Differentially private federated learning: A client level perspective. http://arxiv.org/abs/1712.07557.
I. Giacomelli, et al. 2017. https://eprint.iacr.org/2017/979. https://doi.org/10.1109/SP.2016.30.
Gidney, C., et al. (2019). *Quantum, 3*, 135.
Giovannetti, V., et al. (2008). Quantum random access memory. *Physical Review Letters, 100*, Article 160501.
V. Giovannetti, et al., Advances in quantum metrology. 2011. http://dx.doi.org/10.1038/nphoton.2011.35.
Girvan, M., et al. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America, 99*, 7821.
Glisic, S., & Lorenzo, B. (2022). *Artificial intelligence and quantum computing for advanced wireless communications*. Wiley.
Glisic, S. (2016). *Advanced wireless networks: Technology and business models*. John Wiley & Sons.
Glisic, S. (2023a). *Wireless quantum networks: Intelligent continuous variable technology*. John Wiley.
Glisic, S. (2023b). *Quantum vs post quantum security: Algorithms and design technology*. John Wiley.
Goebel, R., et al. (2009). Hybrid dynamical systems. *IEEE Control Systems Magazine, 29*(2), 28–93.
Goodfellow, I. J., et al. (2014). Generative adversarial nets. *Proceedings advances in neural information processing systems* (pp. 1–9).
I.J. Goodfellow, et al., Generative Adversarial Networks. 2014a. http://arxiv.org/abs/1406.2661.
I.J. Goodfellow, et al. Qualitatively characterizing neural network optimization problems. 2014b. http://arxiv.org/abs/1412.6544.
Goodfellow, I., et al. (2016). *Deep learning, 1*. Cambridge: MIT Press.
Gopnik, A., et al. (2000). *The scientist in the crib: What early learning tells us about the mind*. Harper Paperbacks.
Gottesman, D., et al. (2001). Encoding a qubit in an oscillator. *Physical Review A, 64*, Article 012310.
Graupner, B. (2007). STDP in a bistable synapse model based on CaMKII and associated signaling pathways. *Public Library of Science Computational Biology, 3*(11), e221.
A. Graves, et al. Neural Turing Machines. ArXiv, 2014. http://arxiv.org/abs/1410.5401 [20].
Gregor, K., & LeCun, Y. (2010). Learning fast approximations of sparse coding. In *Proc. 27th int. conf. mach. learn* (pp. 399–406).
K. Gu, et al., Stability of Time-Delay Systems. Boston, MA, USA: Birkhäuser, 2003.
Guan, Z.-H., & Chen, G. (1999). On delayed impulsive Hopfield neural networks. *Neural Networks, 12*(2), 273–280.

Guan, Z.-H., Hu, B., & Shen, X. (2019). *Introduction to hybrid intelligent networks*. Cham, Switzerland: Springer.
Guan, Z. (2010). Synchronization of complex dynamical networks with time-varying delays via impulsive distributed control. *IEEE Transactions on Networks and Learning Systems, 57*(8), 2182–2195.
Guo, C., et al. (2018). Matrix product operators for sequence-to-sequence learning. *Physical Review E, 98*, Article 042114.
Hale, J. K., et al. (1993). *Introduction to functional differential equations*. New York, NY, USA: Springer-Verlag.
Han, Z.-Y., et al. (2018). Unsupervised generative modeling using matrix product states. *Physical Review X, 8*, Article 031012.
Hansel, D., Mato, G., & Meunier, C. (1993). Phase dynamics for weakly coupled Hodgkin–Huxley neurons. *Europhysics Letters, 23*(5), 367.
Hansel, D., Mato, G., & Meunier, C. (1995). Synchrony in excitatory neural networks. *Neural Computation, 7*(2), 307–337.
Harris, K. D. (2008). Stability of the fittest: Organizing learning through retroaxonal signals. *Trends in Neurosciences, 31*(3), 130–136. http://www.sciencedirect.com/science/article/pii/S0166223608000180.
M.E. Hasselmo. The role of acetylcholine in learning and memory. Current Opinion in Neurobiology, doi:10.1016/j.conb.2006.09.002.
Hauke, P., et al. (2013). Quantum simulation of a lattice schwinger model in a chain of trapped ions. *Physical Review X, 3*, Article 041018.
He, X., et al. (2014a). Neural network for solving convex quadratic bilevel programming. *Neural Networks, 51*(3), 17–25.
He, X., et al. (2014b). A Recurrent neural network for solving bilevel linear programming problem. *IEEE Transactions on Neural Networks and Learning Systems, 25*(4), 824–830.
Hebb, D. O. (1949). *The organization of behavior*. New York: Wiley.
Hekmat, R. (2006). *Ad-hoc networks: Fundamental properties and network topologies*. Berlin, Germany: Springer.
A. Hentschel et al., Efficient algorithm for optimizing adaptive quantum metrology processes. 2011. https://link.aps.org/doi/10.1103/PhysRevLett.107.233601.
D.A. Herman, C. Googin, X. Liu, et al., A survey of quantum computing for finance arXiv: 2201.02773v4 [quant-ph] 27 Jun 2022.
J.R. Hershey, J. Le Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," 2014, arXiv:1409.2574.
G.E. Hinton, et al. Transforming auto-encoders. 2011. http://link.springer.com/chapter/10.1007/978-3-642-21735-7_6.
S. Hochreiter et al. Long short-term memory. Neural computation, http://www.ncbi.nlm.nih.gov/pubmed/9377276.
Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology, 117*(4), 500–544.
Hong, H., et al. (2002). Synchronization on small-world networks. *Physical Review E, 65*, Article 026139.
https://en.wikipedia.org/wiki/Chemical_synapse.
https://en.wikipedia.org/wiki/Discrete_group.
https://www.geeksforgeeks.org/mathematics-graph-isomorphisms-connectivity/.
https://en.wikipedia.org/wiki/Molecular_Hamiltonian.
https://encyclopediaofmath.org/wiki/Neutral_differential_equation.
https://encyclopediaofmath.org/wiki/Differential_equations,_ordinary,_with_distributed_arguments.
Hu, B., et al. (2019). Multistability of delayed hybrid impulsive neural networks with application to associative memories. *IEEE Transactions on Neural Networks and Learning Systems, 30*(5), 1537–1551.
Hu, B., et al. (2021). Neuroscience and network dynamics toward brain-inspired intelligence. *IEEE Transactions on Cybernetics*. Pre-Print.
Hu, B., Guan, Z. H., Chen, G., & Chen, C. L. P. (2022). Neuroscience and network dynamics toward brain-inspired intelligence. *IEEE Transactions on Cybernetics, 52*(10).
Hu, C., et al., Exponential stabilization and synchronization of neural networks with time-varying delays via periodically intermittent control. Nonlinearity, 23(10), 2369.
Hu, W. (2018). Towards a real quantum neuron. *Natural Science, 10*, 99–109.
D. Huh and T.J. Sejnowski. Gradient descent for spiking neural networks. arXiv preprint arXiv:1706.04698, 2017.
Ibrahim, M., Bronn, N. T., & Byrd, G. T. (2023). Crosstalk-based parameterized quantum circuit approximation. In *2023 IEEE international conference on quantum computing and engineering (QCE)*. IEEE. | Conference Paper | Publisher:.
Izhikevich, E. M. (2000). Phase equations for relaxation oscillators. *SIAM Journal on Applied Mathematics, 60*(5), 1789–1804.
Izquierdo, Z. G., et al. (2020). Discriminating non isomorphic graphs with an experimental quantum annealer. *Physical Review A, 102*, Article 032622.
Jacob, L., Vert, J.-P., & Bach, F. R. (2009). Clustered multi-task learning: A convex formulation. In *NIPS* (pp. 745–755).
Jarzyna, M., et al. (2015). True precision limits in quantum metrology. *New Journal of Physics, 17*(1), Article 013010. http://stacks.iop.org/1367-2630/17/i=1/a=013010.
Ji, D. H., et al. (2011). Synchronization of neutral complex dynamical networks with coupling time-varying delays. *Nonlinear Dynamics, 65*(4), 349–358.
Jia, Y., & Zhao, L. (2023). Implantable medical devices for wireless optical neuromodulation and neural recording: Energy-efficient integrated circuit and system design. *IEEE Solid-State Circuits Magazine, 15*(4).
Jia, Y., et al. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on multimedia* (pp. 675–678). New York: ACM.
Z. Jiang, et al., arXiv:1812.08190 (2018).
Johansson, J., et al. (2013). QuTiP 2: A Python framework for the dynamics of open quantum systems. *Computer Physics Communications, 1240*.
Jordan, P., et al. (1928). Über das paulische äquivalenzverbot. *Zeitschrift für Physik, 47*, 631–651.
K.P. Körding et al. A learning rule for dynamic recruitment and decorrelation. http://www.sciencedirect.com/science/article/pii/S089360809000088X.
K.P. Körding et al. Supervised and unsupervised learning with two sites of synaptic integration. http://link.springer.com/article/10.1023/A:1013776130161.
Körding, K. (2007). Decision theory: What ``should'' the nervous system do? *Science*. http://science.sciencemag.org/content/318/5850/606.short.
Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research, 4*(1), 237–285.
Kapoor, A., et al. (2016). Quantum perceptron models. *Advances in Neural Information Processing Systems (NIPS 2016), 29*, 3999–4007.
D. Kappel, et al. STDP installs in Winner-Take-All circuits an online approximation to hidden Markov model learning. doi:10.1371/journal.pcbi.1003511.
Karimi, H. R., et al. (2010). New delay-dependent exponential H∞ synchronization for uncertain neural networks with mixed time delays. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 40*(1), 173–185.
Karimi, H. R. (2011). Robust delay-dependent H∞ control of uncertain time-delay systems with mixed neutral, discrete, and distributed time-delays and Markovian switching parameters. *IEEE Transactions on Circuits and Systems I: Regular Papers, 58*(8), 1910–1923.
Kelso, S. R., Ganong, A. H., & Brown, T. H. (1986). Hebbian synapses in hippocampus. *Proceedings of the National Academy of Sciences of the United States of America, 83*, 5326–5330.
Khoromskij, B. (2011a). Tensors-structured numerical methods in scientific computing : Survey on recent advances. *Chemometrics and Intelligent Laboratory Systems, 110*(1), 1–19. Available: http://www.mis.mpg.de/de/publications/preprints/2010/prepr2010-21.html.
Khoromskij, B. (2011b). O(d log N)-quantics approximation of N-d tensors in high-dimensional numerical modeling. *Constructive Approximation, 34*(2), 257–280.
N. Kilbertus, et al., 2018. Blind justice: Fairness with encrypted sensitive attributes. http://proceedings.mlr.press/v80/kilbertus18a.html, 2630–2639.
Killoran, N., et al. (2019). Continuous-variable quantum neural networks. *Physical Review Research, 1*, Article 033063.
M. Kim, et al. 2018. Secure logistic regression based on homomorphic encryption: Design and evaluation. https://doi.org/10.2196/medinform.8805.
A.Y. Kitaev, arXiv:quant-ph/9511026 (1995a).
Kitaev, A.Y. (1995), Preprint at http://arxiv.org/abs/quant-ph/9511026.
I.D. Kivlichan, et al., arXiv:1902.10673 (2019).
Kivlichan, et al. (2019a), arXiv:1902.10673.
Klco, N., et al. (2018). Quantum-classical computation of Schwinger model dynamics using quantum computers. *Physical Review A, 98*, Article 032331.
Kobler, J., et al. (2012). *The graph isomorphism problem: Its structural complexity*. Springer Science & Business Media.
Kolmanovskii, V., et al. (2003). Neutral stochastic differential delay equations with Markovian switching. *Stochastic Analysis and Applications, 21*(4), 819–847.
J. Konecný, H.B. McMahan, D. Ramage, and P. Richtárik. 2016a. Federated optimization: Distributed machine learning for on-device intelligence. http://arxiv.org/abs/1610.02527.
J. Konecný, et al. 2016b. Federated learning: Strategies for improving communication efficiency. http://arxiv.org/abs/1610.05492.
Koseska, A., et al. (2007). Inherent multistability in arrays of autoinducer coupled genetic oscillators. *Physical Review E, 75*, Article 031916.
D. Kressner, et al,"Low-rank tensor completion by Riemannian optimization," arXiv: 1605.08257v1 [cs.LG] 26 May 2016.
K. Kurach, et al. Neural random-access machines. 13, 2015. http://arxiv.org/abs/1511.06392.
Kuramoto, Y. (1975). Self-entrainment of a population of coupled nonlinear oscillators. In H. Araki (Ed.), *Lecture Notes in Physics: 39. International symposium on mathematical problems in theoretical physics* (pp. 420–422). New York, NY,USA: Springer.
Lamata, L., et al. (2018). Digital-analog quantum simulations with superconducting circuits. *Advances in Physics: X, 3*, Article 1457981.
Lamata, L. (2017). Basic protocols in quantum reinforcement learning with superconducting circuits. *Scientific Reports, 7*, 1609.
Lanyon, B. P., et al. (2010). *Nature Chemistry, 2*(2), 106.
M. Larocca, et al., Group-invariant quantum machine learning, arXiv:2205.02261v1 [quant-ph] 4 May 2022.
M. Larocca, et al., Theory of overparametrization in quantum neural networks. 2021.
Laughlin, S. B., & Sejnowski, T. J. (2003). Communication in neuronal networks. *Science, 301*(5641), 1870–1874.
Lebedev, M. A., & Nicolelis, M. A. L. (2006). Brain–machine interfaces: Past, present and future. *TRENDS in Neurosciences, 29*(9), 536–546.
Lebedev, M. A., & Nicolelis, M. A. L. (2017). Brain-machine interfaces: From basic science to neuroprostheses and neurorehabilitation. *Physiological Reviews, 97*(2), 767–837.
Lebedev, M. (2014). Brain-machine interfaces: An overview. *Translational Neuroscience, 5*(1), 99–110.
LeCun, Y., et al. (2015). Deep learning. *Nature, 521*(7553), 436–444.
Y. LeCun, et al. Deep learning. http://dx.doi.org/10.1038/nature14539.
LeCun, Y., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation, 1*, 541.
N. Lee and A. Cichocki, "Fundamental tensor operations for large-scale data analysis in tensor train formats," http://arxiv.org/abs/1405.7786v2.

Levine, Y., et al. (2019). Quantum entanglement in deep learning architectures. *Physical Review Letters, 122*, Article 065301.
Lewis, T. J., & Rinzel, J. (2003). Dynamics of spiking neurons connected by both inhibitory and electrical coupling. *Journal of Computational Neuroscience, 14*(3), 283–309.
Li, K., & Malik, J. (2017). Learning to optimize. In *Proc. int. conf. learn. represent*.
Li, H., Liao, X., & Carin, L. (2009). Multi-task reinforcement learning in partially observable stochastic environments. *The Journal of Machine Learning Research, 10*, 1131–1186.
Li, C., et al. (2014). Impulsive synchronization schemes of stochastic complex networks with switching topology: average time approach. *Neural Networks, 54*, 85–94.
Li, Z., et al. (2019). *The Journal of Chemical Physics, 150*, Article 024302.
Li, Z. (2011). *Scientific Reports, 1*, 88.
Q. Liao, et al. How important is weight symmetry in backpropagation? 2015. http://arxiv.org/abs/1510.05067.
T.P. Lillicrap, et al. Random feedback weights support learning in deep neural networks. 2014. http://arxiv.org/abs/1411.0247.
Lisman, J. E., & Zhabotinsky, A. M. (2001). A model of synaptic memory: CaMKII switch that potentiates transmission by organizing an AMPA receptor anchoring assembly. *Neuron, 31*(2), 191–201.
Lisman, J. (1989). A mechanism for Hebb and anti-Hebb processes underlying learning and memory. *Proceedings of the National Academy of Sciences of the United States of America, 86*, 9574–9578.
Litinski, D. (2019). *Quantum, 3*, 128.
Liu, Y., et al. (2009a). Stability and synchronization of discrete-time Markovian jumping neural networks with mixed mode-dependent time delays. *IEEE Transactions on Neural Networks, 20*(7), 1102–1116.
Liu, et al. (2009b). Stability and synchronization of discrete-time Markovian jumping neural networks with mixed mode dependent time delays. *IEEE Transactions on Neural Networks, 20*(7), 1102–1116.
Liu, X., et al. (2014). Synchronization of nonlinear coupled networks via aperiodically intermittent pinning control. *IEEE Transactions on Neural Networks and Learning Systems*, 2162-237x.
Liu, Q., Liu, B., Zhang, Y., Kim, D. S., & Gao, Z. (2016). Improving opinion aspect extraction using semantic similarity and aspect associations. In *AAAI*.
J.K. Liu et al. Embedding multiple trajectories in simulated recurrent neural networks in a self-organizing manner. http://www.jneurosci.org/content/29/42/13172.short.
D. Liu, et al., arXiv:1710.04833.
Liu, Y., et al. (2013). Synchronization of coupled neutral-type neural networks with jumping-mode-dependent discrete and unbounded distributed delay. *IEEE Transactions on Cybernetics, 43*(1), 102–114.
Y. Liu, et al., Entanglement-guided architectures of machine learning by quantum tensor network, arXiv:1803.09111v3 [stat.ML] 26 Jun 2018.
Z. Liu, et al., The presence and absence of barren plateaus in tensor-network based machine learning, arXiv:2108.08312v1 [quant-ph] 18 Aug 2021.
S. Lloyd, et al., Quantum algorithms for supervised and unsupervised machine learning, arXiv:1307.0411 (2013).
N.B. Lovett, et al. Differential evolution for many-particle adaptive quantum metrology. https://link.aps.org/doi/10.1103/PhysRevLett.110.220501.
G.H. Low and N. Wiebe, arXiv:1805.00675 (2018).
G.H. Low, et al., arXiv:1812.00954 (2018).
Low, G. H., et al. (2019). *Quantum, 3*, 163.
Lu, J., et al. (2009). Pinning stabilization of linearly coupled stochastic neural networks via minimum number of controllers. *Neural Networks, IEEE Transactions on, 20*(10), 1617–1629.
Lu, H., et al. (2011). A survey of multilinear subspace learning for tensor data. *Pattern Recognition, 44*(7), 1540–1551.
Lu, J., et al. (2008). Synchronization in an array of nonlinearly coupled chaotic neural networks with delay coupling. *International Journal of Bifurcation and Chaos, 18*(10), 3101–3111.
Lu, J., et al. (2011). Exponential synchronization of linearly coupled neural networks with impulsive disturbances. *IEEE Transactions on Neural Networks, 22*(2), 329–336.
S. Lu, et al., Tensor networks and efficient descriptions of classical data, arXiv: 2103.06872 (2021).
Lu, C.-Y. (2011). A delay-dependent approach to robust control for neutral uncertain neural networks with mixed interval time-varying delays. *Nonlinearity, 24*(4), 1121–1136.
Mørup, M. (2011). Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1*(1), 24–40.
W.J. Ma, et al. Bayesian inference with probabilistic population codes. 2006. http://dx.doi.org/10.1038/nn1790.
Maclaurin, D., et al. (2015). Autograd: Effortless gradients in Numpy. *ICML 2015 automl workshop*.
Majdandzic, A., et al. (2013). Spontaneous recovery in dynamical networks. *Nature Physics, 10*(1), 34–38.
Mao, X., et al. (2008). Almost surely asymptotic stability of neutral stochastic differential delay equations with Markovian switching. *Stochastic Processes and Their Applications, 118*(8), 1385–1406.
Mao, X. (2011). *Stochastic differential equations and applications* (2nd ed.). Cambridge, U. K.: Woodhead Publishing.
Marblestone, A. H., et al. (2016a). Toward an integration of deep learning and neuroscience. *Frontiers in Computational Neuroscience, 10*, 94.
A.H. Marblestone, et al., Towards an integration of deep learning and neuroscience, arXiv:1606.03813v1 [q-bio.NC] 13 Jun 2016.
A.H. Marblestone et al. Designing tools for assumption-proof brain mapping. http://www.cell.com/article/S0896627314007922/fulltext.
Markram, H., & Sakmann, B. (1995). Action potentials propagating back into dendrites trigger changes in efficacy of single-axon synapses between layer vpyramidal neurons. *Society for Neuroscience - Abstract, 21*, 1–3.
Markram, H., Wang, Y., & Tsodyks, M. (1998). Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences of the United States of America, 95*(9), 5323–5328.
A. Martin, et al., Towards pricing financial derivatives with an IBM quantum computer, arXiv:1904.05803 (2019).
Martinez, E. A., et al. (2016). Real-time dynamics of lattice gauge theories with a few-qubit quantum computer. *Nature, 534*, 516.
Matsuura, S., et al. (2018), arXiv:1810.11511.
S. McArdle, et al., Quantum computational chemistry, arXiv:1808.10402v3 [quant-ph] 27 Jan 2020.
S. McArdle, S. Endo, Aspuru-Guzik, S.C. Benjamin and X. Yuan, Quantum computational chemistry, arXiv:1808.10402v3 [quant-ph] 27 Jan 2020.
McClean, J. R., et al. (2014). *The Journal of Physical Chemistry Letters, 5*(24), 4368. https://doi.org/10.1021/jz501649m. pMID: 26273989.
McClean, J. R., et al. (2018). Barren plateaus in quantum neural network training landscapes. *Nature Communications, 9*, 4812.
McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*, 115–133.
McCulloch, W. S., et al. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical biophysics, 5*, 115–133.
Mehta, M. R., et al. (2002). Role of experience and oscillations in transforming a rate code into a temporal code. *Nature, 417*(6890), 741–746.
K. Meichanetzidis, et al., Quantum natural language processing on near-term quantum computers, arXiv:2005.04147 (2020).
Mel, B. W. (1992). The clusteron: Toward a simple abstraction for a complex neuron. *Advances in neural information processing systems*.
Merolla, P. A., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science, 345*, 668–673.
Mezzacapo, A., et al. (2015). Digital quantum rabi and dicke models in superconducting circuits. *Scientific Reports, 4*, 7482.
Mitarai, K., et al. (2018). Quantum circuit learning. *Physical Review*. https://arxiv.org/abs/1803.00745.
Mocanu, D. C., et al. (2018). Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications, 9*, 2383.
P. Mohassel and P. Rindal. 2018. ABY3: A mixed protocol framework for machine learning. https://doi.org/10.1145/3243734.3243760.
Mohassel, P., & Zhang, Y. (2017). SecureML: A system for scalable privacy-preserving machine learning. In *IEEE symposium on security and privacy* (pp. 19–38). IEEE Computer Society.
P. Mohassel, et al. 2015. Fast and secure three-party computation: The garbled circuit approach. https://doi.org/10.1145/2810106.2813705.
R.C. Moioli, P.H.J. Nardelli, M.T. Barros, W. Saad, et al., Neurosciences and 6G: Lessons from and needs of communicative brains, arXiv:2004.01834v1 [eess.SP] 4 Apr 2020.
Moioli, R. C., Nardelli, P. H. J., Barros, M. T., Saad, W., et al. (2021). Neurosciences and wireless networks: The potential of brain-type communications and their applications. *IEEE Communications Surveys and Tutorials, 23*(3), 1599–1621. Third Quarter.
Moler, C., et al. (2003). Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review, 45*, 3–49.
Monroe, C., & Kim, J. (2013). Scaling the ion trap quantum processor. *Science, 339*, 1164.
Mordatch, I., et al. (2012). Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG), 31*(4), 43.
Morrison, A., Diesmann, M., & Gerstner, W. (2008). Phenomenological models of synaptic plasticity based on spike timing. *Biological Cybernetics, 98*, 459–478. https://doi.org/10.1007/s00422-008-0233-1. Springer, 2008.
M. Motta, et al., arXiv:1808.02625 (2018).
Motta, M., et al. (2019), arXiv:1901.07653.
Moxon, K. A., & Foffani, G. (2015). Brain-machine interfaces beyond neuroprosthetics. *Neuron, 86*(1), 55–67.
A. Neelakantan, et al. Neural programmer: Inducing latent programs with gradient descent. 2015. http://arxiv.org/abs/1511.04834.
Neukart, F., et al. (2013). On Quantum Computers and Artificial Neural Networks. *Journal of Signal Processing Research, 2*.
Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review. E, 69*, Article 026113.
Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review, 45*, 167–256.
Nicol, D. M., et al. (1994). Parallel simulation today. *Annals of Operations Research, 53*, 249–285.
Niebur, E., et al. (1991). Oscillator-phase coupling for different two-dimensional network connectivities. *Physical Review A, 44*, 6895–6904.
Nielsen, M.A., and I. Chuang (2002), "Quantum computation and quantum information,".
Nielsen, M. A., et al. (2000). *Quantum computation and quantum information*. Cambridge, UK: Cambridge University Press.
Nielsen, M. A., et al. (2004). *Quantum computation and quantum information (Cambridge series on information and the natural sciences)*. Cambridge University Press.
Noether, E. (1918). Invariante variationsprobleme. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse, 1918*, 235.
O'Malley, P. J. J., et al. (2016). *Physical Review X, 6*, Article 031007.

R.C. O'Reilly, et al. Learning through time in the thalamocortical loops. 2014b. http://arxiv.org/abs/1407.3432.
R.C. O'Reilly. Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6796552.
T.J. O'Shea, T.C. Clancy, and R. McGwier, "Recurrent neural radio anomaly detection," 2016, arXiv:1611.00301.
Olfati-Saber, R., et al. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control, 49*, 1520–1533.
Y. Ollivier et al. Training recurrent networks online without backtracking. http://arxiv.org/abs/1507.07680.
Olshausen, B. A. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature, 381*, 607–609.
Orus, R. (2012). Exploring corner transfer matrices and corner tensors for the classical simulation of quantum lattice systems. *Physical Review, B85*, Article 205117.
Orus, R. (2013). A Practical introduction to tensor networks: Matrix product states and projected entangled pair states. *The Journal of Chemical Physics*.
Orus, R. (2019). Tensor networks for complex quantum systems. *Nature Reviews Physics, 1*, 538.
Oseledets, I. (2010). Approximation of 2d _ 2d matrices using tensor decomposition. *SIAM Journal on Matrix Analysis and Applications, 31*(4), 2130–2145.
Oseledets, I. V. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing, 33*(5), 2295–2317.
J.S. Otterbach, et al., Unsupervised machine learning on a hybrid quantum computer, arXiv:1712.05771 (2017).
Paesani, S. (2017). *Physical Review Letters, 118*, Article 100503.
Panayotova, G. S., Dimitrov, G. P., Dimitrov, W. A., Petrov, P. S., Petrova, P. V., & Tsvetkova, P. T. (2023). One approach to using R for Bayesian analysis of brain signals. In *2023 58th international scientific conference on information, communication and energy systems and technologies (ICEST)*. IEEE. Conference Paper, Publisher.
Pariño, H. D., et al. (2000). Neural network-based model reference adaptive control system. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 30*(1), 198–204.
A. Paszke, et al., Automatic differentiation in PYTORCH (2017).
Pecora, L. M., et al. (1990). Synchronization in chaotic systems. *Physical Review Letters, 64*(8), 821–824.
E. Pednault, et al., Breaking the 49-qubit barrier in the simulation of quantum circuits, arXiv:1710.05867 (2017).
G. Perea, et al. Tripartite synapses: Astrocytes process and control synaptic information. 2009. http://www.ncbi.nlm.nih.gov/pubmed/19615761.
Peterson, E. C., et al. (2020). Two-qubit circuit depth and the monodromy polytope. *Quantum, 4*, 247.
Phan, A. H., et al. (2011). PARAFAC algorithms for large-scale problems. *Neurocomputing, 74*(11), 1970–1984.
Pietras, B., & Daffertshofer, A. (2019). Network dynamics of coupled oscillators and phase reduction techniques. *Physics Reports, 819*, 1–105.
Poggio, T., & Girosi, F. (1989). Networks for approximation and learning. *Proceedings of the IEEE, 78*(9), 1481–1497.
Pokhrel, S. R., et al. (2020). Federated learning with blockchain for autonomous vehicles: Analysis and design challenges. *IEEE Transactions on Communications, 68*(8), 4734.
M.A. Poter, "Nonlinearity networks: A 2020 vision," 2019. Available: arXiv: 1911.03805.
Pototsky, A., et al. (2009). Synchronization of a large number of continuous one-dimensional stochastic elements with time-delayed mean-field coupling. *Physica D, 238*(2), 175–183.
Poulin, D., et al. (2017). *Physical Review Letters, 121*, Article 010501.
Poulin, D., et al. (2015). *Quantum Information and Computation, 15*, 361.
Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum, 2*, 79.
Qin, J., et al. (2015). Exponential synchronization of complex networks of linear systems and nonlinear oscillators: A unified analysis. *IEEE Transactions on Neural Networks and Learning Systems, 26*(3), 510–521.
Rabinovich, M. I., et al. (2006). Dynamical principles in neuroscience. *Reviews of Modern Physics, 78*(4), 1213–1265.
Rebentrost, P., et al. (2018). Quantum Hopfield neural network. *The Physical Review, 98*, Article 042308.
chapter 4 Recce, M. (1999). Encoding information in neuronal activity. In Maass, & Bishop (Eds.), *Pulsed neural networks*. MIT Press.
Reiher, M., et al. (2017). Elucidating reaction mechanisms on quantum, computers. *Proceedings of the National Academy of Sciences, 114*(29), 7555–7560.
Reiher, M., et al. (2017). In .
Rivest, R. L., et al. (1978). *On data banks and privacy homomorphisms. Foundations of secure computation* (pp. 169–179). Academia Press.
A. Roggero et al., Linear response on a quantum computer, arXiv:1804.01505(2018).
Romanchuk, N., Zakharov, A., Sergeeva, M., Komarova, Y., Shirolapov, I., & Pyatin, V. (2023). EEG correlates of the human brain biological age. In *2023 fifth international conference neurotechnologies and neurointerfaces (CNN)*. IEEE. Conference Paper, Publisher.
A. Romero, et al. Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550, 2014.
Rosenblatt, F. (1957). *The perceptron: A perceiving and recognizing automaton*. Cornell Aeronautical Laboratory. Tech. Rep. Inc. Report No. 85-460-1.
Rossi, M., et al. (2013). Quantum hypergraph states. *New Journal of Physics, 15*, Article 113022.
Rostami, M., Mokari, N., Shariaty, F., & Gelgor, A. (2023). Potential of brain-computer interfaces in dementia. In *2023 international conference on electrical engineering and photonics (EExPolytech)*. IEEE. Conference Paper, Pub.
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition, volume I, chapter learning internal representations by error propagation* (pp. 318–362). MIT Press.
Ruvolo, P., & Eaton, E. (2013). ELLA: An efficient lifelong learning algorithm. In *ICML* (pp. 507–515).
S. Sachdev, "Tensor networks¯a new tool for old problems," 2009. http://link.aps.org/doi/10.1103/Physics.2.90.
Saib, W., Bonet-Monroig, X., Dunjko, V., & Tavernell, I. (2023). Benchmarking adaptive quantum circuit optimization algorithms for quantum chemistry. In *2023 IEEE international conference on quantum computing and engineering (QCE)*. IEEE. Conference Paper, Publisher:.
Samuel, N., et al. (2017). Deep MIMO detection. In *Proc. IEEE 18th int. workshop signal process. adv. wireless commun*.
B.C. Sanders et al., Optimal quantum measurements for phase estimation. 1995. https://link.aps.org/doi/10.1103/PhysRevLett.75.2944.
Santagati, R., et al. (2018). *Science Advances, 4*(1) https://doi.org/10.1126/sciadv.aap9646
Santini, P., et al. (2011). Molecular nanomagnets as quantum simulators. *Physical Review Letters, 107*, Article 230502.
Sasaki, M., & Carlini, A. (2002). Quantum learning and universal quantum matching machine. *Physical Review A, 66*, Article 022303.
Sasaki, M., et al. (2001). Quantum template matching. *Physical Review A, 64*, Article 022317.
Sastry, S., et al. (1989). *Cortical network: Stability, convergence and robustness*. Englewood Cliffs, NJ, USA: Prentice-Hall.
Sawaya, N. P. D., Marti-Dafcik, D., & Ho, Y. (2023). HamLib: A library of hamiltonians for benchmarking quantum algorithms and hardware. In *2023 IEEE international conference on quantum computing and engineering (QCE)*. IEEE. Conference Paper, Publisher.
Schindler, P., et al. (2011). Experimental repetitive quantum error correction. *Science, 332*, 1059.
Schindler, P., Nigg, D., Monz, T., et al. (2013). A quantum information processor with trapped ions. *New Journal of Physics, 15*, Article 123012.
Schmidhuber, J. (2015). Deep learning in neural networks: Overview. *Neural Networks, 61*, 85–117.
Schoelkopf, R. J., & Girvin, S. M. (2008). Wiring up quantum systems. *Nature, 451*, 664–669.
Schuld, M., et al. (2014). The quest for a quantum neural network. *Quantum Information Processing, 13*, 2567–2586.
Schuld, M., et al. (2015). Simulating a perceptron on a quantum computer. *Physics Letters A, 7*, 660–663.
Schuld, M., et al. (2017). Implementing a distance-based classifier with a quantum interference circuit. *Europhysics Letters, 119*, 6002.
Schultz, D.ayan, & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science, 275*, 1593–1599.
T.J. Sejnowski et al. Prospective Optimization. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6803897.
G. Sentís, et al. Quantum learning without quantum memory. 2012. doi:10.1038/srep00708.
A. Sergeevich et al. Optimizing qubit Hamiltonian parameter estimation algorithm using PSO. 10.1109/cec.2012.6252948.
P. Sermanet et al. Pedestrian detection with unsupervised multi-stage feature learning. arXiv:1212.0142v2 [cs.CV] 2 Apr 2013.
T. Serre, et al. A feedforward architecture accounts for rapid categorization. 2007. http://www.pnas.org/content/104/15/6424.long.
Shalev-Shwartz, S., et al. (2014). *Understanding machine learning: From theory to algorithms*. New York, NY, USA: Cambridge University Press. ISBN 1107057132, 9781107057135.
Shepard, D. (1968). A two-dimensional interpolation function for irregularly spaced data. In *Proceedings of the 1968 23rd ACM national conference* (pp. 517–524).
S.M. Sherman. Thalamic relays and cortical functioning. Progress in Brain Research, http://www.ncbi.nlm.nih.gov/pubmed/16226580.
Shouval, H. Z., Bear, M. F., & Cooper, L. N. (2002). A unified model of NMDA receptor dependent bidirectional synaptic plasticity. *Proceedings of the National Academy of Sciences of the United States of America, 99*, 10831–10836.
Si, J. (2004). *Handbook of learning and approximate dynamic programming, 2*. John Wiley & Sons.
N. Sidiropoulos, "Low-rank decomposition of multi-way arrays: A signal processing perspective," http://www.sandia.gov/~tgkolda/tdw2004/Nikos04.pdf.
Silver, D. L., & Mercer, R. E. (2005). The task rehearsal method of life-long learning: Overcoming impoverished data. In *Proceedings of the 15th conference of the canadian society for computational studies of intelligence on advances in artificial intelligence* (pp. 90–101).
Silver, D. L., & Poirier, R. (2004). Sequential consolidation of learned task knowledge. In *Conference of the Canadian society for computational studies of intelligence* (pp. 217–232).
Silver, D. L., & Poirier, R. (2007). Context-sensitive MTL networks for machine lifelong learning. In *FLAIRS conference* (pp. 628–633).
Sjostrom, P., & Turrigiano, N. (2001). Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron, 32*, 1149–1164.
Somma, R., et al. (2002). Simulating physical phenomena by quantum networks. *Physical Review A, 65*, Article 042323.
Sprechmann, P., Litman, R., Yakar, T. B., Bronstein, A. M., & Sapiro, G. (2013). Supervised sparse analysis and synthesis operators. *Advances in neural information processing systems* (pp. 908–916).

Stanoev, A., et al. (2013). Consensus theory in networked systems. In L. Kocarev (Ed.), *Consensus and synchronization in complex networks*. Berlin, Germany: Springer-Verlag.
Stauffer, J., & Zhang, Q. (2023). Spiking neural network with backpropagation learning for brain visual dynamics decoding. In *2023 11th international IEEE/EMBS conference on neural engineering (NER)*. IEEE. Conference Paper, Publisher.
Stoudenmire, E., et al. (2016a). Supervised learning with tensor networks. In D. D. Lee, et al. (Eds.), *Advances in neural information processing systems vol. 29* (pp. 4799–4807). Curran Associates, Inc.
E. Stoudenmire et al., in Advances in neural information processing systems (2016b), pp. 4799–4807.
Strogatz, S. H., et al. (1988). Collective synchronisation in lattices of nonlinear oscillators with randomness. *Journal of Physics A: Mathematical and General, 21*, L699–L705.
Strogatz, S. H. (2001). Exploring complex networks. *Nature, 410*, 268–276.
Strogatz, S. H. (2003). *Sync: The emerging science of spontaneous order*. New York, NY, USA: Hyperion.
Sugisaki, et al. (2016). *The Journal of Physical Chemistry A, 120*(32), 6459. https://doi.org/10.1021/acs.jpca.6b04932. pMID: 27499026.
Sugisaki, K., et al. (2019). *ACS Central Science, 5*(1), 167. https://doi.org/10.1021/acscentsci.8b00788
Sussillo, D., et al. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*. http://www.sciencedirect.com/science/article/pii/S0896627309005479.
Suter, S. K., et al. (2013). Tamresh -tensor approximation multiresolution hierarchy for interactive volume visualization. *Computer Graphics Forum, 32*(3), 151–160.
Sutton, R. S., & Barto, A. G. (1998a). *Reinforcement learning: An introduction*. MIT Press.
Sutton, R. S., & Barto, A. G. (1998b). Reinforcement learning: An introduction. *Adaptive computation and machine learning*. The MIT Press.
Sutton, R. S., et al. (1998). *Introduction to reinforcement learning* (1st ed.). Cambridge, MA, USA: MIT Press. ISBN 0262193981.
Swan, M., dos Santos, R. P., Lebedev, M. A., & Witte, F. (2022). *Quantum computing for the brain* (pp. 1–20). London, U.K.: World Scientific.
L. Sweeney. 2002. K-anonymity: A model for protecting privacy, https://doi.org/10.1142/S0218488502001648.
Szegedy, M. (2004). In *45th annual IEEE symposium on foundations of computer science* (pp. 32–41). IEEE.
Tabbal, J., Kabbara, A., & Hassan, M. (2023). Insights into electrophysiological brain states dynamics. In *2023 seventh international conference on advances in biomedical engineering (ICABME)*. IEEE. Conference Paper, Publisher.
F. Tacchino, et al., An artificial neuron implemented on an actual quantum processor, arXiv:18110.02266v1 [quant-ph] 6 Nov 2018.
Tang, Y., et al. (2012). A constrained evolutionary computation method for detecting controlling regions of cortical networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 9*(6), 1569–1581.
Tavakoli, H., Rostami, R., & Nazem-Zadeh, M.-R. (2023). Assessments of variability in cortical and subcortical measurements and within-network connectivity of the brain using test-retest data. In *2023 45th annual international conference of the IEEE engineering in medicine & biology society (EMBC)*. IEEE. Conference Paper, Publisher.
S.V. Taylor et al. Does the cost function of human motor control depend on the internal metabolic state? http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3240571/.
Thorpe, S., Fize, D., & Marlot, C. (1996). Speed of processing in the human visual system. *Nature, 381*(6582), 520–522.
Thorpe, S., Delorme, A., & Van Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural Networks, 14*, 715–725.
Thrun, S., & Mitchell, T. M. (1995). *Lifelong robot learning*. Springer.
Thrun, S. (1996). Is learning the n-th thing any easier than learning the first?. In *NIPS* (pp. 640–646).
E. Torrontegui, et al., Universal quantum perceptron as efficient unitary approximators, arXiv:18010.00934 (2018).
Troyer, M., & Wiese, U.-J. (2005). Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations. *Physical review letters, 94*, Article 170201.
Tubman, et al. (2018a), arXiv:1809.05523.
N.M. Tubman, et al., arXiv:1809.05523 (2018b).
Turrigiano, G. G., & Nelson, S. B. (2004). Homeostatic plasticity in the developing nervous system. *Nature Reviews Neuroscience, 5*, 97–107.
Udayakumar, P., & Subhashini, R. (2023). Diffusion MRI preprocessing for deep learning analysis of brain structural connectivity. In *2023 12th international conference on advanced computing (ICoAC)*. IEEE. Conference Paper, Publisher:.
S. Ullman, et al. From simple innate biases to complex visual concepts. http://www.pnas.org/content/109/44/18215.full.
Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM, 27*(11), 1134–1142.
Van Vreeswijk, C., Abbott, L. F., & Ermentrout, G. B. (1994). When inhibition not excitation synchronizes neural firing. *Journal of Computational Neuroscience, 1*(4), 313–321.
Vapnik, V. N. (1998). *Statistical learning theory*. Wiley.
Veis, L., et al. (2014). *The Journal of Chemical Physics, 140*(21), Article 214111. https://doi.org/10.1063/1.4880755
A. Veit, et al. Residual networks are exponential ensembles of relatively shallow networks. 2016. http://arxiv.org/abs/1605.06431.
Veletić, M., Barros, M. T., Balasingham, I., & Balasubramaniam, S. (2019). A molecular communication model of exosome-mediated brain drug delivery. In *Proc. 6th annu. ACM int. conf. nanoscale comput. commun.* (pp. 1–7).
Vidal, G. (2003). Efficient classical simulation of slightly entangled quantum computations. *Physical Review Letters, 91*(14), Article 147902.
B. Villalonga, et al., Establishing the Quantum Supremacy Frontier with a 281 Pflop/s Simulation, arXiv:1905.00444 (2019).
Wagemakers, A., et al. (2006). Synchronization of electronic genetic networks. *Chaos, 16*, 3127.
Wang, Y., et al. (2015). *ACS Nano, 9*(8), 7769.
J. Wang, et al., Anomaly detection with tensor networks, arXiv:2006.02516 (2020a).
S. Wang, et al., Noise-induced barren plateaus in variational quantum algorithms, arXiv: 2007.14384 (2020b).
Wang, Q., Wang, H., Chen, S., Deng, H., & Zhu, Y. (2023). Scientific problem solving and brain symmetry index: An exploratory EEG study. In *2023 45th annual international conference of the IEEE engineering in medicine & biology society (EMBC)*. IEEE. Conference Paper, Publisher.
Wang, H., et al. (2005). Out-of core tensor approximation of multi-dimensional matrices of visual data. *ACM Transactions on Graphics, 24*(3), 527–535.
Wang, Y., et al. (2010). Synchronization of stochastic genetic oscillator networks with time delays and Markovian jumping parameters. *Neurocomputing, 73*(13–15), 2532–2539.
Wang, et al. (2013). Exponential synchronization of stochastic perturbed complex networks with time-varying delays via periodically intermittent pinning. *Communications in Nonlinear Science and Numerical Simulation, 18*(11), 3146–3157.
Wang, Y.-W., et al. (2015). Global synchronization of complex dynamical networks through digital communication with limited data rate. *IEEE Transactions on Neural Networks and Learning Systems, 26*(10), 2487–2499.
X.-J. Wang. The prefrontal cortex as a quintessential cognitive-type neural circuit: Principles of frontal lobe function http://oxfordindex.oup.com/view/.
Watanabe, H. C., Raymond, R., Ohnishi, Y.-Y., Kaminishi, E., & Sugawara, M. (2023). Optimizing parameterized quantum circuits with free-axis single-qubit gates. *IEEE Transactions on Quantum Engineering, 4*. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10153660.
Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-word' networks. *Nature, 393*(6684), 440–442.
Watts, D. J. (1999). *Small worlds: The dynamics of networks between order and randomness*. Princeton, NJ, USA: Princeton University Press.
Wecker, D., et al. (2015). *Physical Review A, 92*, Article 062318.
Wen, S., et al. (2013). Global exponential synchronization of memristor-based recurrent neural networks with time-varying delays. *Neural Networks, 48*, 195–203.
Wen, S., et al. (2015). Circuit design and exponential stabilization of memristive neural networks. *Neural Networks, 63*, 48–56.
Wen, R., Song, L., & Lin, Y. (2023). Brain-like intelligent data mining mechanism based on convolutional neural network. In *2023 international conference on mechatronics, IoT and industrial informatics (ICMIII)*. IEEE. Conference Paper, Publisher.
Wen, P., et al., Exponential lag adaptive synchronization of memristive neural networks and applications in Pseudo-random generators. IEEE Transactions on Fuzzy Systems, doi: 10.TFUZZ.2013.2294855.
Wen, S., et al. (2012). H∞ filtering for neutral systems with mixed delays and multiplicative noises. *IEEE Transactions on Circuits and Systems II: Express Briefs, 59*(11), 820–824.
Werbo, P. J. (2009). Intelligence in the brain: A theory of how it works and how to build it. *Neural Networks, 22*(3), 200–212.
P. Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. 1974.
Werbos, P. J. (1990). Backpropagation through time: What it does and how to do it. In *Proceedings of the IEEE*. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=58337.
J. Weston, et al. Memory Networks. 2014. http://arxiv.org/abs/1410.3916.
Whiteway, M. R., et al. (2019). The quest for interpretable models of neural population activity. *Current Opinion in Neurobiology, 58*, 86–93.
S.R. Williams et al. Backpropagation of physiological spike trains in neocortical pyramidal neurons: Implications for temporal coding in dendrites. http://www.ncbi.nlm.nih.gov/pubmed/11069929.
R.I. Wilson and R.A. Nicoll. Endogenous cannabinoids mediate retrograde signalling at hippocampal synapses. http://www.ncbi.nlm.nih.gov/pubmed/11279497.
Winfree, A. T. (1967). Biological rhythms and the behavior of populations of coupled oscillators. *Journal of Theoretical Biology, 16*, 15–42.
Woerner, S., et al. (2019). Quantum risk analysis. *npj Quantum Information, 5*, 15.
Wu, Z.-G., et al. (2013a). Sampled-data exponential synchronization of complex dynamical networks with time-varying coupling delay. *IEEE Transactions on Neural Networks and Learning Systems, 24*(8), 1177–1187.
Wu, Z.-G., et al. (2013b). Stochastic synchronization of Markovian jump neural networks with time-varying delay using sampled data. *IEEE Transactions on Cybernetics, 43*(6), 1796–1806.
Wu, W., He, S., Chen, Y., Chen, C., Fu, Y., Liu, K., & Qu, J. Y. (2023). Minimally invasive microglial and neuronal imaging in mouse spinal cord dorsal horn. *IEEE Journal of Selected Topics in Quantum Electronics, 29*(4), 1177–1187.
Wu, C. (2007). *Synchronization in complex networks of nonlinear dynamical systems*. singapore: World Scientfic.
Xia, Y., et al. (2009). Adaptive backstepping controller design for stochastic jump systems. *IEEE Transactions on Automatic Control, 54*(12), 2853–2859.
X. Xie et al. Equivalence of backpropagation and contrastive Hebbian learning in a layered network. http://www.ncbi.nlm.nih.gov/pubmed/12590814.
Yamamoto, M., et al. (2004). Trajectory control of incompletely restrained parallel-wire-suspended mechanism based on inverse dynamics. *IEEE Transactions on Robotics, 20*(5), 840–850. Oct.
Yan, G., et al. (2017). Network control principles predict neuron function in the Caenorhabditis elegans connectome. *Nature, 550*(7677), 519–523.

Q. Yang, Y. Liu, T. Chen and Y. Tong, Federated machine learning: Concept and applications, arXiv:1902.04885v1 2019.
Yang, T. (2004). A survey of chaotic secure communication systems. *International Journal of Computational Cognition, 2*(2), 81–130.
Yang, X., et al. (2009). Stochastic synchronization of coupled neural networks with intermittent control. *Physics Letters A, 373*(36), 3259–3272.
Yang, X., et al. (2010). Adaptive lag synchronization for competitive neural networks with mixed delays and uncertain hybrid perturbations. *IEEE Transactions on Neural Networks, 21*(10), 1656–1667.
Yang, X., et al. (2012). Synchronization of Markovian coupled neural networks with nonidentical node-delays and random coupling strengths. *IEEE Transactions on Neural Networks and Learning Systems, 23*(1), 60–71.
Yang, X., et al. (2013). Synchronization of coupled reaction-diffusion neural networks with time-varying delays via pinning-impulsive controller. *SIAM Journal on Control and Optimization, 51*(5), 3486–3510.
You, H., et al. (2013). Simulating the transverse Ising model on a quantum computer: Error correction with the surface code. *Physical Review A, 87*, Article 032341.
Yu, X., & Kaynak, O. (2017). Sliding mode control made smarter: A computational intelligence perspective. *IEEE Systems, Man, and Cybernetics Magazine, 3*(2), 31–34.
Yu, J., et al. (2012). Exponential lay synchronization for delayed fuzzy cellular nerual networks via periodically intermittent control. *Mathematics and Computers in Simulation, 82*(5), 895–908.
J. Yuan and S. Yu. 2014. Privacy preserving back-propagation neural network learning made practical with cloud computing. https://doi.org/10.1109/TPDS.2016.18.
Yung, M.-H., et al. (2014). *Scientific Reports, 4*, 3589.
Zhang, et al. (2013a). Exponential synchronization of coupled switched neural networks with mode-dependent impulsive effects. *IEEE Transactions on Networks and Learning Systems, 24*(8), 1316–1326.
Zhang, Y., et al. (2013b). Global exponential adaptive synchronization of complex dynamical networks with neutral-type neural network nodes and stochastic disturbances. *IEEE Transactions on Circuits and Systems I: Regular Papers, 60*(10), 2709–2718.
Q. Zhang, et al. 2016. Privacy preserving deep computation model on cloud for big data feature learning https://doi.org/10.1109/TC.2015.2470255.
W. Zhang, et al., Synchronization of neural networks with stochastic perturbation via aperiodically intermittent control, https://scholar.google.com/scholar?q=Synchronization+of+neural+networks+with+stochastic+perturbation+via+aperiodically+intermittent+control&hl=en&as_sdt=0&as_vis=1&oi=scholart.
W. Zhang, et al. Synchronization of neural networks with stochastic perturbation via aperiodically intermittent control, https://scholar.google.com/scholar?q=Synchronization+of+neural+networks+with+stochastic+perturbation+via+aperiodically+intermittent+control&hl=en&as_sdt=0&as_vis=1&oi=scholart.
Zhang, H., et al. (2008). Global asymptotic stability of recurrent neural networks with multiple time-varying delays. *IEEE Transactions on Neural Networks, 19*(5), 855–873.
Zhang, Y., et al. (2009). Fuzzy impulsive control of chaotic systems based on TS fuzzy model. *Chaos, Solitons Fractals, 39*(4), 2002–2011.
Zhang, H., et al. (2015). Mode-dependent stochastic synchronization for Markovian coupled neural networks with timevarying mode-delays. *IEEE Transactions on Neural Networks and Learning Systems, 26*(11), 2621–2634.
Zhao, C., et al. (2021). Analyzing the barren plateau phenomenon in training quantum neural networks with the ZX-calculus. *Quantum, 5*, 466.
Zhou, W., et al. (2013). Adaptive exponential synchronization in pth moment of neutral-type neural networks with time delays and Markovian switching. *International Journal of Control, Automation and Systems, 11*(4), 845–851.
Zhou, W., et al. (2014). Adaptive synchronization for neutral-type neural networks with stochastic perturbation and Markovian switching parameters. *IEEE Transactions on Cybernetics, 44*(12), 2848–2860.
Zhu, Q., et al. (2012). p-th moment exponential synchronization for stochastic delayed Cohen–Grossberg neural networks with Markovian switching. *Nonlinear Dynamics, 67*(1), 829–845.
Zhuang, Y., et al. (2012). On the uplink MAC performance of a drive-thru internet. *IEEE Transactions on Vehicular Technology, 61*(4), 1925–1935.
Zochowski, M. (2000). Intermittent dynamical control. *Physica D, 145*(3), 181–190.
Zou, D. (2007). Kinetic models of spike-timing dependent plasticity and their functional consequences in detecting correlations. *Biological Cybernetics, 97*(1), 81–97.
Zurada, J. M. (1992). *Introduction to artificial neural systems*. West Group.