



This is the accepted manuscript made available via CHORUS. The article has been published as:

Classifying quantum data by dissipation

Jeffrey Marshall, Lorenzo Campos Venuti, and Paolo Zanardi

Phys. Rev. A **99**, 032330 — Published 20 March 2019

DOI: [10.1103/PhysRevA.99.032330](https://doi.org/10.1103/PhysRevA.99.032330)

Classifying quantum data by dissipation

Jeffrey Marshall, Lorenzo Campos Venuti, and Paolo Zanardi

*Department of Physics and Astronomy, and Center for Quantum Information Science & Technology,
University of Southern California, Los Angeles, CA 90089-0484*

We investigate a general class of dissipative quantum circuit capable of computing arbitrary Conjunctive Normal Form (CNF) Boolean formulas. In particular, the clauses in a CNF formula define a local generator of Markovian quantum dynamics which acts on a network of qubits. Fixed points of this dynamical system encode the evaluation of the CNF formula. The structure of the corresponding quantum map partitions the Hilbert space into sectors, according to decoherence-free subspaces (DFSs) associated with the dissipative dynamics. These sectors then provide a natural and consistent way to classify quantum data (i.e. quantum states). Indeed, the attractive fixed points of the network allow one to learn the sector(s) for which some particular quantum state is associated. We show how this structure can be used to dissipatively prepare quantum states (e.g. entangled states), and outline how it may be used to generalize certain classical computational learning tasks.

I. INTRODUCTION

It has been shown that the dissipative model of quantum computation, described by Ref. [1], is equivalent in computational power to the gate model [1, 2]. In fact, since there is some built-in robustness to errors within this framework of dissipative quantum circuits [1, 3, 4], it provides a potential route to large-scale universal quantum computation. Moreover, since quantum states can be constructed to be attractive fixed points of such a dissipative dynamical system, state preparation is a very natural application of these circuits [5]. Another key area where quantum technology may prove to be beneficial is the field of machine learning [6–9], several techniques of which actively exploit dissipative quantum dynamics [10–14].

In this paper we add to the growing list of dissipation assisted techniques for use in quantum computing. As such, this provides yet another example where engineered dissipation can be understood as a resource. In particular we describe a dissipative quantum network capable of computing Conjunctive Normal Form (CNF) [and also Disjunctive Normal Form (DNF)] formulas, which connects all three of the aforementioned topics of interest: computation, state preparation, machine learning. Formulas of this type are of particular interest in the context of computational learning theory [15].

We achieve this by introducing a local Lindbladian which defines the dynamics of a quantum system. The associated fixed points of this dynamics encode the evaluation of the clauses in the CNF formula. Recently similar techniques have been used to efficiently transfer quantum states [16].

The circuit construction we introduce preserves quantum coherence between ‘classical’ states (computational basis states) which have the same evaluation on the clauses in the CNF formula. As such,

this provides a natural way to categorize quantum data, i.e., according to the resulting subspace partitioning of the Hilbert space, upon evolving the dissipative network. These partitions are in fact decoherence-free subspaces (DFSs) of the underlying dissipative dynamics [17, 18]. As will be demonstrated in Sect. III A, this level of classification can be particularly useful from the perspective of learning theory.

The inspiration for this work comes partly from the classical Hopfield recurrent neural network. In this model, data is classified according to a set of ‘memories’ or ‘patterns’, to which the network dynamically evolves towards, i.e. attractive fixed points of the dynamics. In this manner, the space of classical bit strings is partitioned according to the set of memory states. In our model, which is similar in spirit, data is classified according to the fixed points of a dissipative quantum evolution, and the full space is partitioned into sectors according to DFSs. Our model however in fact goes further in the sense that it can in addition classify quantum data, i.e. superpositions of ‘classical’ states.

Based on the model we introduce, we generalize certain classical notions of data classification, providing a consistent framework from which to classify quantum data. We provide a basic outline as to how this can be used in the context of computational learning theory, where the data is quantum in nature.

Another use of this partitioning of the Hilbert space is in the preparation of quantum states – such as entangled states – by dissipation alone. This technique is somewhat different to previous dissipative state preparation techniques, such as described in Refs. [4, 5], which effectively use dissipation to enact the desired unitary. Our method relies on finding a CNF formula which will split the space into DFSs containing the desired states. In fact, one could pre-

pare an ensemble of quantum states in this manner. We also demonstrate how to prepare quantum Probably Approximately Correct (PAC) states [19, 20].

We discuss and bound errors within this network, which shows it actually benefits from the engineered dissipation being as strong as possible, i.e., we are working in the strongly dissipative regime.

After briefly introducing some well known results, we will discuss our general framework, and two propositions from which all of the applications discussed above are essentially special cases.

A. Background and Notation

Consider the set of all bit strings of length n , $\mathcal{X} := \{0, 1\}^n$. We denote the binary variables associated with $x \in \mathcal{X}$ by $x_i \in \{0, 1\}$, where $x = (x_1, \dots, x_n)$.

A CNF formula C is a conjunction (\wedge) of clauses, where each of the clauses are disjunctions (\vee) of literals, l . Each literal l is either a variable x_j or its negation $\neg x_j$, for some $1 \leq j \leq n$. For example, $C = C_1 \wedge \dots \wedge C_N$, with $C_i = l_1^{(i)} \vee \dots \vee l_{k_i}^{(i)}$, where $k_i \in [n]$. We will throughout use the notation $[n] := \{1, 2, \dots, n\}$.

We state some standard results pertaining to CNF formulas and Boolean functions (see e.g. Ref. [15]):

- i) Any Boolean function, $f : \mathcal{X} \rightarrow \{0, 1\}$ can be represented by a CNF formula (although the number of clauses may be exponential in n).
- ii) Any function $f : \mathcal{X} \rightarrow \{0, 1\}^m$ can be represented using m CNF formulas; one formula for each bit of $f(x)$.
- iii) A clause C_i of a CNF formula containing k_i literals can be represented, using $O(k_i) = O(n)$ additional variables, by an equivalent, equisatisfiable CNF formula where each of the clauses contain at most 3 literals. Such a formula is also known as a ‘3-CNF’.

Throughout we will make use of the above three points, and as such will focus much of our attention on computing Boolean functions ($m = 1$) with a 3-CNF representation, C . We will refer to the explicit evaluation of C on some particular $x \in \mathcal{X}$, using the notation $C(x) \in \{0, 1\}$. Similarly, for the evaluation of a literal l on bit string x we will write $l(x) \in \{0, 1\}$.

From the perspective of machine learning, one may be directly interested in the clauses themselves since these can be thought of as the relevant (abstract) ‘features’ being used by, for example, a machine learning algorithm to classify data. Imagine the task of learning a Boolean function f according to some particular algorithm. Each time the

algorithm makes an update, it can be thought of as changing the clauses in a CNF formula relating to the estimation of f . This induces a partitioning over the space \mathcal{X} , according to the following definition.

Definition 1. Let $C = \wedge_i C_i$ be a 3-CNF formula over \mathcal{X} . Then, $x, y \in \mathcal{X}$, belong to the same ‘partition’, or ‘sector’, of \mathcal{X} iff $C_i(x) = C_i(y)$, $\forall i$.

For a CNF formula with N clauses, this defines a partitioning of the space \mathcal{X} into at most 2^N sectors, i.e. according to every possible set of features (clause evaluations) in the data. Note that x, y belonging to the same sector of course implies $C(x) = C(y)$, but the converse will not necessarily be true.

We will upgrade these general notions to define partitions over the Hilbert space, where it will be shown that Def. 1 provides a natural starting point for defining quantum data classification by dissipation. Practically speaking, one is often interested only in the evaluation of C itself, regardless of the individual clauses. We will in a similar manner extend the definition of the Boolean formula C over classical bit strings, and give it a meaning over the Hilbert space, allowing us to write such quantities as $C(\psi)$ where $|\psi\rangle$ is a quantum state.

Since the clauses are in this context the most fundamental objects, we will first look at the ‘quantum evaluation’ of individual clauses, $C_i(\psi)$, and then generalize this to computing 3-CNF formulas over the Hilbert space, using dissipation.

II. THEORY

We map the above into the quantum realm, where instead of using n binary variables to encode data, one considers a system of n qubits. We will occasionally refer to the i -th qubit itself as q_i for $i = 1, \dots, n$. The Hilbert space $\mathcal{H}_{\mathcal{X}} = \text{span}\{|x\rangle\}_{x \in \mathcal{X}} \cong \mathbb{C}_2^{\otimes n}$ is therefore of dimension 2^n . Here $|x\rangle := \otimes_{i=1}^n |x_i\rangle$, with $|0\rangle, |1\rangle$ eigen-states of the Pauli z operator $\sigma^z := |1\rangle\langle 1| - |0\rangle\langle 0|$. We represent an arbitrary pure state in $\mathcal{H}_{\mathcal{X}}$ as $|\psi\rangle = \sum_{x \in \mathcal{X}} a_x |x\rangle$, for complex coefficients (amplitudes) a_x which are normalized $\sum_{x \in \mathcal{X}} |a_x|^2 = 1$. Occasionally we will refer explicitly to density operators $\rho \in \mathcal{L}(\mathcal{H}_{\mathcal{X}})$ which are Hermitian, positive-semidefinite, trace one, linear operators acting over the Hilbert space.

A. Dissipative evaluation of 3-CNF clause

Central to the dissipative network which is to be described, is the ability to dissipatively evaluate a clause of 3-variables, i.e., $C = l_1 \vee l_2 \vee l_3$, where the l_i are either x_k or $\neg x_k$, for some $k \in [n]$. That is,

given some $|x\rangle \in \mathcal{H}_{\mathcal{X}}$, with $x \in \mathcal{X}$, we will compute $C(x)$.

Let us assume the only three bits of x involved in C are the i_1, i_2, i_3 -th bits, where $i_j \in [n]$. Without loss of generality we will assume these three bits are distinct. In the quantum setting, this corresponds to three qubits, q_{i_j} for $j = 1, 2, 3$. One can construct a small dissipative network, coupling these three qubits to an ancilla qubit, a , via a 4-local Lindbladian [21, 22], as in Fig. 1.

We define this Linbladian

$$\mathcal{L} \cdot = \gamma \left(L \cdot L^\dagger - \frac{1}{2} \{L^\dagger L, \cdot\} \right) \quad (1)$$

by a single Lindblad (jump) operator which acts on these 4 qubits as

$$\begin{aligned} L &= \Pi^\neg \otimes \sigma^-, \\ \Pi^\neg &:= |\neg \bar{l}\rangle \langle \neg \bar{l}|, \\ |\neg \bar{l}\rangle &:= |l_1 = 0, l_2 = 0, l_3 = 0\rangle. \end{aligned} \quad (2)$$

The second term in the tensor product of L acts on the ancilla qubit a [23], Π^\neg depends entirely on the form of C and acts on the q_{i_j} ($j = 1, 2, 3$), and $\gamma > 0$ is the strength of the dissipation. The initial state of the ancilla qubit is assumed to be $|a\rangle = |1\rangle$.

The notation should be interpreted as follows. If for example a clause is given by

$$C = x_i \vee \neg x_j \vee x_k, \quad (3)$$

the projector will be given by $\Pi^\neg = |0, 1, 0\rangle \langle 0, 1, 0|$, acting on the i, j and k -th qubits ($q_{i,j,k}$). This follows by writing $l_{1,2,3} = 0$, where here $l_1 = x_i$, $l_2 = \neg x_j$, $l_3 = x_k$, which in our notation gives

$$|\neg \bar{l}\rangle = |x_i = 0, x_j = 1, x_k = 0\rangle, \quad (4)$$

i.e. $\Pi^\neg = |0, 1, 0\rangle \langle 0, 1, 0|$ acting on the qubits $q_{i,j,k}$ respectively.

The projector Π^\neg projects out any state not of the form $|l_1 = 0, l_2 = 0, l_3 = 0\rangle$, i.e., the case in which the clause C evaluates to 0. In this case, the ancilla state is accordingly flipped to $|0\rangle$ by σ^- .

We make the following claim.

Proposition 1. *Let $\mathcal{E}_t := e^{t\mathcal{L}}$ be the evolution operator, under Lindbladian \mathcal{L} defined by single Lindblad operator Eq. (2). Then the evolution of a matrix element $|x\rangle \langle y| \otimes |1\rangle \langle 1|$, where $x, y \in \mathcal{X}$, and the second term in the tensor product refers to the ancilla qubit a , is*

$$\mathcal{E}_t (|x\rangle \langle y| \otimes |1\rangle \langle 1|) = |x\rangle \langle y| \otimes \begin{cases} e^{-\gamma t} |1\rangle \langle 1| + (1 - e^{-\gamma t}) |c\rangle \langle c| & \text{if } C(x) = c = C(y) \\ e^{-\gamma t/2} |1\rangle \langle 1| & \text{if } C(x) \neq C(y). \end{cases} \quad (5)$$

Note, if $C(x) = 1 = C(y)$, there is strictly no evolution, since in this case, by construction, $L|x, 1\rangle = L|y, 1\rangle = 0$.

Proof. Consider a disjunction of three literals, $C = l_1 \vee l_2 \vee l_3$, where l_j is either x_{i_j} , or $\neg x_{i_j}$, for some $i_j \in [n]$. That is, $C(x)$, for $x \in \mathcal{X}$, is fully determined by the (distinct) bits i_j in x , for $j = 1, 2, 3$.

Let \mathcal{L} be a Lindbladian defined by Eqs. (1), (2) with the Lindblad operator L acting on (distinct) qubits i_1, i_2, i_3 , and an ancilla qubit.

Let $x, y \in \mathcal{X}$. Then

$$\begin{aligned} \mathcal{L}|x\rangle \langle y| \otimes |1\rangle \langle 1| &= \\ |x\rangle \langle y| \otimes \begin{cases} \gamma(|0\rangle \langle 0| - |1\rangle \langle 1|) & \text{if } C(x) = 0 = C(y) \\ -\frac{\gamma}{2}|1\rangle \langle 1| & \text{if } C(x) \neq C(y) \\ 0 & \text{if } C(x) = 1 = C(y) \end{cases} \end{aligned} \quad (6)$$

The conditions on the RHS of the above all come directly from the form of L :

- i) If $C(x) = 0 = C(y)$, then $l_j(x) = 0 = l_j(y)$ for $j = 1, 2, 3$. Hence, $L|x, 1\rangle = |x, 0\rangle$, and $L|y, 1\rangle = |y, 0\rangle$. Similarly, $L^\dagger L|x, 1\rangle = |x, 1\rangle$, and $L^\dagger L|y, 1\rangle = |y, 1\rangle$.
- ii) If $C(x) \neq C(y)$, either $L|x, 1\rangle = 0$ and $L^\dagger L|y, 1\rangle = |y, 1\rangle$, or $L^\dagger L|x, 1\rangle = |x, 1\rangle$ and $L|y, 1\rangle = 0$.
- iii) If $C(x) = 1 = C(y)$, then $\exists j, k$ such that $l_j(x) = 1 = l_k(y)$. Hence $L|x, 1\rangle = 0 = L|y, 1\rangle$.

Moreover, since

$$\mathcal{L}|x\rangle \langle y| \otimes |0\rangle \langle 0| = 0, \quad (7)$$

the form of Eq. (5) is clear. \square

We provide some additional comments regarding this result:

- 1) For input state $|x\rangle \langle x|$ (with $x \in \mathcal{X}$), in order to guarantee with probability of $1 - \epsilon$ of obtaining result $C(x)$ upon measuring the ancilla qubit,

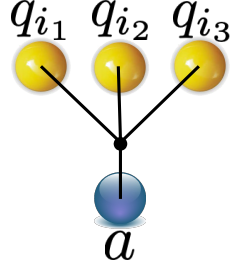


FIG. 1. Three input qubits q_{i_j} (yellow, top), coupled dissipatively to an ancilla qubit a (blue, bottom), via a 4-local Lindbladian (represented by the four solid lines, and the solid dot in the center).

it is sufficient to evolve the network for time $t > \frac{1}{\gamma} \log \frac{1}{\epsilon}$.

That is, in order to guarantee

$$\langle C(x) | \text{Tr}_{\mathcal{X}} \mathcal{E}_t(\rho_x) | C(x) \rangle \geq 1 - \epsilon, \quad (8)$$

one requires $t > \frac{1}{\gamma} \log \frac{1}{\epsilon}$, where $\rho_x := |x\rangle\langle x| \otimes |1\rangle\langle 1|$, and $\text{Tr}_{\mathcal{X}}$ is the partial trace, tracing out $\mathcal{H}_{\mathcal{X}}$.

- 2) Coherences $|x\rangle\langle y|$ ($x \neq y$) are preserved, unless $C(x) \neq C(y)$.
- 3) If one wishes to compute a clause of just one or two literals (instead of three), the construction and general result is exactly the same except the projector Π^- of the Lindblad operator (2) acts on the appropriate one or two qubits respectively.
- 4) One can compute a disjunctive normal form (DNF) clause $l_1 \wedge l_2 \wedge l_3$ in a similar manner by replacing Π^- by $\Pi = |\vec{l}\rangle\langle \vec{l}|$ and σ^- by σ^+ in Eq. (2). In this case the ancilla qubit should be initialized as $|0\rangle$. As such, all of our subsequent results can be equivalently phrased through DNF formulas.

Point 2) in particular implies the existence of decoherence-free subspaces [17, 18], defined according to

$$\text{DFS}_c := \text{span} \{ |x\rangle : C(x) = c \}, \quad (9)$$

for $c = 0, 1$. That is, for $|\psi\rangle \in \text{DFS}_c$, one has $\text{Tr}_a[\mathcal{E}_t(|\psi\rangle\langle\psi| \otimes |1\rangle\langle 1|)] = |\psi\rangle\langle\psi|$, where the partial trace Tr_a traces out the ancilla qubit a .

The Hilbert space is partitioned in accordance with this observation: given an arbitrary quantum input $\rho \in \mathcal{L}(\mathcal{H}_{\mathcal{X}})$ to the dissipative network, the quantum map \mathcal{E}_t will evolve to the fixed point

$$\lim_{t \rightarrow \infty} \mathcal{E}_t(\rho \otimes |1\rangle\langle 1|) = \sum_{c \in \{0,1\}} \Pi_c \rho \Pi_c \otimes |c\rangle\langle c| \quad (10)$$

where $\Pi_c := \sum_{x: C(x)=c} |x\rangle\langle x|$ is a projector onto DFS_c (note $\sum_{c \in \{0,1\}} \Pi_c = \mathbb{I}$).

That is, the Hilbert space $\mathcal{H}_{\mathcal{X}}$ contains two ‘coherent sectors’ (DFSs) defined by C . States belonging to these sectors are preserved under map \mathcal{E}_t , and coherences between the sectors decay away exponentially as $e^{-\gamma t/2}$.

This simple dissipative network and the resulting decoherence-free subspaces can therefore be used to not only evaluate the clause $C(x)$ for $x \in \mathcal{X}$. One can now give meaning to and compute quantities such as $C(\psi)$ where $|\psi\rangle \in \text{DFS}_c$. In particular in this case the network will output $C(\psi) = c$. We will provide a more rigorous definition below in Sect. II B.

We lastly comment, that in the infinite time limit, upon tracing out the ancilla qubit, the state of the system $\mathcal{H}_{\mathcal{X}}$, is the same as under a purely dephasing Lindbladian defined by jump operator $O = \sum_x C(x) |x\rangle\langle x|$. In particular, our scheme (in the long-time limit) is equivalent to measurement of the observable O . That is, the engineered dissipation can be interpreted as *performing* the required measurement on the system.

B. Dissipative evaluation of arbitrary 3-CNF

Consider now the more general case where one wishes to evaluate a 3-CNF formula consisting of N clauses $C = \bigwedge_{i=1}^N C_i$. We show how to construct a dissipative network to achieve this, which uses N ancilla qubits, each of which is coupled to at most 3 of the input qubits, and evolves in a similar manner as Eq. (5). The full Hilbert space, of the ‘input’ qubits, and ancilla qubits is therefore of the form $\mathcal{H} = \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_a$, where \mathcal{H}_a is of dimension 2^N . We provide an illustration of this set-up in Fig. 2.

The structure is now much richer, and we will see the Hilbert space is actually partitioned into (at most) 2^N DFSs, according to the output of each of the N clauses.

The full Lindbladian acting on the system of $n+N$ qubits is now given by $\mathcal{L} = \sum_{i=1}^N \mathcal{L}_i$, where \mathcal{L}_i is a Lindblad generator defined by a single jump operator, of the form Eq. (2), which acts on the qubits defined by the i -th clause in C .

Proposition 2. For $\mathcal{E}_t := e^{t\mathcal{L}}$, with \mathcal{L} defined as above, the infinite time evolution of matrix element $|x\rangle\langle y| \otimes |1\rangle\langle 1|$, with $x, y \in \mathcal{X}$, is given by

$$\begin{aligned} \lim_{t \rightarrow \infty} \mathcal{E}_t(|x\rangle\langle y| \otimes |1\rangle\langle 1|^{\otimes N}) = \\ \begin{cases} |x\rangle\langle y| \otimes \bigotimes_{i=1}^N |c_i\rangle\langle c_i| & \text{if } \forall i \ C_i(x) = c_i = C_i(y) \\ 0 & \text{if } \exists i \text{ s.t. } C_i(x) \neq C_i(y). \end{cases} \end{aligned} \quad (11)$$

Moreover, under a finite time evolution of input state $|x\rangle$ ($x \in \mathcal{X}$), upon measuring the N ancilla qubits, to successfully obtain $C_i(x)$, $\forall i$ with probability $1 - \epsilon$, it is sufficient to evolve the network for time $t > O(\frac{1}{\gamma} \log \frac{N}{\epsilon})$.

Proof. Given a 3-CNF, $C = \bigwedge_{i=1}^N C_i$, we construct a 4-local Lindbladian $\mathcal{L} = \sum_{i=1}^N \mathcal{L}_i$, where \mathcal{L}_i is defined for each clause, with single jump operator of

the form Eq. (2). Here, \mathcal{L}_i acts on the three qubits defined by clause C_i , and the i th ancilla qubit.

The full evolution is in fact easy to calculate exactly, since the individual Lindbladians, \mathcal{L}_i all commute with each other. Moreover, as can be seen from Eq. (5), the action of a single \mathcal{L}_i only changes the state of the i th ancilla qubit.

These two facts and Prop. 1 imply

$$\begin{aligned} \mathcal{E}_t(|x\rangle\langle y| \otimes |1\rangle\langle 1|^{\otimes N}) &= \prod_{i=1}^N \mathcal{E}_t^{(i)}(|x\rangle\langle y| \otimes |1\rangle\langle 1|^{\otimes N}) \\ &= |x\rangle\langle y| \bigotimes_{i=1}^N \begin{cases} e^{-\gamma t} |1\rangle\langle 1| + (1 - e^{-\gamma t}) |c_i\rangle\langle c_i| & \text{if } C_i(x) = c_i = C_i(y) \\ e^{-\gamma t/2} |1\rangle\langle 1| & \text{if } C_i(x) \neq C_i(y) \end{cases} \end{aligned} \quad (12)$$

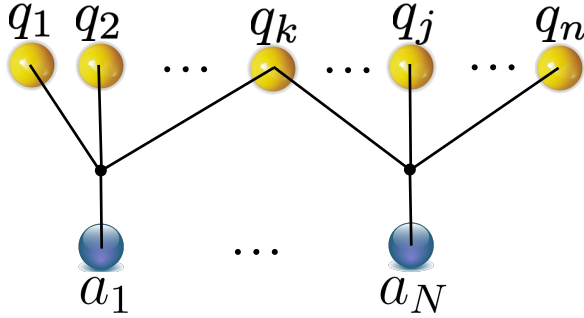


FIG. 2. Illustration of a dissipative computational network, capable of computing a 3-CNF with N clauses, on n variables. The q_i represent input qubits, and the a_i are ancilla qubits (where the i -th clause is evaluated). In this particular example, we show explicitly the clauses $C_1 = l_1 \vee l_2 \vee l_k$ and $C_N = l_k \vee l_j \vee l_n$, where l_i is either x_i or $\neg x_i$.

where $\mathcal{E}_t := e^{t \sum_{i=1}^N \mathcal{L}_i}$, and $\mathcal{E}_t^{(i)} := e^{t \mathcal{L}_i}$.

Taking $t \rightarrow \infty$ gives precisely Eq. (11).

We comment on errors associated with finite-time evolution, $t < \infty$. In particular, for input of the form $|x\rangle\langle x|$ ($x \in \mathcal{X}$), the probability of correctly obtaining via a projective measurement outcome $C_i(x)$ for the i -th clause is $1 - e^{-\gamma t}$. This is precisely the same calculation as in point 1) following Prop. 1. Since these N clauses are all independent, in order to correctly evaluate $C(x) = \bigwedge_{i=1}^N C_i(x)$ with probability $1 - \epsilon$, one requires

$$(1 - e^{-\gamma t})^N > 1 - \epsilon, \quad (13)$$

and hence $t > O(\frac{1}{\gamma} \log \frac{N}{\epsilon})$ [24].

□

One can see from Eq. (12) that this construction completely generalizes the results of the previous subsection (i.e. where $N = 1$), and therefore provides a richer structure than previously described. In particular, there are now up to 2^N decoherence-free subspaces, defined according to binary representation

$$\bar{C}(x) := (C_1(x), \dots, C_N(x)), \quad (14)$$

which uniquely determines the DFS to which $|x\rangle$ ($x \in \mathcal{X}$) belongs. In particular, the generalization of (9) is

$$\text{DFS}_{\bar{C}} := \text{span} \{ |x\rangle : \bar{C}(x) = \bar{C} \}. \quad (15)$$

If indeed $\bar{C}(x) = \bar{C}(y)$, then coherences of the form $|x\rangle\langle y|$ will be preserved under the time evolution of the dissipative quantum network, and if $|\psi\rangle \in \text{DFS}_{\bar{C}}$, then

$$\text{Tr}_{\bar{a}} [\mathcal{E}_t(|\psi\rangle\langle\psi| \otimes |1\rangle\langle 1|^{\otimes N})] = |\psi\rangle\langle\psi| \quad (16)$$

where $\text{Tr}_{\bar{a}}$ traces out the N ancilla qubits.

Similarly, for an arbitrary input $\rho \in L(\mathcal{H}_{\mathcal{X}})$, the infinite time evolved state is of the form

$$\begin{aligned} \rho_{\infty} &= \sum_{\bar{C} \in \{0,1\}^N} \Pi_{\bar{C}} \rho \Pi_{\bar{C}} \otimes |\bar{C}\rangle\langle\bar{C}|, \\ \Pi_{\bar{C}} &:= \sum_{x \in \mathcal{X} : \bar{C}(x) = \bar{C}} |x\rangle\langle x|, \end{aligned} \quad (17)$$

where $\Pi_{\bar{C}}^2 = \Pi_{\bar{C}}$ are projectors ($\sum_{\bar{C} \in \{0,1\}^N} \Pi_{\bar{C}} = \mathbb{I}$) over $\mathcal{H}_{\mathcal{X}}$.

In order for the final state $\rho_t := \mathcal{E}_t(\rho \otimes |1\rangle\langle 1|^{\otimes N})$ to be ϵ -close [25] to the steady state ρ_{∞} ,

$$\|\rho_t - \rho_{\infty}\|_1 < \epsilon, \quad (18)$$

requires (as shown in Appendix A)

$$t > O\left(\frac{n}{\gamma} + \frac{1}{\gamma} \log \frac{N}{\epsilon}\right). \quad (19)$$

From Eq. (17) it is apparent that upon evolving an initial state $|\psi\rangle \in \text{DFS}_{\bar{C}}$ (i.e. where $\Pi_{\bar{C}}|\psi\rangle = |\psi\rangle$), that one can determine \bar{C} without directly measuring or disturbing sub-system $\mathcal{H}_{\mathcal{X}}$ itself. That is, one can learn \bar{C} passively, whilst still retaining the state $|\psi\rangle$, since here $\rho_{\infty} = |\psi\rangle\langle\psi| \otimes |\bar{C}\rangle\langle\bar{C}|$.

In this case it is apparent that one can ‘classify’ a quantum state $|\psi\rangle \in \text{DFS}_{\bar{C}}$ according to $\bar{C}(|\psi\rangle) = \bar{C}$. The definition is ‘consistent’ in the sense that $\bar{C}(|x\rangle) = \bar{C}(x)$, for $x \in \mathcal{X}$, where $\bar{C}(x)$ is defined by Eq. (14). As such we will also write $\bar{C}(|\psi\rangle) = \bar{C}(\psi)$.

One can generalize this notion as follows:

Definition 2. Let $|\psi\rangle \in \mathcal{H}_{\mathcal{X}}$. Then the ‘DFS-classification’ of state $|\psi\rangle$ is given by a function $\tilde{C} : \mathcal{H}_{\mathcal{X}} \rightarrow [0, 1]^N$, defined by

$$\tilde{C}(|\psi\rangle) \equiv \tilde{C}(\psi) := \sum_{\bar{C} \in \{0, 1\}^N} \|\Pi_{\bar{C}}|\psi\rangle\|^2 \bar{C}, \quad (20)$$

where $\Pi_{\bar{C}}$ is given in Eq. (17).

The interpretation of Def. 2 should be clear: Each DFS in the Hilbert space corresponds to a unique vertex of the hypercube $[0, 1]^N$. Non-vertex points correspond to states $|\psi\rangle$ which are superpositions between various DFSs, which, according to Eq. (17), are the states that become mixed during the time evolution of the associated dissipative quantum circuit.

This motivates the straightforward observation:

Lemma 1. Let $|\psi\rangle \in \mathcal{H}_{\mathcal{X}}$. Then, $\tilde{C}(\psi) = \bar{C}$ $\iff |\psi\rangle \in \text{DFS}_{\bar{C}}$.

We now see that Def. 2 completely generalizes Def. 1; $x, y \in \mathcal{X}$ belong to the same partition of \mathcal{X} according to Def. 1, iff $\bar{C}(x) = \bar{C}(y)$ in Def. 2.

The i -th clause C_i evaluated on $|\psi\rangle$, is given by the i -th entry of $\tilde{C}(\psi)$ from Def. 2 – which we denote $\tilde{C}_i(\psi) \in [0, 1]$ – and shows that in the quantum case, according to this definition, Boolean logic is replaced by fuzzy logic [26], in accordance with the DFS partitioning of the space. One would of course require multiple samples to estimate this quantity for a given state.

One can now also lift the definition of the Boolean function C itself over classical bit strings to a function over the Hilbert space:

Definition 3. Let $|\psi\rangle \in \mathcal{H}_{\mathcal{X}}$. The function $\hat{C}_1 : \mathcal{H}_{\mathcal{X}} \rightarrow [0, 1]$ is defined by

$$\hat{C}_1(|\psi\rangle) \equiv \hat{C}_1(\psi) := \|\Pi_{\bar{e}}|\psi\rangle\|^2 \quad (21)$$

where $\bar{e} := (1, \dots, 1)$.

Def. 3 is also fully consistent with the definition of the underlying classifier C , and therefore generalizes this Boolean function. In particular, for $x \in \mathcal{X}$, $\hat{C}_1(|x\rangle) = C(x)$. Moreover, if $|\psi\rangle$ is of the general form

$$|\psi\rangle = \sum_{x: C(x)=c} a_x |x\rangle, \quad (22)$$

the network will output $\hat{C}_1(\psi) = c \in \{0, 1\}$, with probability $1 - \epsilon$.

Inspired by the classical evaluation of a CNF formula, $C(x) = \prod_{i=1}^N C_i(x)$ for $x \in \mathcal{X}$, one may alternatively be interested in perhaps the more natural (fuzzy) generalization:

Definition 4. Let $|\psi\rangle \in \mathcal{H}_{\mathcal{X}}$. The function $\hat{C}_2 : \mathcal{H}_{\mathcal{X}} \rightarrow [0, 1]$ is defined by

$$\hat{C}_2(|\psi\rangle) \equiv \hat{C}_2(\psi) := \prod_{i=1}^N \tilde{C}_i(\psi) \quad (23)$$

where $\tilde{C}_i(\psi)$ is the i -th entry of vector $\tilde{C}(\psi)$ in Def. 2.

This is also fully consistent with the underlying classical definition, i.e. $\hat{C}_2(|x\rangle) = C(x)$, for $x \in \mathcal{X}$. Moreover, under Def. 4, for $|\psi\rangle \in \text{DFS}_{\bar{C}}$, $\hat{C}_2(\psi) = \prod_{i=1}^N \bar{C}_i = \bar{C}$ as expected.

In this section, we have shown how one can lift classical notions pertaining to data classification in a natural manner to the quantum case. We provided a practical construction for implementing this physically, defining a local dissipative computational network, which naturally evaluates such classifiers, as in Prop. 2. Not only can the network evaluate the clauses $C_i(x)$ for $x \in \mathcal{X}$ of a CNF formula, but also give a fuzzy logic meaning to $C_i(\psi)$ for quantum states $|\psi\rangle$, and therefore, similarly to the CNF formula $C(\psi)$ itself. As discussed in Sect. I A, this framework applies also in the more general case where $C(x) \in \{0, 1\}^m$.

We now show how to use these ideas for practical applications relating to quantum learning theory and state preparation.

III. APPLICATIONS AND EXAMPLES

A. A dissipative quantum data classifier

A common classical machine learning task is, given labeled samples $(x, f(x))$ where $x \in \mathcal{X}$, and $f : \mathcal{X} \rightarrow \{0, 1\}^m$, to build a classifier $C : \mathcal{X} \rightarrow \{0, 1\}^m$, such that given a new sample, y , that $C(y) = f(y)$ with high probability.

There are many approaches which attempt to solve this problem classically, including linear regression models, Bayesian networks and artificial neural networks [27].

One such quantum generalization of this type of task is to learn $f(x)$ given samples of $n + m$ qubits of the form

$$|\psi_i\rangle = \sum_x a_x^{(i)} |x, f(x)\rangle, \quad (24)$$

where i labels the sample. Many well-known results in quantum learning theory are phrased in this manner, and quantum speed-ups have been demonstrated [6, 20, 28–31]. Using $|\psi_i\rangle$ as an input into our quantum network, acting only on the n qubits encoding x and the N ancilla qubits defined by the network (i.e. leaving alone the m qubits encoding $f(x)$), following from Eq. (17) the infinite time evolved state is

$$\sum_{\bar{C} \in \{0,1\}^N} (\Pi_{\bar{C}} \otimes \mathbb{I}) |\psi_i\rangle \langle \psi_i| (\Pi_{\bar{C}} \otimes \mathbb{I}) \otimes |\bar{C}\rangle \langle \bar{C}|. \quad (25)$$

Upon measurement of \bar{C} in the N ancilla qubits, the state of the $n + m$ qubits is (up to normalization)

$$\sum_{x: \bar{C}(x)=\bar{C}} a_x^{(i)} |x, f(x)\rangle. \quad (26)$$

That is, the sample $|\psi_i\rangle$ is projected into subspace $\text{DFS}_{\bar{C}}$.

We will show by example how using the general construction outlined in the previous section, one can build a trainable quantum network, which can be used to classify quantum data, in accordance with Defs. 2, 3, 4.

At a high level, the prescription of using our scheme to learn quantum data consists of four basic steps:

- i) Initialize network configuration: Define a system of 4-local Lindbladians which each act on up to three input qubits, and one unique ancilla qubit, as described by Eq. (2).
- ii) Evolve the network: Input is a quantum state of the general form Eq. (24). Evolve network for time $O(\frac{n}{\gamma} + \frac{1}{\gamma} \log \frac{N}{\epsilon})$, where N is the number of ancilla qubits, and ϵ the error associated with finite time evolution. The resulting state will be ϵ -close to Eq. (25).
- iii) Measurement: Perform post-processing of the quantum data, e.g. by performing a POVM.
- iv) Update: Based on the measurement result in step iii), update the network configuration and

return to step ii). If no update is required and the network has converged, subsequent evaluations of the network will correctly classify new quantum data (to error ϵ).

We demonstrate these general ideas by a simple and well known classical example (see e.g. Ref. [15]), where a priori the space of possible functions from which one is learning is of exponential size 2^{2^n} , but where the problem class can be learned efficiently, and moreover will require at most $2n$ ancilla qubits in our construction.

Note that the goal of the proceeding example is *not* to demonstrate a quantum advantage. Rather, it is to show that using the tools outlined in this work, certain classical tasks can be generalized to a quantum setting. Indeed, in the example below, the update procedure we use in step iv) is entirely classical in nature. In principle one could achieve the same outcome (learning the target function f) by first measuring each of the qubits in the computational basis – destroying any quantum coherence – and running the classical algorithm. By using our network however, we show, as the algorithm progresses the amount of quantum coherence preserved after step iii) of the procedure just mentioned, is increased. An interesting question is whether one can gain performance advantages using an algorithm which is inherently quantum.

1. Example: Learning the class of conjunctions

Consider the task of learning the function $f : \mathcal{X} \rightarrow \{0,1\}$ where f is guaranteed to be a pure conjunction. That is, f is of the form $f = f_1 \wedge \dots \wedge f_k$, where the clauses f_i contain just a single literal l_i that are of the form x_j or $\neg x_j$ for $j \in [n]$. Let us also write for f the corresponding vector of clauses $\bar{f} = (f_1, \dots, f_k)$. Then $\bar{f}(x) = (f_1(x), \dots, f_k(x))$ describes the evaluation of the k clauses on input $x \in \mathcal{X}$.

This is in fact an easy problem, and the classical algorithm to solve it is to start with the hypothesis $C = x_1 \wedge \neg x_1 \wedge \dots \wedge x_n \wedge \neg x_n$ (i.e. a conjunction of all possible literals), and whenever a positive labeled sample $(y, 1)$ is observed, for all $y_i = 1$, remove the literal $\neg x_i$ from C , and similarly for all $y_i = 0$, remove the literal x_i from C .

We will demonstrate the above in a dissipative quantum network, by utilizing $2n$ ancilla qubits, and therefore $3n$ qubits in total. The initial CNF formula defining the network is as in the previous paragraph, from which one constructs the 2-local Lindbladian $\mathcal{L} = \sum_{i=1}^{2n} \mathcal{L}_i$, where \mathcal{L}_i acts on input qubit $[i/2]$, and the i -th ancilla qubit. These \mathcal{L}_i are defined in a similar manner as in Eq. (2), instead now with

just 2 qubits. We show an example of this initial network in Fig. 3 (top), where the odd ancilla qubits (top in figure) compute the positive variables x_k , and the even ancillas compute the negations $\neg x_k$ ($k = 1, \dots, n$). Explicitly, the Lindbladian \mathcal{L}_{2k-1} is defined by Eq. 2 with $\Pi^\Gamma = |0\rangle\langle 0|$ acting on the k -th input qubit (yellow). Similarly, the Lindbladian \mathcal{L}_{2k} is defined by $\Pi^\Gamma = |1\rangle\langle 1|$ also acting on the k -th input qubit.

Let us first consider the case where labeled states are promised to be of the form $(|x\rangle, f(x))$, where $x \in \mathcal{X}$. One evolves the network under \mathcal{L} , and measures the $2n$ ancilla qubits, obtaining $C_i(x)$ where i labels each ancilla qubit. We will assume that the network is evolved for a sufficiently long time at each step so that the probability of incorrectly computing $C_i(x)$ is negligible.

Given a positive sample, $f(y) = 1$, one can delete any \mathcal{L}_i from the network which results in the i -th ancilla qubit measuring 0, since this implies $C(y) = 0$. The network is then re-set and run again, possibly now with fewer ancilla qubits. Repeating this process guarantees the convergence of C to f . The total number of times one must update the network in this manner is upper bound by $2n$. We show an example of the network training process in Fig. 3.

It is at this point interesting to note that one can perform the exact same algorithm if one receives labeled quantum states $(|\psi\rangle, f(\psi))$ where $|\psi\rangle$ is guaranteed to be a superposition of classical states all with the same evaluation on the clauses of f . In this case, following Defs. 3, 4, $f(\psi) \in \{0, 1\}$, and the input state is of the general form

$$|\psi\rangle = \sum_{x: \bar{f}(x)=\bar{f}'} a_x |x\rangle, \quad (27)$$

for some $\bar{f}' \in \{0, 1\}^k$ such that $f(\psi) = f'_1 \wedge \dots \wedge f'_k$, and a_x arbitrary normalized complex amplitudes. Note, these samples can also be described equivalently as in Eq. (24), where one first measures the qubit encoding $f(x)$, since in this case $f(x) \equiv f(\psi)$, for all x in the sum Eq. (27).

Evolving the network with input $|\psi\rangle$, and measuring the ancilla qubits – obtaining outcome $\bar{C} = (C_1, \dots, C_N) \in \{0, 1\}^N$ – results in a projection $|\psi_{\bar{C}}\rangle := \Pi_{\bar{C}}|\psi\rangle / \|\Pi_{\bar{C}}|\psi\rangle\|$ (as in Eq. (17)), where N is the number of ancilla qubits in the current network. Since by assumption $f(\psi) = f(\psi_{\bar{C}})$, if $C(\psi_{\bar{C}}) = C_1 \wedge \dots \wedge C_N$ disagrees with $f(\psi)$, one updates the network as above, by deleting the conflicting \mathcal{L}_i which define C .

Once the network is fully trained, given an unlabeled state $|\psi\rangle$ as above, one can evaluate $f(\psi)$ without directly measuring $|\psi\rangle$ itself, i.e., one only needs to observe the ancilla qubits. Moreover, since by assumption an input state $|\psi\rangle \in \text{DFS}_{\bar{f}'}$ for some

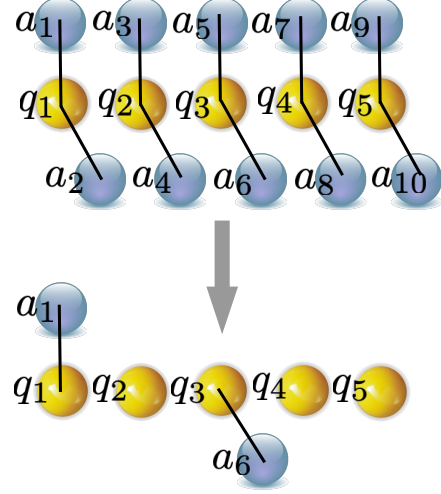


FIG. 3. Example of a network learning the conjunction $f = x_1 \wedge \neg x_3$, for $n = 5$. Ancilla qubits are in blue labeled a_i , and the n data qubits by q_i . The initial network (top) is configured to compute $x_1 \wedge \neg x_1 \wedge \dots \wedge x_5 \wedge \neg x_5$ as described by the learning algorithm of the main text. The top row of ancilla qubits (odd) compute the literals x_i , and the bottom row (even) the negations $\neg x_i$. After training for a sufficiently long time so that the network has converged, the final network is given by the bottom diagram, which now has just two ancilla qubits (blue), corresponding to the computation of $x_1 \wedge \neg x_3$. The Hilbert space therefore has 4 associated DFSs. Input (data) qubits are shown in yellow.

\bar{f}' , and therefore $\Pi_{\bar{f}'}|\psi\rangle = |\psi\rangle$, no coherence is lost under the network evolution, and the initial state of the input qubits is the same as the final state; that is, one can passively classify the state $|\psi\rangle$ (i.e. evaluate $f(\psi)$), whilst still retaining the quantum state $|\psi\rangle$. One can therefore classify $|\psi\rangle$, and still use $|\psi\rangle$ for subsequent computations.

This is quite a unique occurrence, since typically to obtain any information about a quantum state requires at least some destructive measurement to take place.

B. Dissipative preparation of PAC states

In the quantum version [20] of the Probably Approximately Correct (PAC) model [19], copies of the state $|\psi\rangle = \sum_{x \in \mathcal{X}} \sqrt{p(x)} |x, f(x)\rangle$ are provided, where $p(x)$ are probabilities, and $f(x) \in \{0, 1\}$ a Boolean function which is to be learned. Quantum states of this type, arising from a ‘quantum example oracle’, are typically represented by some oracular unitary U_f such that $U_f|x, b\rangle = |x, b \oplus f(x)\rangle$. We show here that one can produce these states in a fundamentally different manner using a dissipative

(non-unitary) network of the type outlined in this work.

The goal is to prepare quantum PAC states given $\sum_{x \in \mathcal{X}} \sqrt{p(x)} |x\rangle$. The procedure which follows also applies to the more general setting where the amplitudes are complex.

We make a small modification to the Lindblad operators L in Eq. (2) as follows:

$$L = \Pi^\top \otimes \sigma^- + (\mathbb{I} - \Pi^\top) \otimes \sigma^+. \quad (28)$$

Then we have:

Proposition 3. *Let L be as in Eq. (28) defined by clause $C = l_1 \vee l_2 \vee l_3$. Then the infinite time evolution of matrix element $|x\rangle\langle y|$ under the network, given the ancilla state initialized as $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ is*

$$\mathcal{E}_\infty(|x\rangle\langle y| \otimes |+\rangle\langle +|) = |x\rangle\langle y| \otimes |C(x)\rangle\langle C(y)|. \quad (29)$$

Proof. To see this use that \mathcal{E}_t is linear, and consider $\mathcal{E}_\infty(|x\rangle\langle y| \otimes |i\rangle\langle j|)$, for $i, j \in \{0, 1\}$ separately.

Consider the four cases:

$$\begin{aligned} L|x, 0\rangle &= \begin{cases} 0 & \text{if } C(x) = 0 \\ |x, 1\rangle & \text{if } C(x) = 1 \end{cases} \\ L|x, 1\rangle &= \begin{cases} |x, 0\rangle & \text{if } C(x) = 0 \\ 0 & \text{if } C(x) = 1, \end{cases} \end{aligned} \quad (30)$$

and similarly $L^\dagger L|x, i\rangle = \delta_{i, -C(x)} |x, i\rangle$ where $i \in \{0, 1\}$.

One can then write

$$\begin{aligned} \mathcal{L}(|x\rangle\langle y| \otimes |i\rangle\langle j|) &= \\ \gamma |x\rangle\langle y| \otimes [\delta_{i, -C(x)} \delta_{j, -C(y)} |C(x)\rangle\langle C(y)| - & \\ \frac{1}{2} |i\rangle\langle j| (\delta_{i, -C(x)} + \delta_{j, -C(y)})]. & \end{aligned} \quad (31)$$

From this one can check – in a similar manner as the proof of Prop. 1 – that if $C(x) = C(y)$, then

$$\mathcal{E}_\infty(|x\rangle\langle y| \otimes |i\rangle\langle j|) = \delta_{i, j} |x\rangle\langle y| \otimes |C(x)\rangle\langle C(y)|. \quad (32)$$

Similarly, if $C(x) \neq C(y)$, one has

$$\mathcal{E}_\infty(|x\rangle\langle y| \otimes |i\rangle\langle j|) = \delta_{i, -j} |x\rangle\langle y| \otimes |C(x)\rangle\langle C(y)|. \quad (33)$$

That is, in this latter case, coherences $|x\rangle\langle y|$ remain only when $i \neq j$.

The result follows noting that $|+\rangle\langle +| = \sum_i \frac{1}{2} |i\rangle\langle i| + \sum_{i \neq j} \frac{1}{2} |i\rangle\langle j|$. \square

From Prop. 3 it immediately follows that under the network with N clauses one has in the long time limit [32] ($t \rightarrow \infty$)

$$\sum_{x \in \mathcal{X}} \sqrt{p(x)} |x\rangle |+\rangle^{\otimes N} \rightarrow \sum_{x \in \mathcal{X}} \sqrt{p(x)} |x\rangle |\bar{f}(x)\rangle \quad (34)$$

where $|\bar{f}(x)\rangle = \otimes_{i=1}^N |f_i(x)\rangle$ with $f_i(x)$ the i -th clause of $f(x)$ realized as a 3-CNF. Since $|\bar{f}(x)\rangle$ contains all the information required to compute $f(x)$, states of this type can equivalently be used in the PAC framework. The only complication is that one now has N additional qubits in the scheme, instead of just 1. Nevertheless, if one is interested in machine learning involving CNF or DNF formulas [33] this framework provides an alternative mechanism for obtaining quantum PAC states. This is therefore relevant even in the original formulation of the quantum PAC model by Ref. [20] for which formulas in DNF over the uniform distribution ($p(x) = 1/2^n$) can be efficiently PAC learned on a quantum computer (i.e. in polynomial time), as compared to the best known quasi-polynomial classical algorithm [34].

C. Probabilistic preparation of quantum states

By Eq. (17), upon measuring state $|\bar{C}\rangle$ of the ancilla qubits, the resulting time-evolved input state is, with probability $1 - \epsilon$, $\Pi_{\bar{C}} \rho \Pi_{\bar{C}} / \text{Tr}(\Pi_{\bar{C}} \rho)$, where ρ was the initial state.

One can use this to prepare quantum states. In particular, to prepare state $|\psi\rangle$, one requires a suitable, easy to prepare input state, $|\psi_0\rangle$, and to define an appropriate CNF formula so that, for some $\bar{C} \in \{0, 1\}^N$, one has $\Pi_{\bar{C}} |\psi_0\rangle \propto |\psi\rangle$. Then, upon measuring \bar{C} in the ancilla qubits, which occurs with probability $\langle \psi_0 | \Pi_{\bar{C}} | \psi_0 \rangle$, one has prepared state $|\psi\rangle$.

One could also prepare an ensemble of states in this manner.

1. Preparation of entangled states

Consider for simplicity, two qubits, which we wish to entangle, that are initially prepared in the product state $|\psi_0\rangle = |+\rangle \otimes |+\rangle$. Here, in the z -eigenbasis, $|+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. We wish to prepare the Bell state $|\psi^+\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

This procedure will be effectively the same as performing a projective measurement on $|\psi_0\rangle$, although we will in fact not perform any measurement over $\mathcal{H}_{\mathcal{X}}$. This example is interesting therefore for two reasons: 1) dissipation is used directly to create entanglement, and 2) it provides a novel manner in which to actually perform a measurement of a quantum system.

To achieve this task, we take the 2-CNF, $C = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) =: C_1 \wedge C_2$, with associated

projectors

$$\begin{aligned}\Pi_{(0,0)} &= 0, \Pi_{(0,1)} = |01\rangle\langle 01|, \Pi_{(1,0)} = |10\rangle\langle 10|, \\ \Pi_{(1,1)} &= |00\rangle\langle 00| + |11\rangle\langle 11|.\end{aligned}\quad (35)$$

From this, utilizing two ancilla qubits that are initialized in the $|1\rangle$ state, one can construct a following 3-local Lindbladian, $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$. These Lindbladians are defined by Lindblad operators $L_1 = |01\rangle\langle 01| \otimes \sigma_1^-$, $L_2 = |10\rangle\langle 10| \otimes \sigma_2^-$, where the second term in the tensor product acts on a single ancilla qubit, labeled by 1, 2 respectively. See inset of Fig. 4 for a schematic of the network.

This evolution will partition the space into three DFSs, and in particular, the infinite time evolved state is

$$\rho_\infty = \sum_{c_1, c_2 \in \{0,1\}} P_{c_1, c_2} |\psi_{c_1, c_2}\rangle\langle \psi_{c_1, c_2}| \otimes |c_1, c_2\rangle\langle c_1, c_2|, \quad (36)$$

with the probability of being in each sector after measurement of the two ancilla qubits $P_{\bar{C}} = \langle \psi_0 | \Pi_{\bar{C}} | \psi_0 \rangle$,

$$P_{0,0} = 0, P_{0,1} = \frac{1}{4} = P_{1,0}, P_{1,1} = \frac{1}{2}. \quad (37)$$

The states in each sector are given by

$$\begin{aligned}|\psi_{0,1}\rangle &= \frac{\Pi_{(0,1)}|\psi_0\rangle}{\sqrt{P_{0,1}}} = |01\rangle \\ |\psi_{1,0}\rangle &= \frac{\Pi_{(1,0)}|\psi_0\rangle}{\sqrt{P_{1,0}}} = |10\rangle \\ |\psi_{1,1}\rangle &= \frac{\Pi_{(1,1)}|\psi_0\rangle}{\sqrt{P_{1,1}}} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).\end{aligned}\quad (38)$$

Note, the $(c_1, c_2) = (0,0)$ sector is empty since at least one of the two clauses in C must be satisfied by construction. That is, in this case, there is no such state $|\psi_{0,0}\rangle$.

After evolving the network for a sufficiently long time, one can post-select on the ancilla qubits with success probability $\frac{1}{2}$ to pick out the Bell state:

$$\rho_{1,1}(t) := \text{Tr}_{\bar{a}} \left[\tilde{\Pi}_{1,1} \rho(t) \tilde{\Pi}_{1,1} \right] \xrightarrow{t \rightarrow \infty} |\psi^+\rangle\langle \psi^+| \quad (39)$$

where

$$\begin{aligned}\rho(t) &:= \mathcal{E}_t(|\psi_0\rangle\langle \psi_0| \otimes |1,1\rangle\langle 1,1|), \\ \tilde{\Pi}_{1,1} &:= \mathbb{I}_4 \otimes |1,1\rangle\langle 1,1|.\end{aligned}\quad (40)$$

In Fig. 4 we provide a numerical verification of this scheme, in which the state of the qubits, upon post-selecting on $(1,1)$ on the two ancilla qubits, converges exponentially quickly to the desired maximally entangled state.

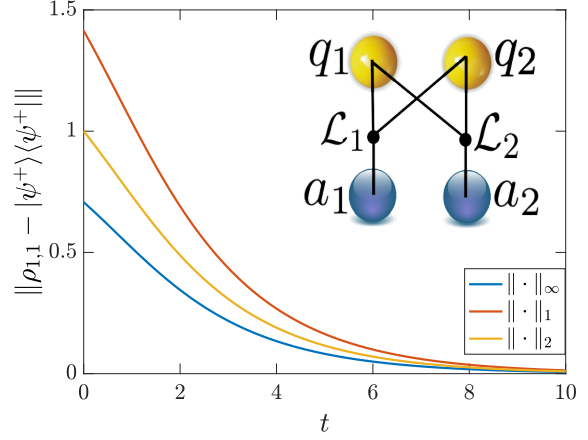


FIG. 4. Distance to maximally entangled Bell state $|\psi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, under dissipative evolution for time t (in units of $1/\gamma$). Here, $\rho_{1,1}$ is the state after post-selecting on measuring 1 in both ancilla qubits Eq. (39), which occurs with probability $1/2$. Initial state is the product state $|+, +\rangle$. We use three different norms to evaluate the distance [35]. Inset: the connectivity of the 2+2 qubit dissipative network. $q_{1,2}$ label the two qubits which we wish to entangle, and $a_{1,2}$ are the ancillas used in our scheme. The solid dots represent the Lindbladian connectivity, labeled respectively as $\mathcal{L}_{1,2}$.

2. Generating a superposition of all solutions to 3-SAT problem

3-SAT problems are NP-hard optimization problems, where one must find the solution(s) $C(x) = 1$ to a 3-CNF formula C .

Given a 3-SAT problem C over n literals, one can construct a dissipative network (of 4-local Lindbladians) with N ancilla qubits, where N is the number of clauses in the 3-SAT problem.

The input state to the network is initialized in the maximal superposition state, $|\psi\rangle = \frac{1}{2^{n/2}} \sum_{i=1}^{2^n} |i\rangle$. After evolving the network for time $t > O(\frac{n}{\gamma} + \frac{1}{\gamma} \log \frac{N}{\epsilon})$, upon measurement of $|1\rangle^{\otimes N}$ of the ancilla qubits, the resulting state of the input qubits is guaranteed (with probability $1 - \epsilon$) to be

$$|S\rangle = \frac{1}{\sqrt{N_s}} \sum_{s=1}^{N_s} |s\rangle \quad (41)$$

where each classical bit string s is such that $C(s) = 1$. Moreover, there are no other bit-strings with this property. That is, $|S\rangle$ is the uniform superposition of all solutions to the 3-SAT problem. We strongly stress however that in general it is exponentially unlikely to observe such a state; we are not claiming to have an efficient algorithm to solve SAT problems.

Nevertheless, this general methodology shows po-

tential promise in obtaining a fair sampling for certain problems, which is typically challenging on current quantum optimization devices [36].

IV. DISCUSSION

We have shown how to evaluate 3-CNF (and DNF) clauses (and hence arbitrary Boolean functions) using dissipative quantum dynamics. We did this by providing a way to construct a dissipative network consisting of 4-local Lindbladians, where upon measuring a subset of the qubits in the network, one can evaluate the CNF formula. We also showed that errors associated with a finite time evolution can be made arbitrarily small. The CNF structure naturally partitions the Hilbert space into sectors in which coherence is preserved, i.e., decoherence-free subspaces. This provides a route to generalize classical notions of data classification, lifting the definitions to the case where the data is quantum in nature. In particular, we achieve this task by classifying quantum data according to the partition, or DFS, to which a particular state belongs, which itself is defined by the dissipative network.

This has applications in state preparation, and perhaps more interestingly, machine learning. We provided one such example, but have hopefully have demonstrated the general applicability of these techniques. Indeed, one can make a connection between the work presented here, and a Hopfield recurrent neural network, which we now comment on.

A Hopfield network is a classical pattern recognition and classification machine learning model, where the ‘patterns’ or ‘memories’ one wishes to associate new data, are local minima of a pre-defined energy function. In particular, these patterns act like fixed points of the total space \mathcal{X} of a dynamical system (where the dynamics is described entirely by the energy function). We make an analogy between this classical dynamical system, and the quantum dynamical systems described in this work defined entirely by dissipative dynamics with attractive fixed points.

In our model, which is morally similar, data is classified according to the DFSs of the computational network, which are analogous to the patterns of the corresponding Hopfield network. However, not only can the dissipative networks presented in this work classify classical data, e.g. compute a function $C(x) \in \{0, 1\}$ for $x \in \mathcal{X}$, they can also classify quantum data in a consistent manner. Moreover, given a quantum state $|\psi\rangle$ associated to a particular memory pattern (i.e. a DFS), one can evaluate $C(\psi)$ passively, without disturbing the state $|\psi\rangle$.

We lastly mention that the same type of construction given in this work may of course be demonstrated through local unitary evolutions and single-qubit projective measurements alone. Though this is true, it misses perhaps the main point; a demonstration that engineered dissipation can be used as a resource for certain machine learning and classification tasks. We have given a precise construction for implementing the tasks described above, where the engineered dissipation is essentially performing the measurements of the relevant observables, required for classifying quantum data according to Defs. 2, 3, 4. Indeed, similar arguments can be made against *any* type of dissipative based scheme. In particular, it is known that the dissipative model of quantum computation is no more powerful than the unitary gate model, and that therefore the latter can efficiently simulate the former [2]. Nevertheless, it is a conceptually interesting observation that purely non-unitary dynamics can be used to carry out tasks relevant for machine learning.

This work provides a well defined and consistent starting point for generalizing certain aspects of classical machine learning theory, of which numerous applications could benefit. We highlighted simple examples for demonstrative purposes, but hope it is clear that the general protocols can be modified in a multitude of ways to be made applicable for different applications. It would be interesting for example to apply these techniques to learning quantum data in the PAC framework [15, 19, 20, 37].

V. ACKNOWLEDGMENTS

We thank Evgeny Mozgunov for reading the paper and providing useful comments. P.Z. acknowledges partial support from the NSF award PHY-1819189. L.C.V. acknowledges partial support from the Air Force Research Laboratory award no. FA8750-18-1-004. L.C.V. also acknowledges that this research is based upon work (partially) supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the U.S. Army Research Office contract W911NF-17-C-0050. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

-
- [1] F. Verstraete, M. M. Wolf, and J. Ignacio Cirac, “Quantum computation and quantum-state engineering driven by dissipation,” *Nat. Phys.* **5**, 633 (2009).
 - [2] M. Kliesch, T. Barthel, C. Gogolin, M. Kastoryano, and J. Eisert, “Dissipative Quantum Church-Turing Theorem,” *Phys. Rev. Lett.* **107**, 120501 (2011).
 - [3] P. Zanardi, J. Marshall, and L. Campos Venuti, “Dissipative universal Lindbladian simulation,” *Phys. Rev. A* **93**, 022312 (2016).
 - [4] J. Marshall, L. Campos Venuti, and P. Zanardi, “Modular quantum-information processing by dissipation,” *Phys. Rev. A* **94**, 052339 (2016).
 - [5] B. Kraus, H. P. Büchler, S. Diehl, A. Kantian, A. Micheli, and P. Zoller, “Preparation of entangled states by quantum Markov processes,” *Phys. Rev. A* **78**, 042307 (2008).
 - [6] E. Bernstein and U. Vazirani, “Quantum Complexity Theory,” in *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC ’93 (ACM, New York, NY, USA, 1993) pp. 11–20.
 - [7] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum Algorithm for Linear Systems of Equations,” *Phys. Rev. Lett.* **103**, 150502 (2009).
 - [8] S. Lloyd, S. Garnerone, and P. Zanardi, “Quantum algorithms for topological and geometric analysis of data,” *Nat. Commun.* **7**, 10138 (2016).
 - [9] P. Rebentrost, T. R. Bromley, C. Weedbrook, and S. Lloyd, “Quantum Hopfield neural network,” *Phys. Rev. A* **98**, 042308 (2018).
 - [10] A. Monras, A. Beige, and K. Wiesner, “Hidden Quantum Markov Models and non-adaptive read-out of many-body states,” [arXiv:1002.2337](https://arxiv.org/abs/1002.2337) (2010).
 - [11] M. Schuld, I. Sinayskiy, and F. Petruccione, “Quantum walks on graphs representing the firing patterns of a quantum neural network,” *Phys. Rev. A* **89**, 032333 (2014).
 - [12] M. Schuld, I. Sinayskiy, and F. Petruccione, “The quest for a Quantum Neural Network,” *Quantum Inf. Process.* **13**, 2567 (2014).
 - [13] P. Rotondo, M. Marcuzzi, J. P. Garrahan, I. Lesanovsky, and M. Müller, “Open quantum generalisation of Hopfield neural networks,” *J. Phys. A* **51**, 115301 (2018).
 - [14] J.ichi Inoue, “Pattern-recalling processes in quantum Hopfield networks far from saturation,” *J. of Phys.: Conference Series* **297**, 012012 (2011).
 - [15] M. J. Kearns and U. V. Vazirani, *An Introduction to Computational Learning Theory* (MIT Press, 1994).
 - [16] C. Wang and J. M. Gertler, “Directional Transfer of Quantum Information by Dissipation Engineering,” [arXiv:1809.03571](https://arxiv.org/abs/1809.03571) (2018).
 - [17] P. Zanardi and M. Rasetti, “Noiseless quantum codes,” *Phys. Rev. Lett.* **79**, 3306 (1997).
 - [18] D. A. Lidar, I. L. Chuang, and K. B. Whaley, “Decoherence-free subspaces for quantum computation,” *Phys. Rev. Lett.* **81**, 2594 (1998).
 - [19] L. G. Valiant, “A Theory of the Learnable,” *Commun. ACM* **27**, 1134 (1984).
 - [20] N. H. Bshouty and J. C. Jackson, “Learning DNF over the Uniform Distribution Using a Quantum Example Oracle,” *SIAM J. Comput.* **28**, 1136 (1999).
 - [21] G. Lindblad, “On the Generators of Quantum Dynamical Semigroups,” *Comm. Math. Phys.* **48**, 119 (1976).
 - [22] V. Gorini, A. Kossakowski and E.C.G Sudarshan, “Completely positive dynamical semigroups of N-level systems,” *J. Math. Phys.* **17**, 821 (1976).
 - [23] $\sigma^- = |0\rangle\langle 1|$ is defined in the eigen-basis of $\sigma^z = |1\rangle\langle 1| - |0\rangle\langle 0|$.
 - [24] We wish to find t such that $(1 - e^{-\gamma t})^N > 1 - \epsilon$. Noting that $e^{-\epsilon} > 1 - \epsilon$, we can instead bound $(1 - e^{-\gamma t})^N > e^{-\epsilon}$. Using the identity $-\log(1 - x) < \frac{x}{1-x}$ with $x = e^{-\gamma t}$ gives $\frac{Ne^{-\gamma t}}{1 - e^{-\gamma t}} < \epsilon$ which can be rearranged to give the result in the main text, $t > O(\frac{1}{\gamma} \log \frac{N}{\epsilon})$.
 - [25] Here $\|\cdot\|_1$ is the trace norm: $\|X\|_1 = \text{Tr}|X|$ with $|X| = \sqrt{X^\dagger X}$.
 - [26] L. A. Zadah, “Fuzzy sets,” *Information and Control* **8**, 338 (1965).
 - [27] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. (Pearson, 2010).
 - [28] E. Aïmeur, G. Brassard, and S. Gambs, “Machine Learning in a Quantum World,” in *Advances in Artificial Intelligence* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 431–442.
 - [29] E. Aïmeur, G. Brassard, and S. Gambs, “Quantum speed-up for unsupervised learning,” *Machine Learning* **90**, 261 (2013).
 - [30] D. R. Simon, “On the power of quantum computation,” in *Proc. 35th Annual Symposium on Foundations of Computer Science* (1994) pp. 116–123.
 - [31] R. A. Servidio, “Separating Quantum and Classical Learning,” in *Automata, Languages and Programming* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2001) pp. 1065–1080.
 - [32] Similar arguments as in Appendix A apply to Prop. 3 regarding the evolution time as a function of error ϵ , which is therefore still $t > O(\frac{n}{\gamma} + \frac{1}{\gamma} \log \frac{N}{\epsilon})$.
 - [33] As mentioned in point 4 following Prop. 1, our framework can equivalently be used to evaluate both CNF and DNF formulas.
 - [34] S. Arunachalam and R. de Wolf, “A Survey of Quantum Learning Theory,” [arXiv:1701.06806](https://arxiv.org/abs/1701.06806) (2017).
 - [35] Here, $\|\cdot\|_1$ is the trace norm, $\|\cdot\|_2$ the Hilbert-Schmidt norm, and $\|\cdot\|_\infty$ the maximum singular value norm.
 - [36] B. H. Zhang, G. Wagenbreth, V. Martin-Mayor, and I. Hen, “Advantages of Unfair Quantum Ground-State Sampling,” *Sci. Rep.* **7**, 1044 (2017).
 - [37] S. Arunachalam and R. de Wolf, “Optimal Quantum Sample Complexity of Learning Algorithms,” in *Proc. 32nd Computational Complexity Conference*, CCC ’17 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Germany, 2017) pp. 25:1–25:31.

Appendix A: Proof of Eq. (19)

Let $\rho_t^{(xy)} = \mathcal{E}_t(|x\rangle\langle y| \otimes |1\rangle\langle 1|^{\otimes N})$. By the triangle inequality and linearity of \mathcal{E}_t , we have $\|\rho_t - \rho_\infty\|_1 \leq 2^{2n} \|\rho_t^{(xy)} - \rho_\infty^{(xy)}\|_1$. We set $\|\rho_t^{(xy)} - \rho_\infty^{(xy)}\|_1 < \epsilon/2^{2n}$ and directly compute this norm. There are two general cases i) $\bar{C}(x) = \bar{C}(y)$, ii) $\bar{C}(x) \neq \bar{C}(y)$.

In i), let us first consider the case where $\bar{C}_i(x) = 0 = \bar{C}_i(y), \forall i$. Then $\rho_\infty^{(xy)} = |x\rangle\langle y| \otimes |\bar{0}\rangle\langle \bar{0}|$ where $|\bar{0}\rangle \equiv |0\rangle^{\otimes N}$. Then, using Eq. (12),

$$\begin{aligned} & (\rho_t^{(xy)} - \rho_\infty^{(xy)})^\dagger (\rho_t^{(xy)} - \rho_\infty^{(xy)}) \\ &= \sum_{\bar{C} \in \{0,1\}^N} \alpha_{\bar{C}}^2 |y\rangle\langle y| \otimes |\bar{C}\rangle\langle \bar{C}| \end{aligned} \quad (\text{A1})$$

with

$$\begin{aligned} \alpha_{\bar{C} \neq \bar{0}} &= e^{-h(\bar{C})\gamma t} (1 - e^{-\gamma t})^{N-h(\bar{C})} \\ \alpha_{\bar{0}} &= \sqrt{(1 - e^{-\gamma t})^{2N} - 2(1 - e^{-\gamma t})^N + 1} \\ &= 1 - (1 - e^{-\gamma t})^N \end{aligned} \quad (\text{A2})$$

where $h(\bar{C})$ is the Hamming weight of bit-string \bar{C} . The form of $\alpha_{\bar{0}}$ comes from the cross terms in Eq. (A1).

Taking the matrix square root of Eq. (A1) (which is diagonal), followed by the trace, gives

$$\begin{aligned} \|\rho_t^{(xy)} - \rho_\infty^{(xy)}\|_1 &= 1 - (1 - e^{-\gamma t})^N + \\ &\sum_{k=1}^N \binom{N}{k} e^{-k\gamma t} (1 - e^{-\gamma t})^{N-k} \end{aligned} \quad (\text{A3})$$

which can be rearranged (by the Binomial theorem) to

$$\|\rho_t^{(xy)} - \rho_\infty^{(xy)}\|_1 = 2(1 - (1 - e^{-\gamma t})^N). \quad (\text{A4})$$

Similarly as in the proof of Prop. 2, to guarantee this is less than $\epsilon/2^{2n}$ requires $t > O(\frac{n}{\gamma} + \frac{1}{\gamma} \log \frac{N}{\epsilon})$ as required.

Note, if instead $\bar{C}(x) = \tilde{C} = \bar{C}(y) \neq \bar{0}$, the only difference is that in Eq. (A4), N is replaced by $N - h(\tilde{C})$, and therefore the final conclusion remains unchanged.

For case ii), $\rho_\infty^{(xy)} = 0$, and if there are m clauses which differ between x and y (and all others evaluate to 0), one instead bounds

$$\begin{aligned} e^{-\gamma t m/2} \left[\sum_{k=0}^{N-m} \binom{N-m}{k} e^{-\gamma k t} (1 - e^{-\gamma t})^{N-m-k} \right] \\ < \epsilon/2^{2n}, \end{aligned} \quad (\text{A5})$$

which by the Binomial theorem gives $t > O(\frac{1}{\gamma} \log \frac{2^n}{\epsilon})$. As before, in the case where the are clauses $C_i(x) = C_i(y) = 1$, one instead replaces N in Eq. (A5) with $N - H$, where H is the number of such clauses. The result is unchanged.

Overall then, to achieve $\|\rho_t - \rho_\infty\|_1 < \epsilon$, one requires evolution time $t > O(\frac{n}{\gamma} + \frac{1}{\gamma} \log \frac{N}{\epsilon})$, i.e. Eq. (19).