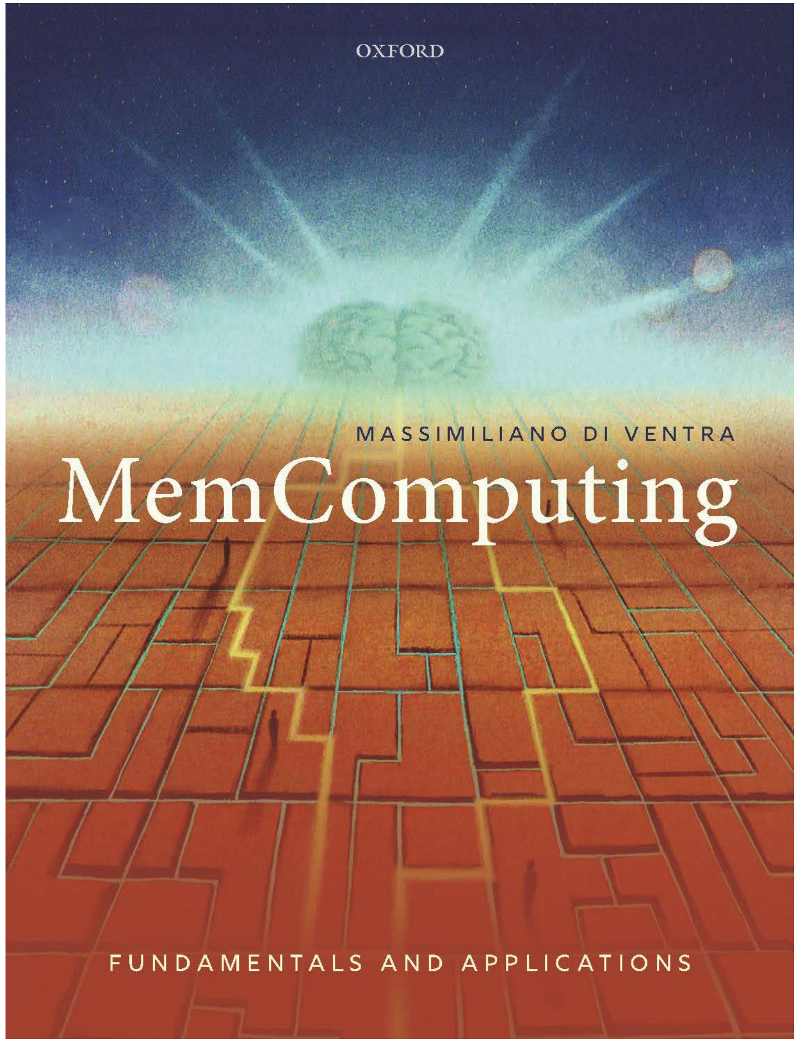
Massimiliano Di Ventra



www.memcpu.com

(SaaS available)

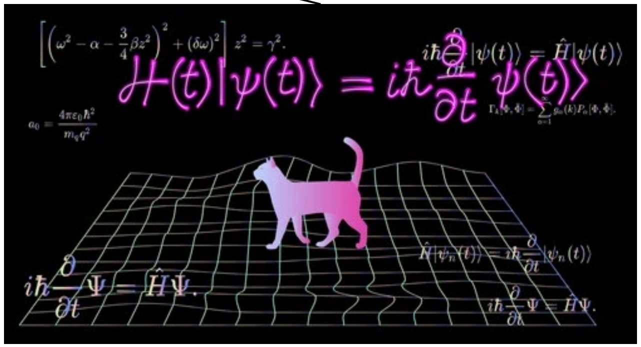
What is it useful for?

Tough, real-life problems

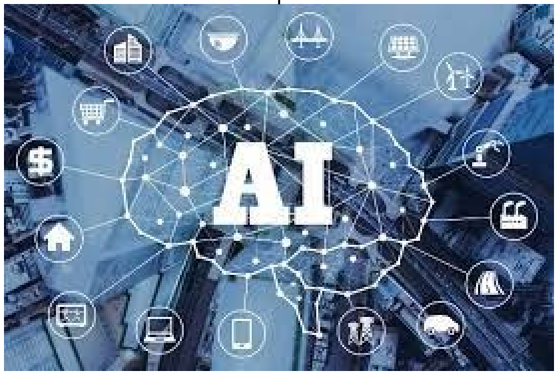
Many more…

results

Logistics, transportation, encryption…



Quantum phenomena, Materials Science, Devices,…



Machine Learning autonomous vehicles…

The Turing way

- What does it mean "to compute", the Turing way?

?

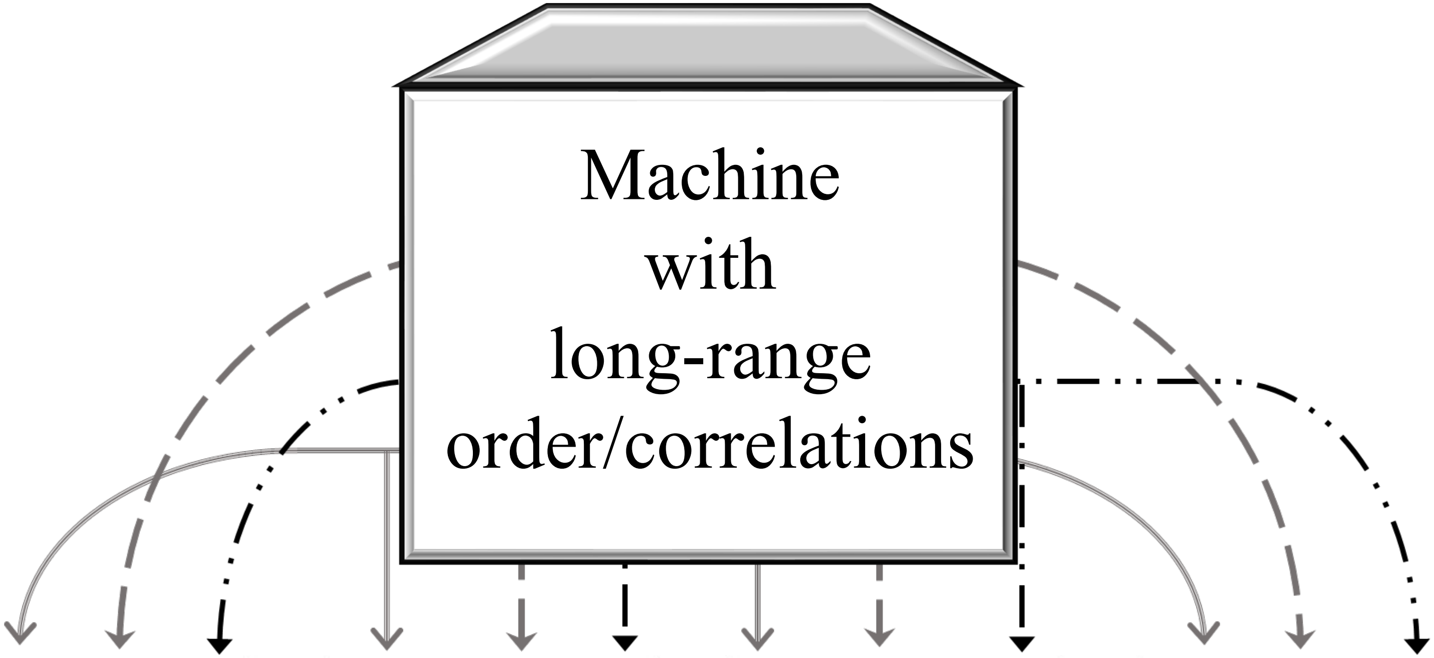
$$Z^n \xrightarrow{\delta} Z^n$$

<u>Algorithms</u>

Alan Turing (1912 – 1954)

A. M. Turing, "On computational numbers, with an application to the entscheidungsproblem," *Proc. of the London Math. Soc.*, vol. 42, pp. 230–265, 1936.

Why some problems are so difficult to solve?
(a physicist perspective)



TRUE

$$1 \quad = (\neg v_1 \vee v_2 \vee v_3) \wedge \underbrace{(\neg v_1 \vee \neg v_2 \vee v_3)}_{\text{clause}} \wedge (v_1 \vee \neg v_2 \vee \neg v_3) \wedge (\neg v_1 \vee v_2 \vee \neg v_3)$$

variable   OR   AND

from <u>short</u> to <u>long</u> wavelengths

(high energy to low energy, Renormalization Group)

Critical dynamics?

Physics to compute?

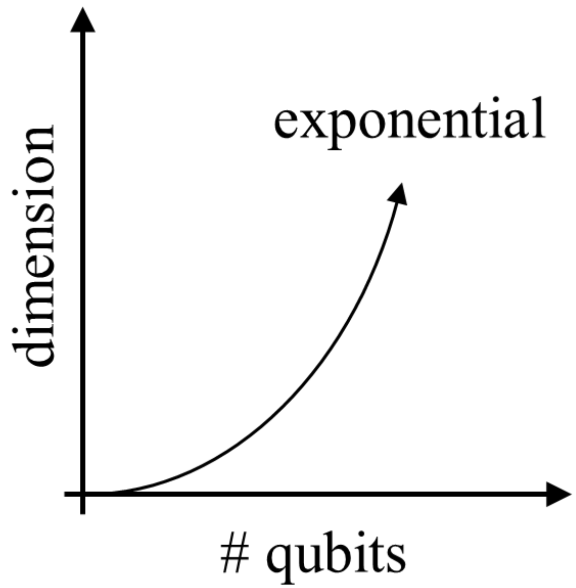
—The theory of computation has traditionally been studied almost entirely in the abstract, as a topic in pure Mathematics. This is to miss the point of it. Computers are physical objects, and computations are physical processes. What computers can or cannot compute is determined by the laws of Physics alone, and not by pure Mathematics.—
David Deutsch (1953–)

what Physics?

Quantum vs. non-quantum systems

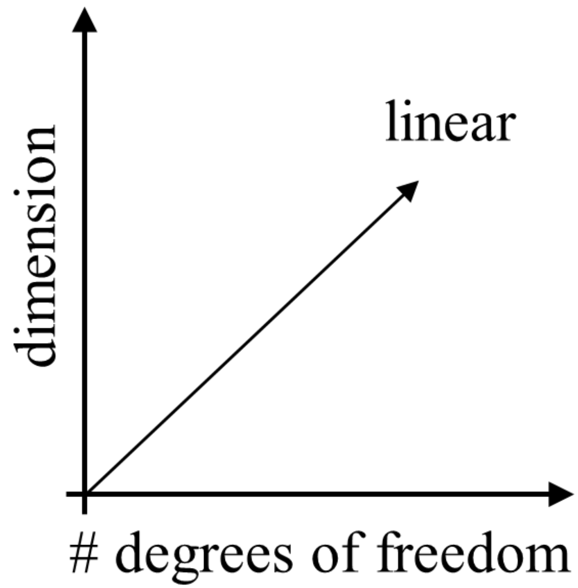
UCSD

Hilbert Space



cannot be
simulated efficiently
(limited, difficult to scale)

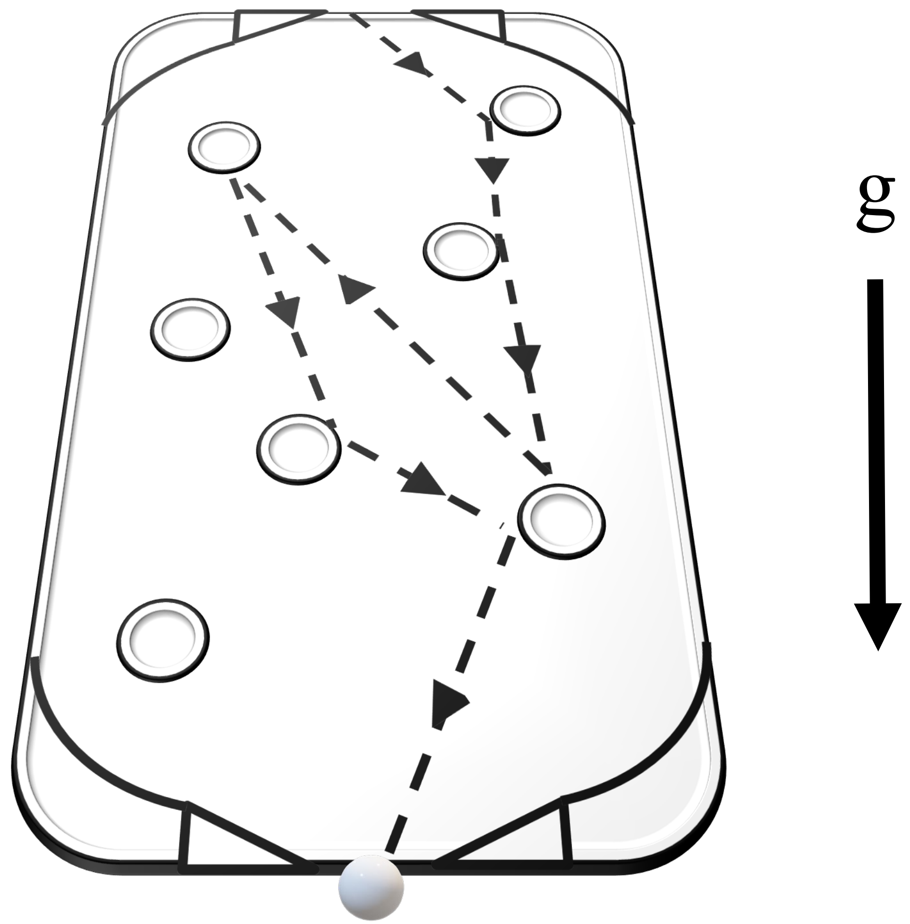
Phase Space



can be
simulated efficiently!
(broader, scalable)

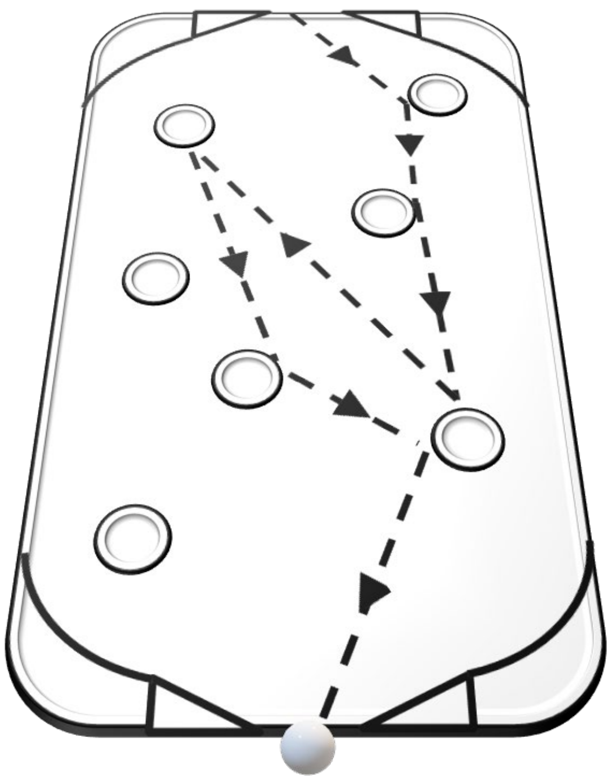
Physics to compute?

UCSD



The pinball “calculates” its own trajectory!

Physics to compute?



- A physical system has <u>no choice</u>!

$$\delta S_{Eucl} = 0$$

- A physical system has *no knowledge* about the difficulty of a problem

Desired Properties

$$\boxed{Z^n \xrightarrow{\delta = ?} Z^n}$$

Physics!

$$\underbrace{\qquad\qquad}_{\dot{x}(t) = F(x(t))}$$

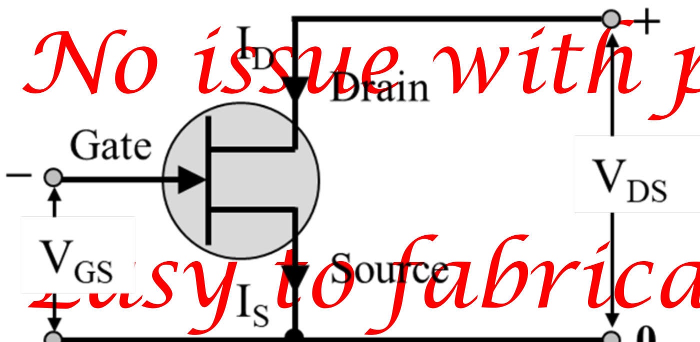
$$\{0s, 1s\} \longrightarrow \{0s, 1s\}$$

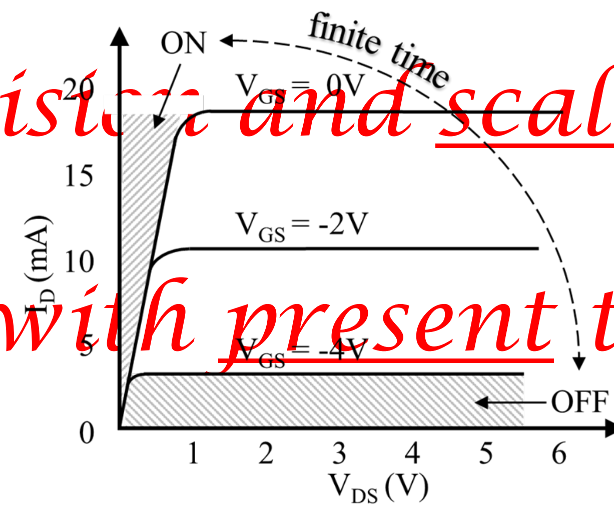
- *Digital input and output*

  (reading and writing error independent of the size)

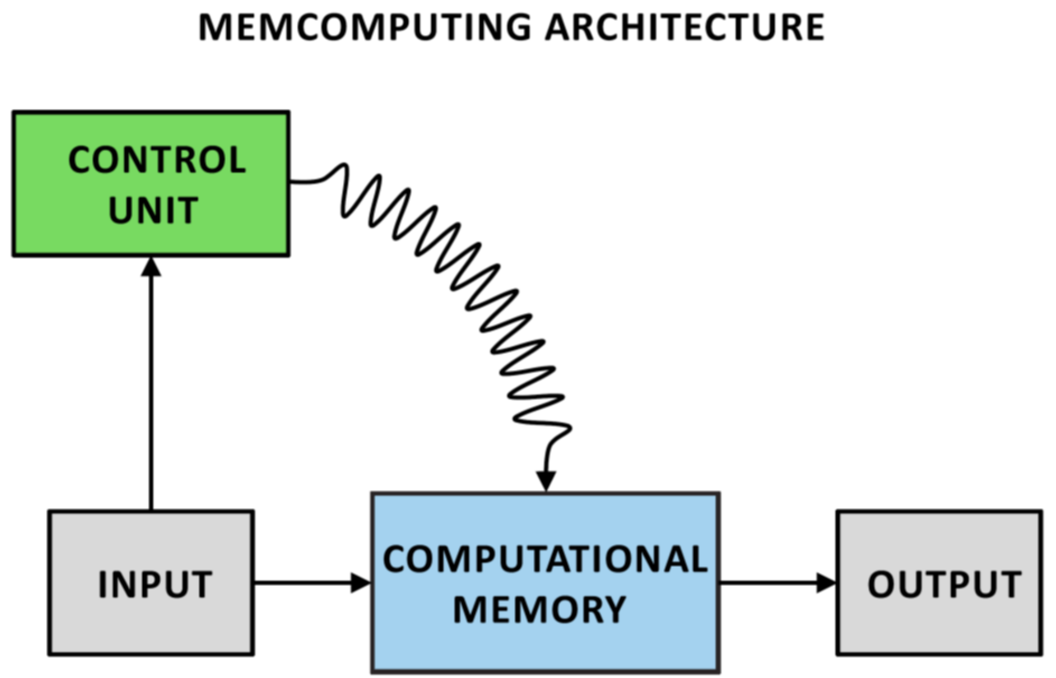
- *No issue with precision and scalable*

- *Easy to fabricate (with present tech.)*





Universal Memcomputing Machines



*mem*processors

formal definition

$$UMM = (M, \Delta, \mathcal{P}, S, \Sigma, p_0, s_0, F)$$

$$\delta_\alpha : M^{m_\alpha} \backslash F \times \mathcal{P} \to M^{m'_\alpha} \times \mathcal{P}^2 \times S$$

F.L. Traversa and M. Di Ventra, *IEEE Trans. Neur. Net. & Learn. Sys. (2015)*

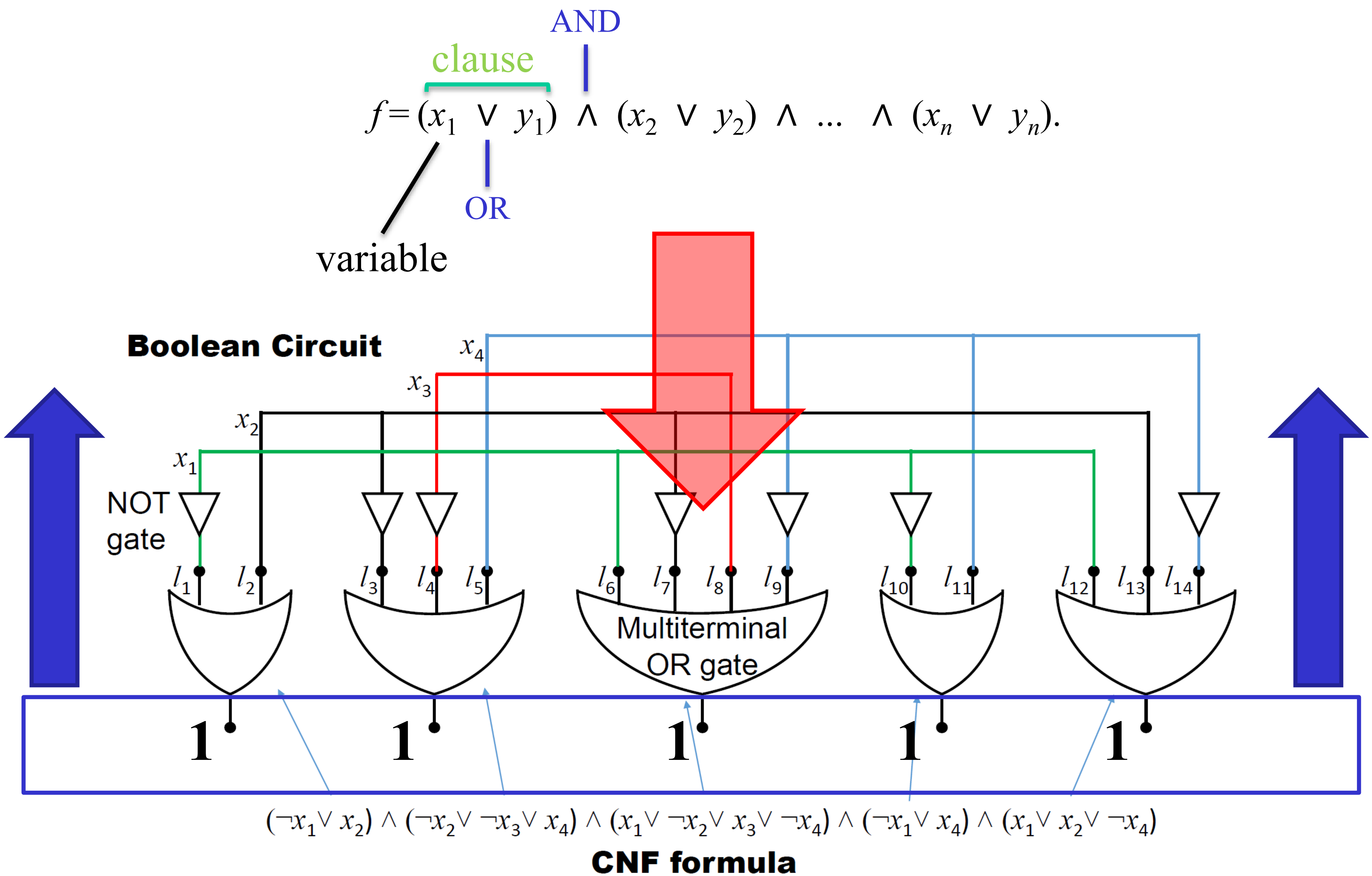
Reality Check

UCSD

—If you tell me precisely what it is a machine cannot do, then I can always make a machine which will do just that.—
John von Neumann (1903 – 1957)

Only Math
or
actual physical systems?

Boolean representation



Combinatorial/Optimization problems and self-organizing logic circuits

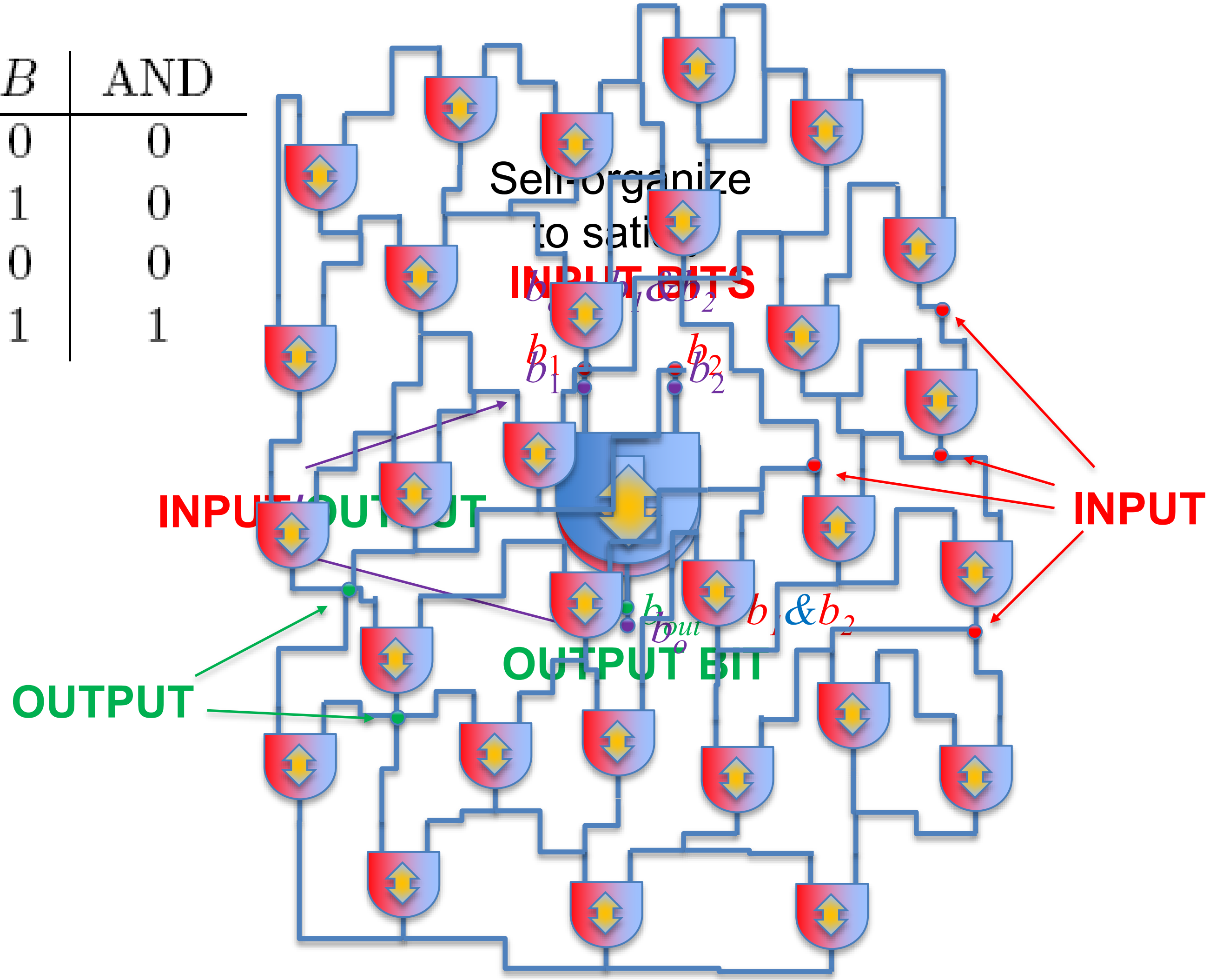
Any combinatorial/optimization problem can be written as a Boolean circuit (not even unique)

↓

We use the same circuit but made of <u>special gates</u>

standard vs. self-orgainizing gates

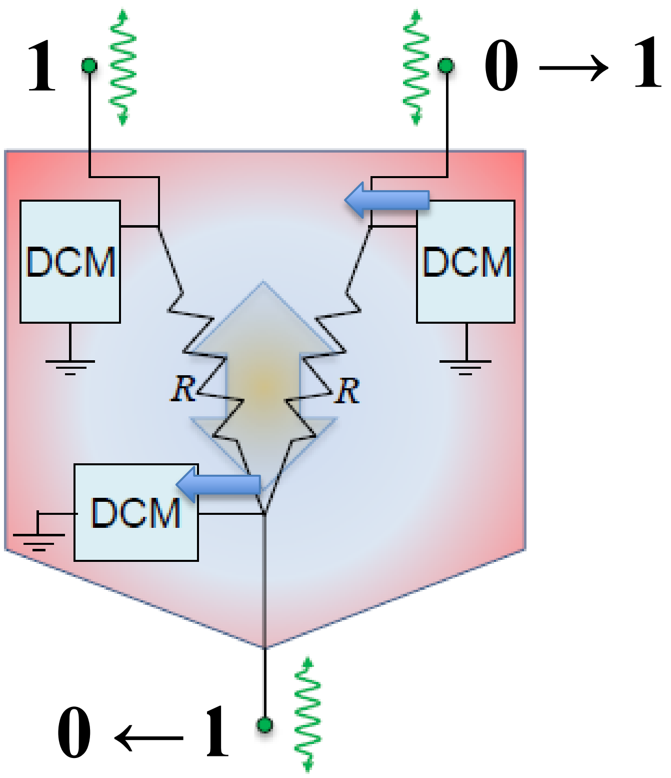
A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1



F.L. Traversa and M. Di Ventra, *UCSD Patent, Chaos* (2017)

Self-organizing gates

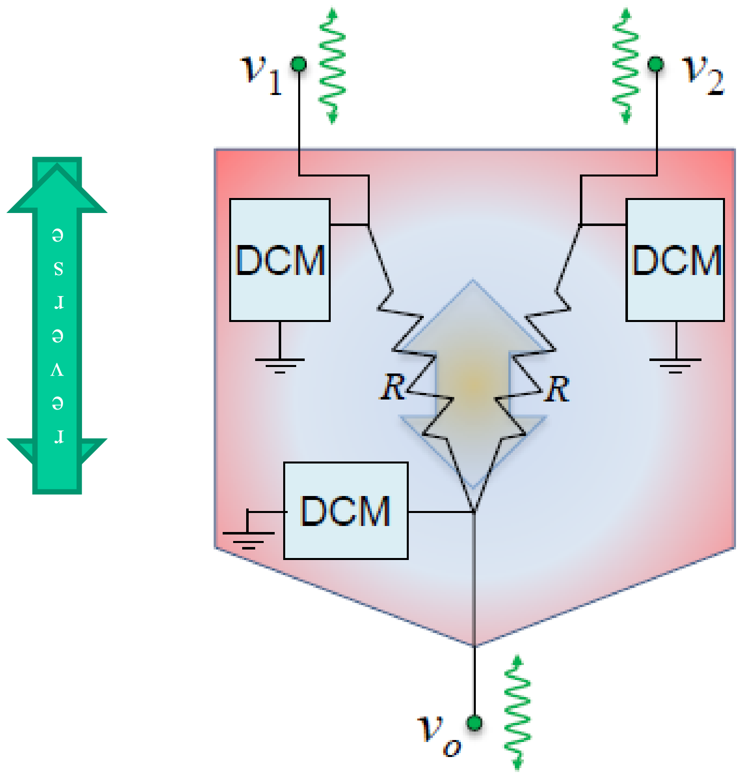
A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1



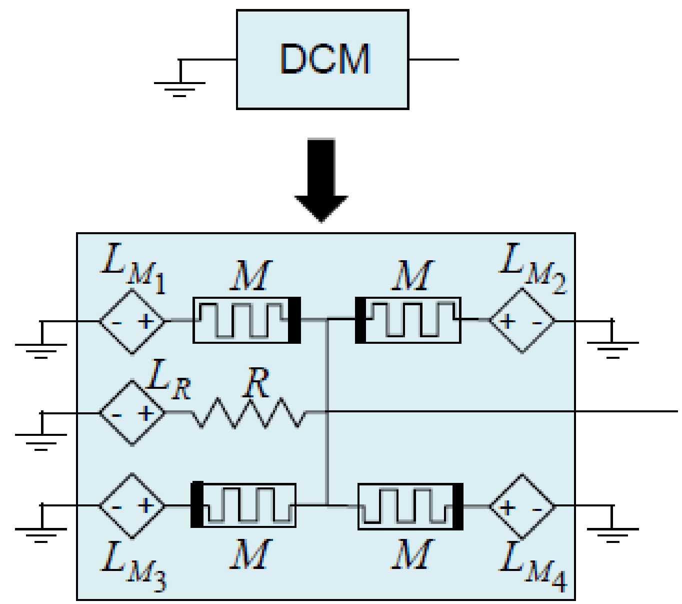
F.L. Traversa and M. Di Ventra, *UCSD Patent, Chaos* (2017)

Self-organizing gates

SO UNIVERSAL GATE



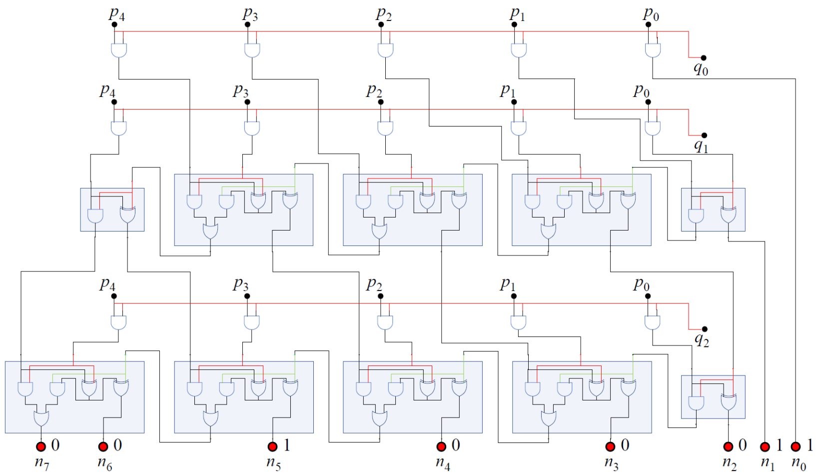
DYNAMIC CORRECTION MODULE



use memory elements and/or standard transistors

F.L. Traversa and M. Di Ventra, *UCSD Patent, Chaos* (2017)

Coarse-graining from hardware to software



non-linear dynamical systems

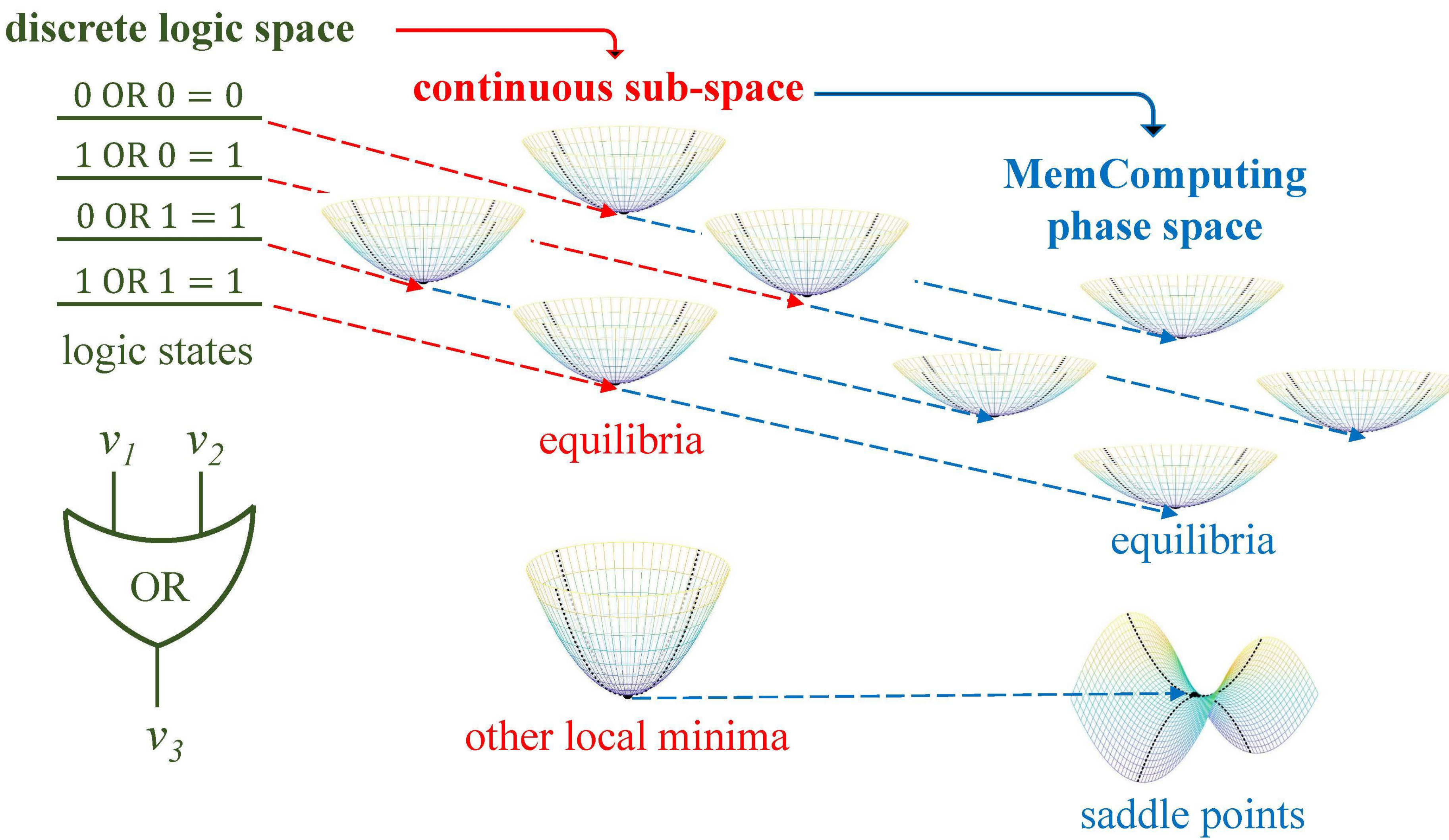
$$\boxed{\dot{x} = F(x) \qquad \text{with initial value } x_0}$$

$x = I, V$, internal states

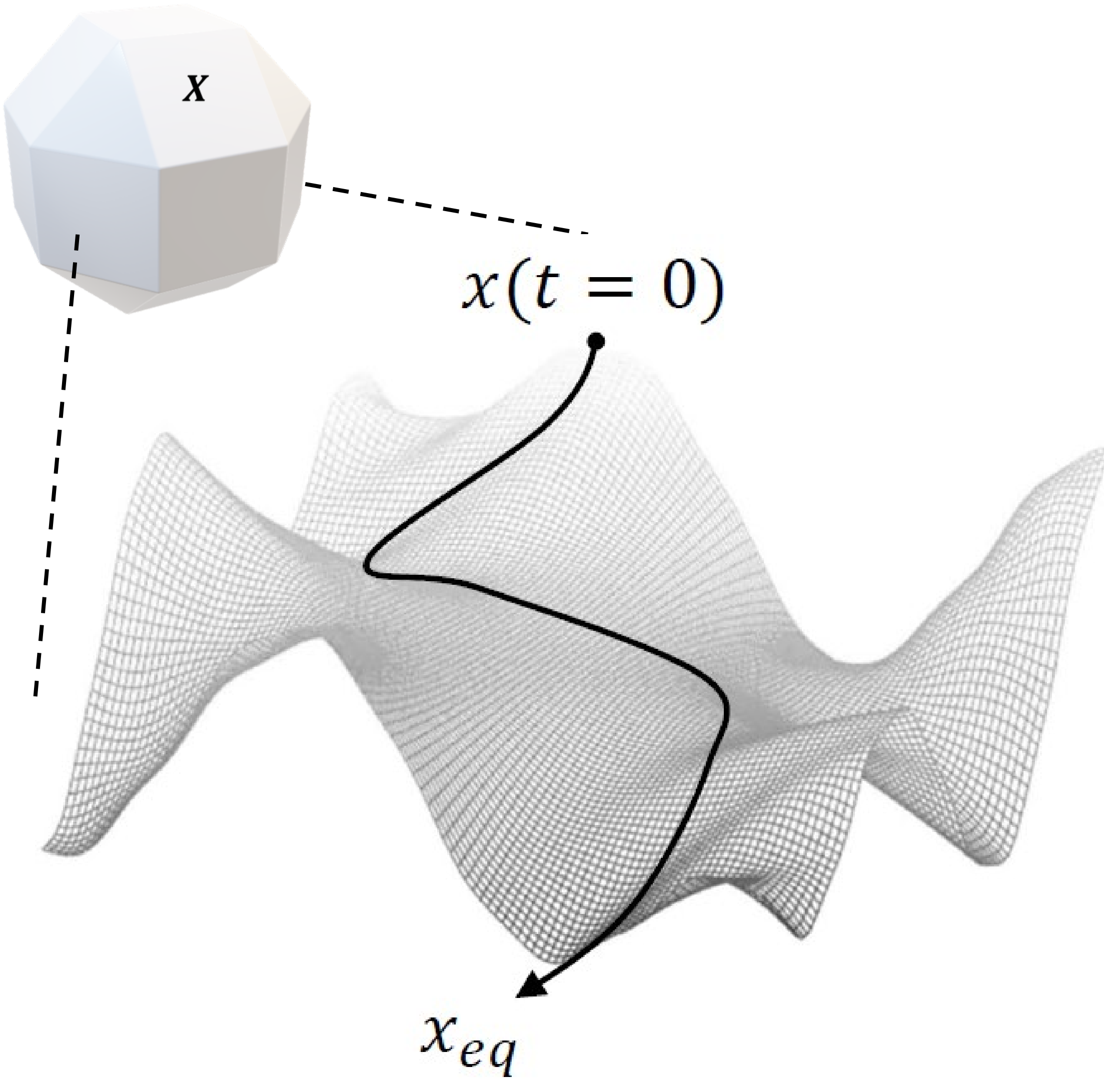
F.L. Traversa and M. Di Ventra, *UCSD Patent, Chaos* (2017)

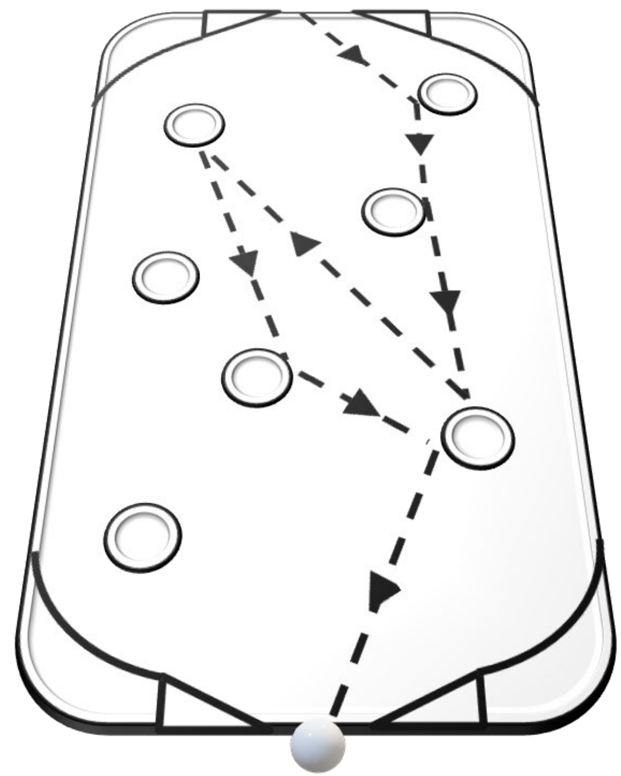
From logic to dynamics

UCSD



From logic to dynamics



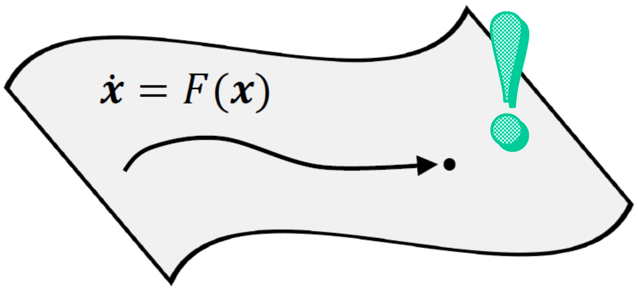


Attractors

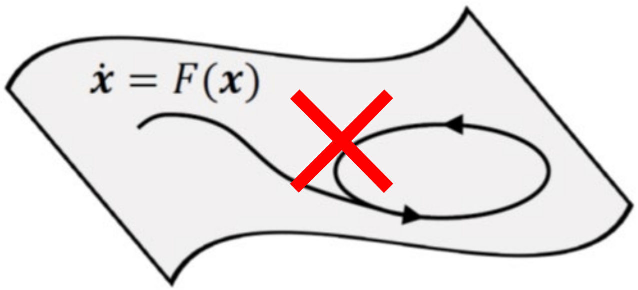
$\dot{x} = F(x)$ with initial value x_0

$x = I, V$, internal states

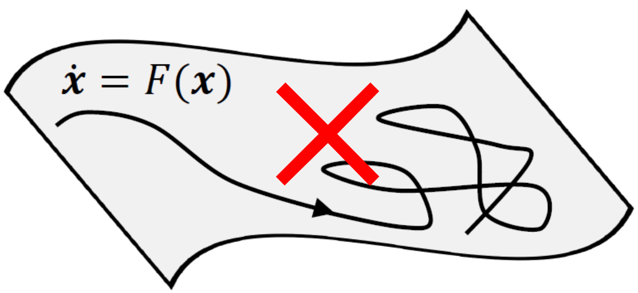
Equilibrium point



Periodic orbit



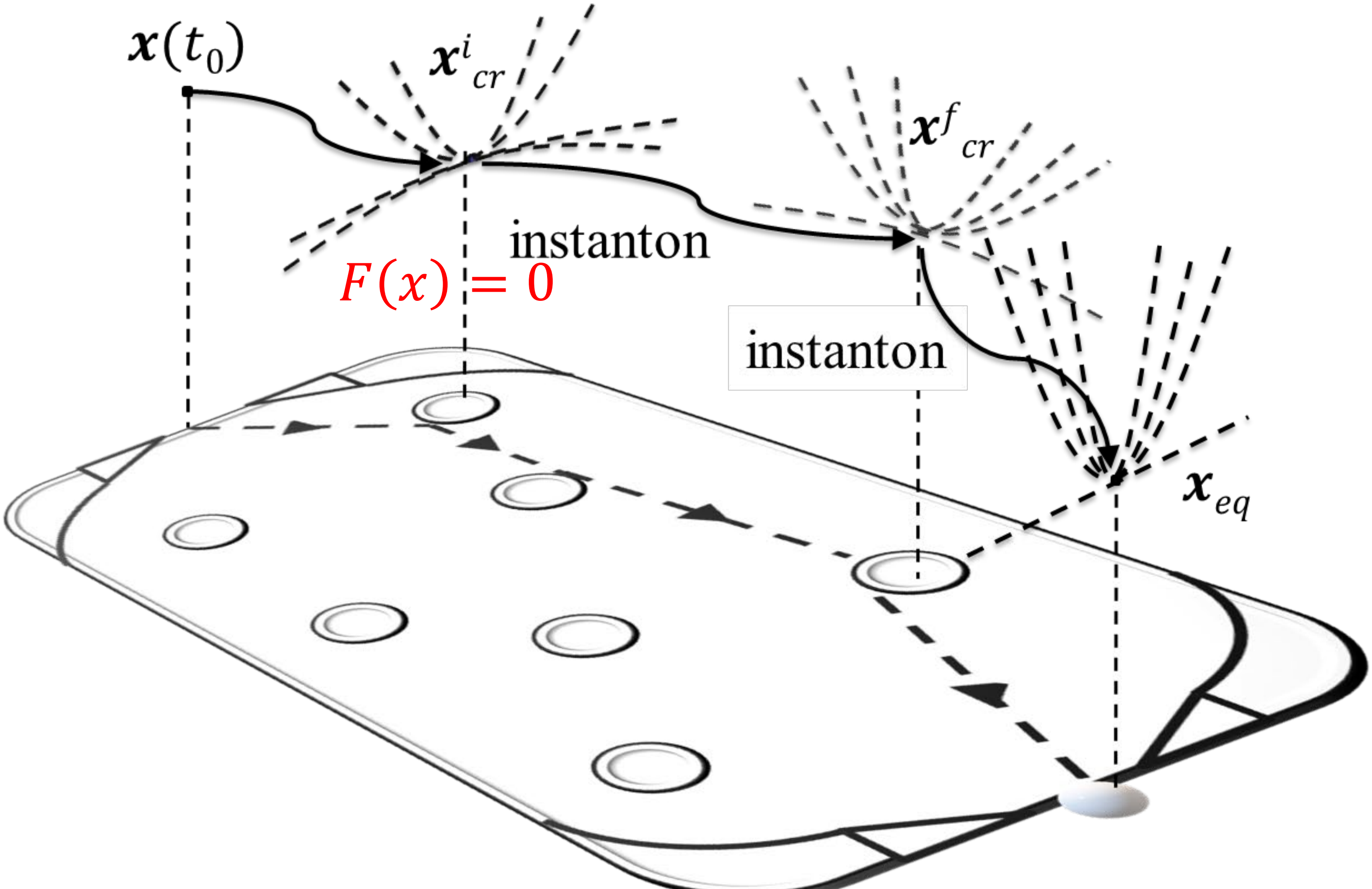
Chaos



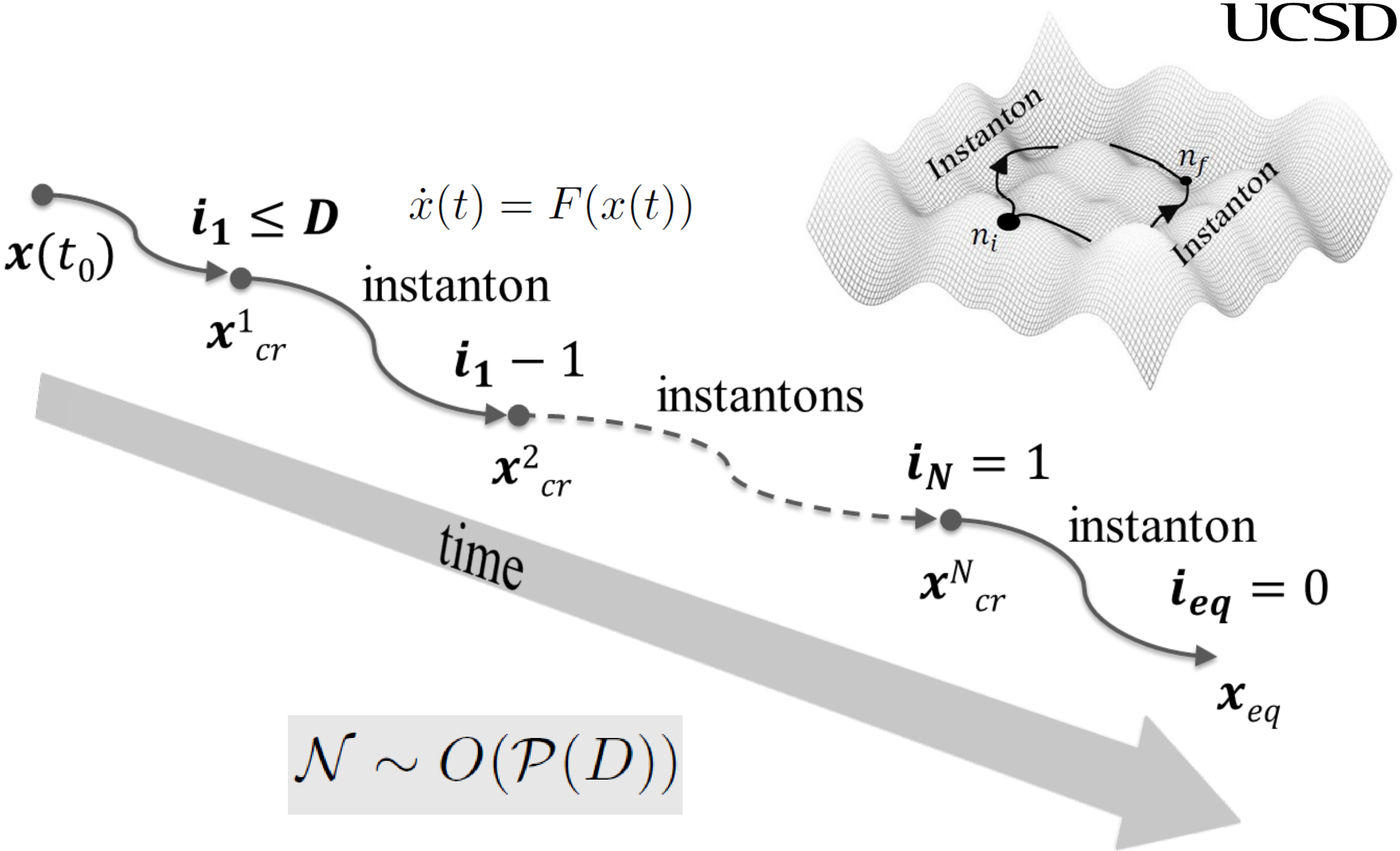
F.L. Traversa and M. Di Ventra, *IEEE Trans. Neur. Net. & Learn. Sys. (2015)*

Low-energy dynamics
(from supersymmetric topological field theory)

$\dot{x}(t) = F(x(t))$ DMM's trajectory in phase space



Instantonic steps to solution



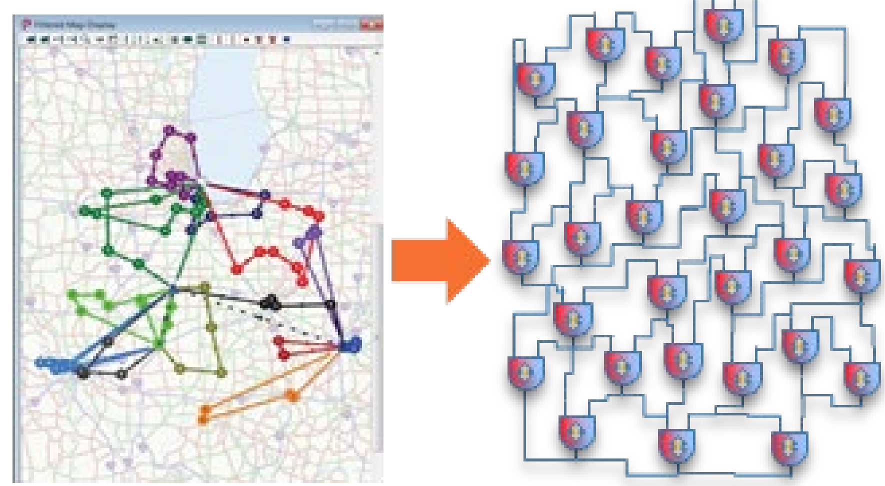
$$\mathcal{N} \sim O(\mathcal{P}(D))$$

It does NOT prove NP=P!

M. Di Ventra, I.V. Ovchinnikov, *Ann. Phys*. (2019)

From problem to solution

UCSD



1. Select the problem
2. Convert it to binary representation
3. Substitute standard gates with SOLGs
4. Build or *simulate* its dynamics
5. Read the solution

F.L. Traversa and M. Di Ventra, *Chaos (2017)*

What I will show

Simulations of ordinary differential equations

$$C\frac{d}{dt}v_M = (A_v + B_v\mathcal{D}[g(x)])v_M + A_i i_{DCG} + b,$$

$$\frac{d}{dt}x = -\alpha\mathcal{D}\left[h(x, v_M)\right]\mathcal{D}[g(x)]v_M,$$

$$\frac{d}{dt}i_{DCG} = \mathcal{D}[\rho(s)]f_{DCG}(v_M) - \gamma\mathcal{D}\left[\rho(1-s)\right]i_{DCG},$$

$$\frac{d}{dt}s = f_s(i_{DCG}, s),$$

MatLab code to search for the steady-states/solutions to the problem

F.L. Traversa and M. Di Ventra, *UCSD Patent, Chaos* (2017)

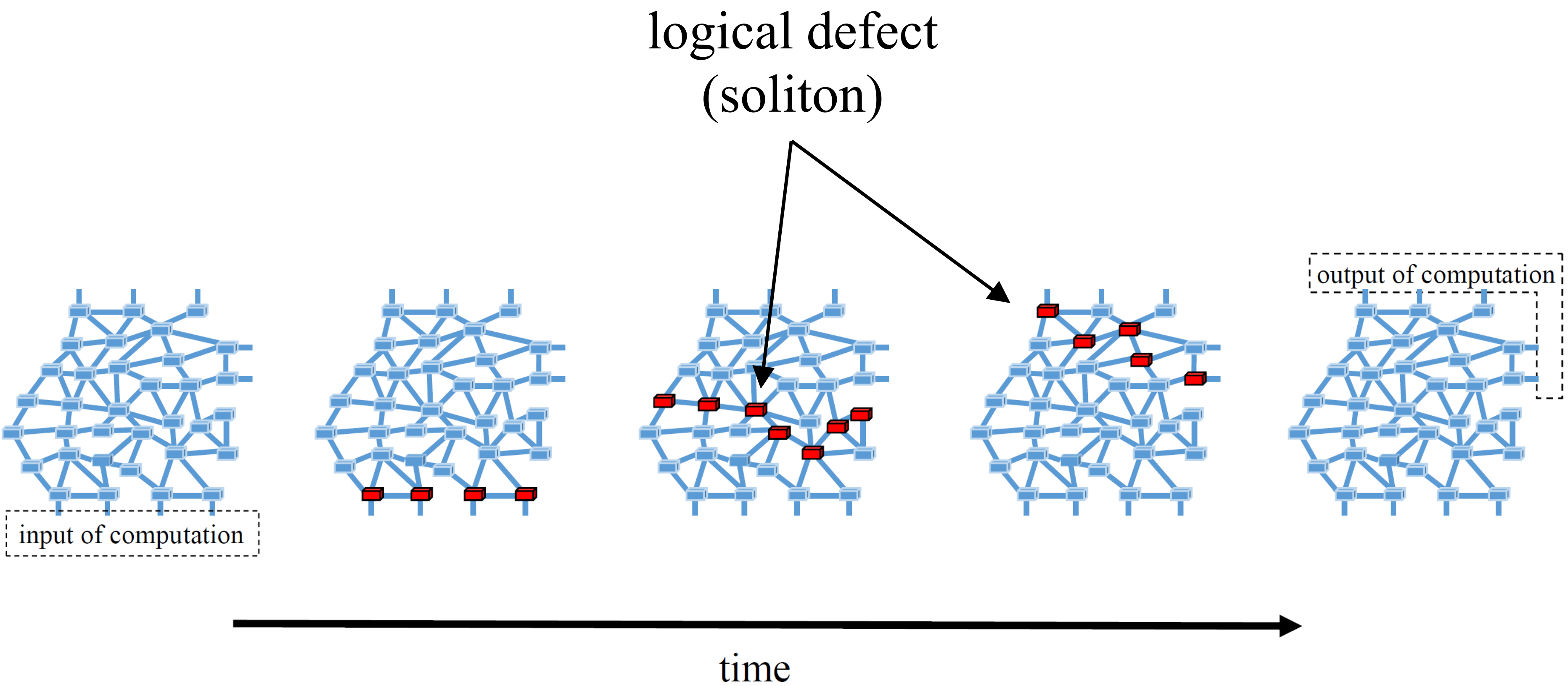
Example: 11 bits $1073 = 29 \times 37$

Initial conditions: random

29

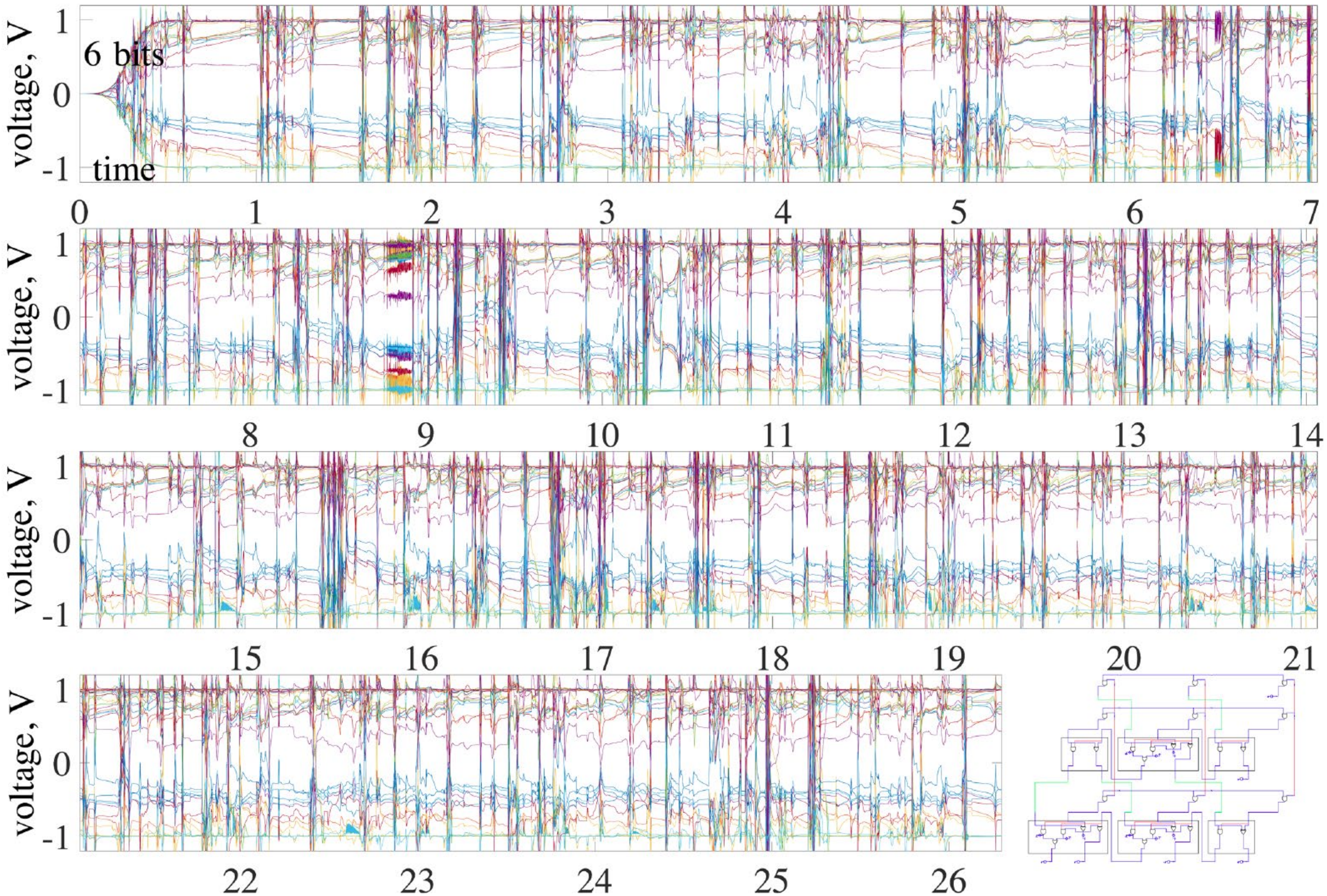
37

Solitons and Logical Defects



M. Di Ventra, I.V. Ovchinnikov, *Ann. Phys*. (2019)

Factorization: no solution



Factorization of the prime number 47

F.L. Traversa and M. Di Ventra, *UCSD Patent*, *Chaos* (2017)

Hardware scalability comparison for factorization

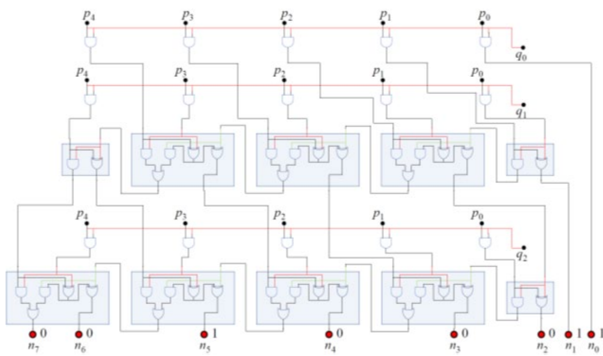
UCSD

N= # bits

Quantum Computing *vs.* **MemComputing**

Shor's algorithm

(P.W. Shor, *SIAM* (1997))



probabilistic

deterministic

$O(N^2 \log_2 N \log_2^2 N)$

on a quantum computer

+

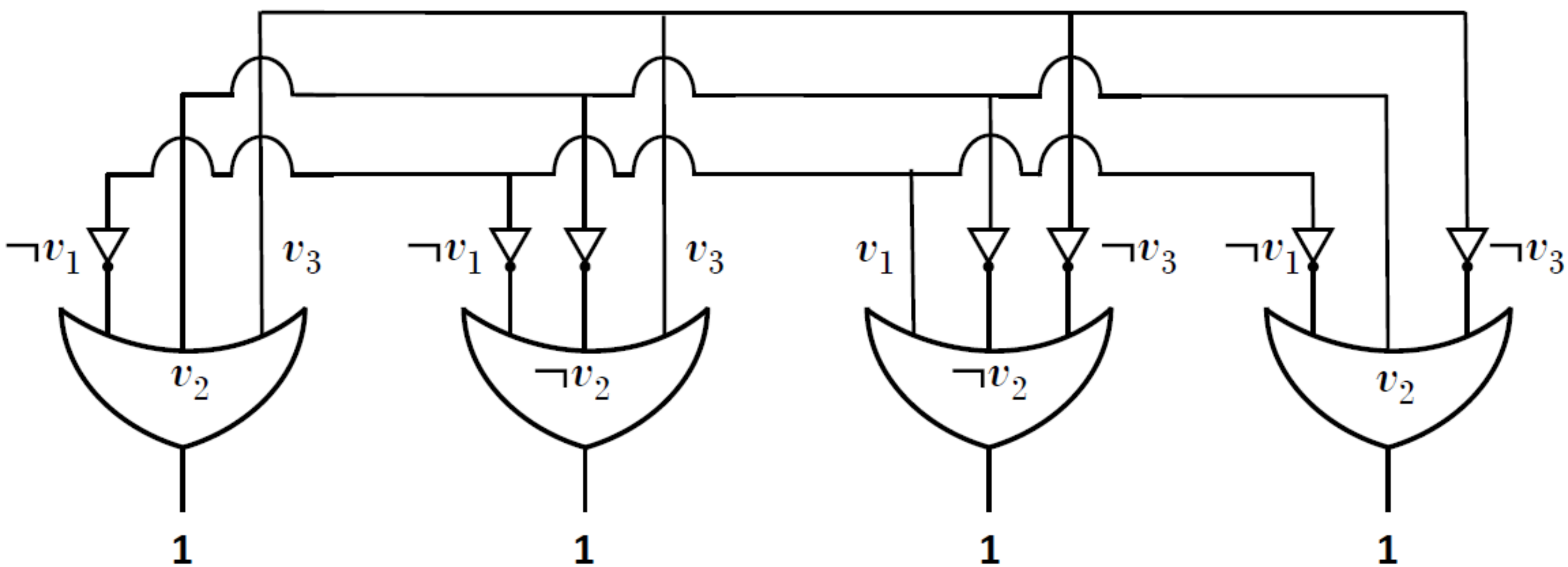
$O(\mathrm{poly}(N))$

on a classical computer

$O(N^2)$

F.L. Traversa and M. Di Ventra, *Chaos* (2017)

An NP-complete problem: 3-SAT



$$\varphi(v) = (\neg v_1 \vee v_2 \vee v_3) \wedge (\neg v_1 \vee \neg v_2 \vee v_3) \wedge (v_1 \vee \neg v_2 \vee \neg v_3) \wedge (\neg v_1 \vee v_2 \vee \neg v_3)$$

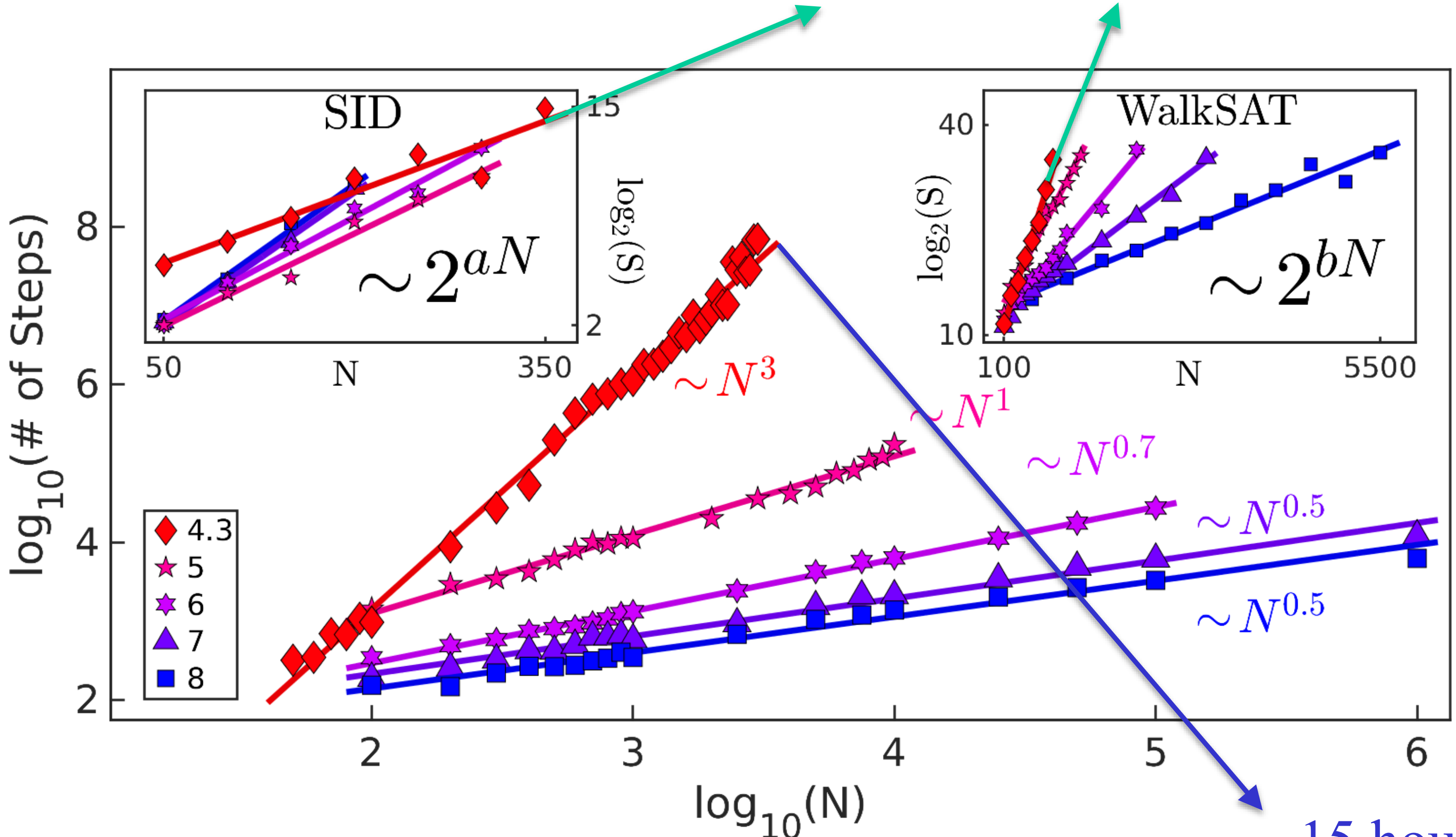
Barthel instances (Barthel, et al. *PRL* 2001): <u>hard satisfiable instances</u>

<u>Simulations</u> performed using forward Euler implemented in MatLab running on a *single* AMD processor (*no optimization*)

S. Bearden, Y.R. Pei, and M. Di Ventra, *Scientific Reports (2020)*

An NP-complete problem: 3-SAT

> age of the Universe (>10^{10} years)



~ 15 hours

S. Bearden, Y.R. Pei, and M. Di Ventra, *Scientific Reports (2020)*

UCSD

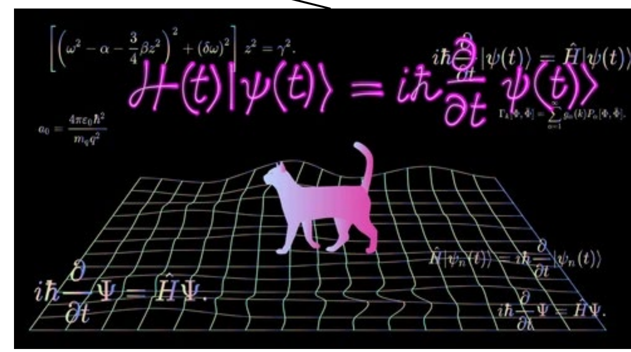
Broader Impact

Problems tackled so far

>100K instances

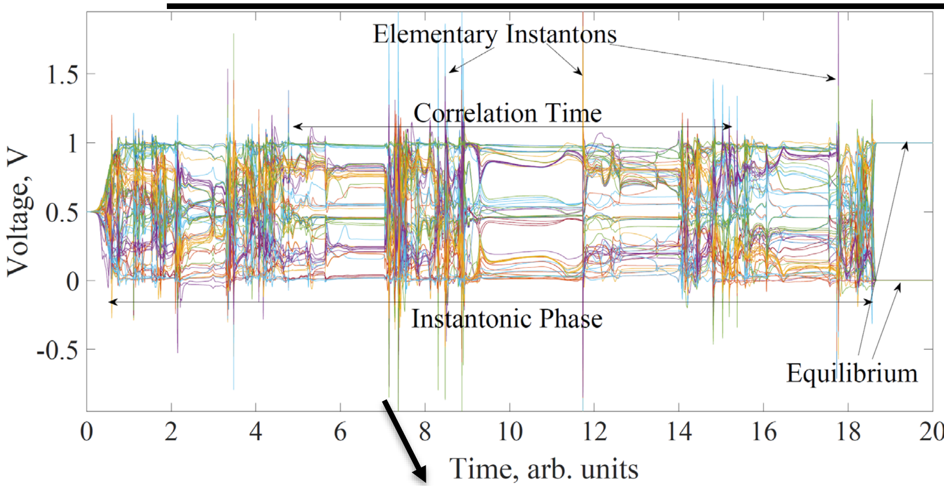
Random 2 Max-Sat
Max-Cut
Forced Random Binary problem
Max-clique
Integer Linear Programming
Traveling Salesman
Factorization
5^{th} Airbus problem…

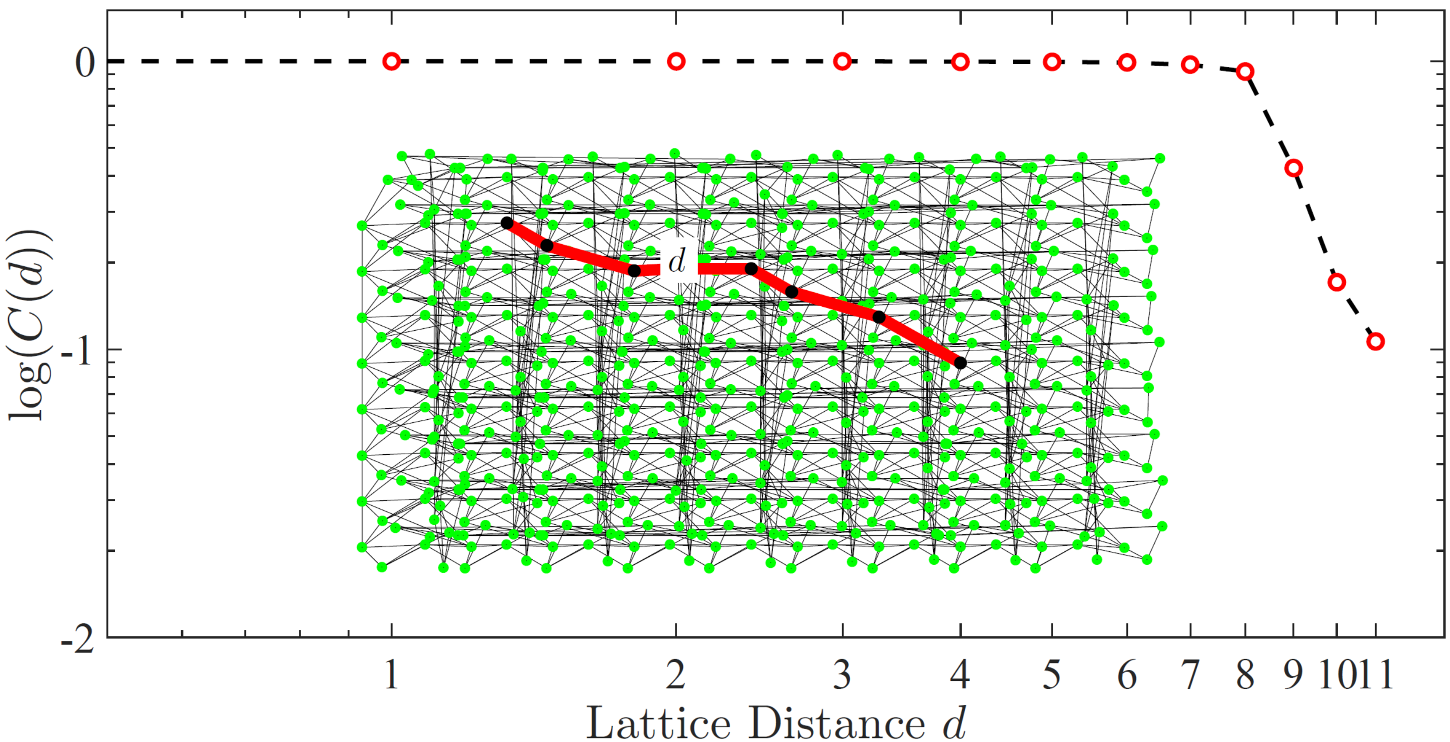
Supervised Learning
Unsupervised Learning
of RBMs and deep NNs



Spin glasses
Quantum state tomography
Ground state of correlated
quantum systems….

Scale-free correlations

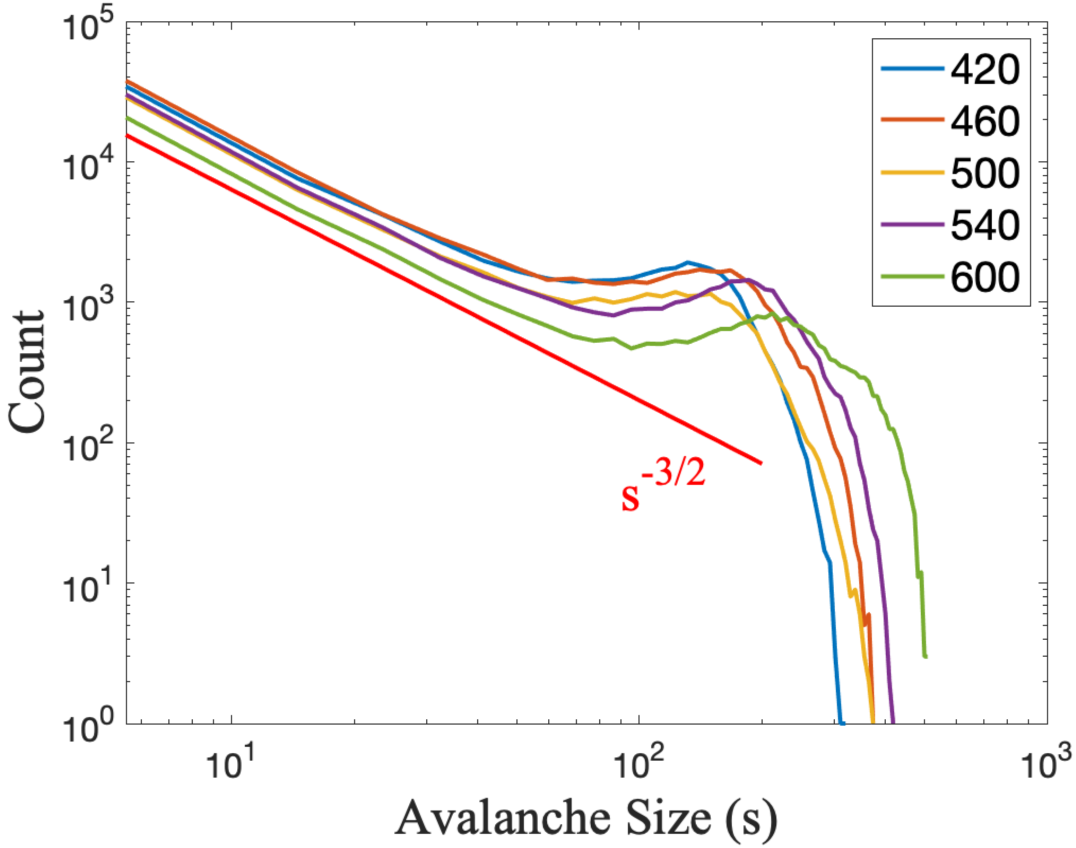




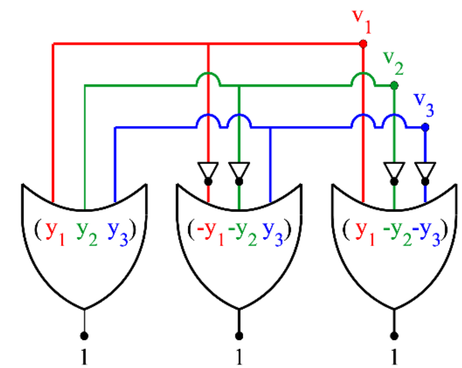
M. Di Ventra, F.L. Traversa, I.V. Ovchinnikov, *Annalen der Physik* (2017)

Critical branching processes

critical Borel distribution

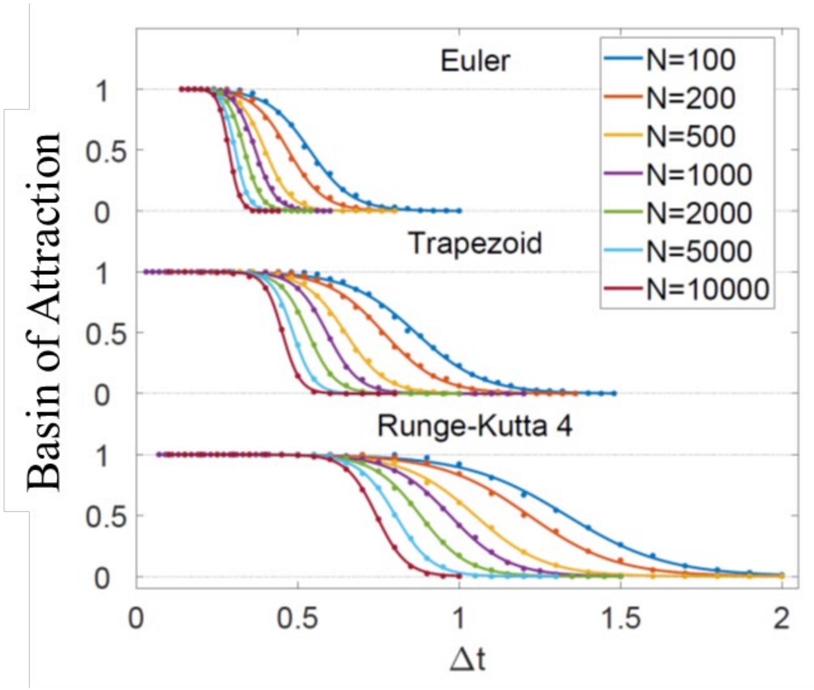


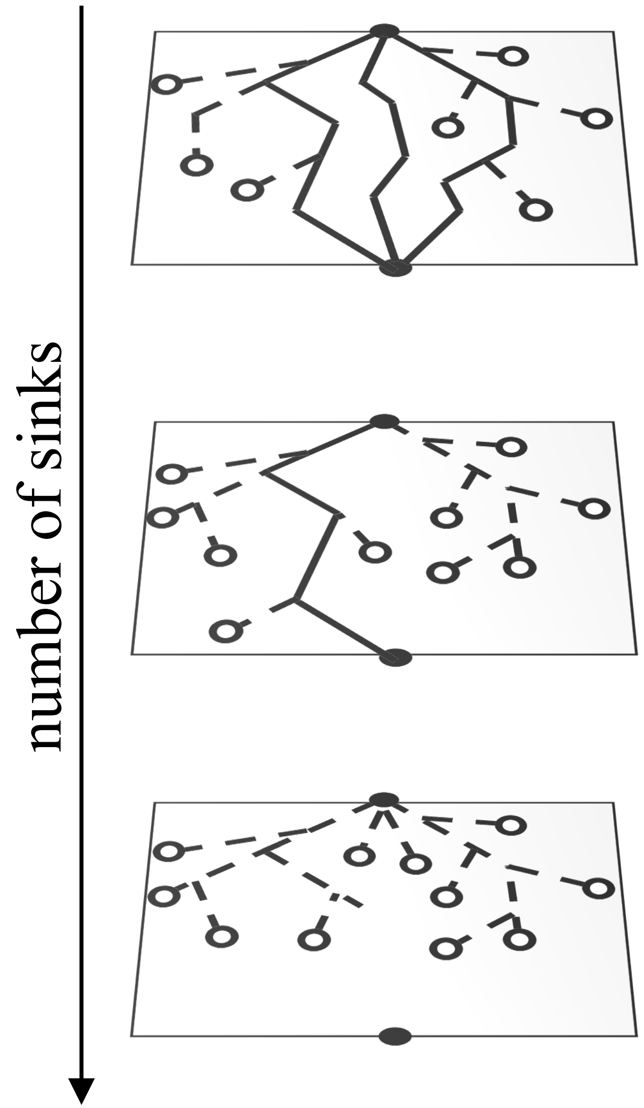
Random 3-SAT



S. Bearden, F. Sheldon, and M. Di Ventra, *Eur. Phys. Lett.* (2019)

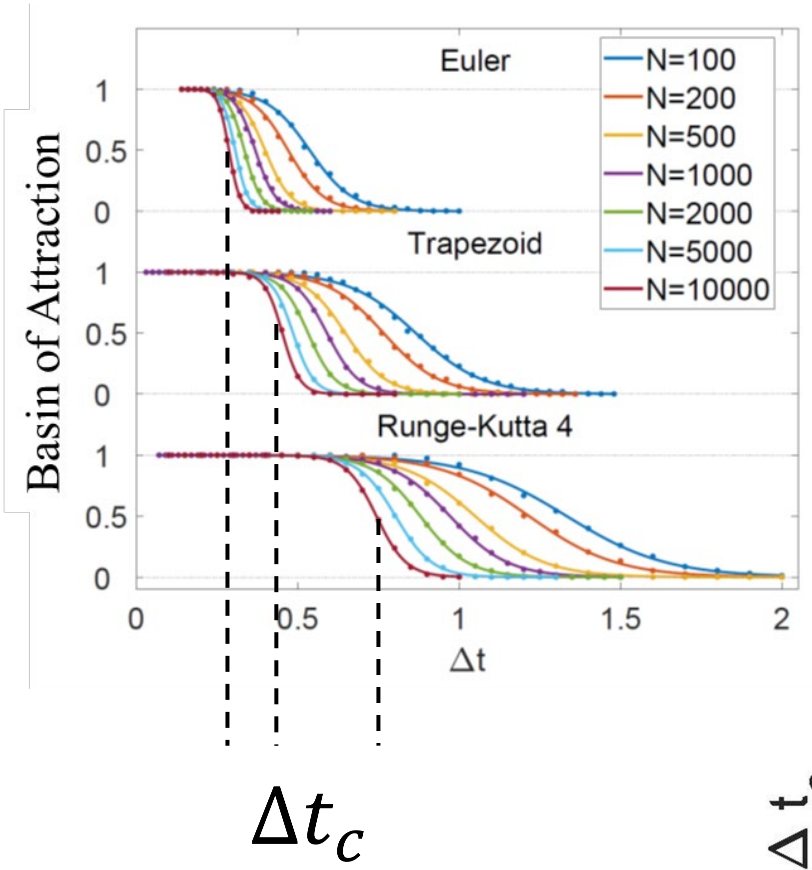
Topological robustness

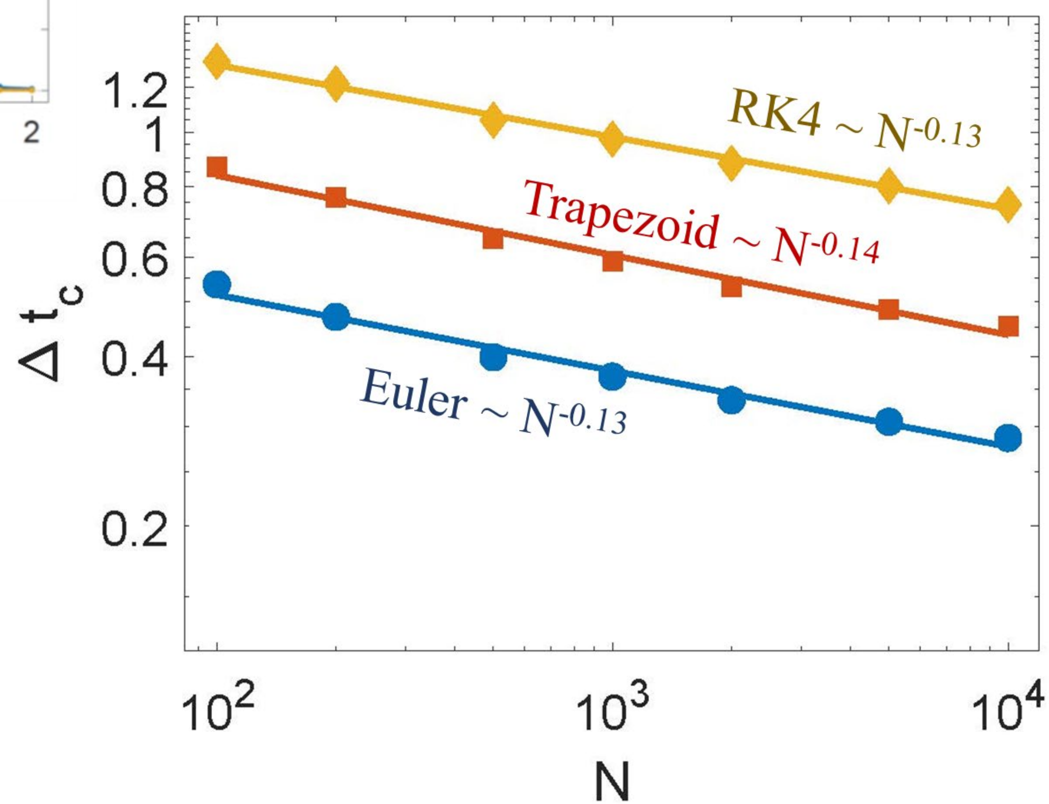




Y.-H. Zhang and M. Di Ventra *Chaos* (2021)

Topological robustness





Y.-H. Zhang and M. Di Ventra *Chaos* (2021)

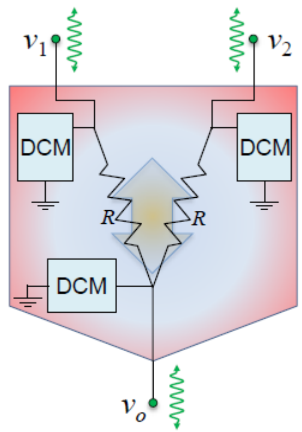
Conclusions

UCSD

Universal Memcomputing Machines

compute complex problems efficiently

Practical realization



Hardware
or
Software!

Computing and Topology

(instantons in phase space)

Duality between Computational Complexity and Algebraic Topology?

- Machine learning
- Optimization
- Quantum Hamiltonians
- Real-time computing (needs hardware)
- …….

Acknowledgments

Sean Bearden

Haik Manukian

Yan Ru Pei

Yuanhang Zhang

Daniel Primosch

Michael Levy

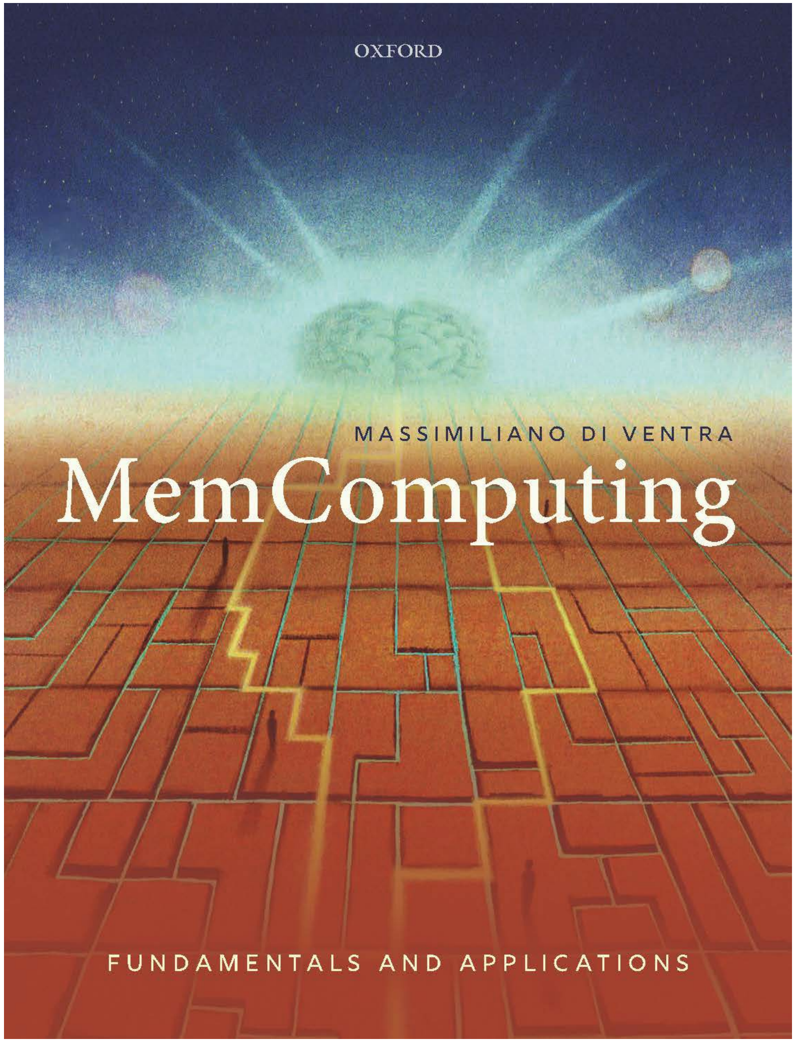
DARPA

NSF

CMRR
Center for Magnetic Recording Research

Office of Science
U.S. DEPARTMENT OF ENERGY

Further Info



Thank you !

MemComputing, Inc.