Neuromorphic computing : AI needs new hardware

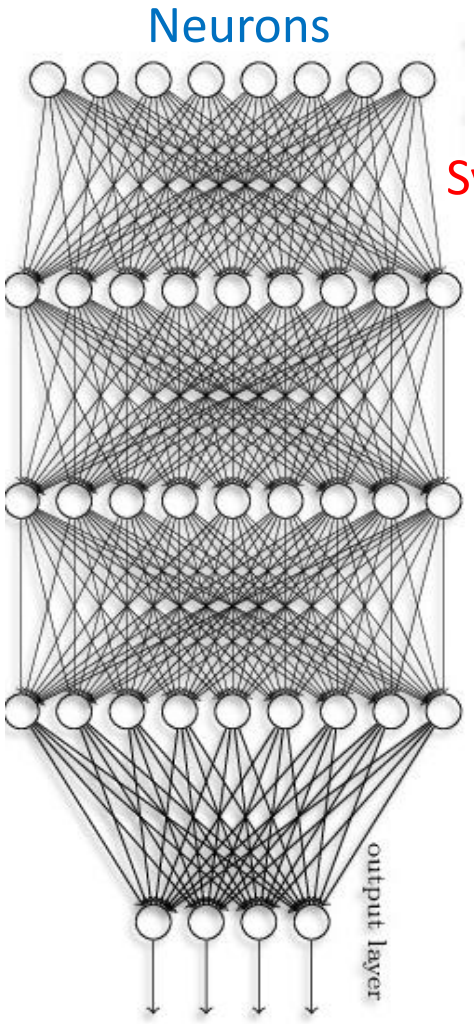
Julie Grollier[1]

Mathieu Riou[1], Philippe Talatchian[1], Jacob Torrejon[1], Miguel Romera[1], Flavio Abreu Araujo[1], Paolo Bortolotti[1], Juan Trastoy[1], Vincent Cros[1], Guru Khalsa[2], Mark Stiles [2], Sumito Tsunegi [3], Kay Yakushiji [3], Akio Fukushima [3], Hitoshi Kubota [3], Shinji Yuasa [3], Damir Vodenicarevic[4], Tifenn Hirtzlin[4], Maxence Ernoult [4], Nicolas Locatelli[4], Damien Querlioz[4]

[1]CNRS/Thales, France [2]NIST, USA [3]AIST, Japan [4]C2N, France

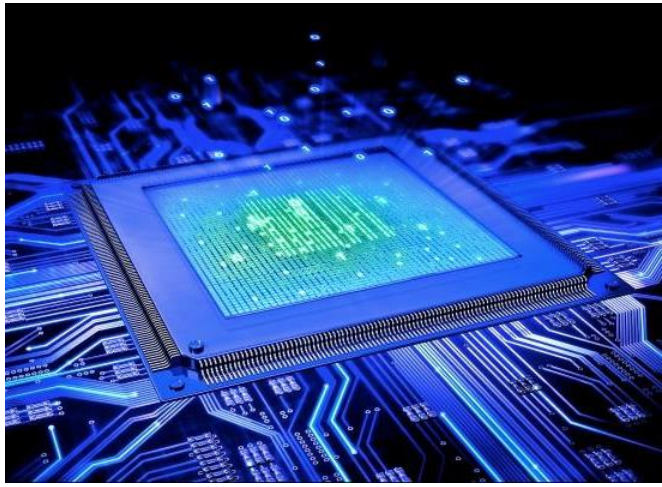
bio*SPIN*spired

Today, neural networks are translated to code, binarized, then sent to a digital processor

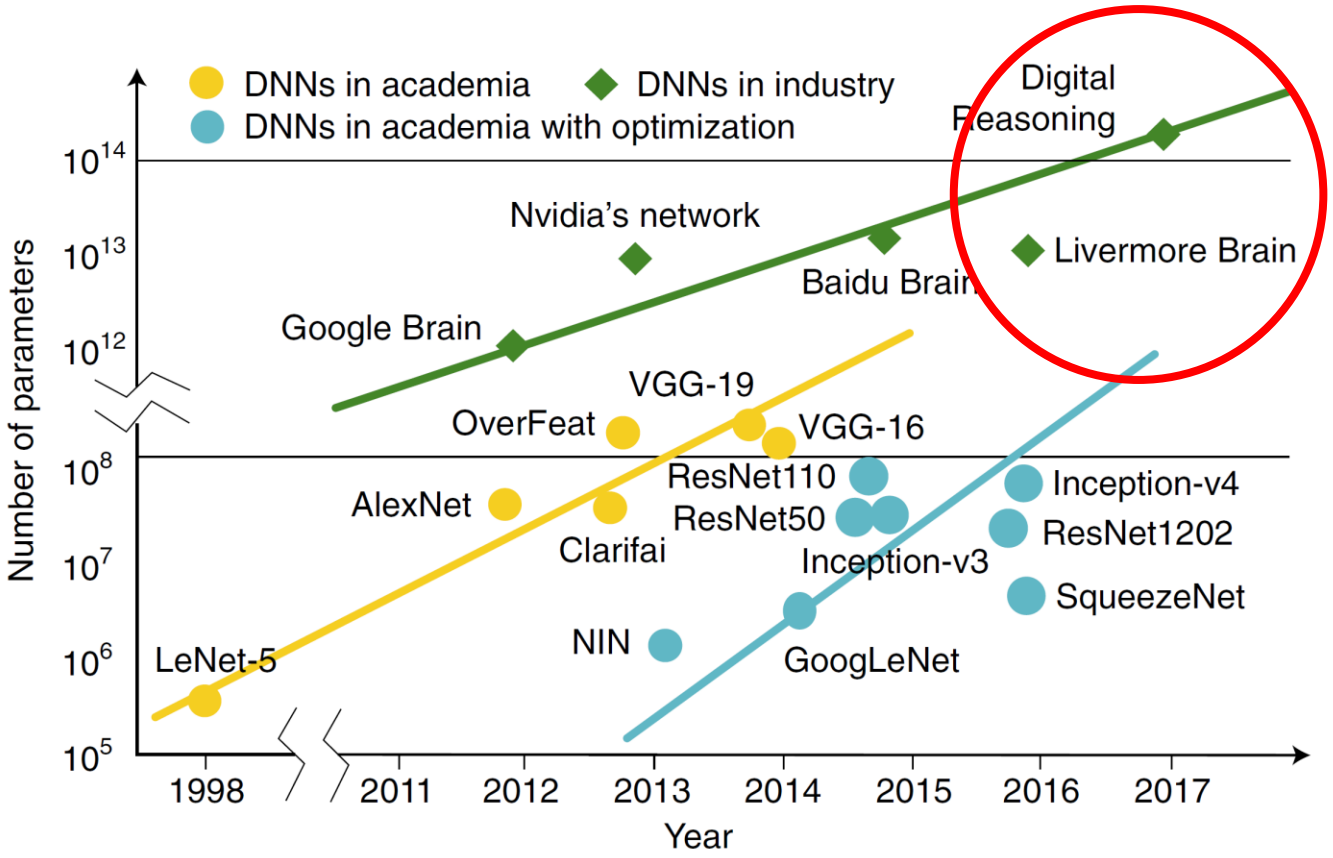



```
 * internal xml or external
 * external is needed when running in static mode
 *
 * @var boolean
 */
define('PSI_INTERNAL_XML', false);

if (version_compare("5.2", PHP_VERSION, ">")) {
    die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcre extension to php in order to work
        properly.");
}

require_once APP_ROOT.'/includes/autoloader.inc.php';

// Load configuration
require_once APP_ROOT.'/config.php';

if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {
    $tpl = new Template("/templates/html/error_config.html");
    echo $tpl->fetch();
    die();
```

0011011



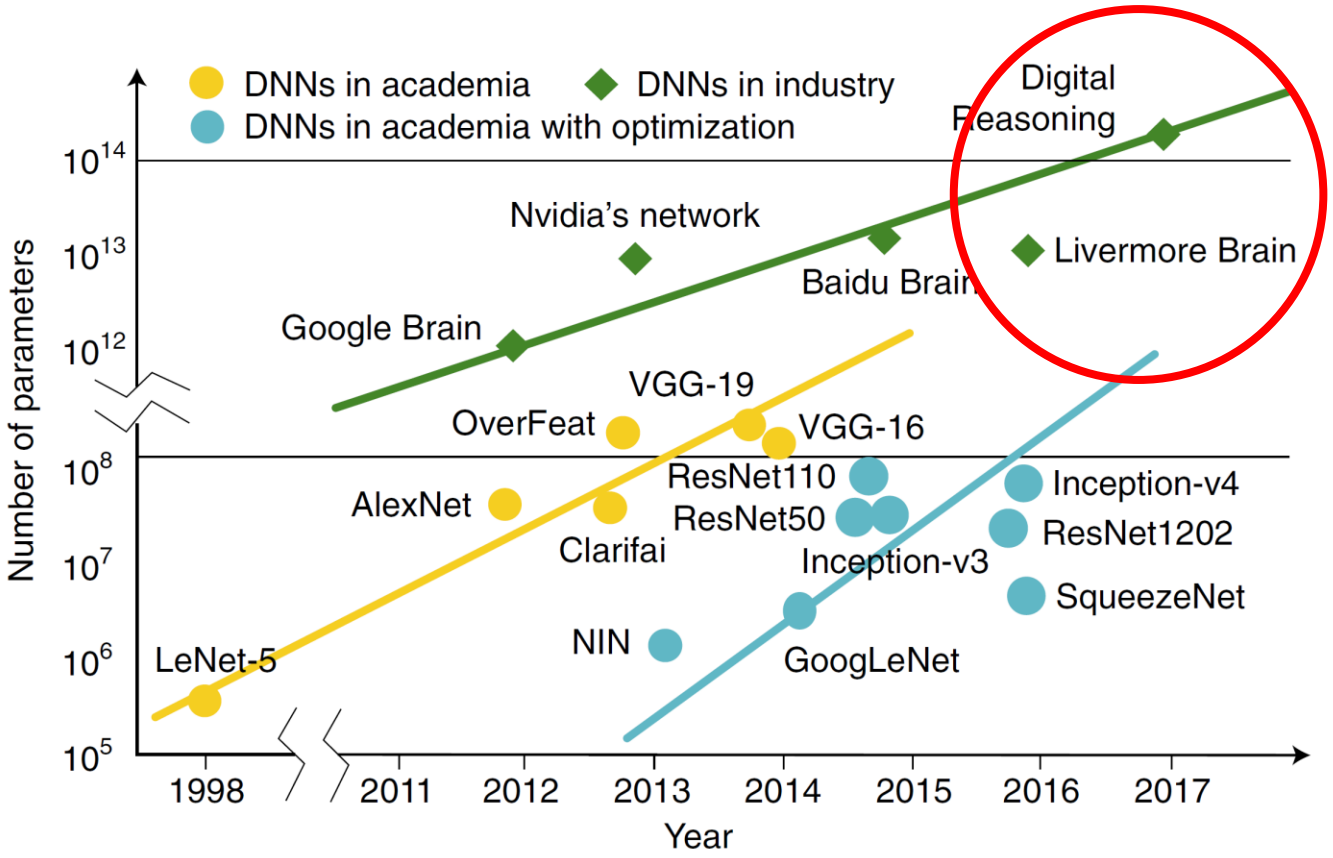
Current processors cannot sustain AI demands



Xu et al, Nature Electronics 1, 216 (2018)

MWatts, slow, no real time learning

The brain is larger than all these neural networks and consumes only 20W



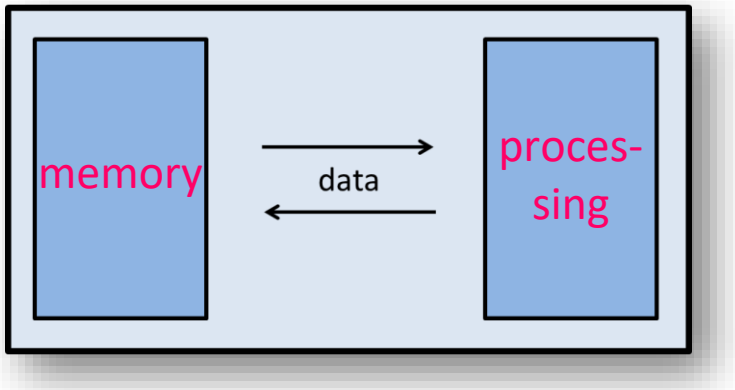
Xu et al, Nature Electronics 1, 216 (2018)

10^{11} neurons, 10^{15} synapses: 20 Watts

Entangling memory and processing allows for fast and energy efficient computing

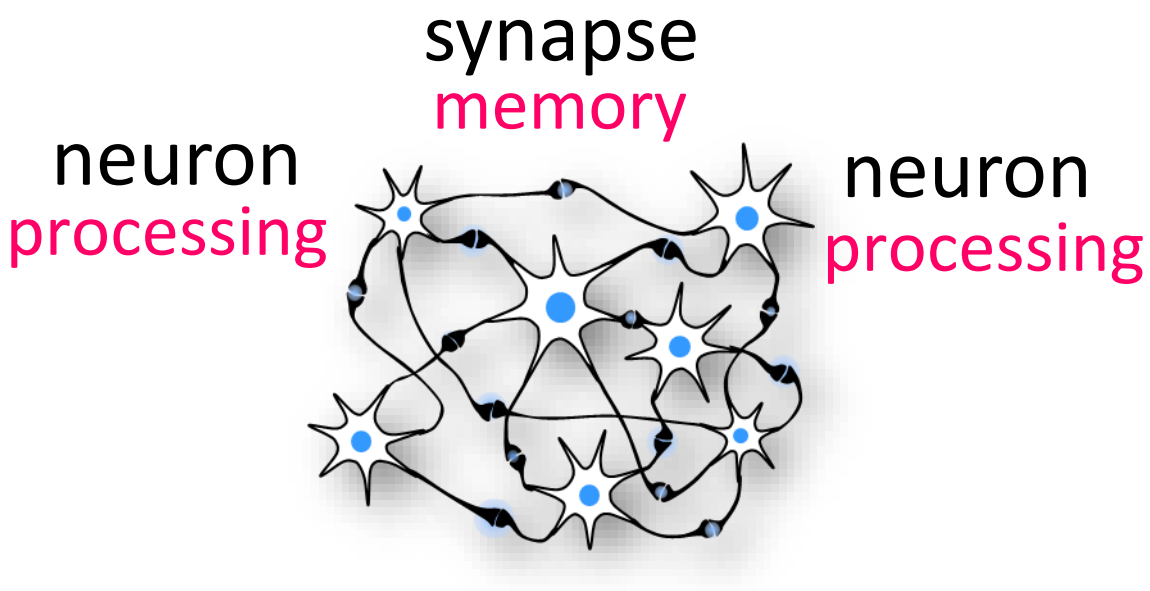
Digital computer

CPUs, GPUs, TPUs



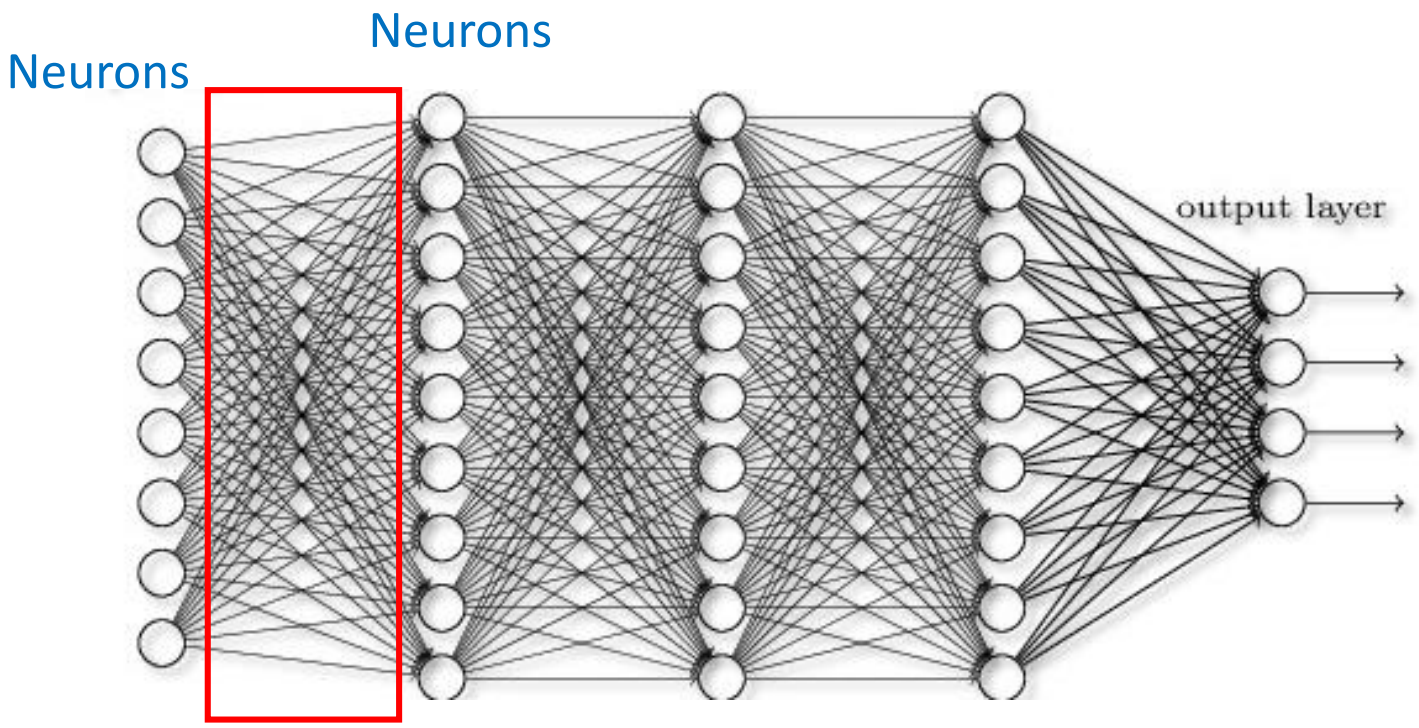
100 W/cm^2

Brain



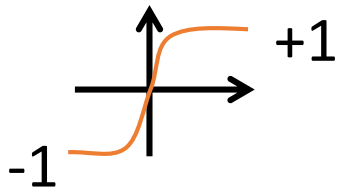
20 W in total !

Chips implementing the layered structure of the network can enable fast, low energy computing with real-time learning

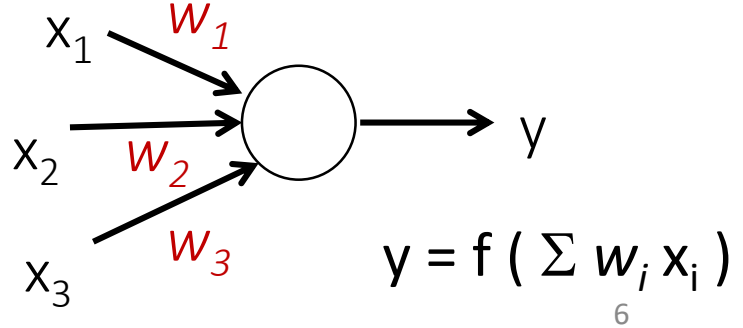


Synapses = in-situ memory

Hardware neurons: non-linear



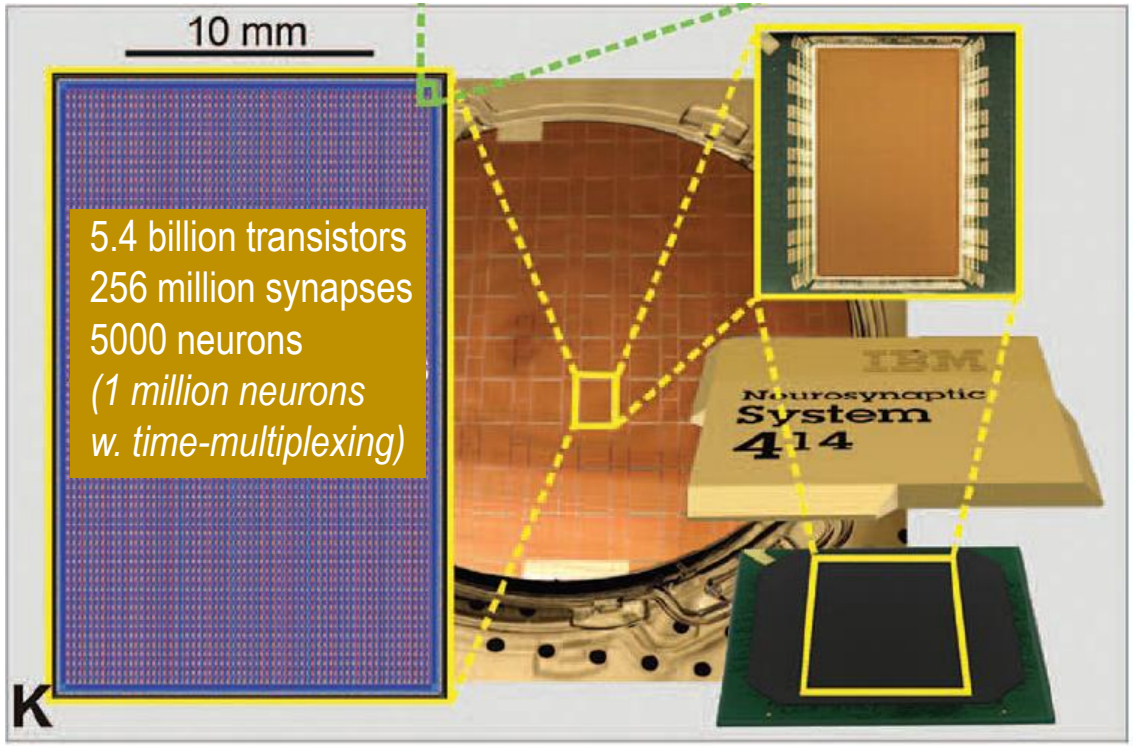
Hardware synapses: analog memory valves (weights w)

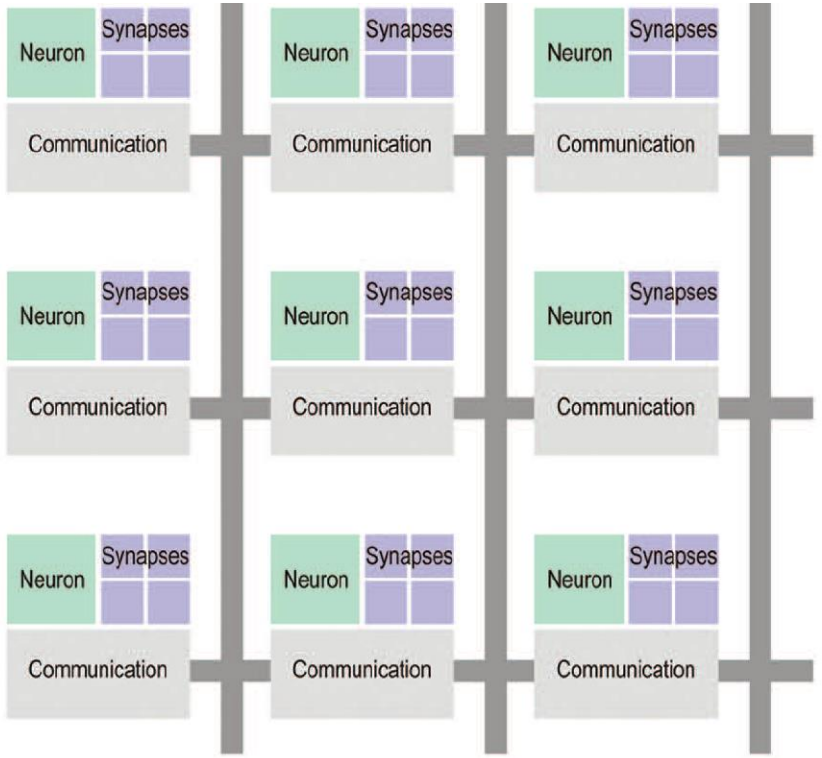


- **Limitations of CMOS chips**
- Current efforts to implement low energy deep learning with emerging nanodevices
- Futuristic ideas

IBM's TrueNorth neuromorphic CMOS chip

- Highly parallel, colocalized memory and processing
- Low power consumption 20 mW/cm^2 (processor 100 W/cm^2)
- Cannot learn



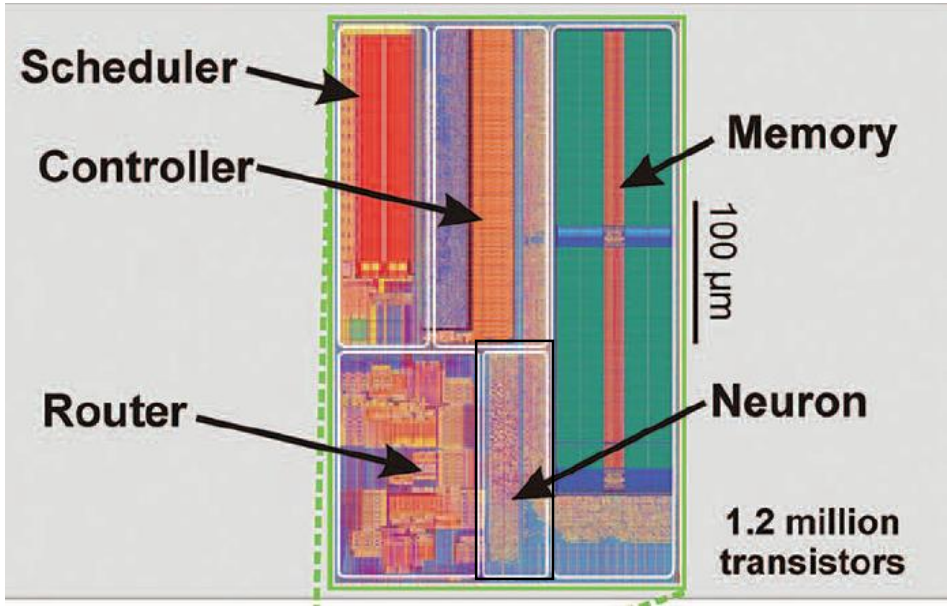


Merolla et al, *Science* **345**, 668 (2014)

CMOS neurons and synapses are complex circuits

- A transistor is nanoscale but it is just a switch
- CMOS does not provide memory (volatile)

CMOS neuron **10-100 µm**
CMOS synapse **10 µm**



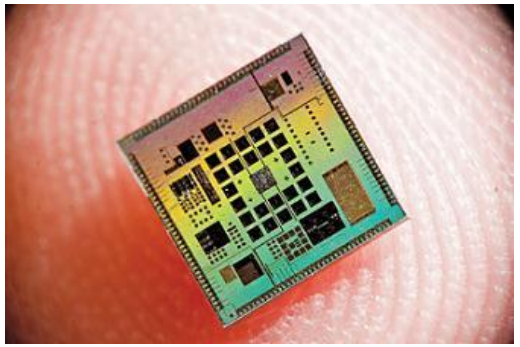
Merolla et al, *Science* **345**, 668 (2014)

Brainscales 20 wafer machine. 4M neurons, 1B synapses

Building small neuromorphic chips requires to complement CMOS with nanoscale devices emulating synapses and neurons

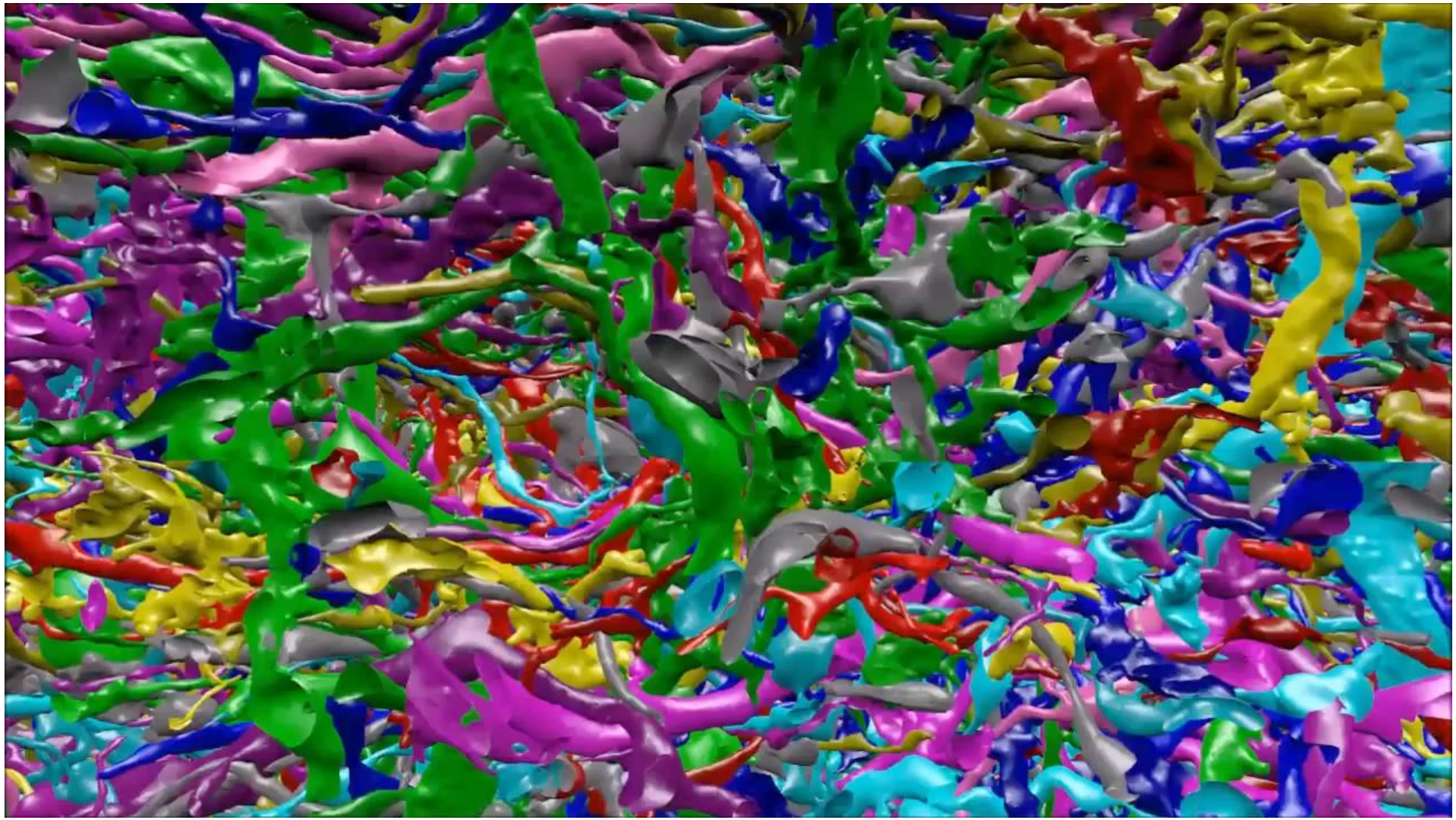
Hundred millions of neurons and synapses in a 1 cm^2 chip

→ Each device smaller than 1 μm^2



→ nanosynapses, nanoneurons

10^4 synapses / neurones = 10^4 wires/neurons



Moritz Helmstaedter lab, retina flight 2013

- Limitations of CMOS chips
- **Current efforts to implement low energy deep learning with emerging nanodevices**
- Futuristic ideas

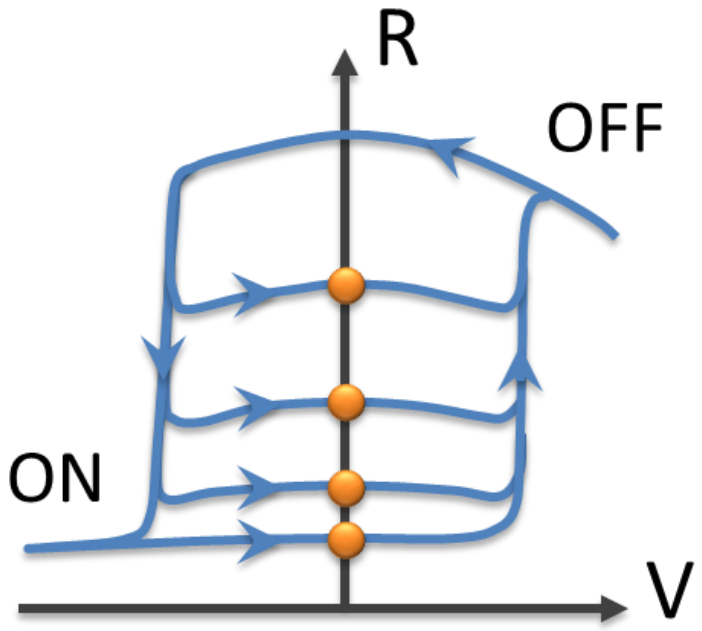
- Limitations of CMOS chips
- **Current efforts to implement low energy deep learning with emerging nanodevices**
- Futuristic ideas

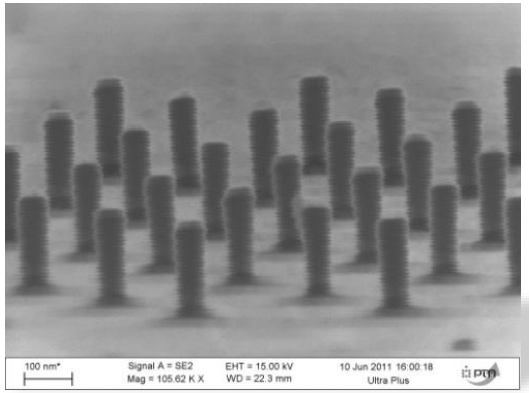
This talk: electronic nanodevices

Memristors implement the fundamental ingredients of synapses

Tunable nano-resistors with memory

Chua, *IEEE Trans. Circuit Theory* (1971)

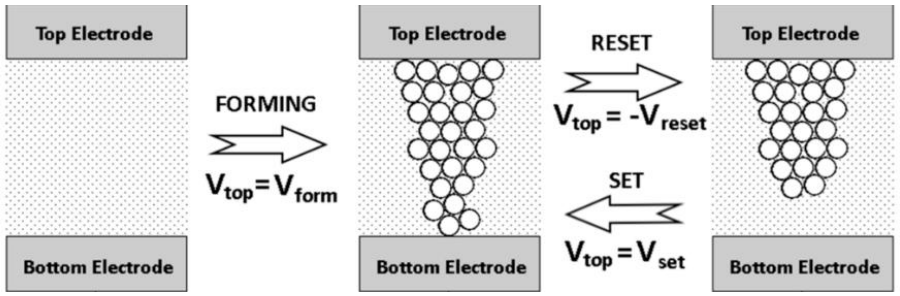




There are many different flavors of memristors

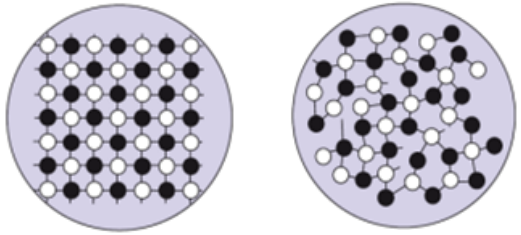
Standard memristors

Red-Ox



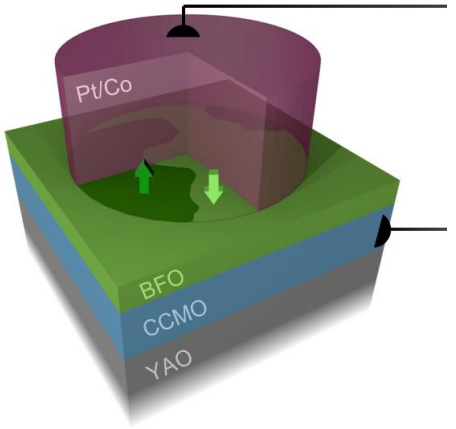
Yang et al., *Nat. Nano*. 8, 13 (2013)

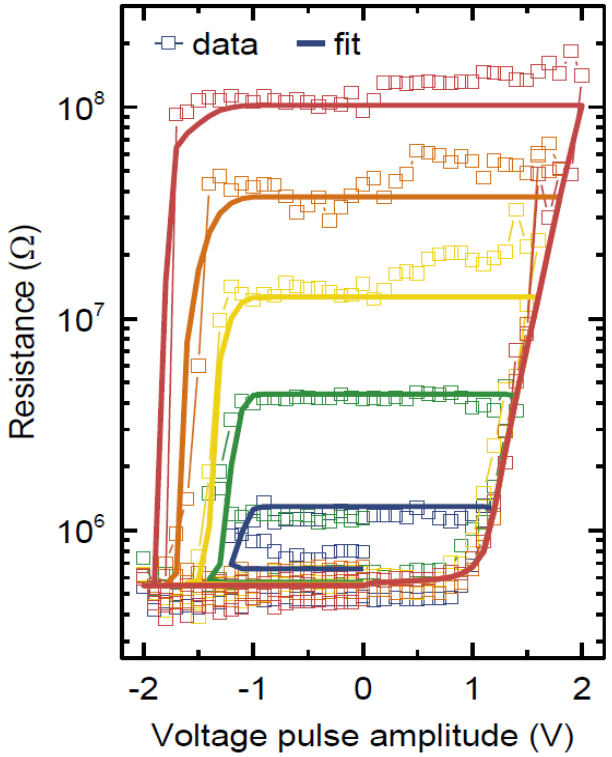
Phase change



Kuzum et al, *Nanotechnology 24, 382001* (2013)

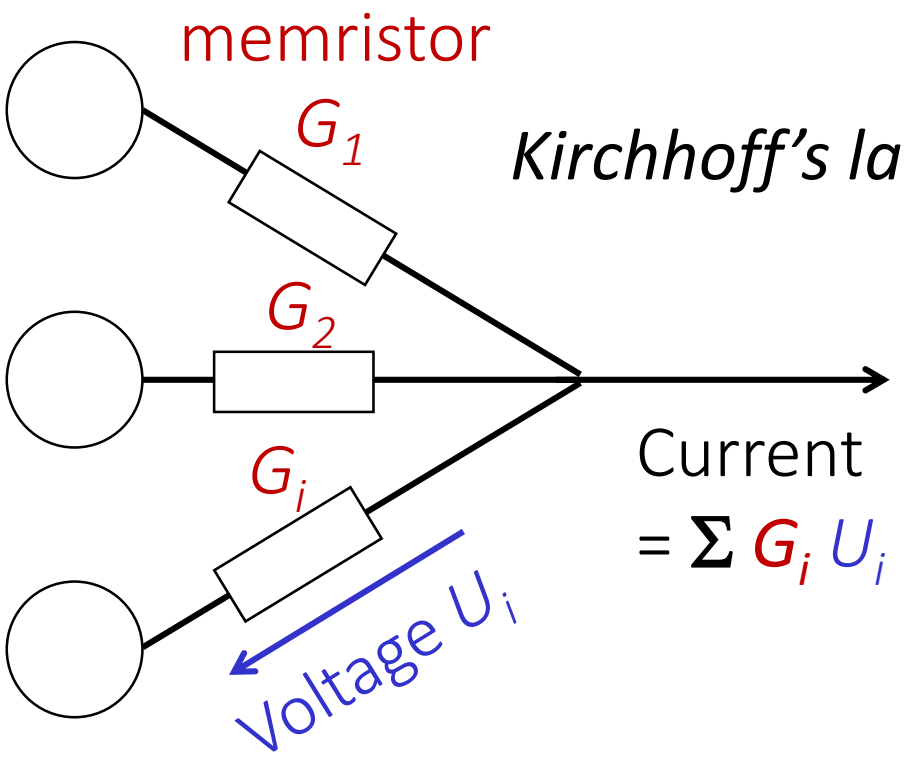
Ferroelectric memristors

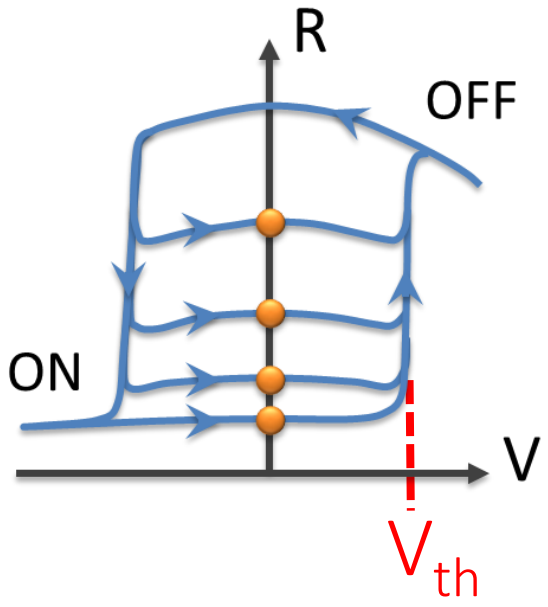




André Chanthbouala, **JG** et al, *Nat. Mat.* 11, 860 (2012)

Memristors emulate electronic synapses: the weight is their tunable conductance G

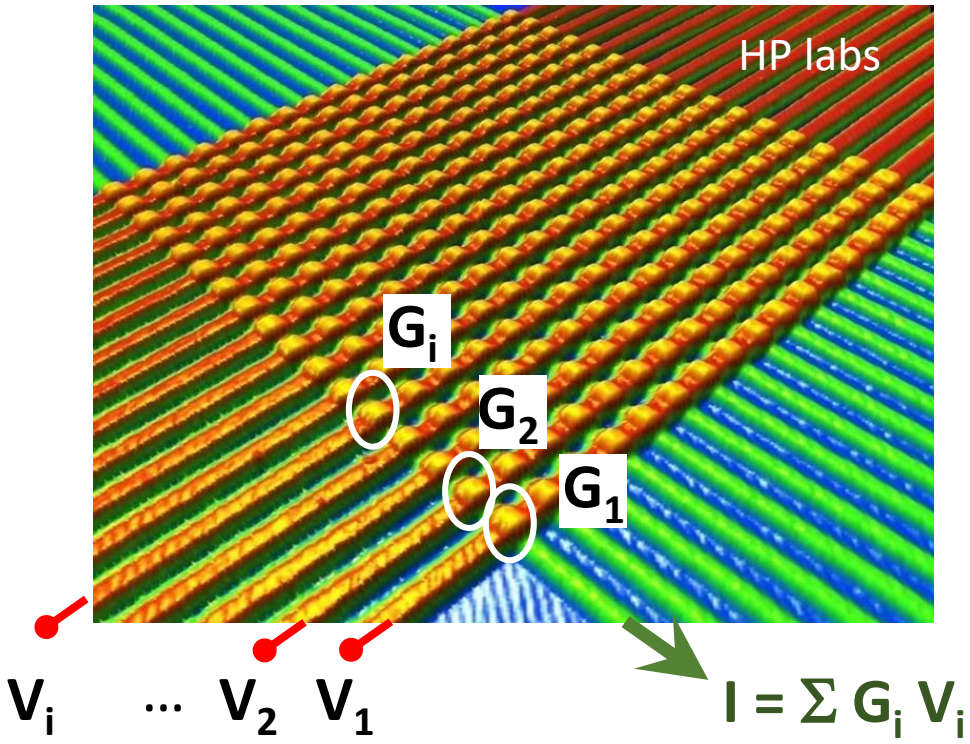


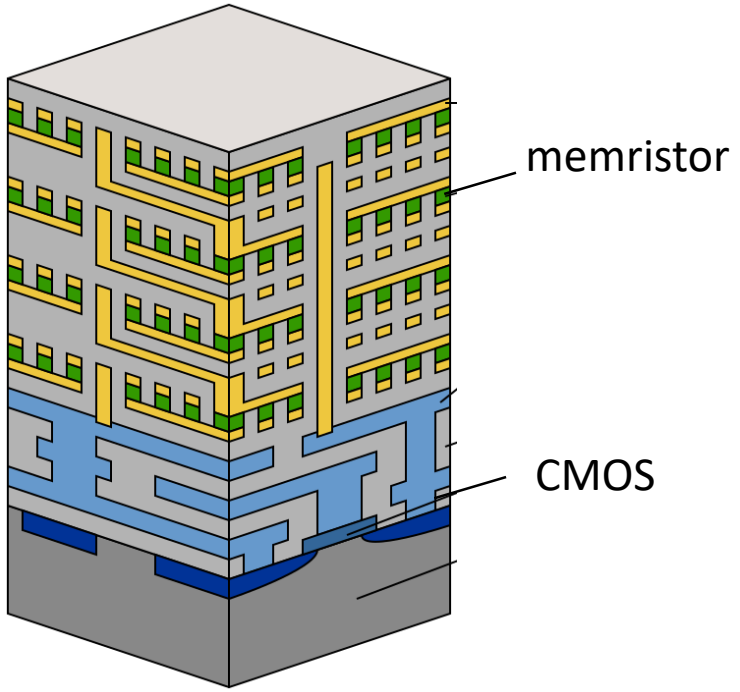


$U_i < V_{th}$: calculating mode

$U_i > V_{th}$: learning mode

Memristor crossbar arrays can fully connect layers of > 100 neurons



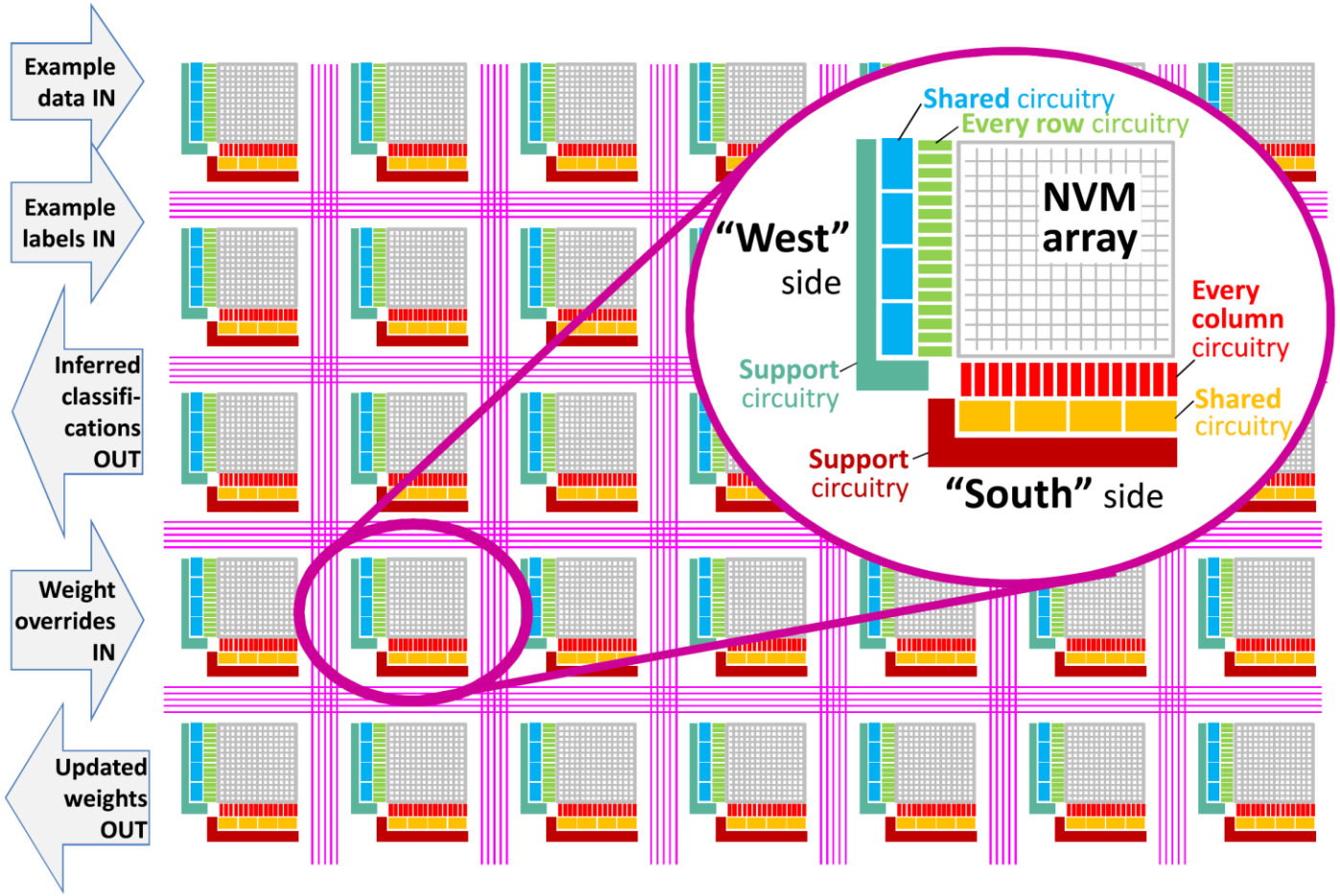


10 000 synapses per neuron ?

Strukov and Williams, PNAS 106, 20155 (2009)

Memristors open a path to energy-efficient real-time learning, but there is no chip yet

Expected : 100 x faster than GPUs and 100 x less energy consumption



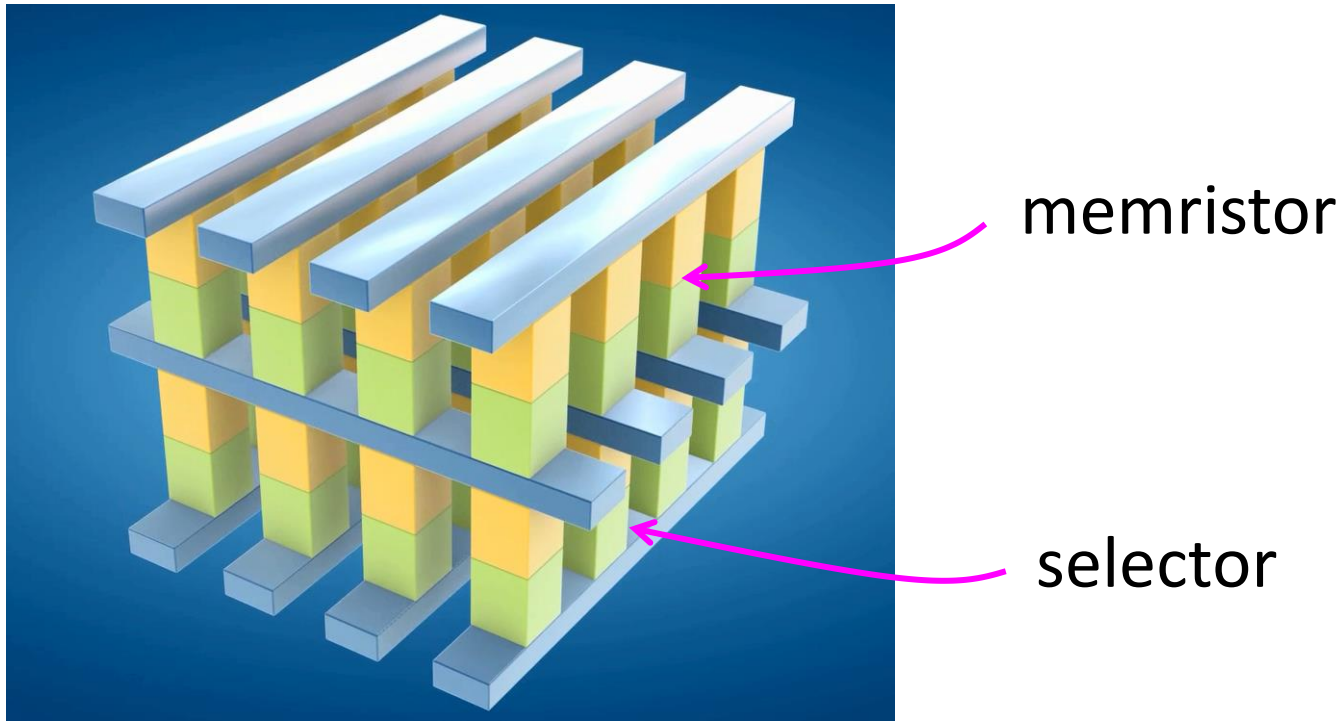
Narayan et al, IBM J. RES. & DEV. 61, NO. 4/5, 11 (2017)

Ambrogio et al, Nature 558, 60 (2018)

One challenge being currently tackled is building large arrays of memristors

Recent progress has been achieved towards the commercialization of binary memories made of memristor arrays

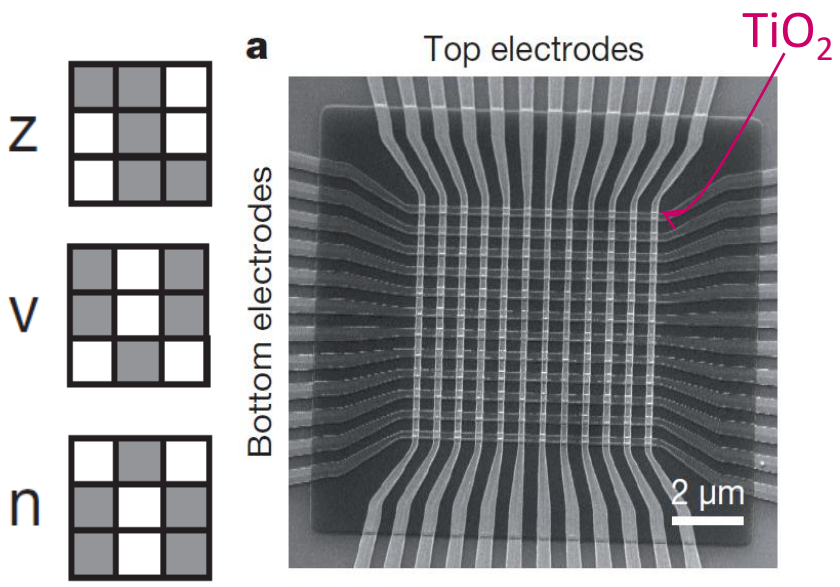
- 3D Xpoint, Intel/Micron Optane Lenovo 32 Gbits



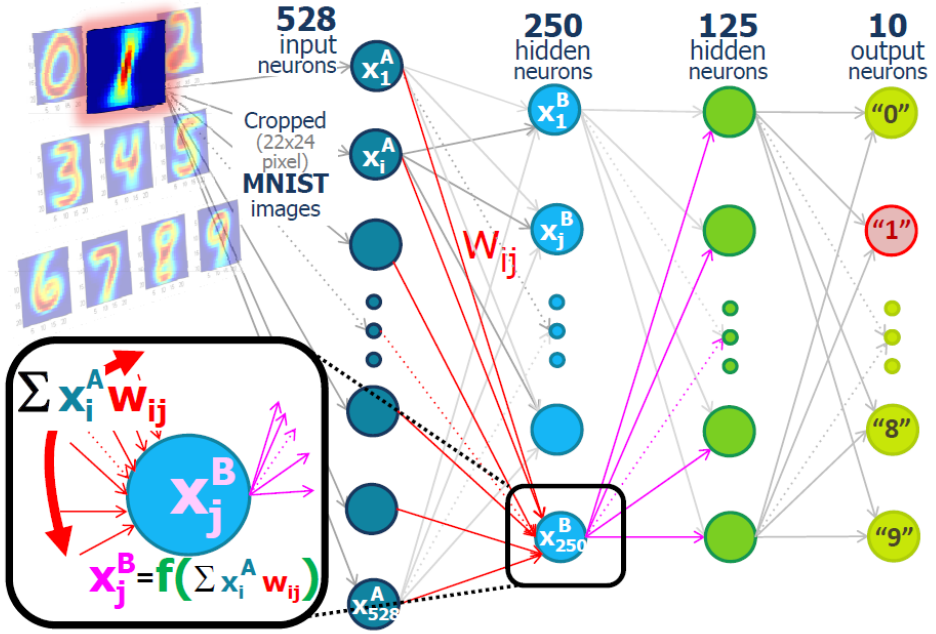
http://www.theregister.co.uk/2017/01/04/optane_arrives_at_ces/

First results of supervised learning through back-propagation have been obtained but the accuracy is low due to devices imperfections (non-linear, imprecise, noisy)

Prezioso et al, *Nature* 521, 61 (2015)



Burr et al, *IEEE IEDM* (2014)

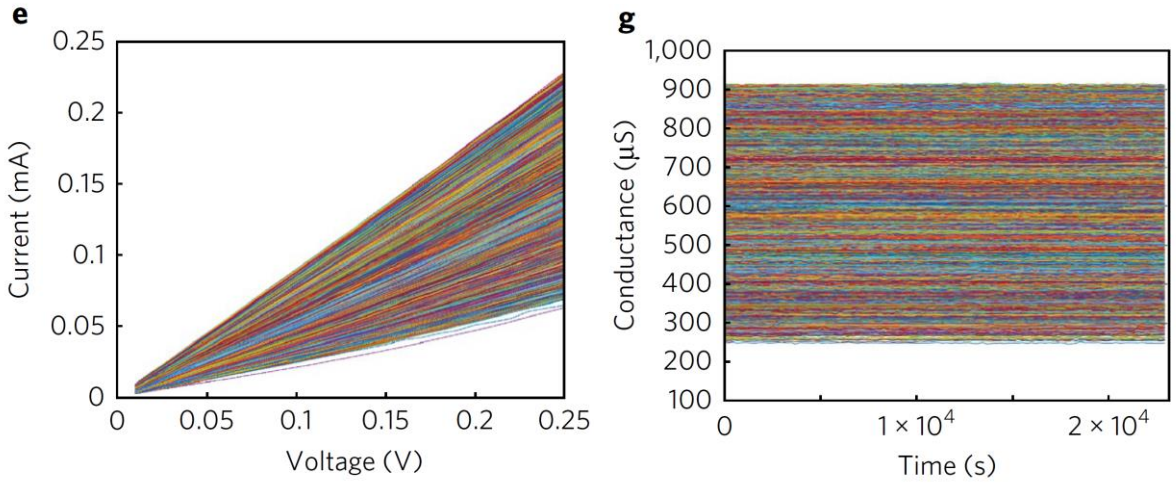


Accuracy on MNIST 82%

IBM experiments: Handwritten digit recognition with ~ 165 000 synapses (phase change with selector)

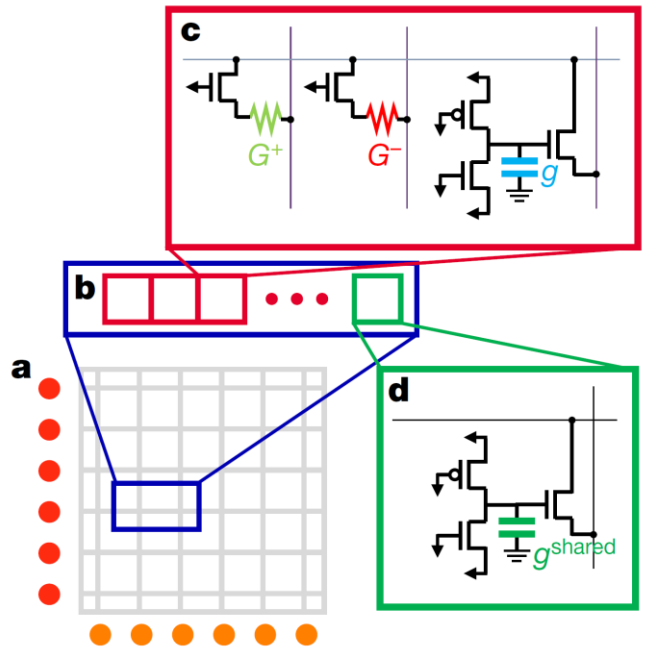
A solution towards implementing backpropagation is to improve nanodevice properties

- Optimizing memristor properties



Li et al, Nature electronics **1**, 52–59 (2018)

- Complementing memristors with high accuracy weight

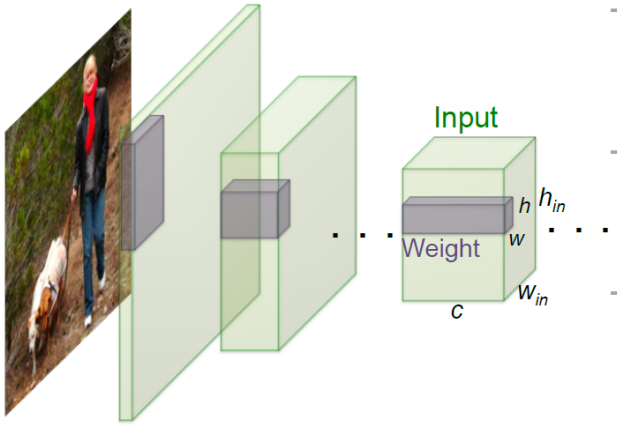


Accuracy on MNIST 98%

Ambrogio et al, Nature 558, 60 (2018)

In parallel, there is a lot of work on finding algorithms that can be accelerated on hardware

Ex : XNOR nets, Rastegari et al, arXiv:1603.05279



	Network Variations	Operations used in Convolution	Memory Saving (Inference)	Computation Saving (Inference)	Accuracy on ImageNet (AlexNet)
Standard Convolution	Real-Value Inputs 0.11 -0.21 ... -0.34 / -0.25 0.61 ... 0.52; Real-Value Weights 0.12 -1.2 ... 0.41 / -0.2 0.5 ... 0.68	+ , − , ×	1x	1x	%56.7
Binary Weight	Real-Value Inputs 0.11 -0.21 ... -0.34 / -0.25 0.61 ... 0.52; Binary Weights 1 -1 ... 1 / -1 1 ... 1	+ , −	~32x	~2x	%56.8
BinaryWeight Binary Input **(XNOR-Net)**	Binary Inputs 1 -1 ... -1 / -1 1 ... 1; Binary Weights 1 -1 ... 1 / -1 1 ... 1	XNOR , bitcount	~32x	~58x	%44.2

Dates n/a. @ None

Poster

Training Deep Neural Networks with 8-bit Floating Point Numbers

NIPS 2018

Naigang Wang · Jungwook Choi · Daniel Brand · Chia-Yu Chen · Kailash Gopalakrishnan

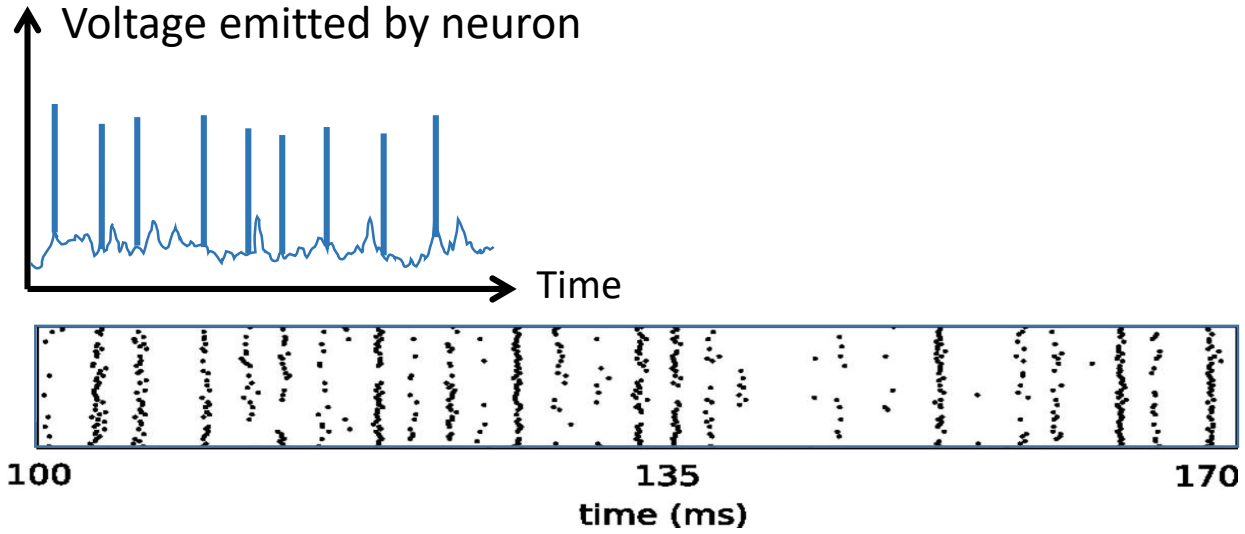
But nanodevice noise and non-linearity remain a problem

- Limitations of CMOS chips
- Current efforts to implement low energy deep learning with emerging nanodevices
- **Futuristic ideas**

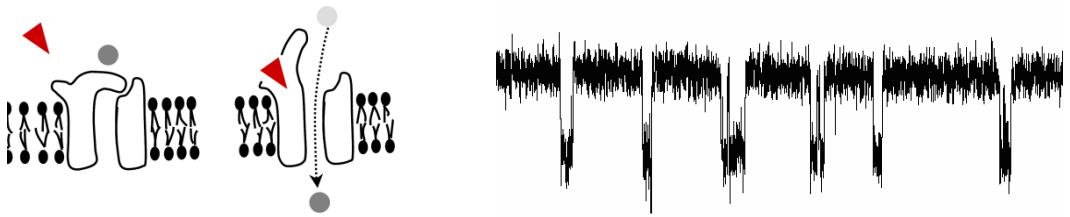
Can noise be used instead of mitigated ?

Biological synapses and neurons are noisy: the brain seems to operate at the thermal limit to minimize its power consumption

Neural spikes in response to the same input recorded 50 times

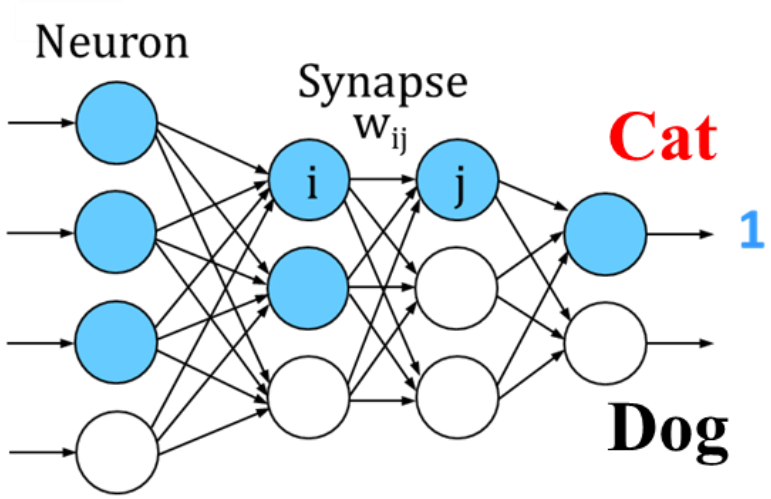


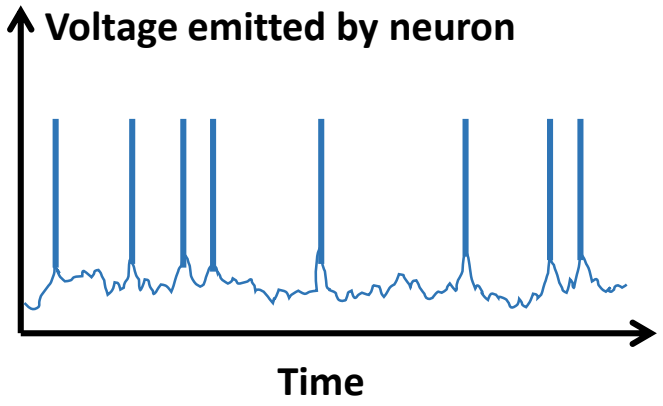
Synaptic ionic channels

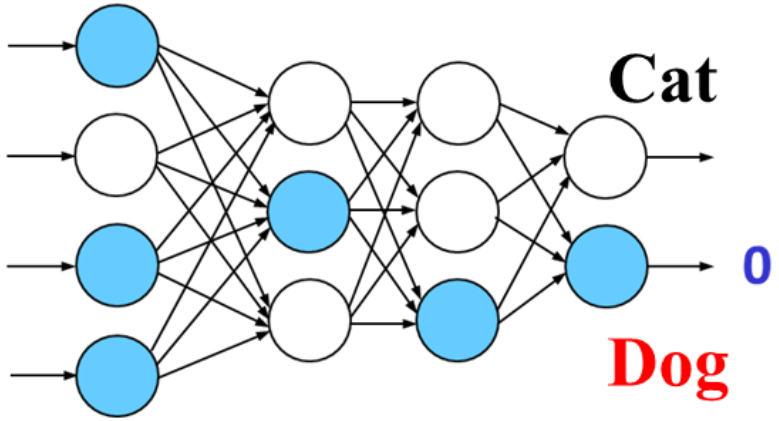


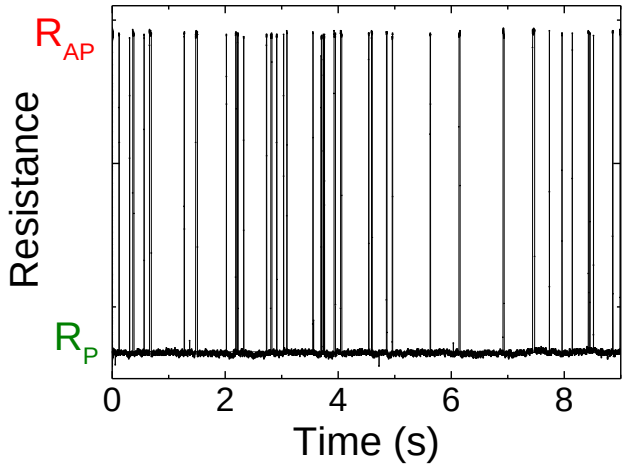
Computing at low power with stochastic components is possible

Can we achieve reliable computations with noisy devices ?







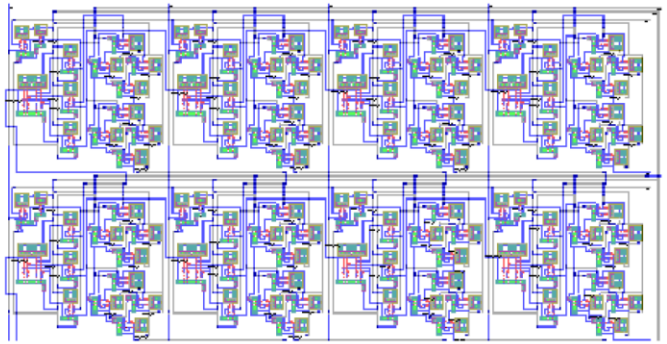


Low energy consumption

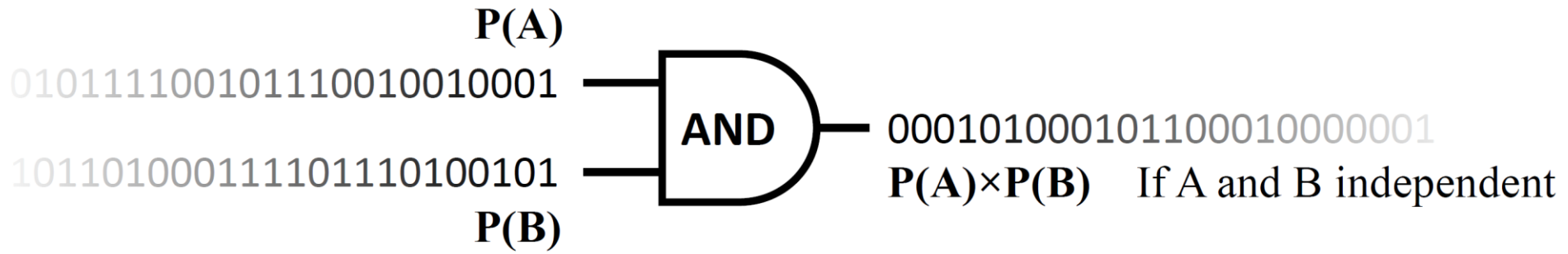
First strategy for computing with noisy devices: accumulating evidence in time by sampling the same device several times

Example: stochastic computing

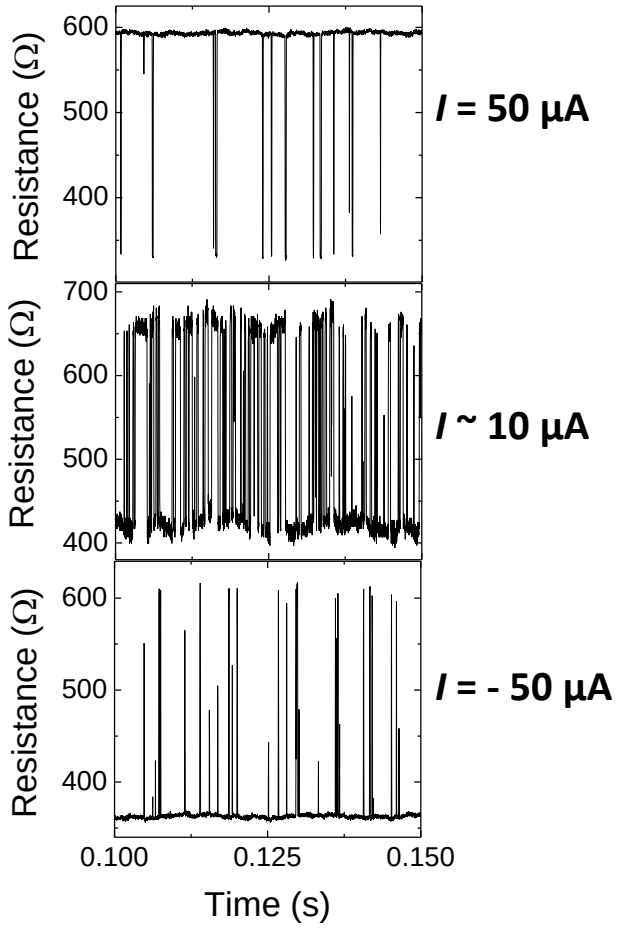
Conventional multiplier:



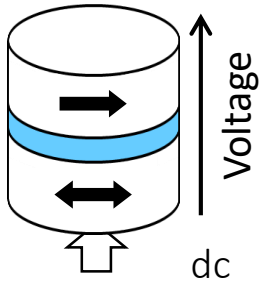
Stochastic computing multiplier:

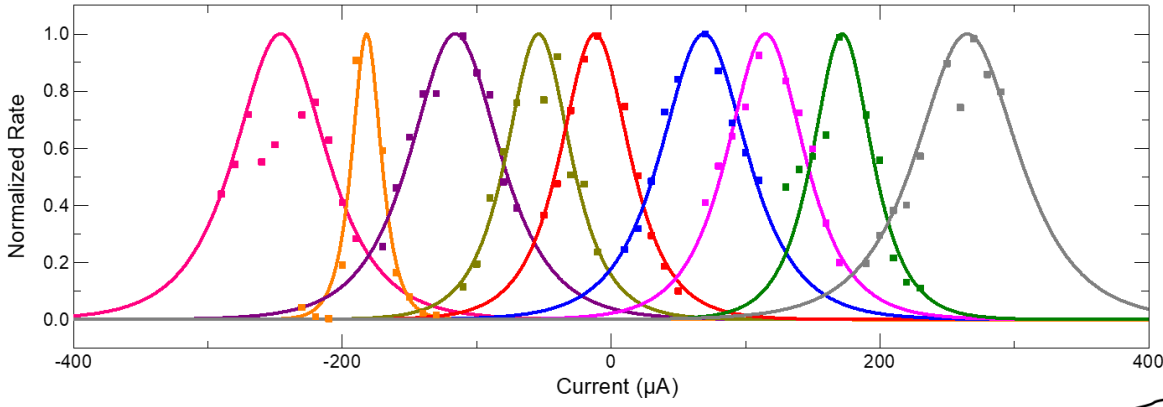


2nd strategy for computing with noisy devices: sampling from populations of nanodevices (redundance)

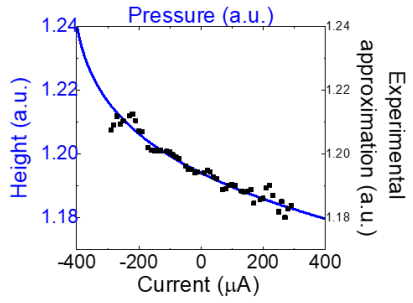


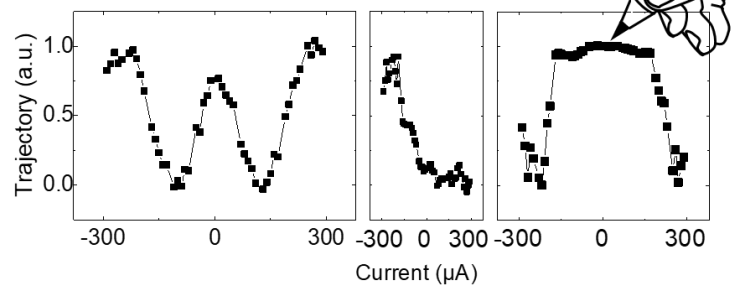
Superparamagnetic tunnel junctions





$$H(I) = \sum_{i=1}^{9} w_i r_i(I)$$





A. Mizrahi et al, Nature Com. 9:1533 (2018)

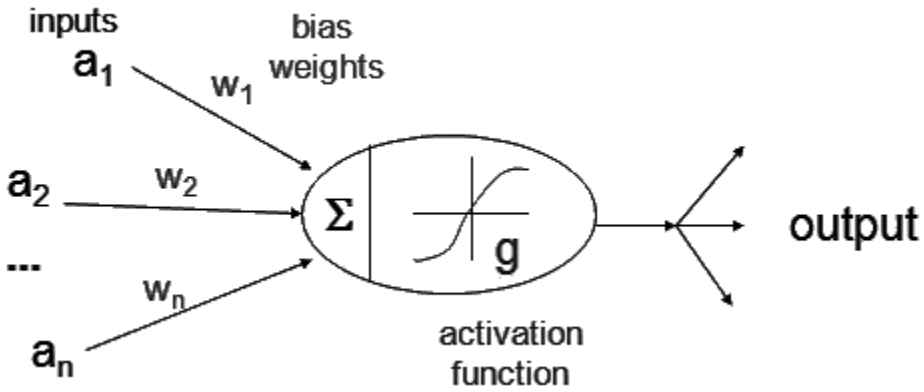
Could we leverage noise for computing ?

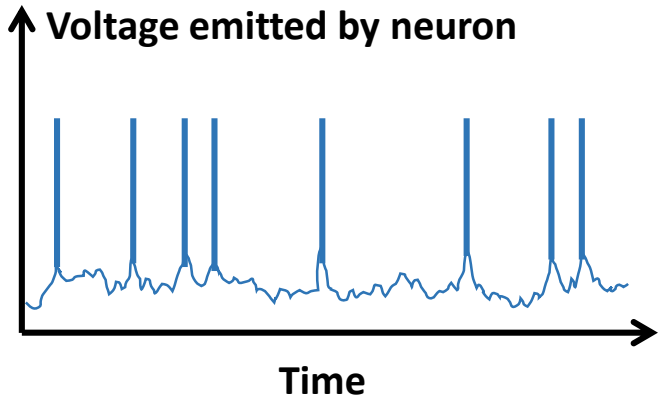
Randomness can be useful - Example: stochastic resonance

Low noise **Optimal noise** High noise

Gammaitoni et al, Reviews Of Modern Physics, 70(1), 223–287 (1998)

Locatelli, Mizrahi, JG et al, Phys. Rev. Appl. 2014

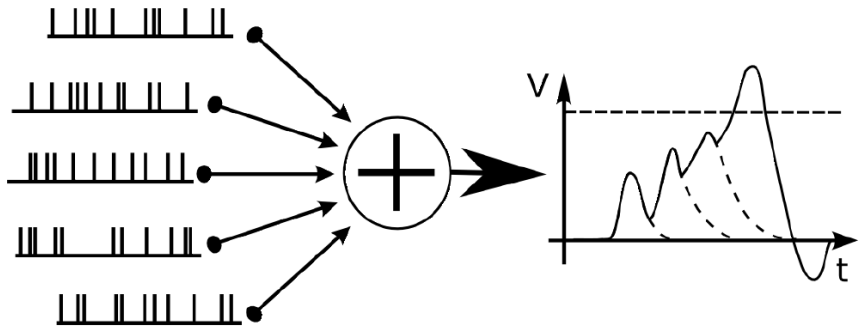


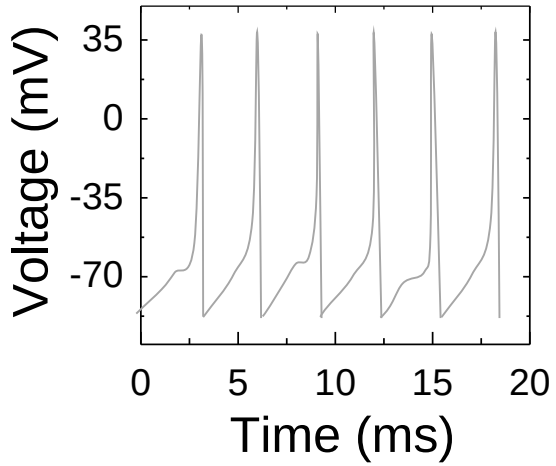


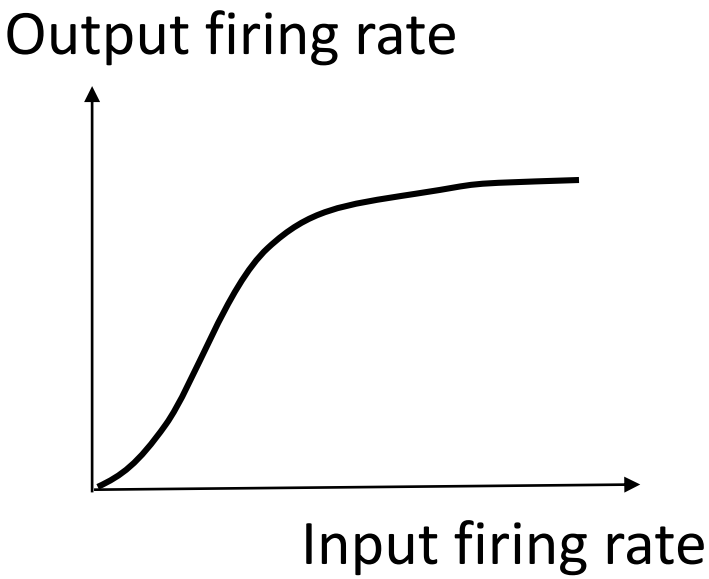
Can spikes help computing with nanodevices?

Biological neurons are more than non-linear functions : they spike !

Integration, spikes

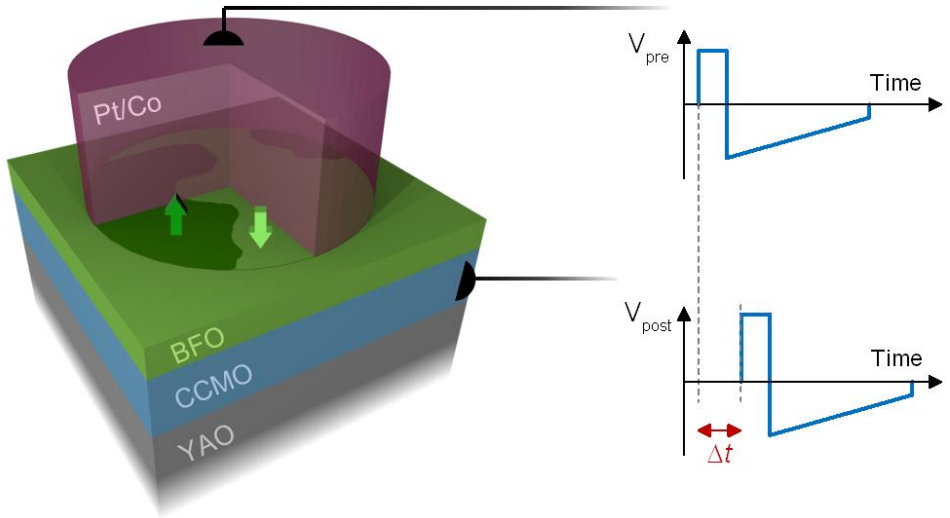






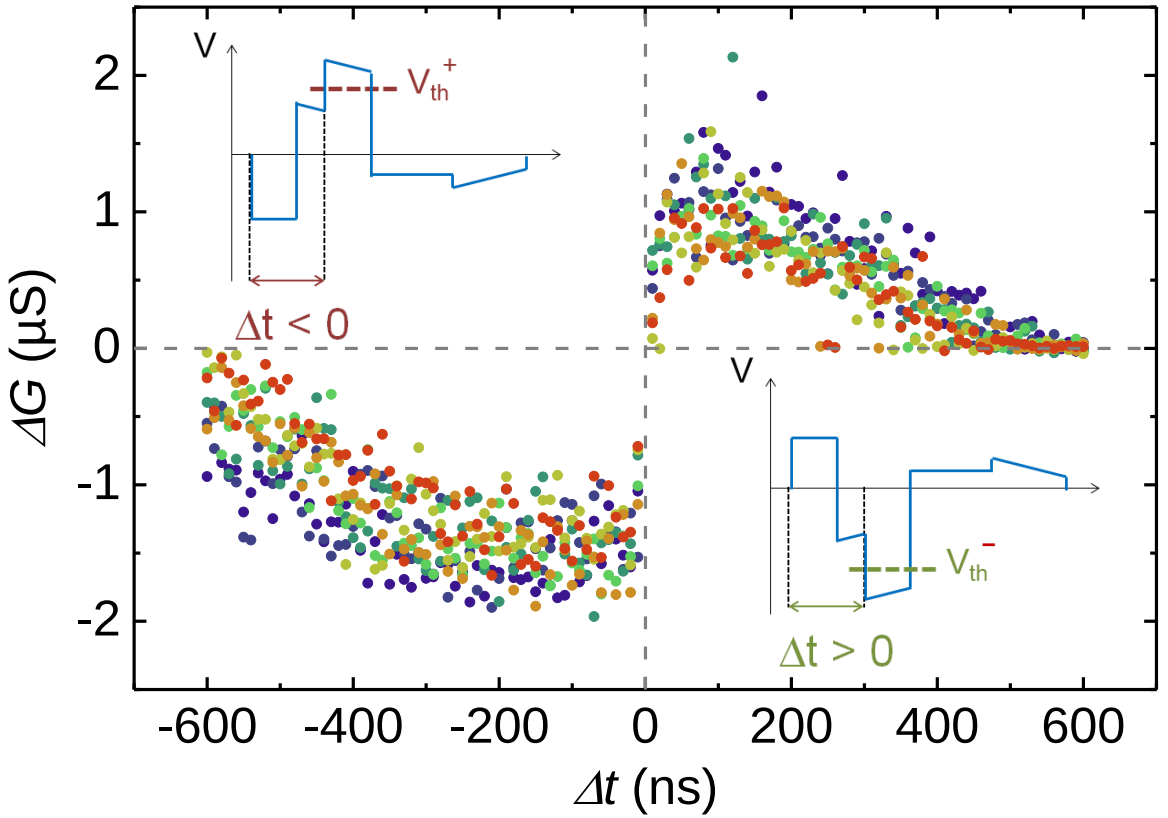
Spikes allow 1) unsupervised learning 2) autonomous evolution of memristor conductance

pre-synaptic neuron



post-synaptic neuron

Spike timing plasticity

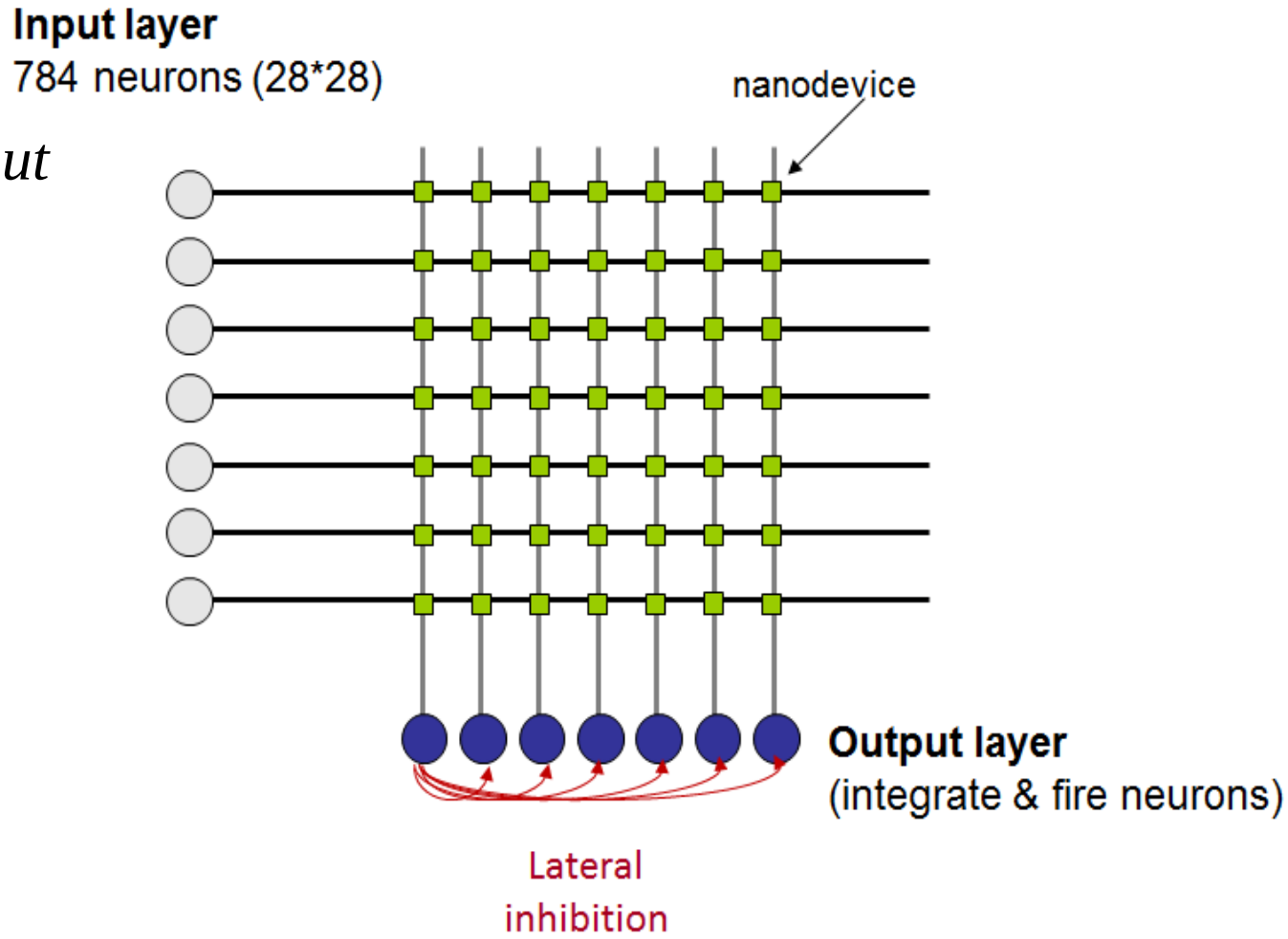


Jo et al, *Nanoletters* 10, 1297 (2010) Zamarrenos-Ramos et al, *Frontiers in Neuroscience* 5, 26 (2011)

Sören Boyn, **JG** et al, Nature Com. 8, 14736 (2017)

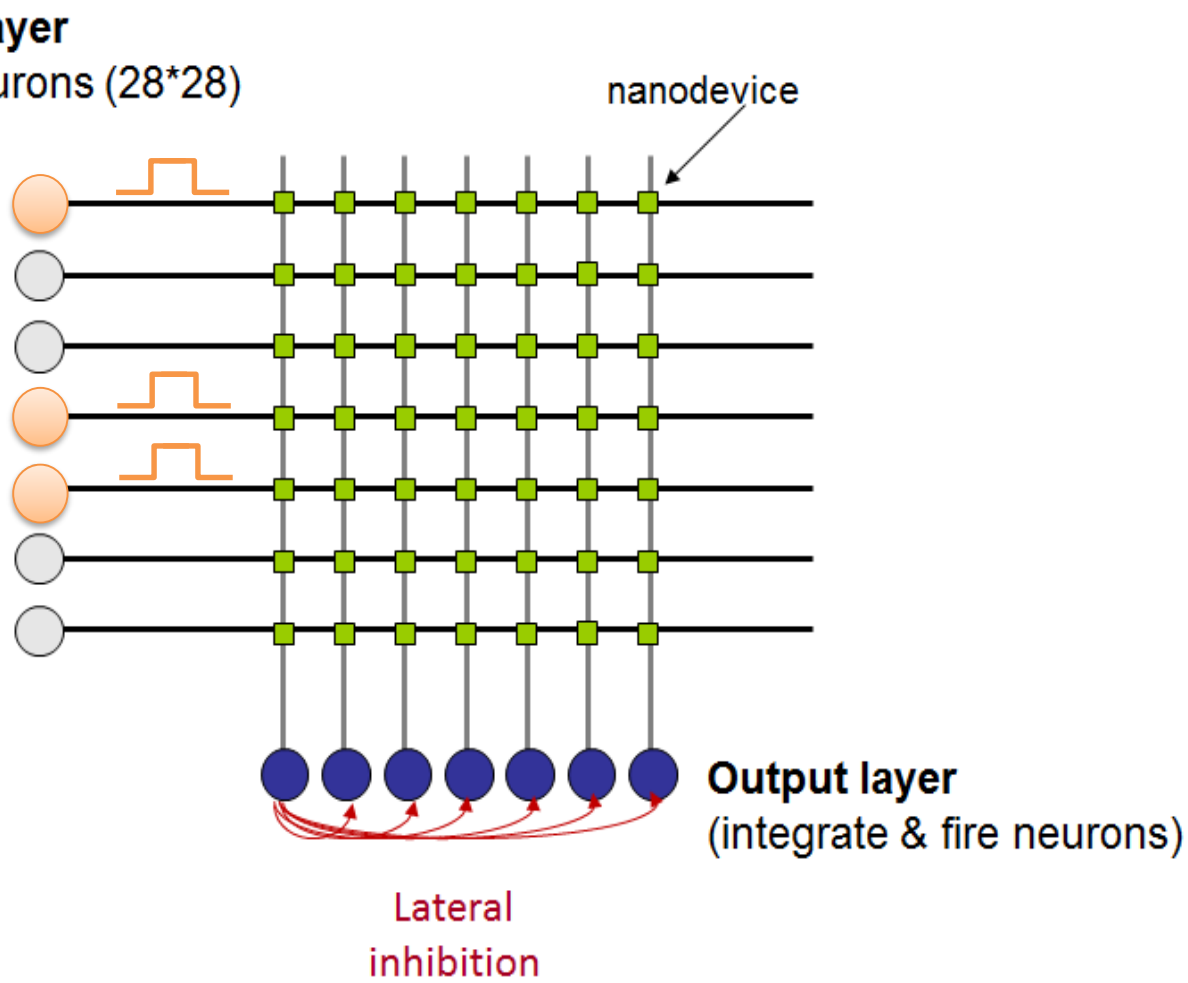
How memristor learn without supervision through STDP

Presented input



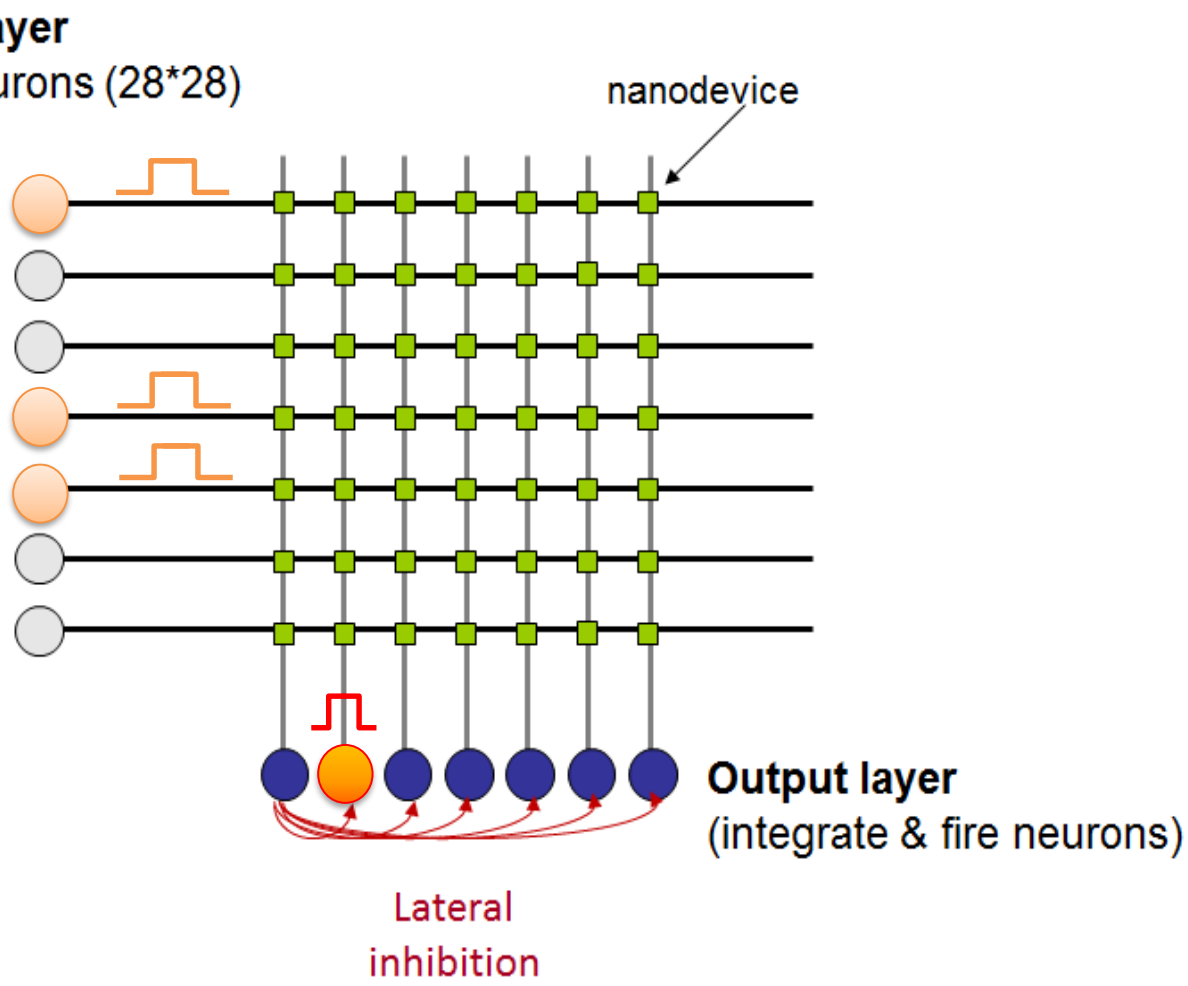
How memristor learn without supervision through STDP

Presented input



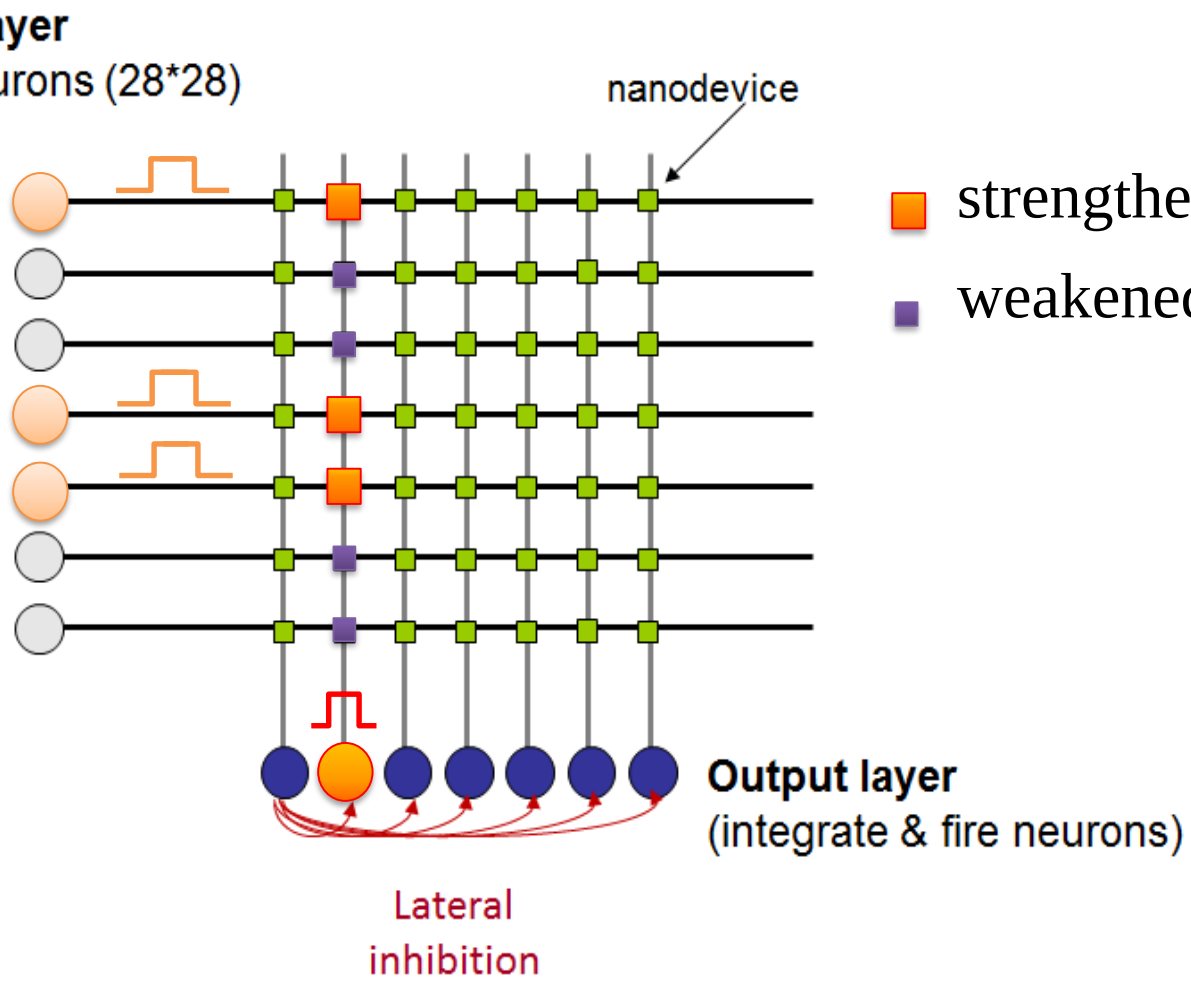
How memristor learn without supervision through STDP

Presented input



How memristor learn without supervision through STDP

Presented input

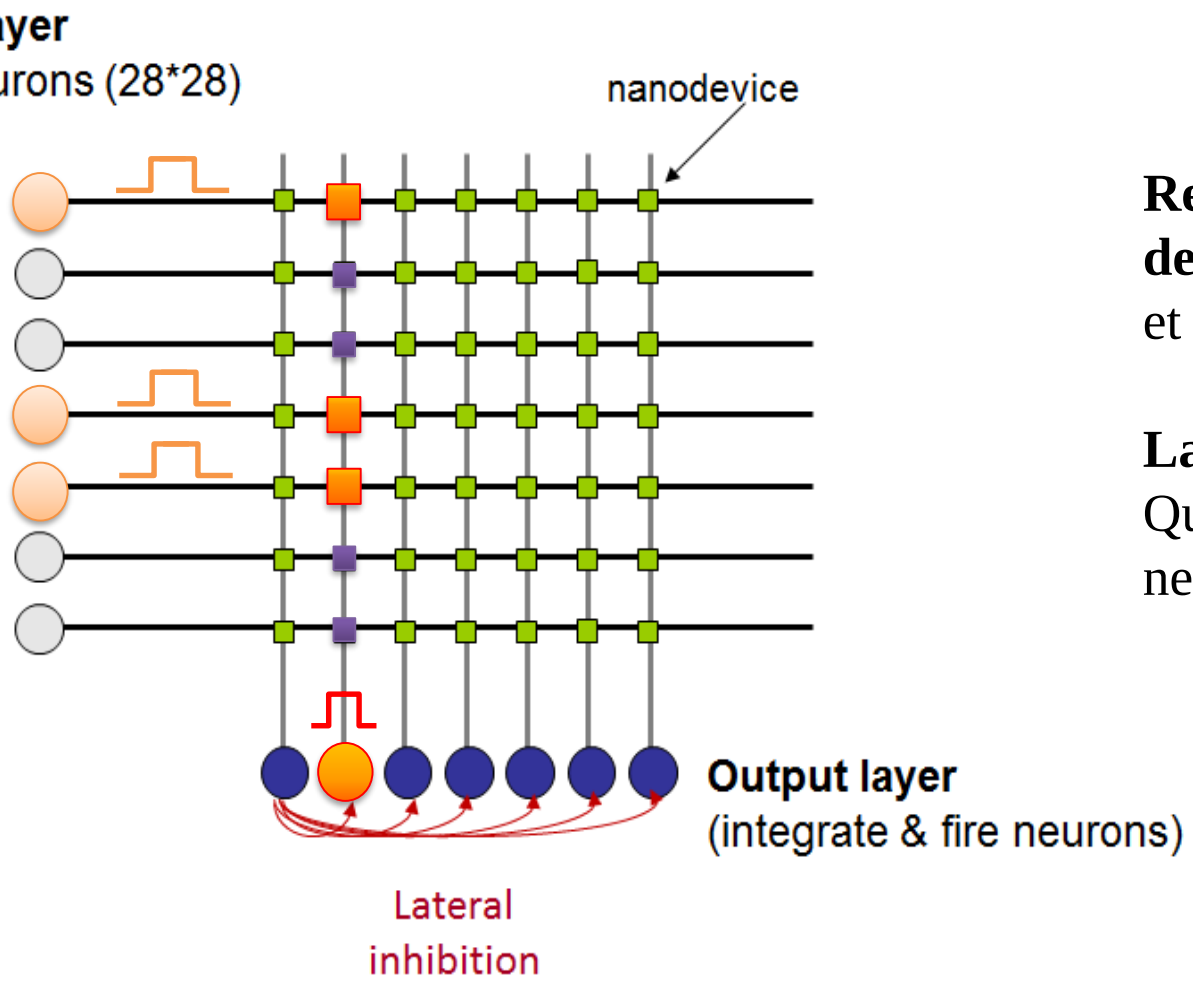


strengthened synapse

weakened synapse

How memristor learn without supervision through STDP

Presented input

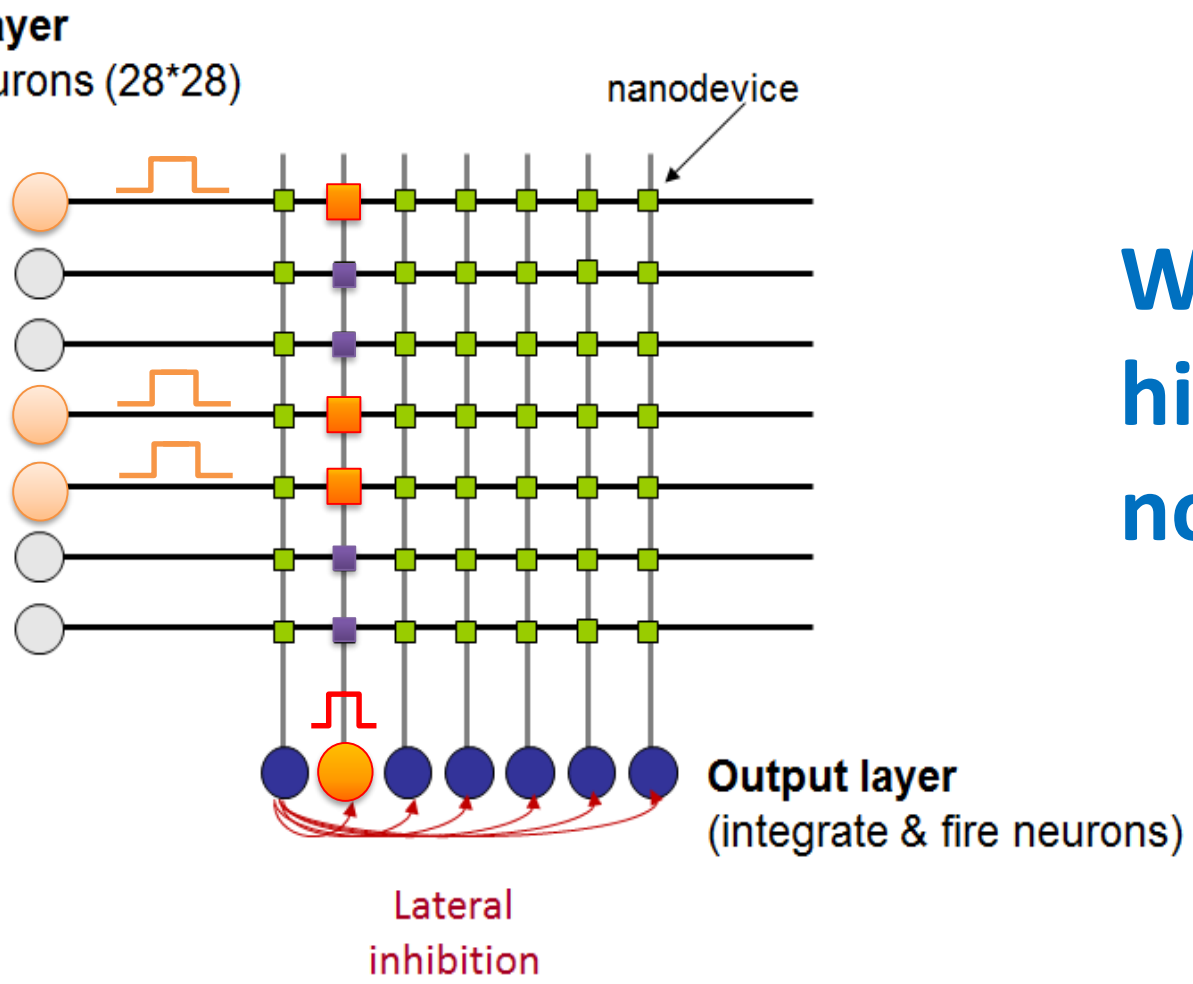


Recent small scale demonstration: Pedretti, Ielmini et al, Sci Rep 2017

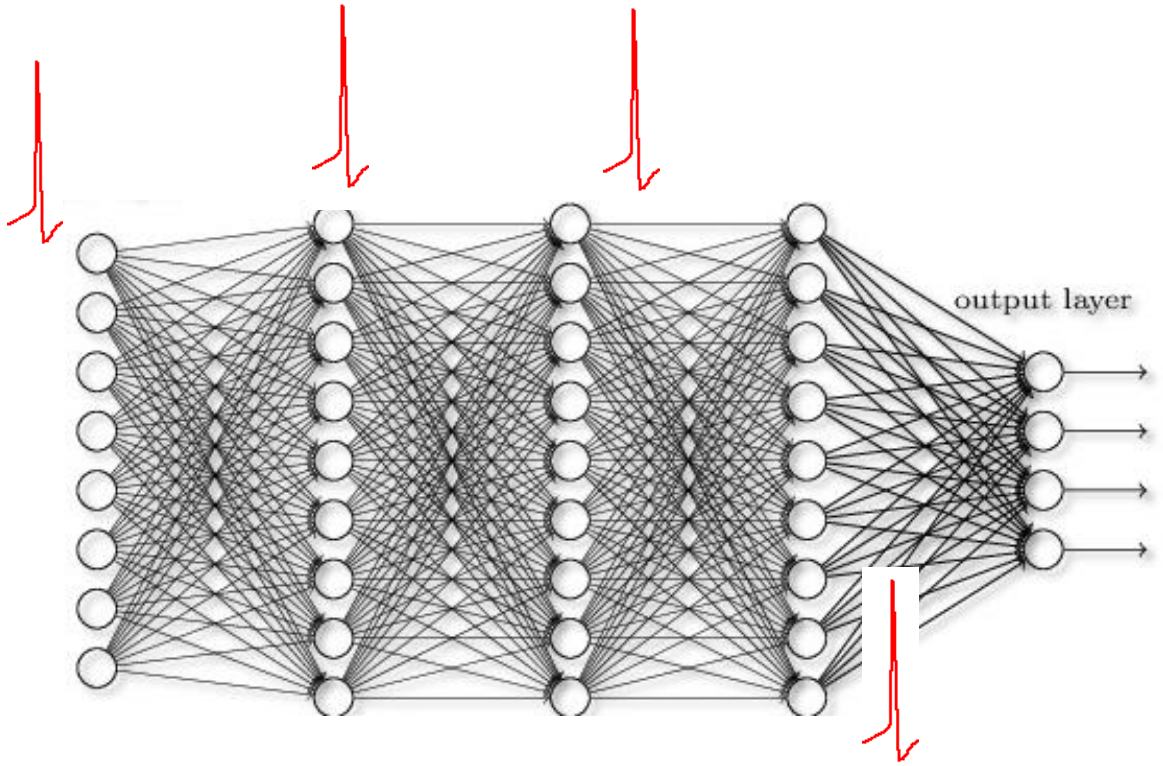
Large scale simulations: Querlioz, Bichler et al, Neural networks 2012

How memristor learn without supervision through STDP

Presented input



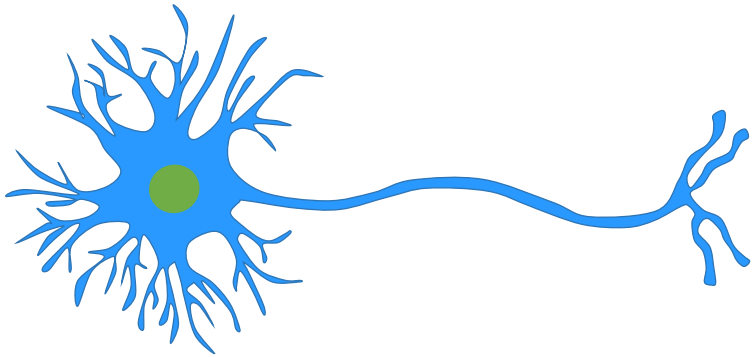
Works even with highly non-linear, noisy devices !

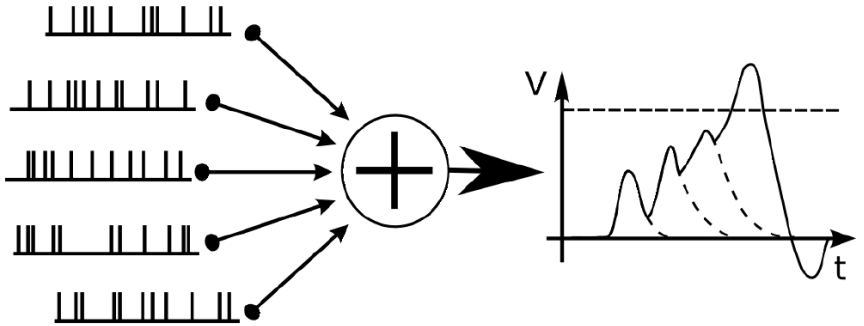


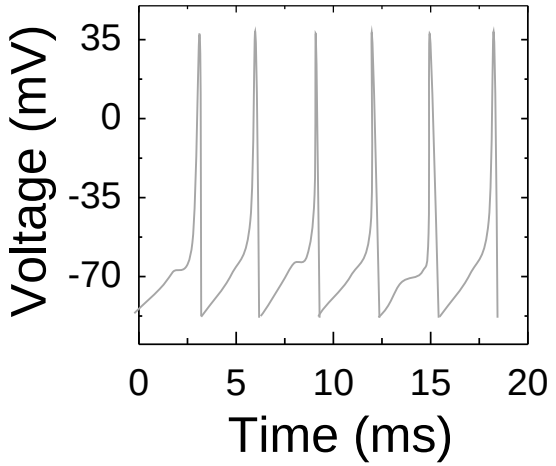
Can non-linear dynamics help ?

Biological neurons are auto-oscillators: can we enhance perfomances of neural networks by giving them dynamical properties ?

Spikes, Non-linear oscillations, Synchronization



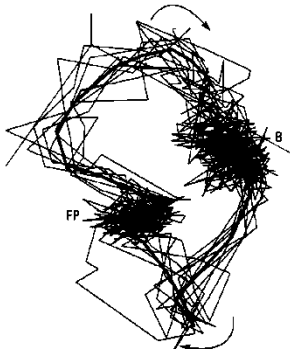




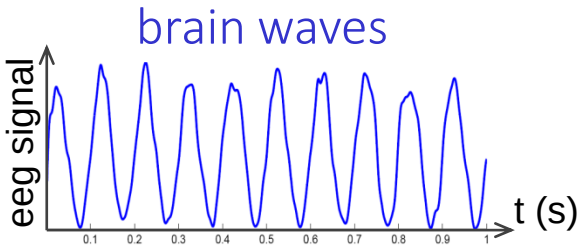
Can we reduce the number of nanodevices by giving them dynamical features ?

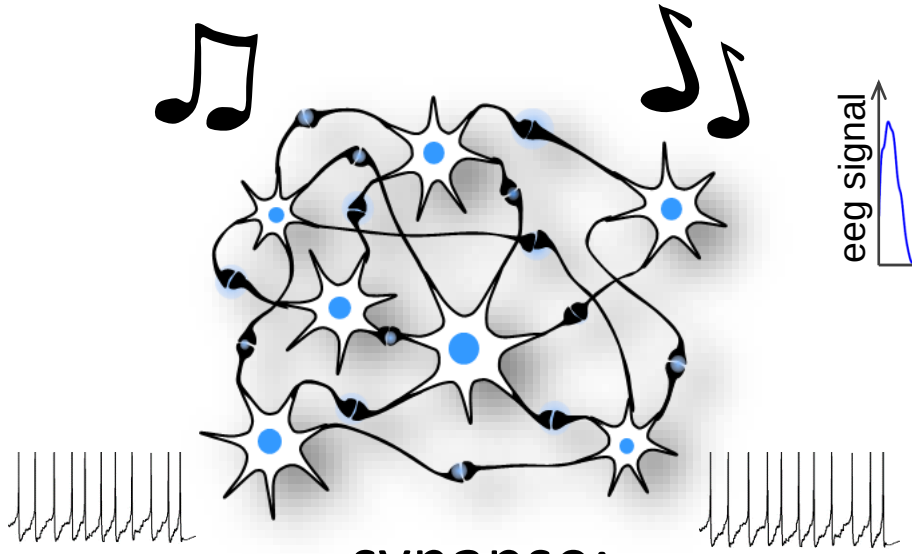
Non-linear dynamics in the brain is probably useful for computing

Complex transients



Synchronization





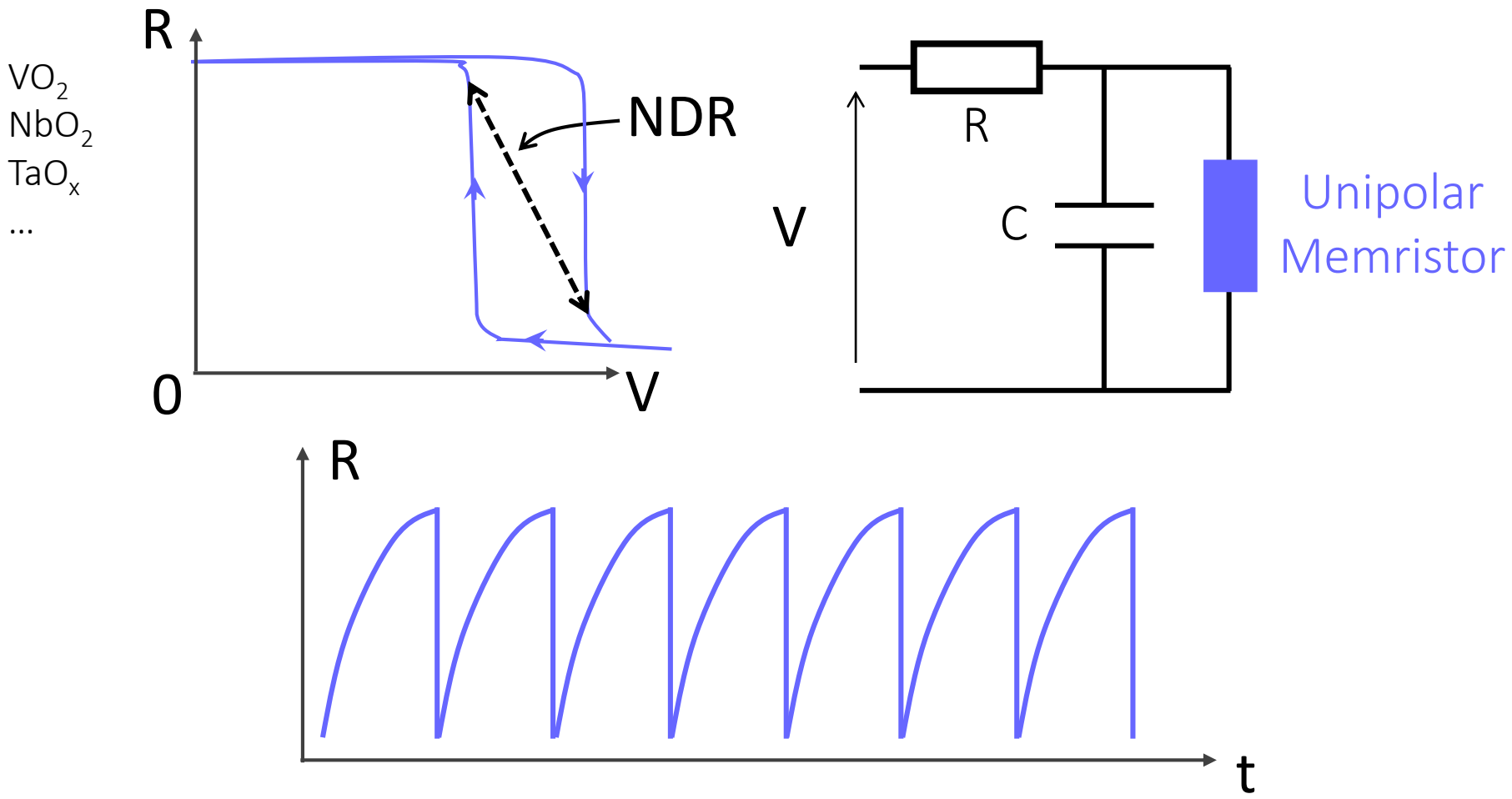
neuron: oscillator

synapse: coupling

neuron: oscillator

Jaeger et al, *Science* (2004), Hoppensteadt et al, *PRL* (1999), Aonishi et al, *PRL* (1999)

There is currently a lot of work towards creating nanoscale dynamical neurons

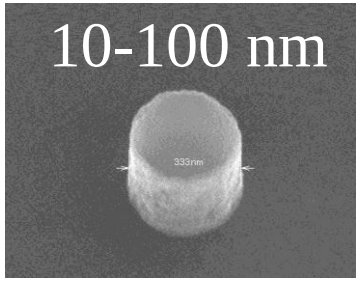


Pickett et al, Nat. Mater. (2013), N. Shukla et al, Sci. Rep. (2014)
A. Sharma, et al, IEEE J. Explor. Solid-State Comput. Devices Circuits (2015)

Magnetic nano-oscillators are non-linear nano-radios

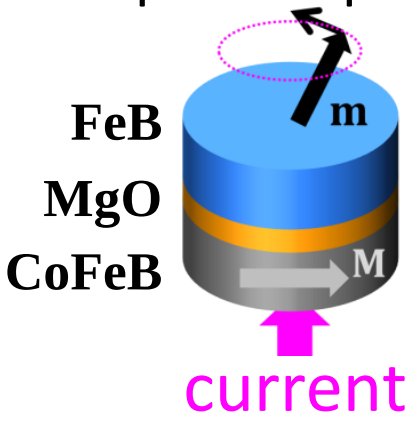
Nanoscale, fast (GHz), non-linear and easily measurable

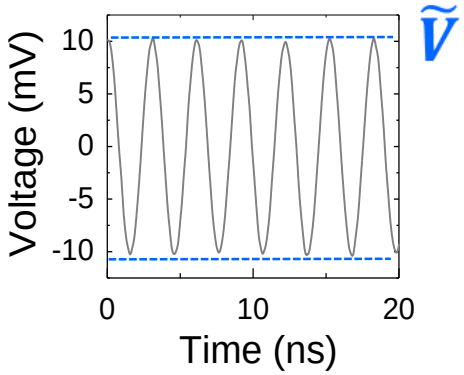
magnetic tunnel junction

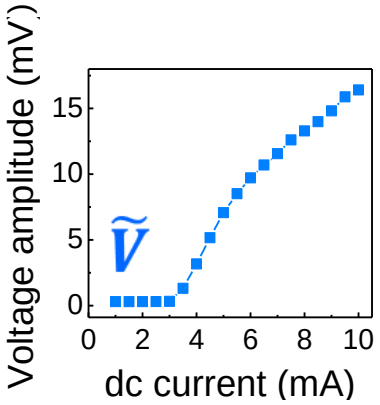


compatible with CMOS

spin torque





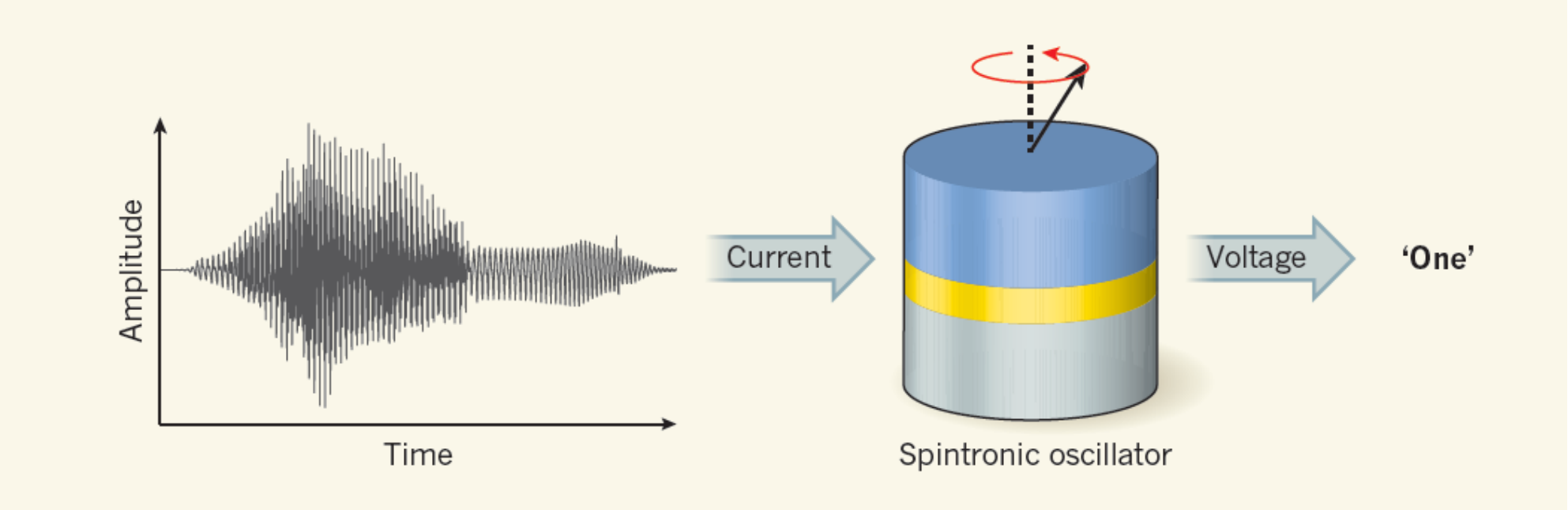


Same structure as magnetic memories

N. Locatelli, V. Cros and J. Grollier, Spin-torque building blocks, Nature Mat. 13, 11 (2014)

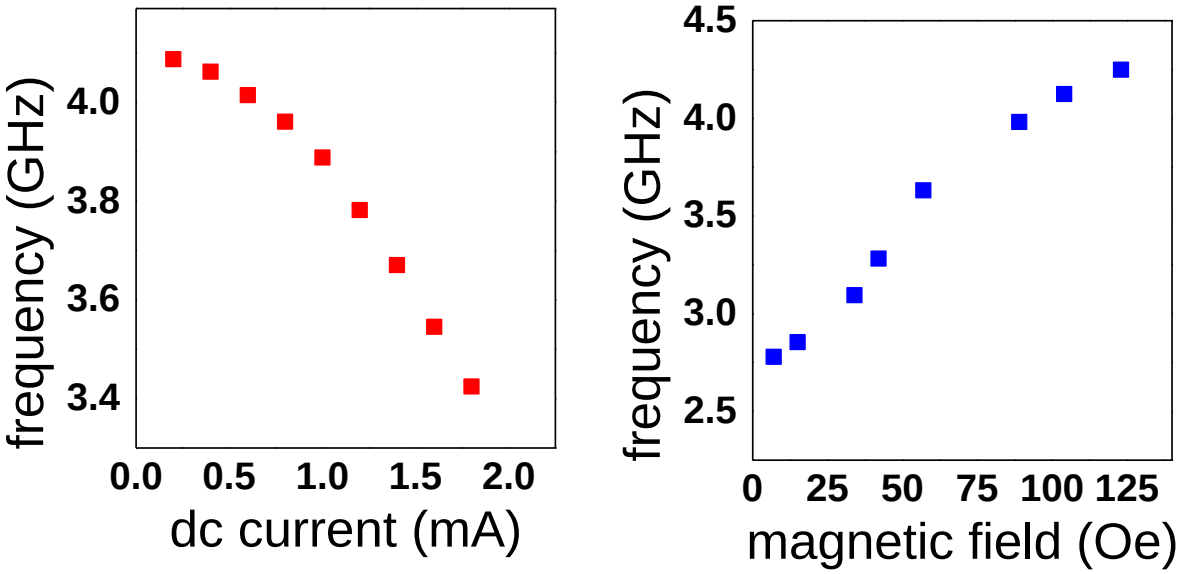
Due to its stability and non-linearity, a single magnetic oscillator can emulate an assembly of neurons and perform neuromorphic computing

Spoken digit recognition through reservoir computing

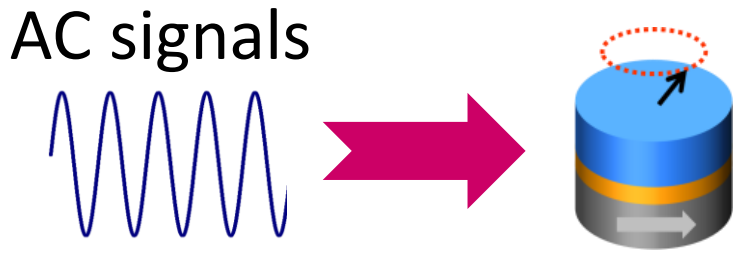


J. Torrejon, M. Riou, F. Abreu Araujo et al, Nature 547, 428 (2017)

Spin-torque nano-oscillators have a high tunability : they are radio-receivers



Enhanced sync ranges

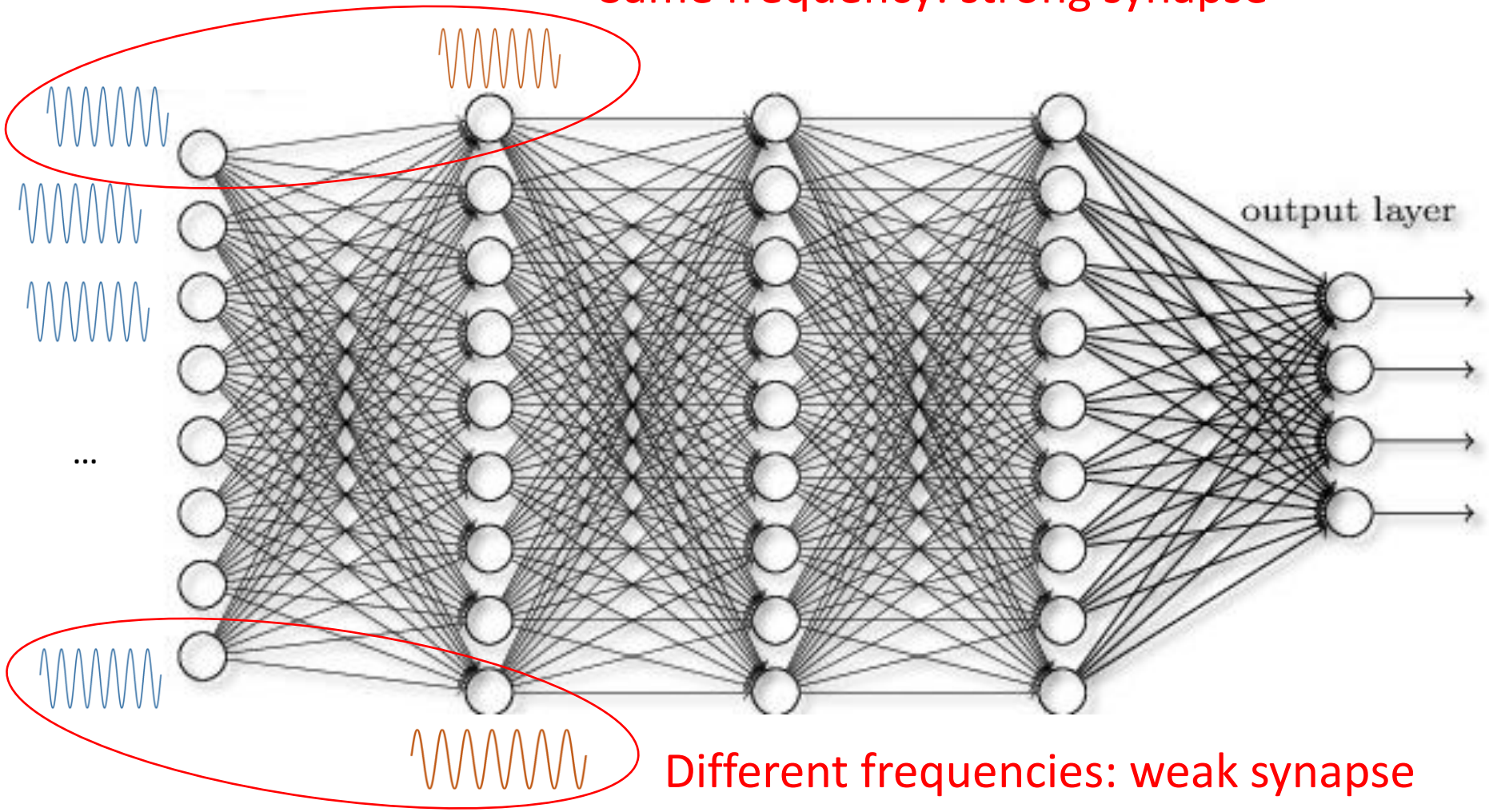


A. Slavin and V. Tiberkevich, IEEE TM 45, 1875 (2009)

W. H. Rippard et al., PRL. 95, 067203 (2005), R. Lebrun, JG et al, PRL 115, 017201 (2015)
A. A. Awad et al, Nature Phys. (2016)

The oscillators ability to mutually interact opens the path to RF on-chip communication between neuron layers

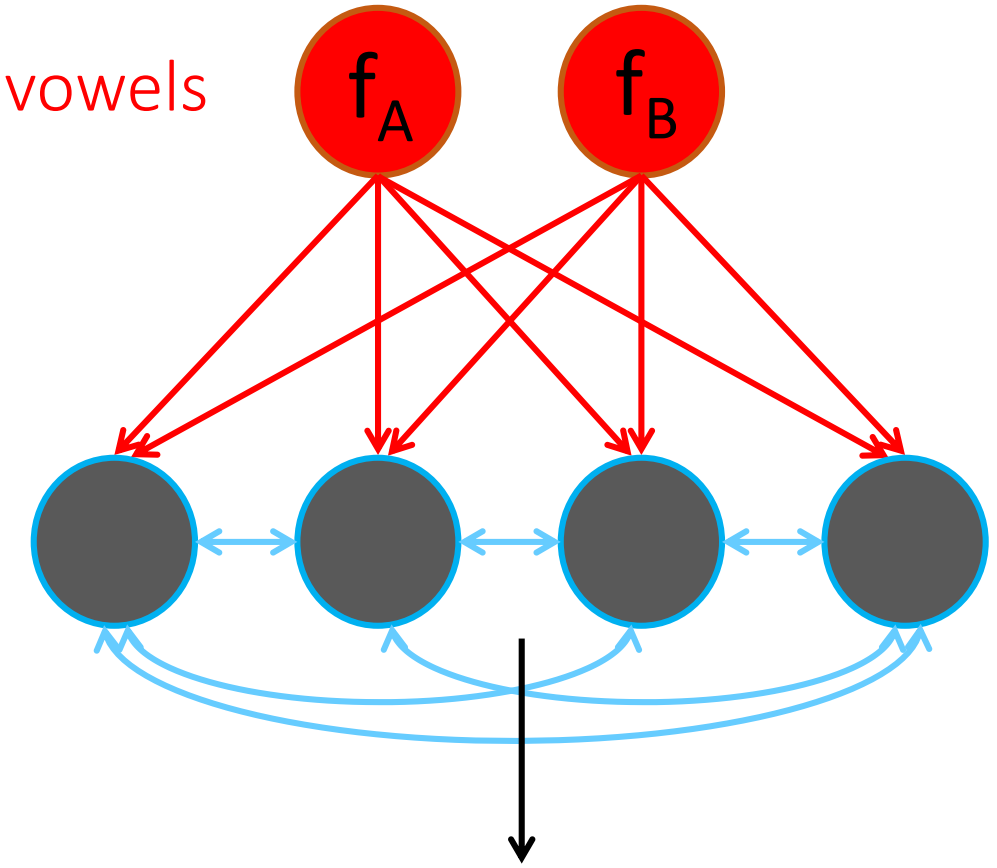
Same frequency: strong synapse



Different frequencies: weak synapse

Vowels classification with spin-torque oscillator neural network

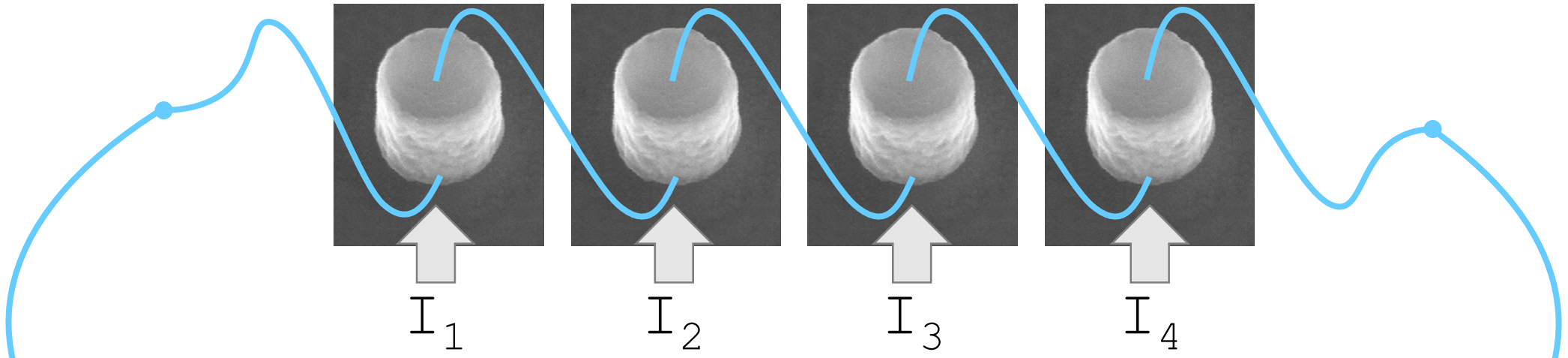
Inputs: vowels



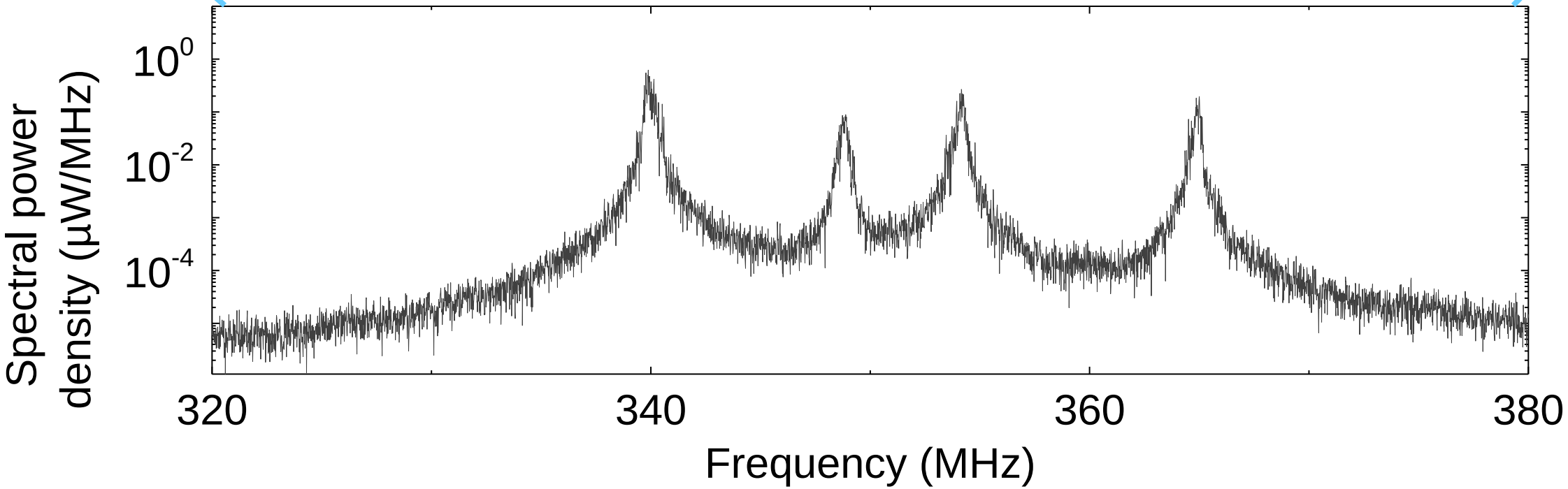
Hillebrands Michingan database

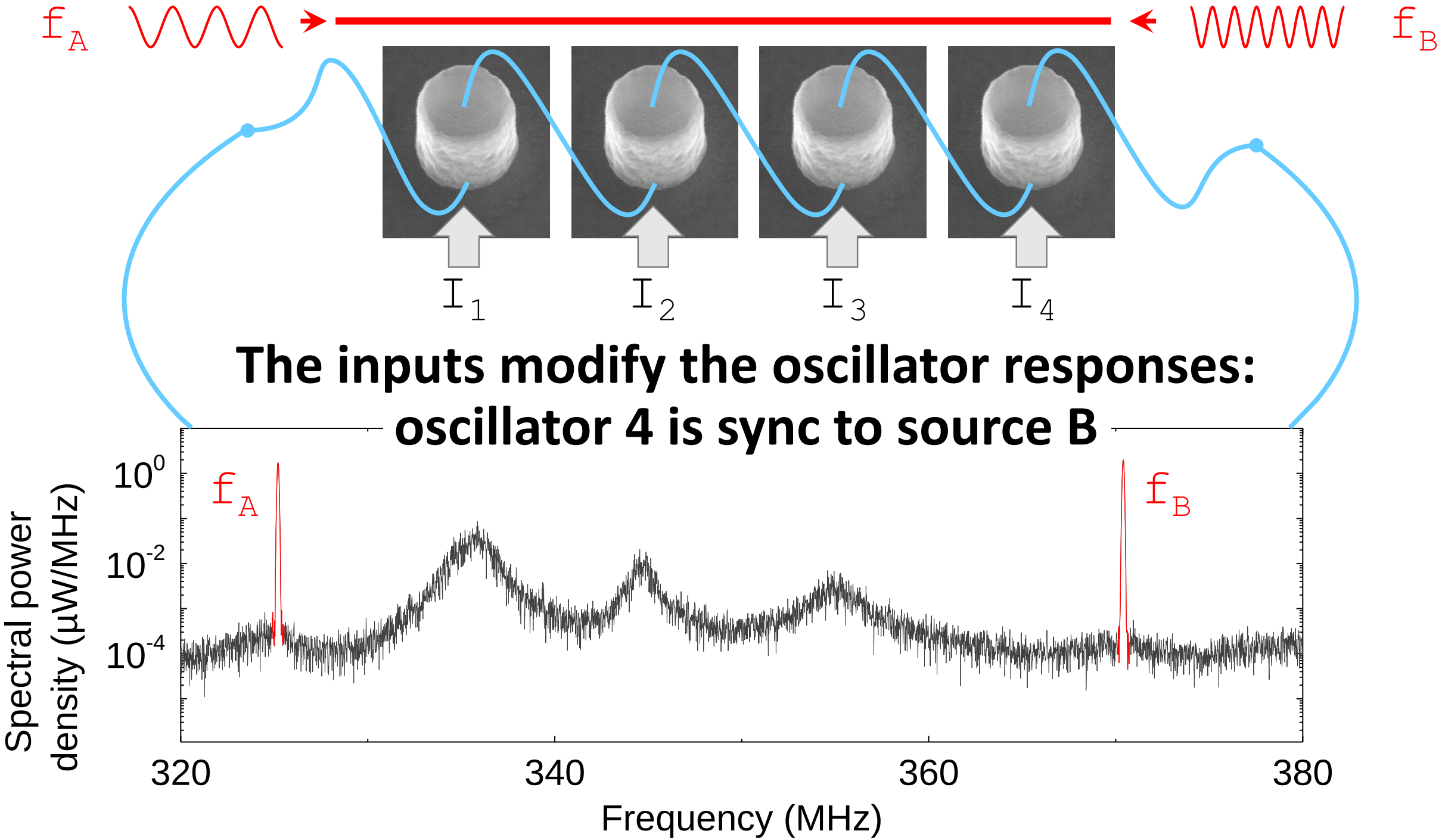
Magnetic Nano-Oscillators

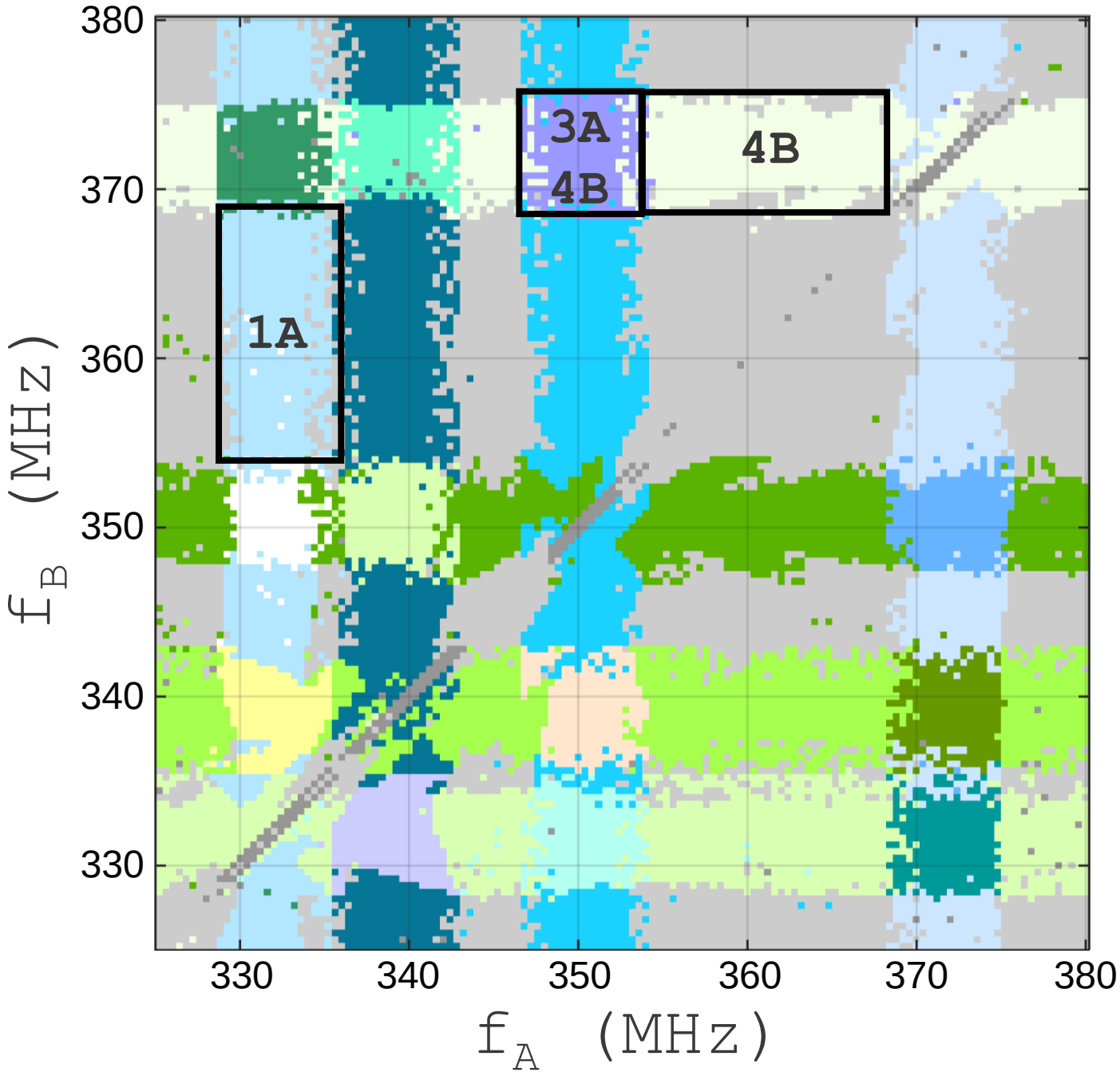
Outputs: synchronized states



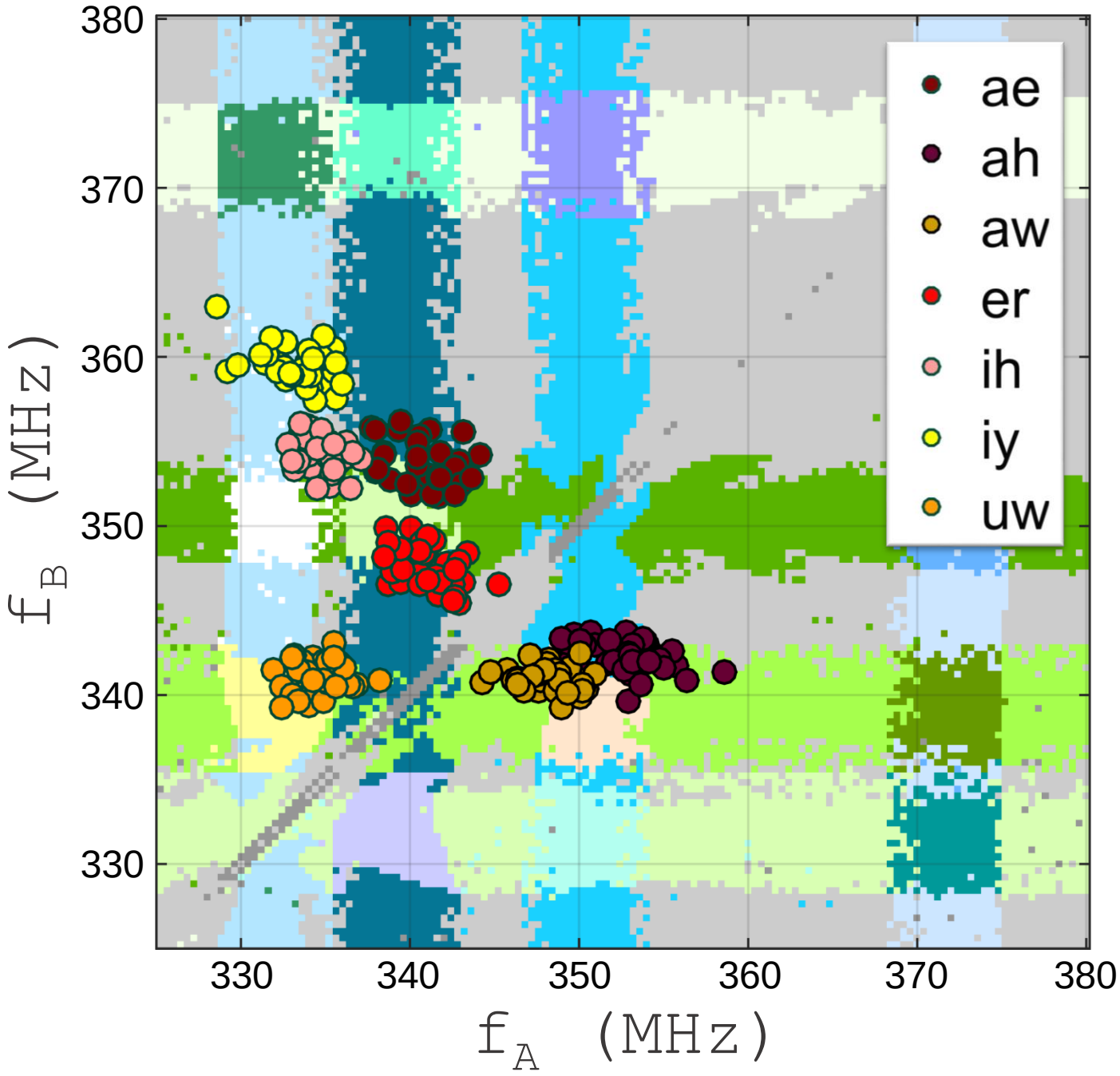
Response of the neural network without inputs




f_A
f_B
I_1
I_2
I_3
I_4
The inputs modify the oscillator responses:
oscillator 4 is sync to source B
Spectral power density (μW/MHz)
10^0
10^{-2}
10^{-4}
f_A
f_B
320
340
360
380
Frequency (MHz)



We summarize all these measurements in a map where the different synchronization states have different colors

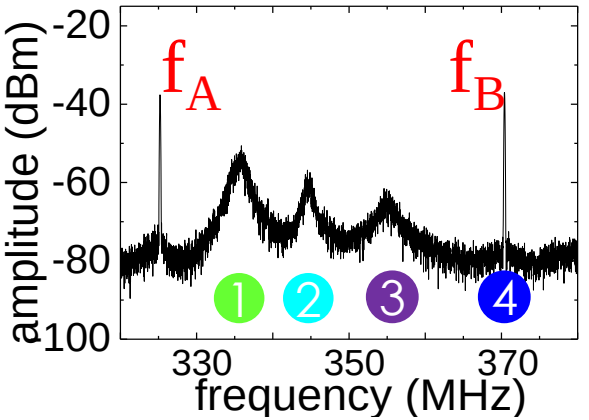


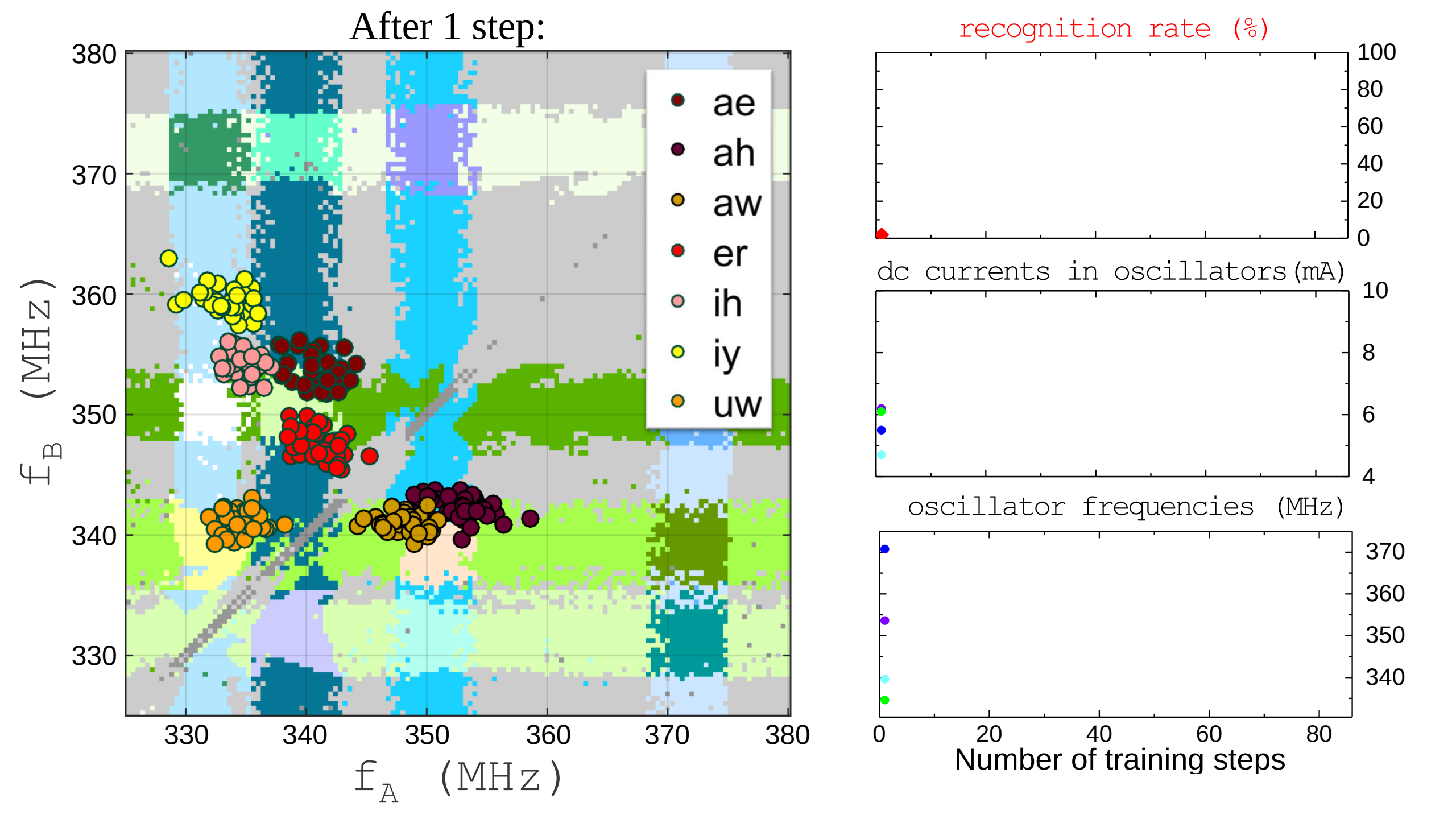
For classification, all the points corresponding to one vowel should fall in a single synchronization region

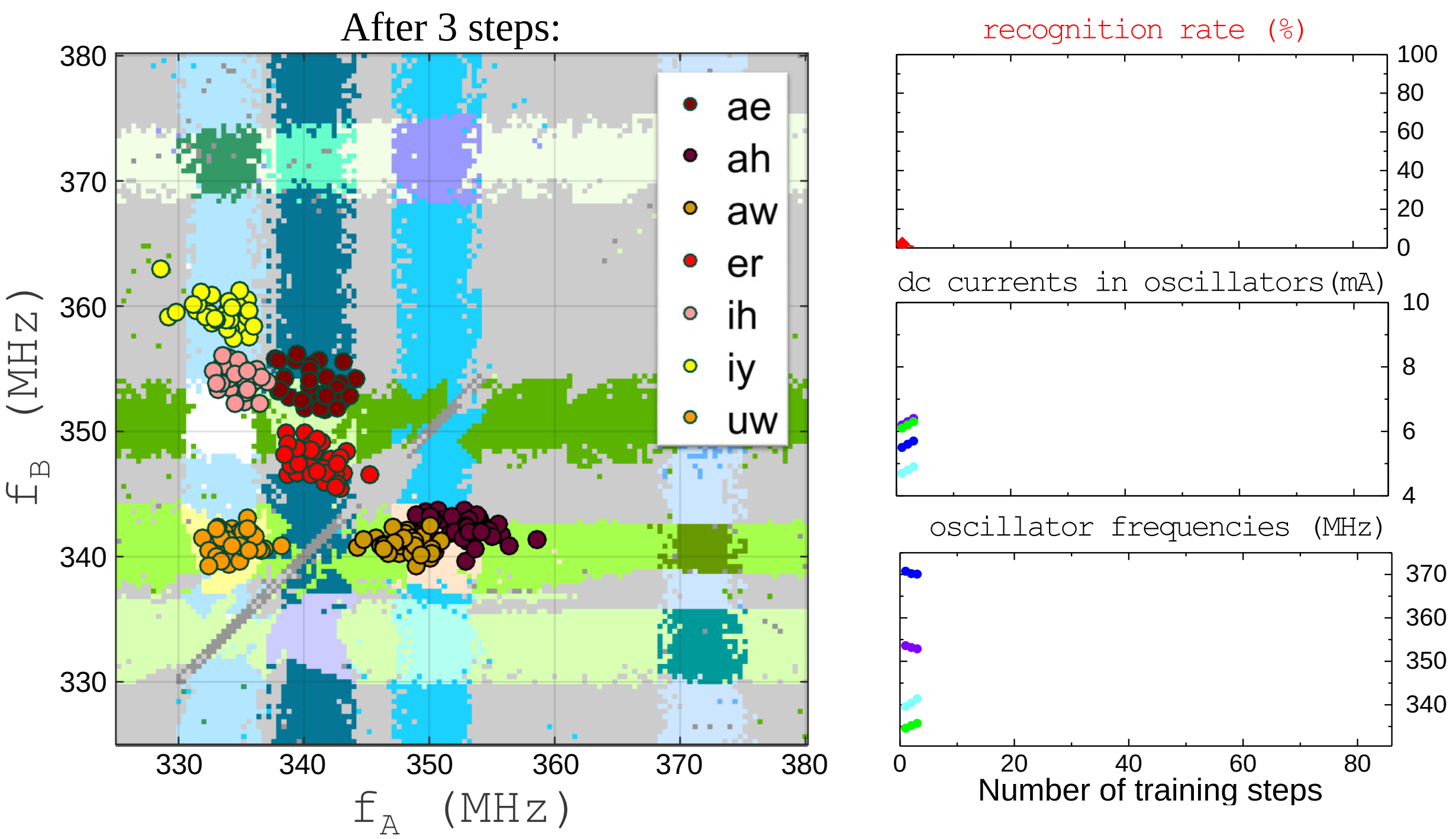
We train the network by tuning the currents through the oscillators according to an online learning rule

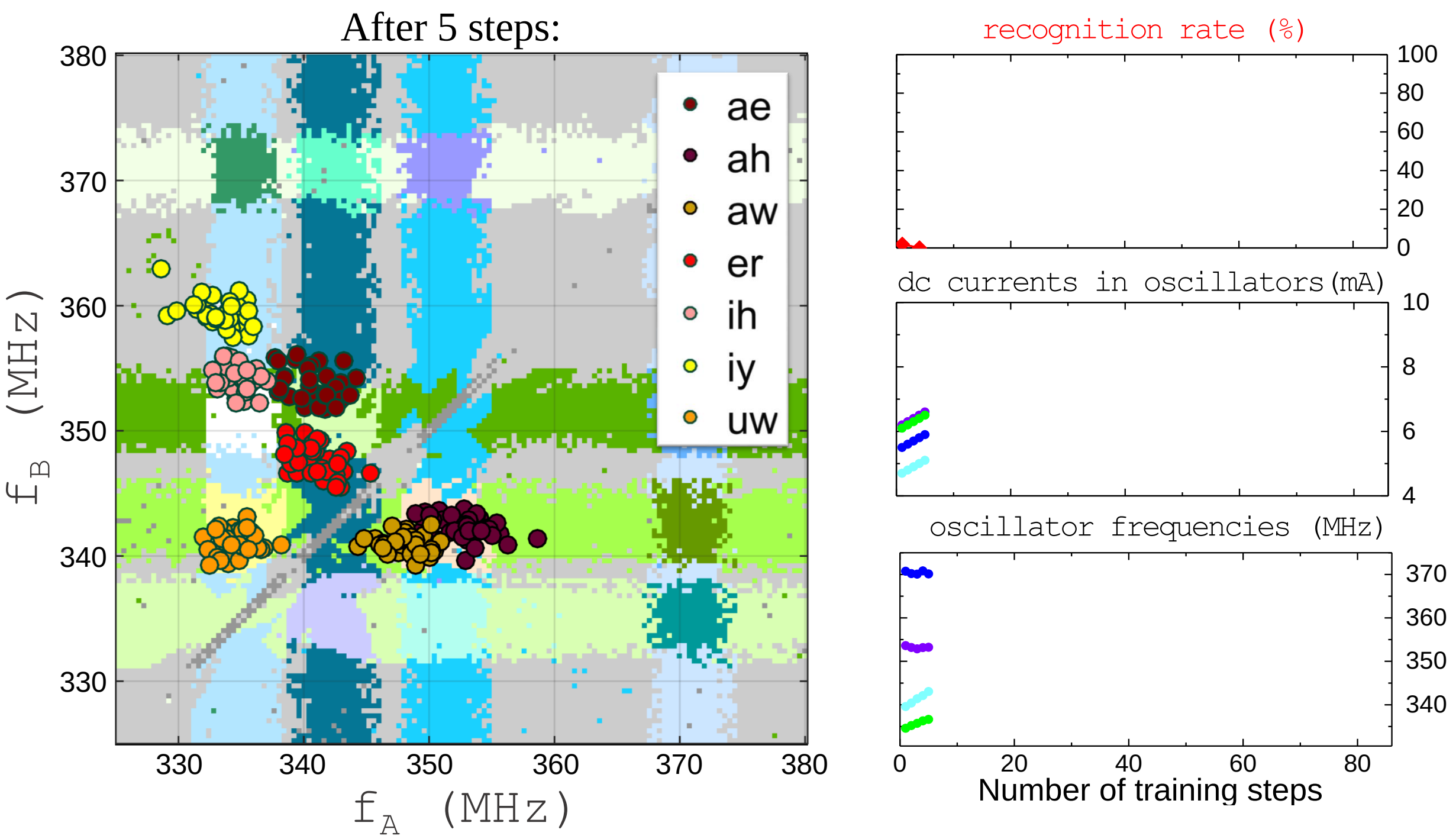
Experimental set-up ⟵ ⟶ Computer (Learning algorithm)

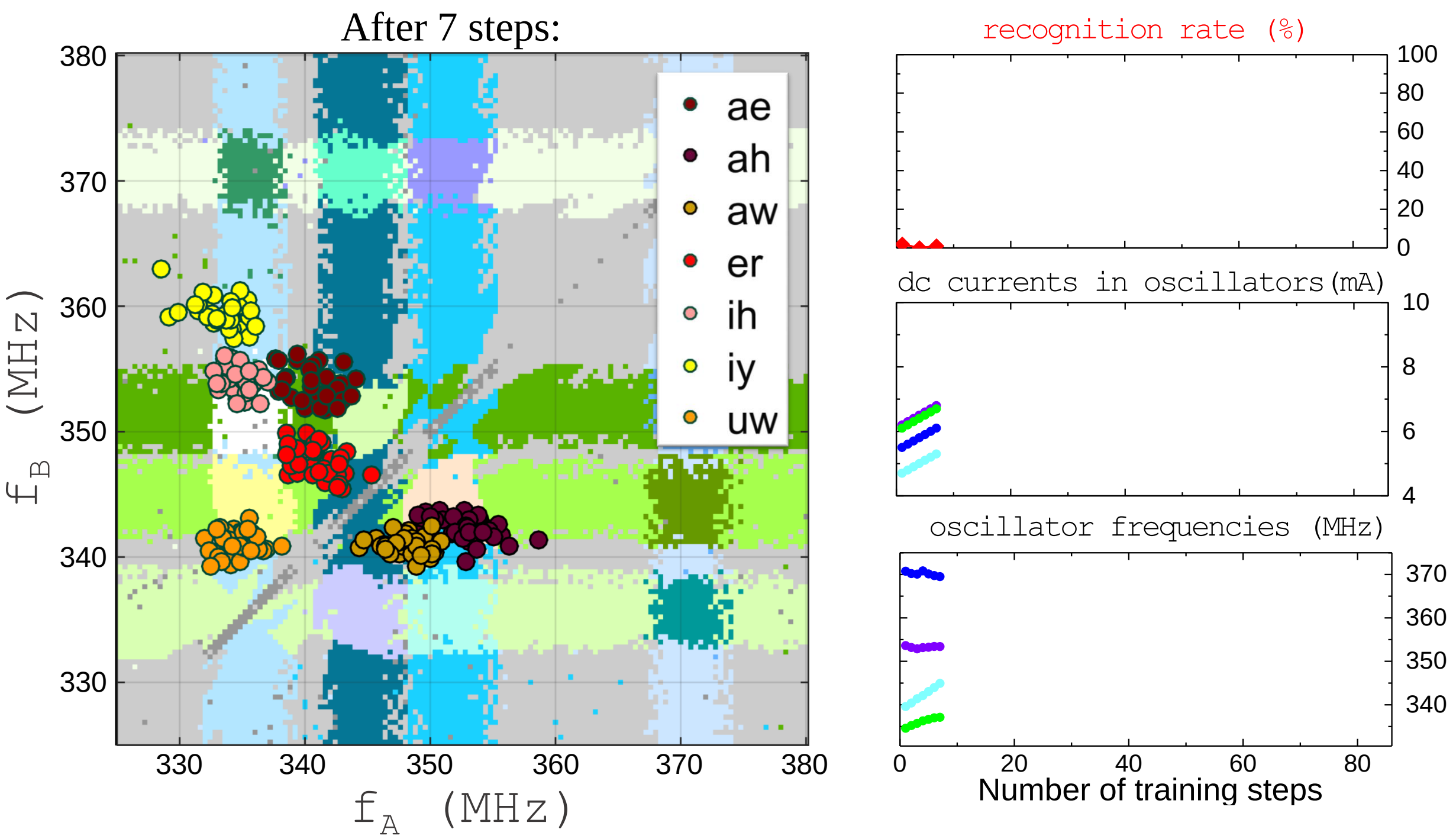
ex: « iy » 4B

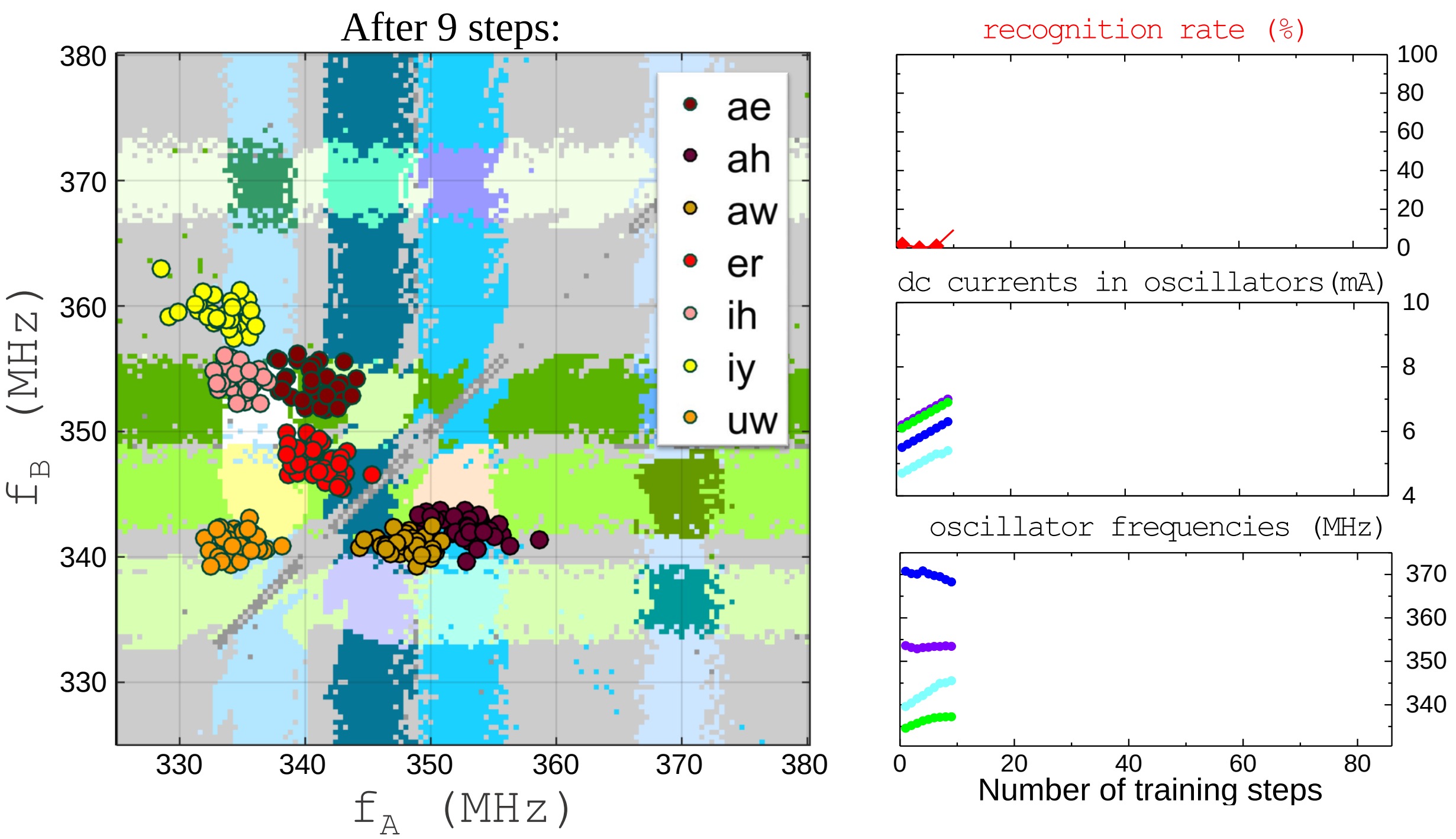


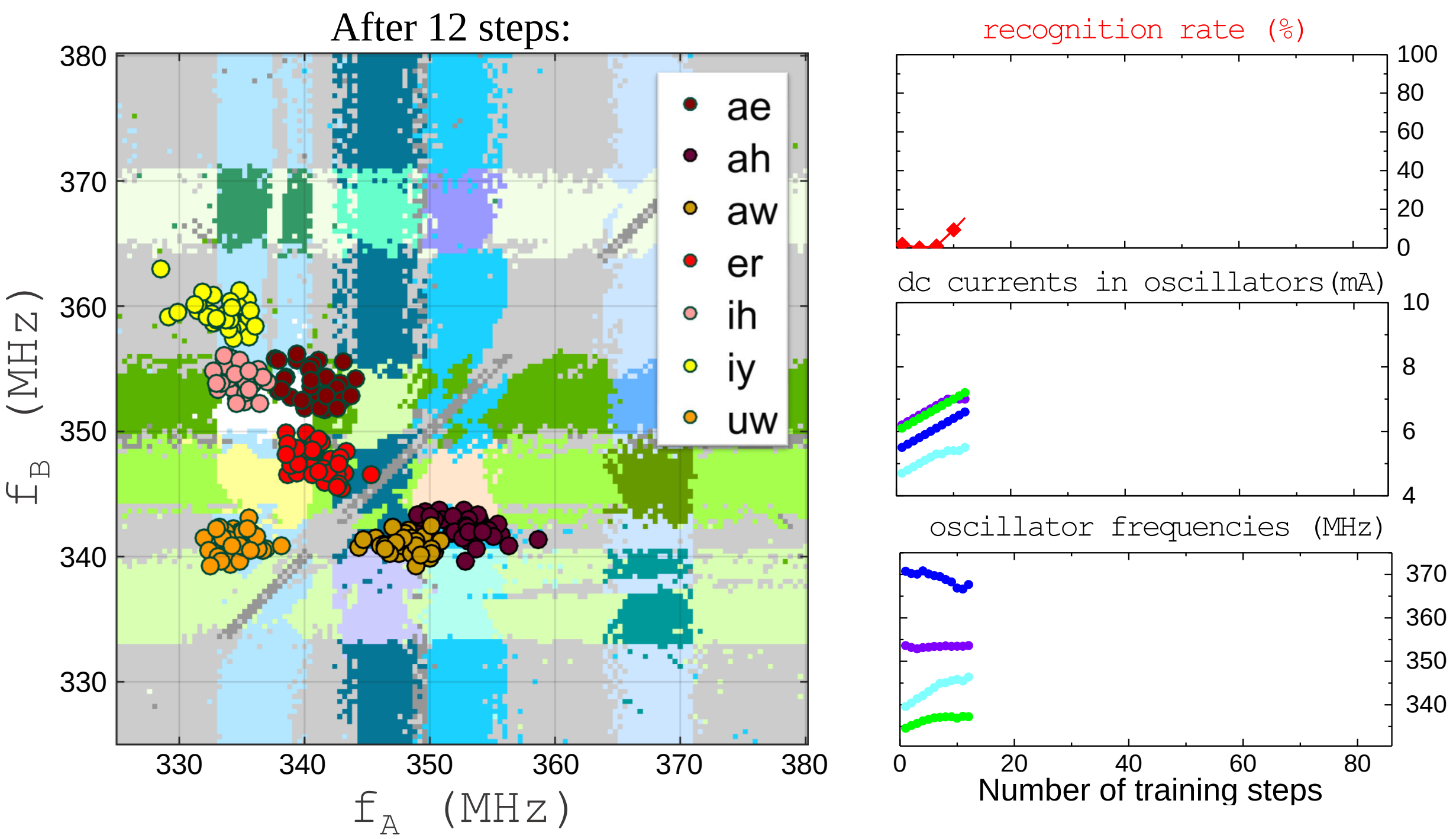

After 1 step:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators(mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

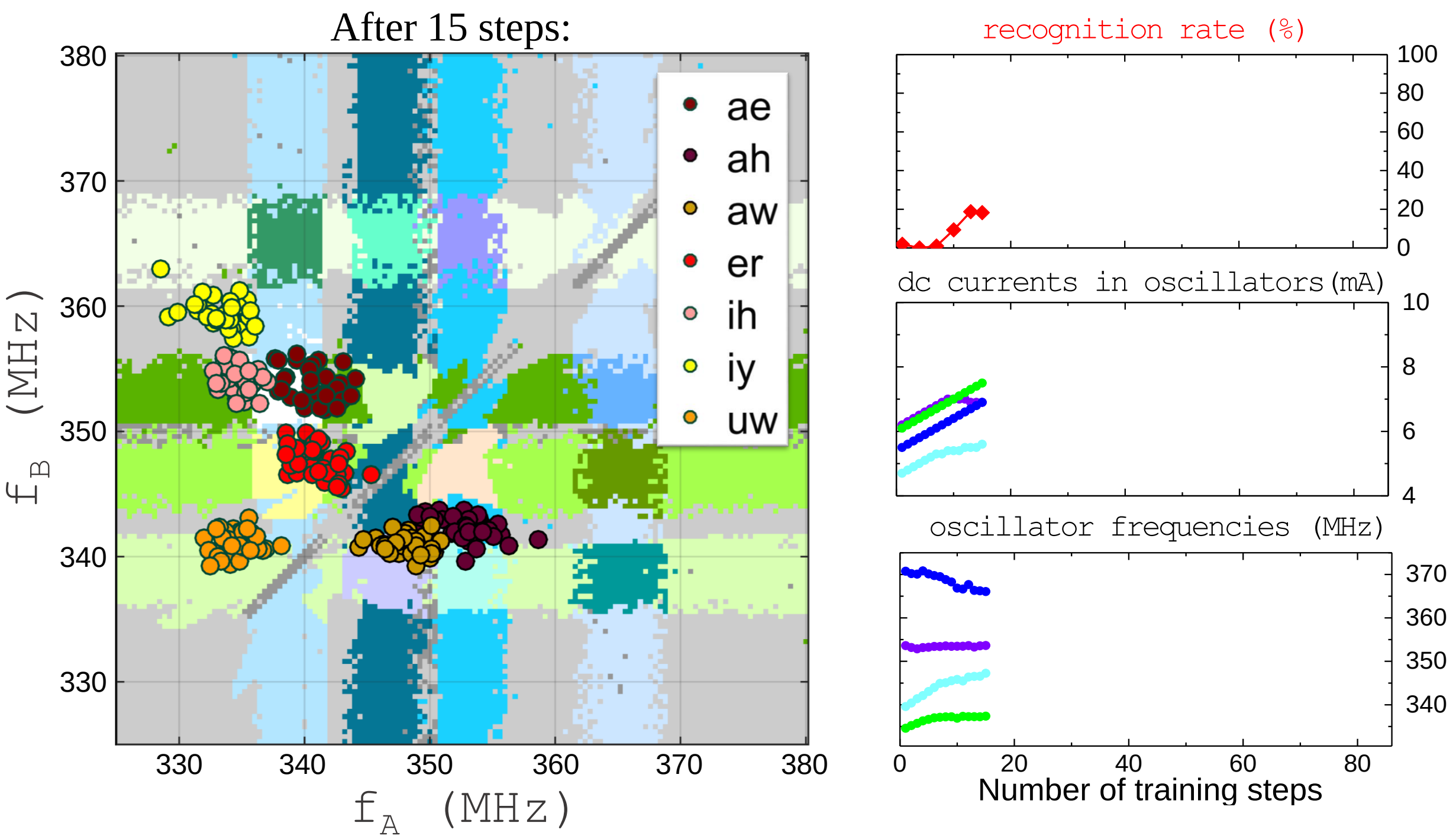

After 3 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

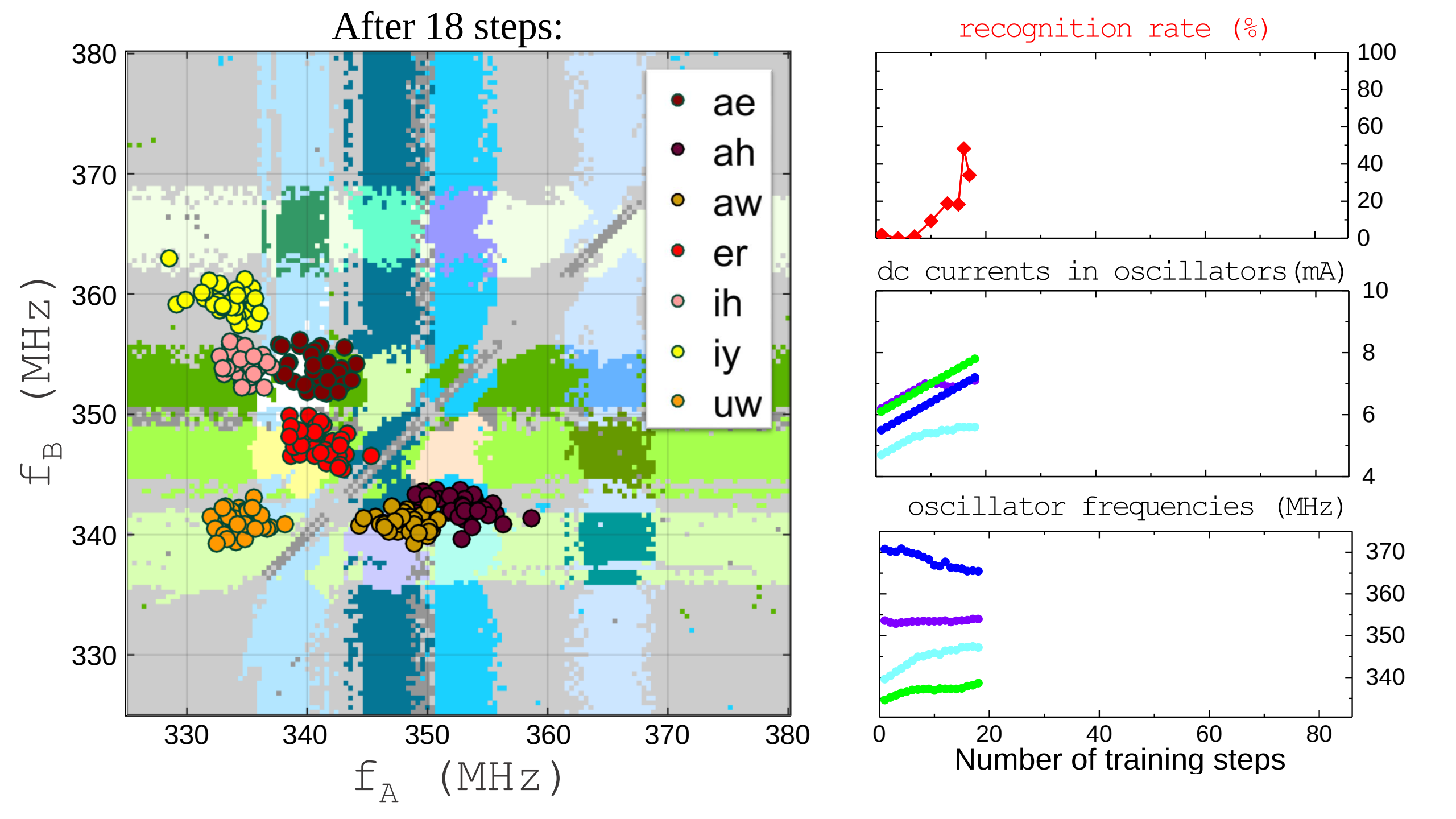

After 5 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

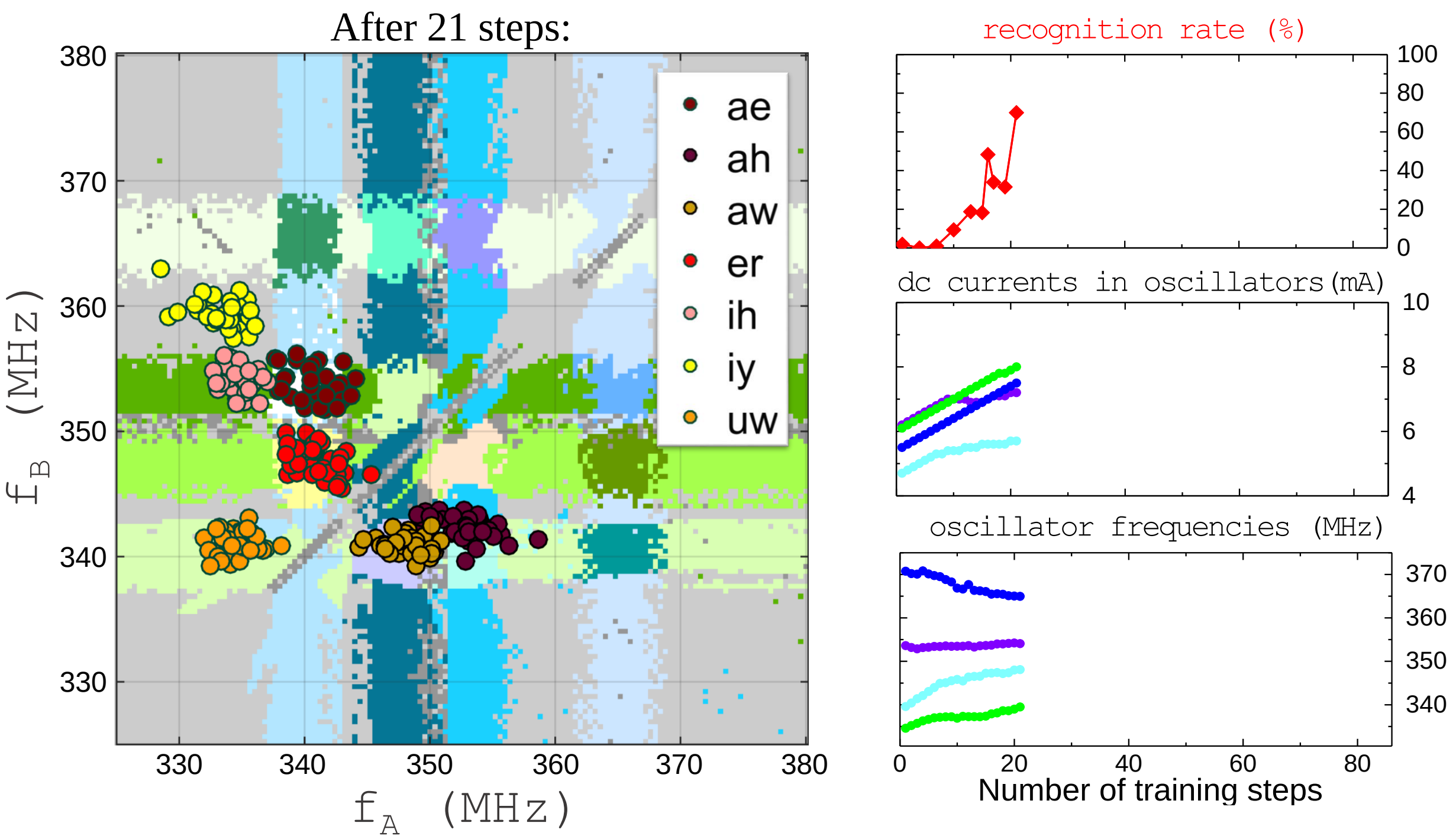

After 7 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

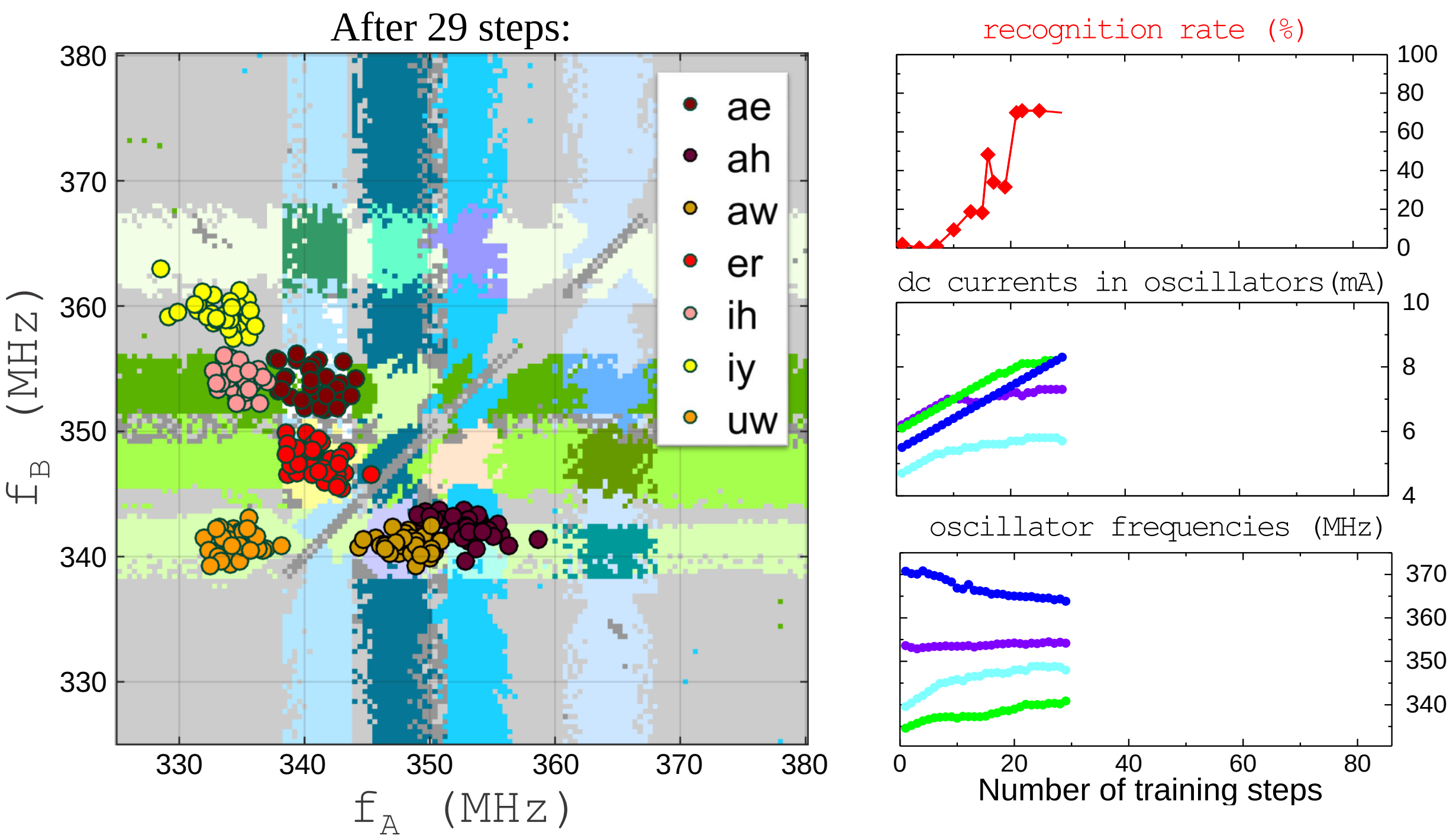

After 9 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators(mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

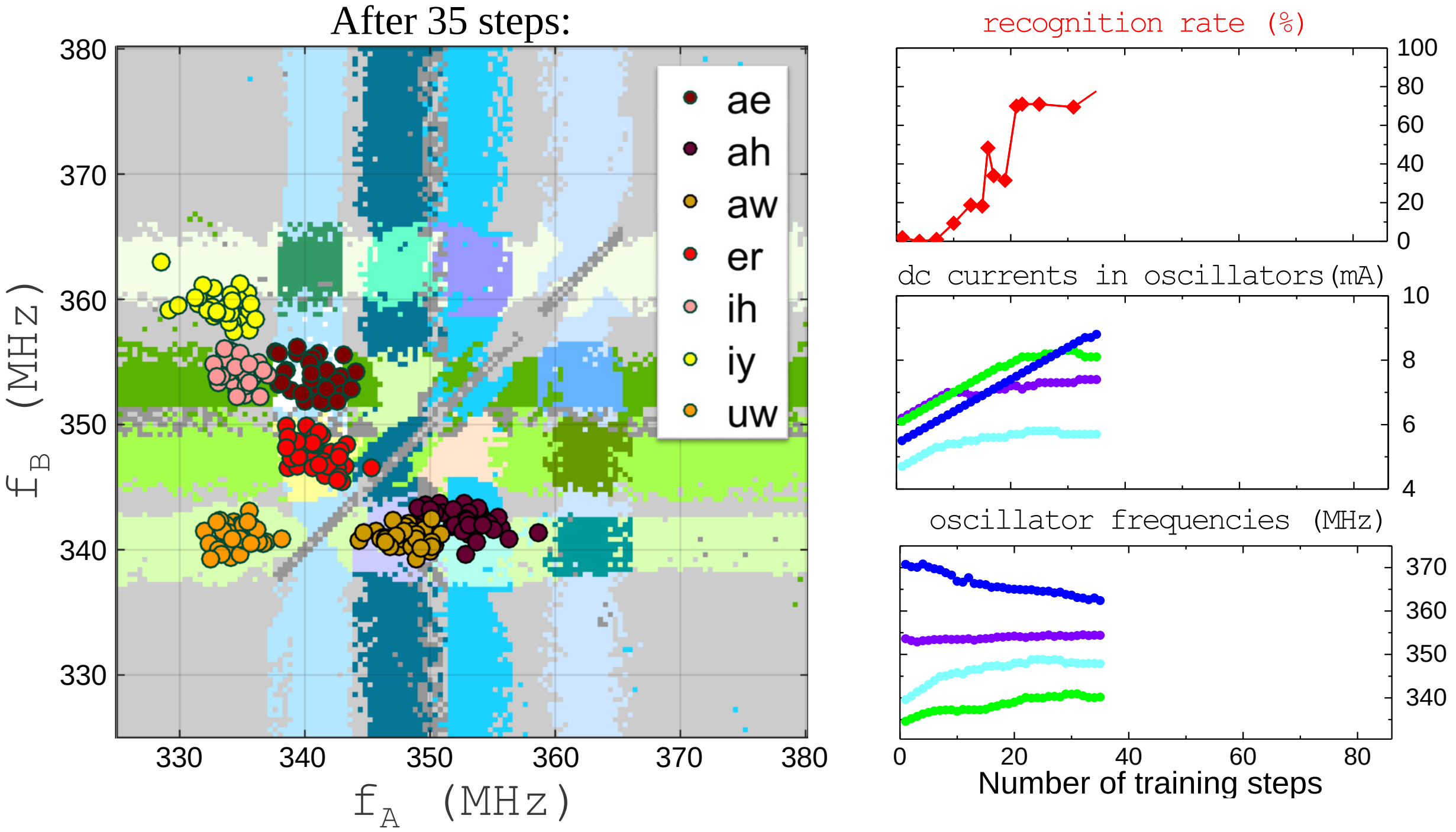

After 12 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

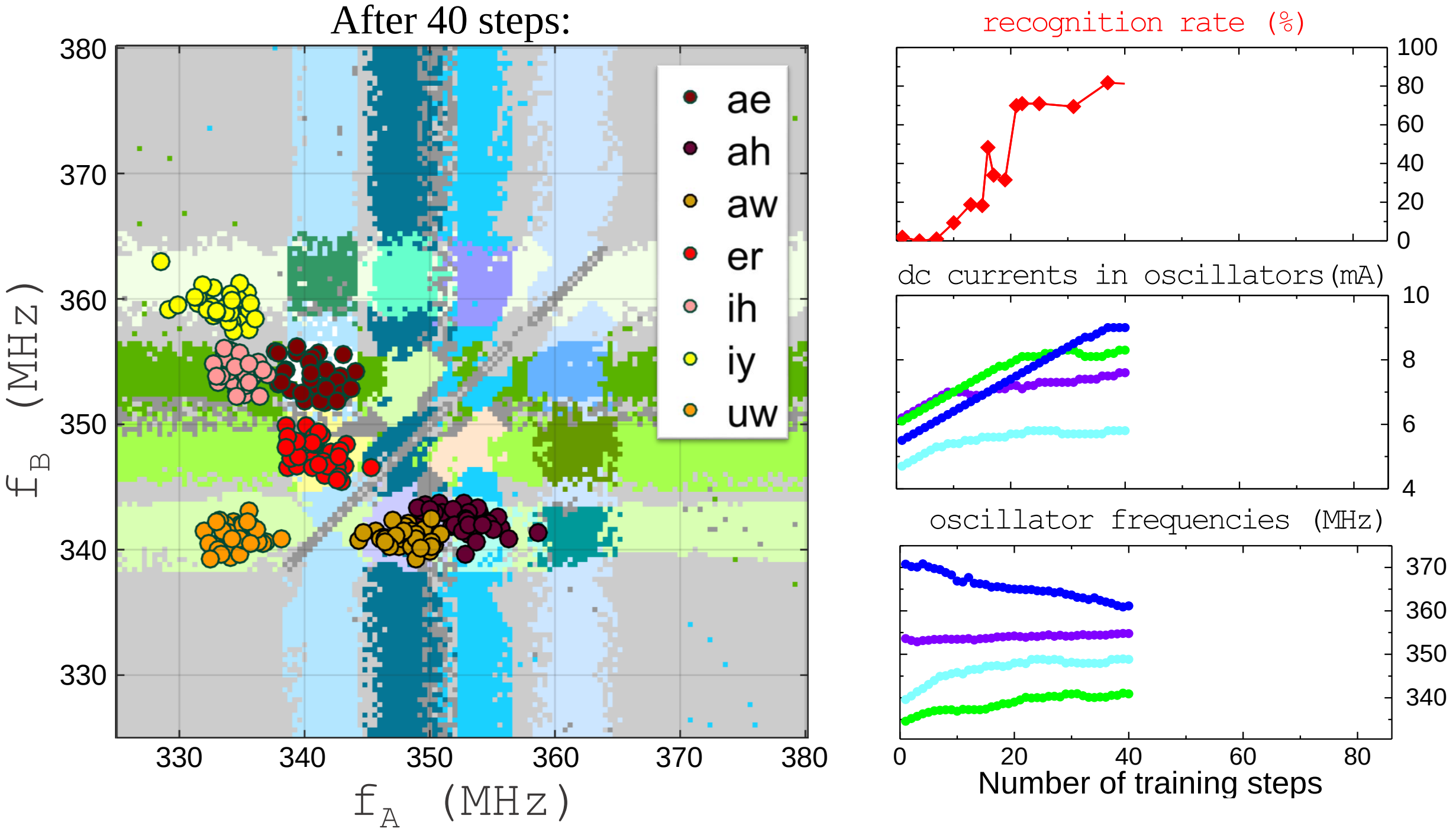

After 15 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

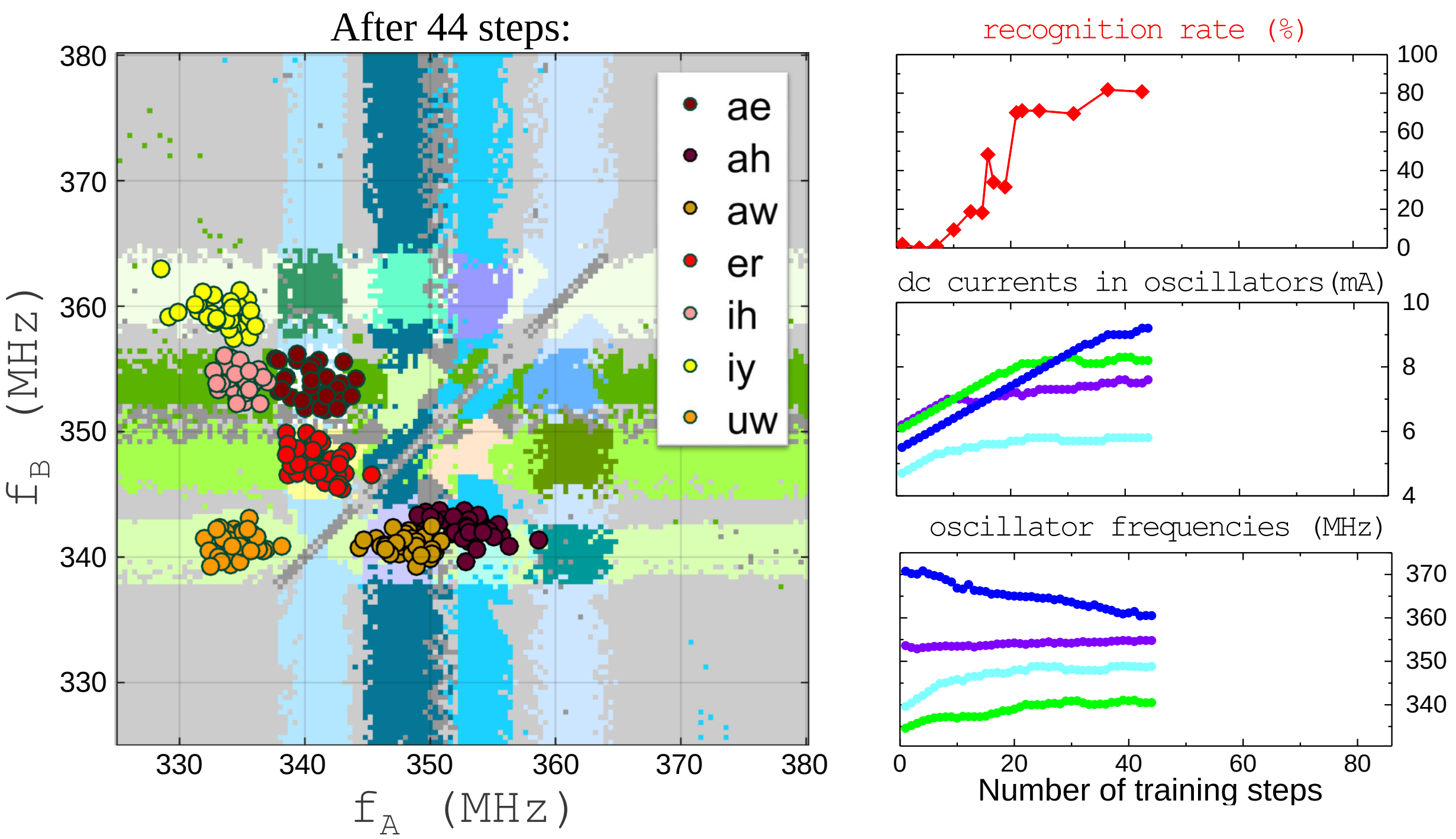

After 18 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

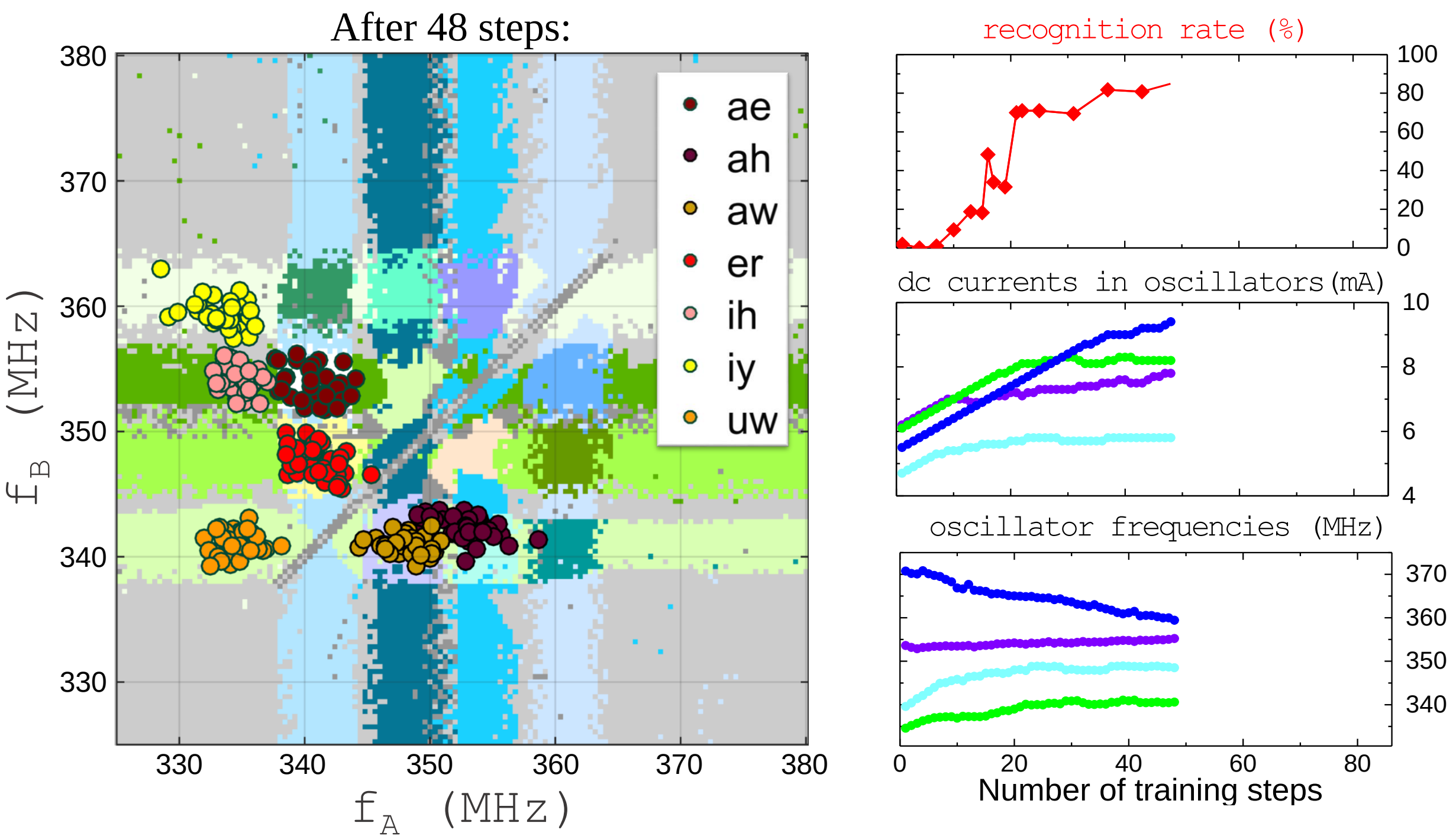

After 21 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

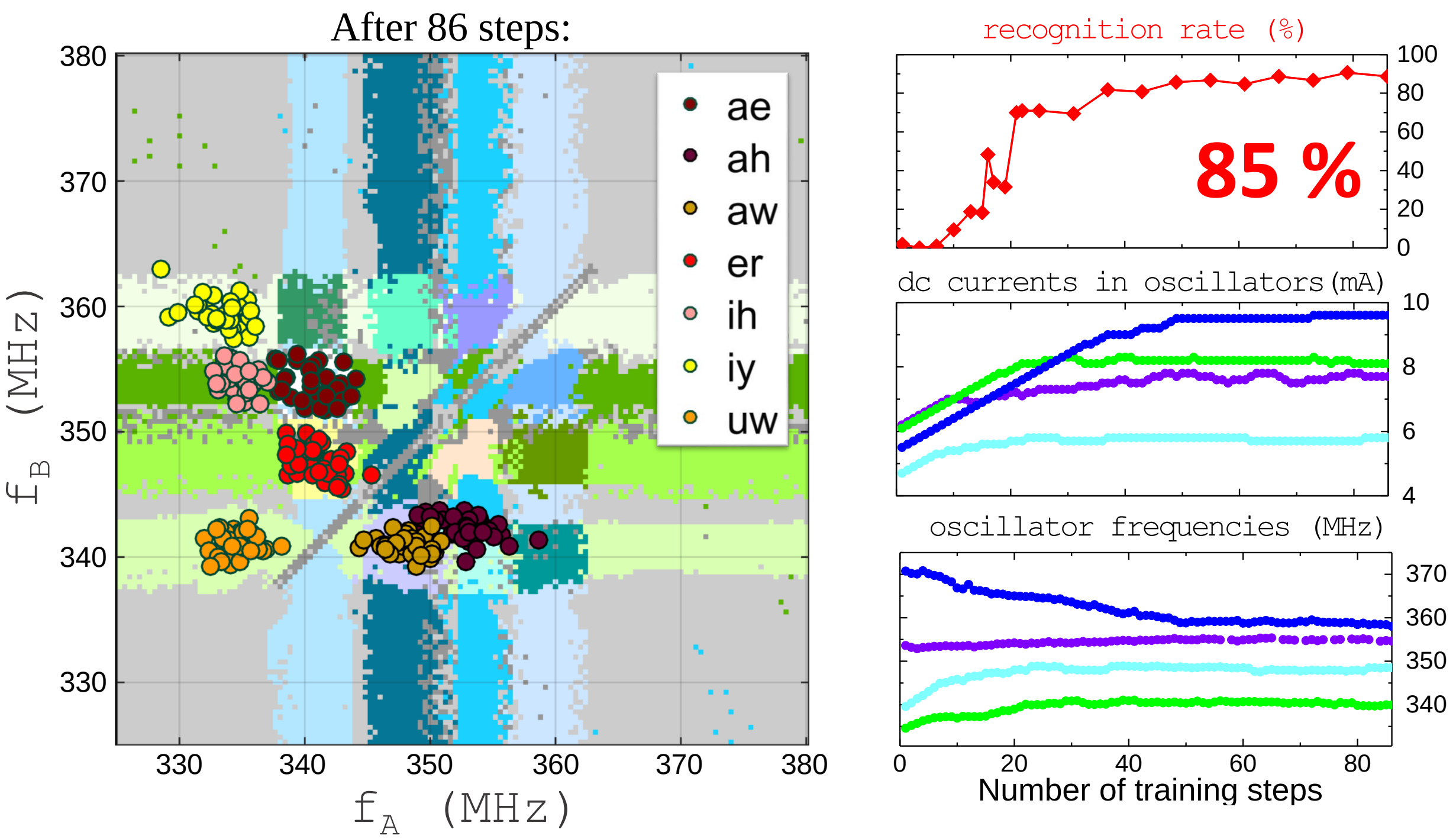

After 29 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps


After 35 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

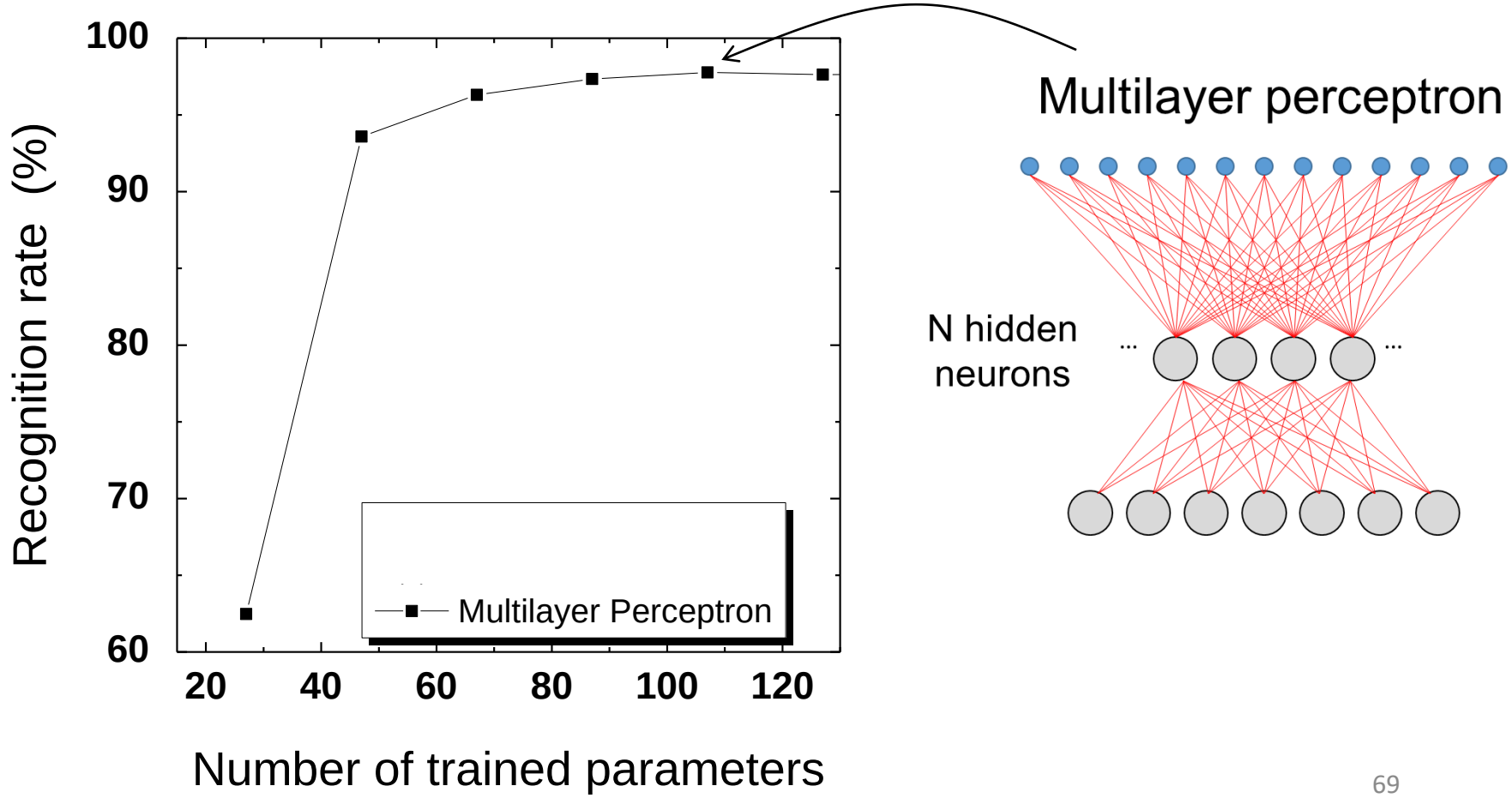

After 40 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps


After 44 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps


After 48 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps


After 86 steps:
ae
ah
aw
er
ih
iy
uw
380
370
360
350
340
330
330
340
350
360
370
380
f_B (MHz)
f_A (MHz)
recognition rate (%)
85 %
100
80
60
40
20
0
dc currents in oscillators (mA)
10
8
6
4
oscillator frequencies (MHz)
370
360
350
340
0
20
40
60
80
Number of training steps

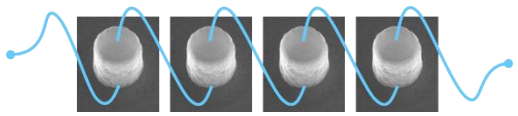
For comparison, we trained a multilayer perceptron on the same database

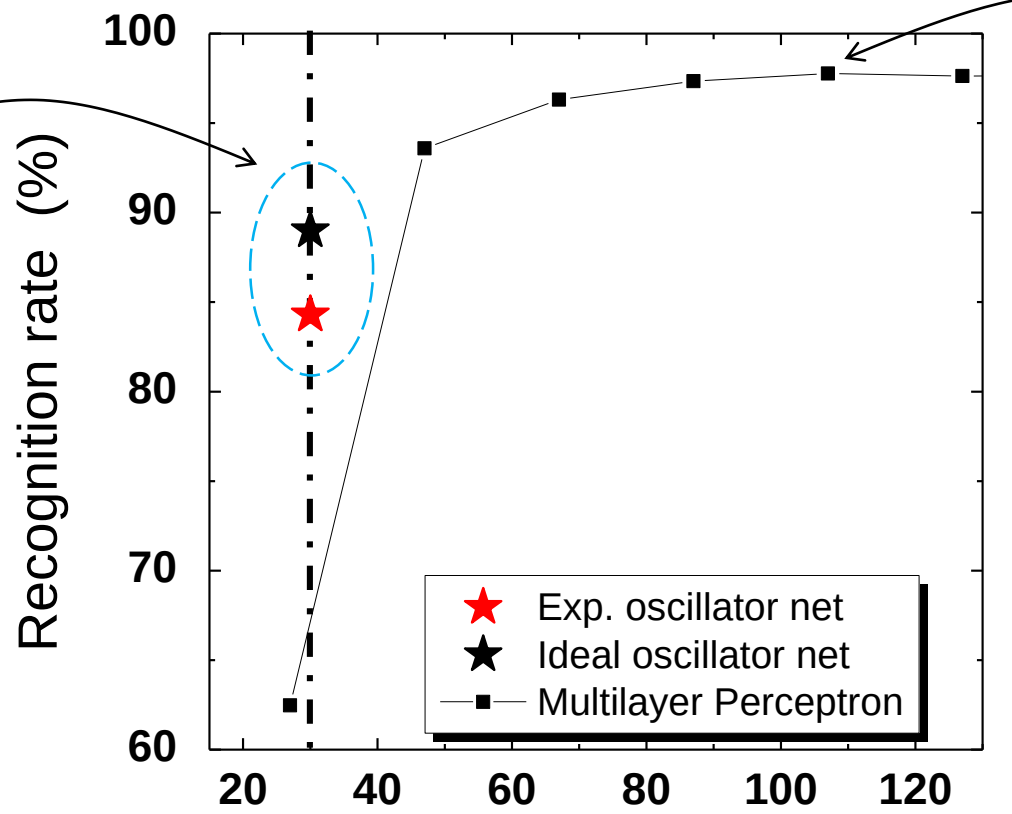


M. Romera, P. Talatchian et al, arXiv:1711.02704

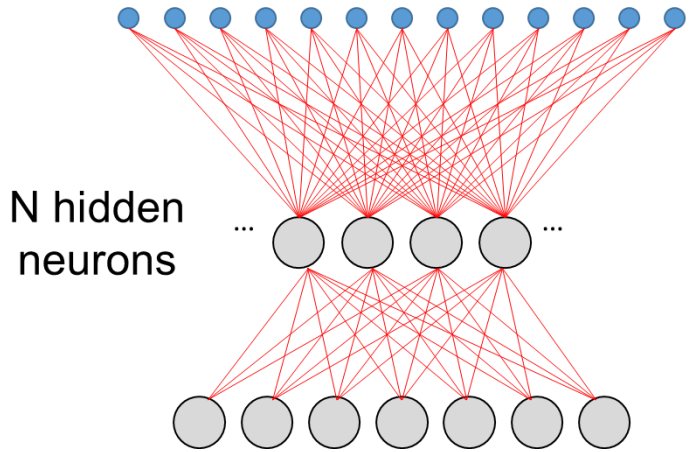
Coupled oscillators achieve similar recognition rates slightly higher than conventional neural networks with the same number of parameters

4 oscillators



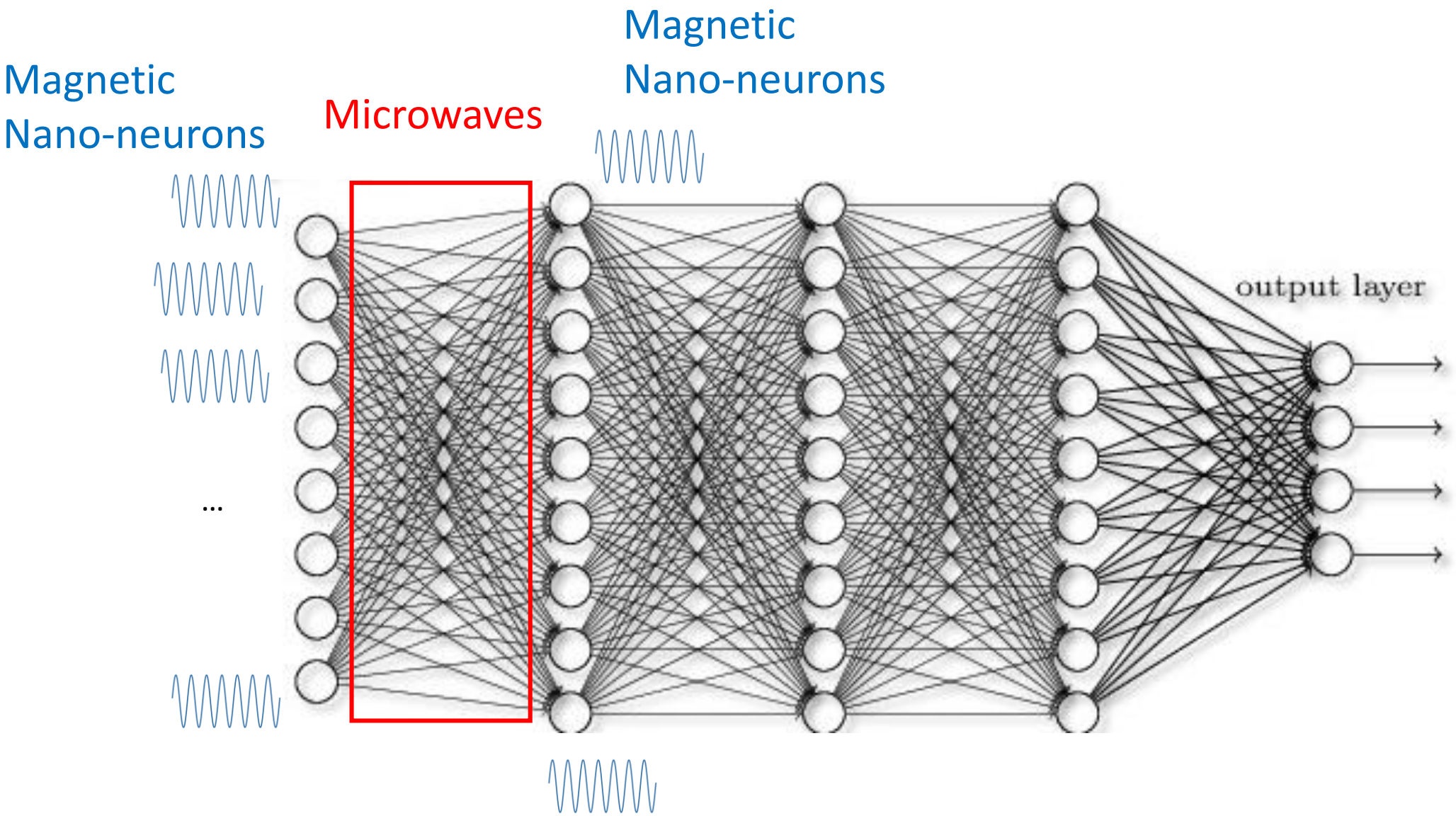


Multilayer perceptron



M. Romera, P. Talatchian et al, arXiv:1711.02704

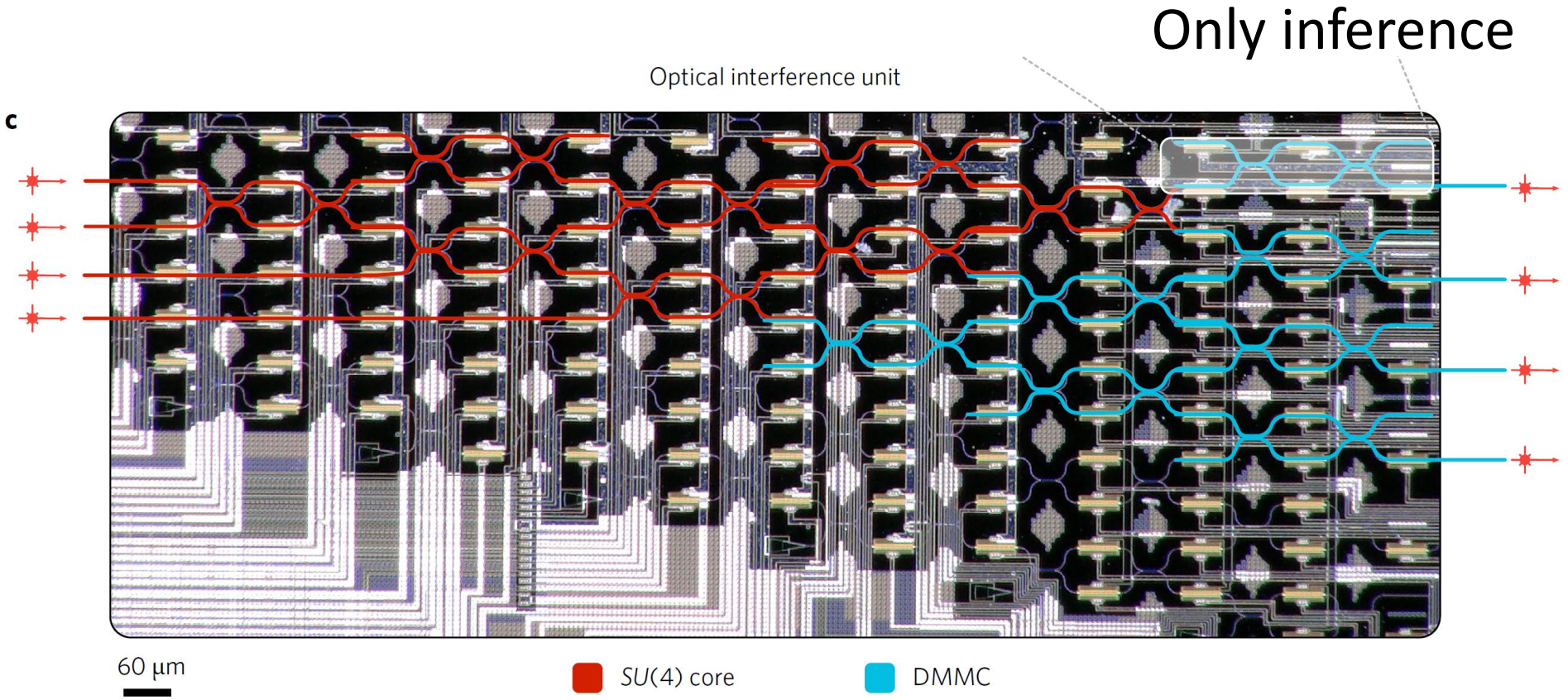
Perspectives: deep learning with spintronic nanodevices ?



Conclusion

- Limitations of CMOS chips
- Current efforts to implement low energy deep learning with emerging nanodevices
- Futuristic ideas

Deep learning in optics

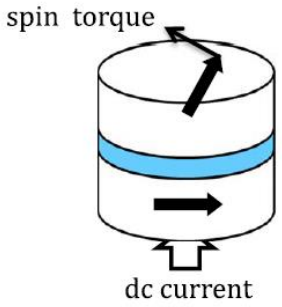


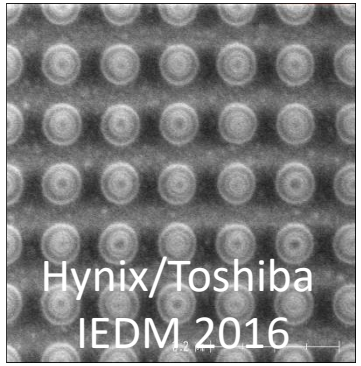
Accuracy for recognizing 4 vowels: 77%

Shen et al, *Nature Photonics*, **11**, 441–446 (2017)

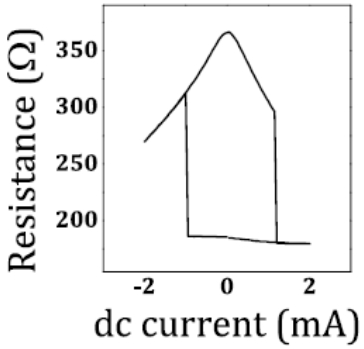
Spintronics is a toolbox for neuromorphic computing

J. Grollier, D. Querlioz and M. D. Stiles, PIEEE 104, 2024 (2016)

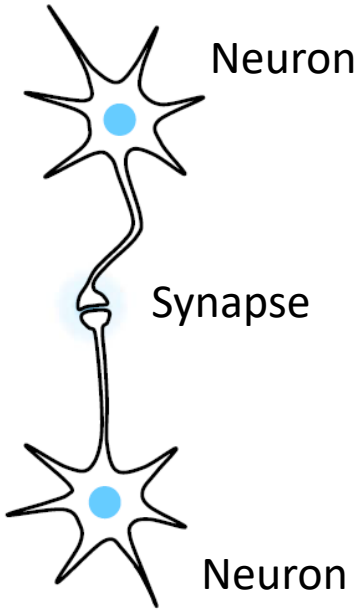




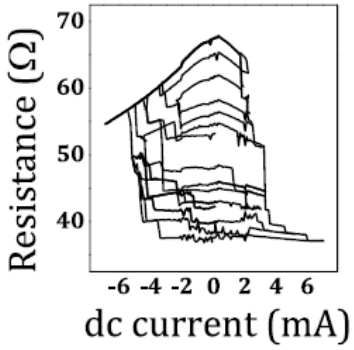
Binary Memory



Ma, Endoh *et al* 2016 *Jpn. J. Appl. Phys.* **55** 04EF15

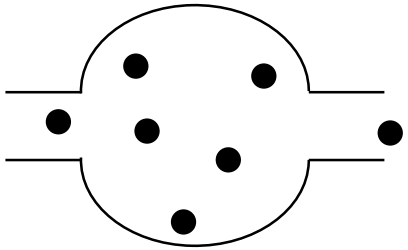


Memristor



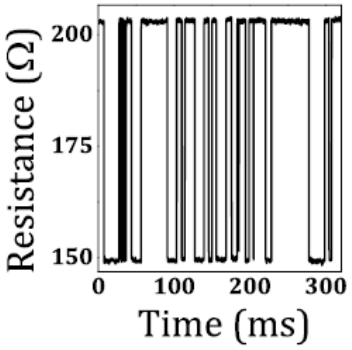
S. Lequeux et al, *Sci. Rep*. (2016)

Skyrmions



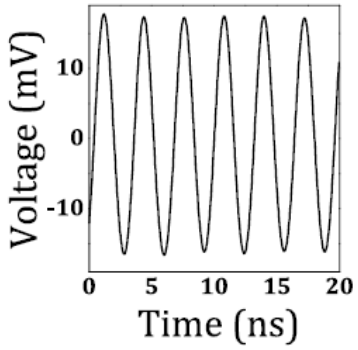
D. Pinna et al, *PRAppl (2018)*
D. Prychynenko et al, PRAppl (2018)

Stochasticity



A. Mizrahi et al, *Nature Com (2018)*

Non-linear dynamics



J. Torrejon, M. Riou et al, *Nature* (2017)