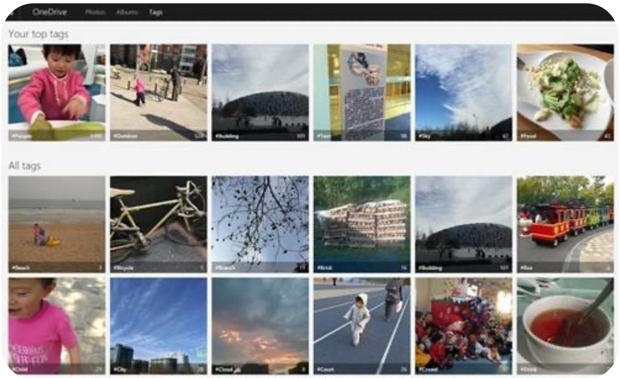
Lifelong Learning

Inspired by the Brain

Hava Siegelmann
Umass Amherst

AI: Super Human? Robust?

Beyond human capabilities



© Ibm: i2.kknews.cc/SIG=29vnh65/2175/3455714929.jpg

©DeepMind Technologies

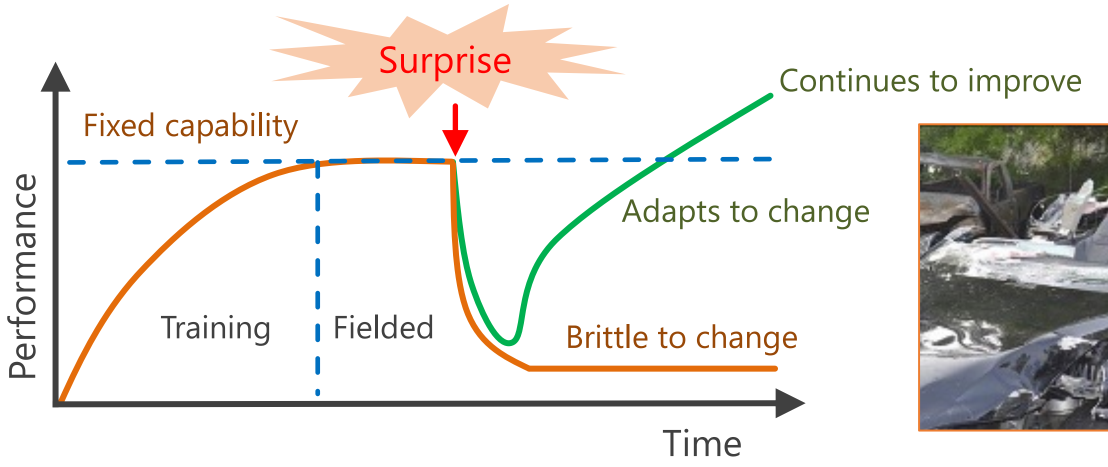
Not fully trustworthy in unstructured environments!

www.bbc.com/news/technology-44300952

Chief source of Untrustworthy AI

Mainly 2 type of AI:
  Programmed (**If** fever>38 **then** covid test)
  Trained (provide: dataset, network, training)
Frozen once ready: AI fully relies on what it was taught advance

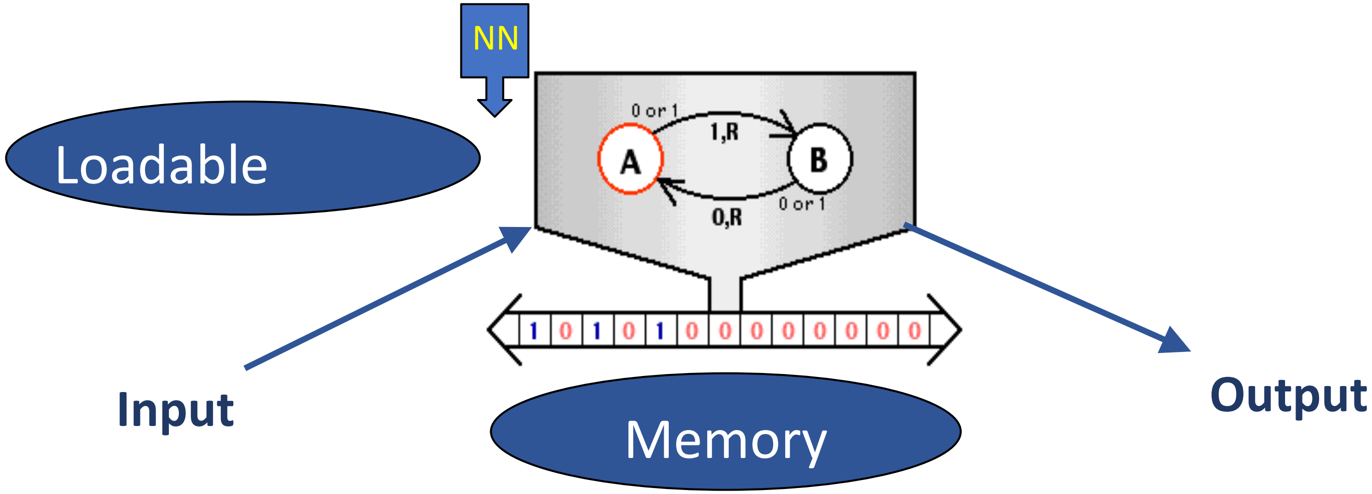
- No way to prepare a training set capturing all possible futures
- Malfunctions in unseen circumstances
- It is easy to attack a non-changing system



Current AI
Lifelong learning AI

Background:

- Universal **Turing Machine** (1936): Does what the **program** fed in to it tells to do
- **Neural Networks** AI (1940's), 200) 0's "Deep learning" : Does what the **database** showed it as knowledge



1. Program and rules (coding expert behavior)
2. Network, dataset (training on Data)

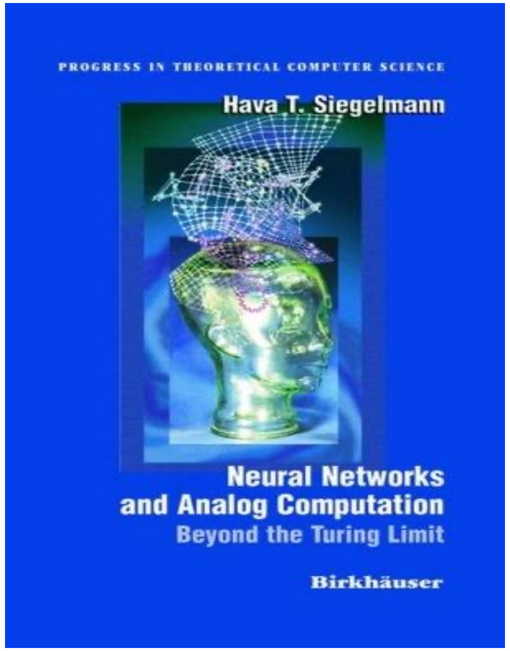
1990's: Super-Turing Continuum Hierarchy: Biological Type Computation

Continuum of computational hierarchy

Turing –computation: Fixed programs ——— **Super Turing, variable programs**

1. Discrete (Q)
2. Deterministic
3. Pre-programmed

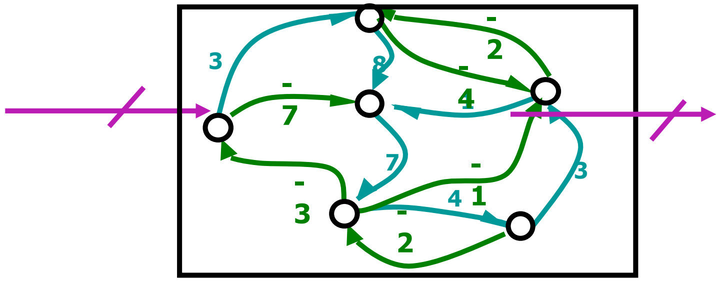
- Center model: Turing machines



Kolmogorov[f(n),g(n)] UTM
calculates α[n-prefix]
from *f(n)* bits in *g(n)* time
P=K[1,p(n)] AnalogP=K[n,n]

1. Physical values (Real)
2. Randomness/asynchronous
3. Lifelong Learning, evolving
4. Series of TM's

- Center model: Recurrent NN



DARPA's Lifelong Learning Machines (L2M)

DARPA's large group effort to advance AI 2017-2022

Create AI systems that can adapt during runtime

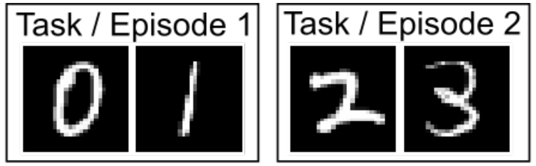
Looking at neuroscience to find how lifelong learning is possible:
We came up with a number of ways to advance AI learning from experience (not from pre-prepared datasets)

"Biological Underpinnings for Lifelong Learning Machines," ***Nature Machine Intelligence***,   03/2022

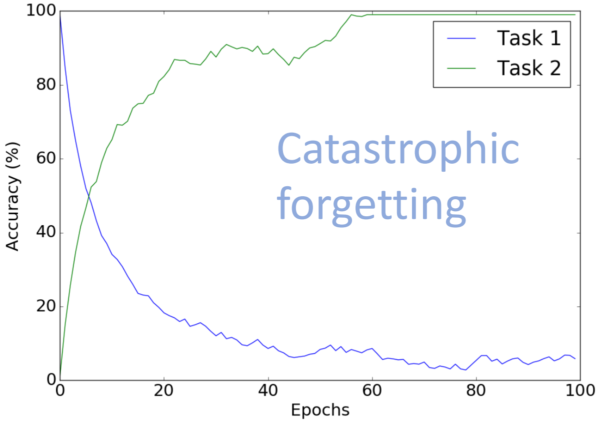
Most successful solutions:

- bio-inspired Replay
- self Supervision
- adaptive modularity

Continual learning



Task-Incremental Learning (Task-IL)
Choice between 2 known digits (e.g., '0



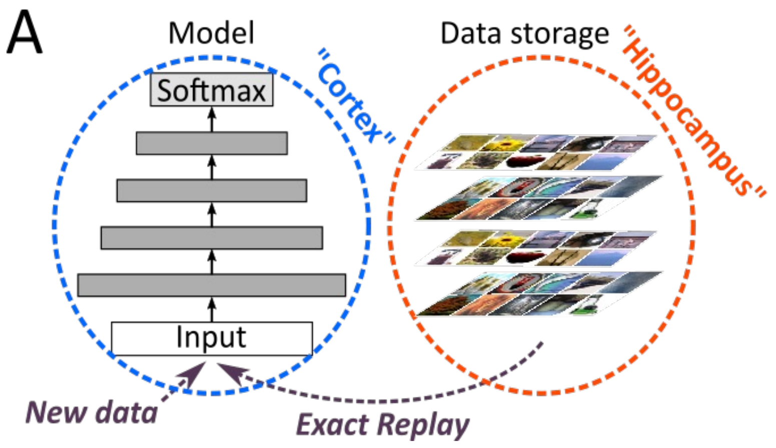
Improvement: Brain Inspired Replay

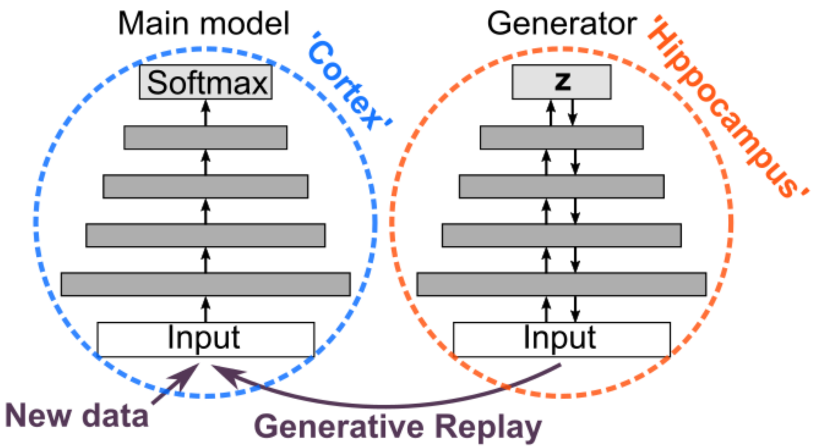
(Nature Communication August 2020: with Gido van de Ven & Andreas Tolias, Baylor)

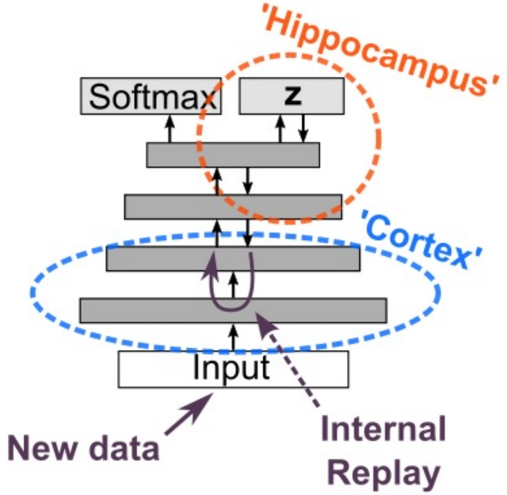
Store and retrain

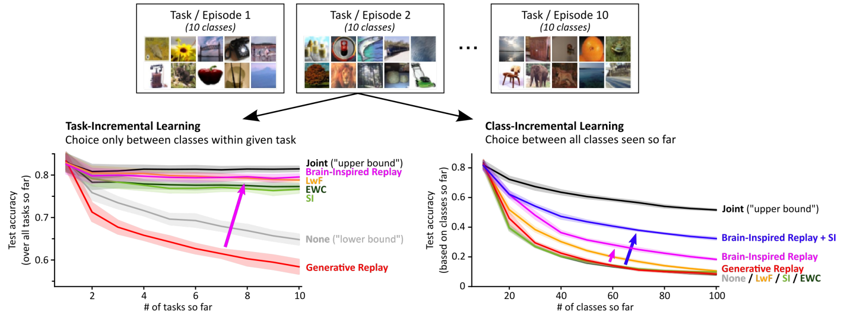
Generate and replay

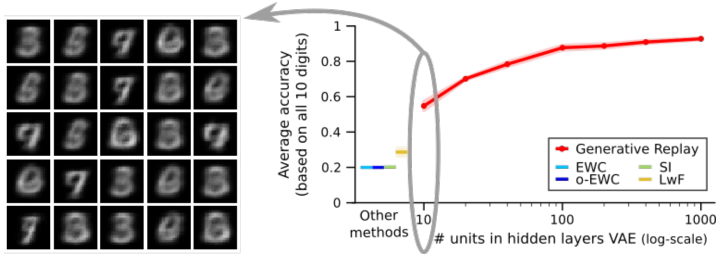
Brain-like
Generate **abstractions** and replay









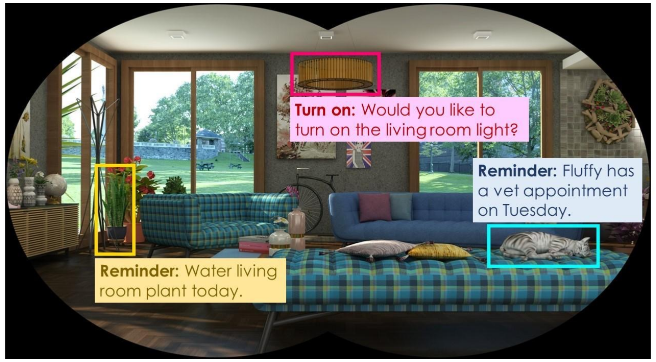


low quality generative model could suffice – better “abstraction

When and What to Replay?

1. When to replay: Pervasive applications - mobile phones, virtual/augmented reality (VR/AR) headsets, household robots: that learn the identity of the individuals, pets, and objects in the house:

Learning online incorporates experience immediately
Offline consolidation while the mobile device charged or its owner is asleep.



Virtual/Augmenter reality: input arrives naturally, not one class after the other (From Chris Kanan)

2. What to replay? *Similarity-weighted interleaved learning (SWIL):*
Only previous memories with **high similarity to the new data**: substantially less data

Rajat Saxena , Justin L. Shobe, and Bruce L. McNaughton “Learning in deep neural networks and brains with similarity-weighted interleaved learning ” **PNAS 06/2022**

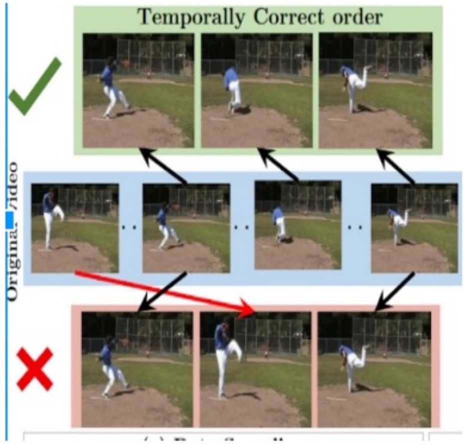
Internal Explorations: Self Directed Learning

Agent uses downtimes to challenge itself with tasks: learn even in the absence of explicit labels **(Toyota, Umass, NYU/Facebook)**

Fill in the blank

Colorization

Temporal order



Reference Image

Image matching

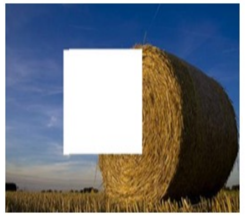
Intelligent search – apply self-learned (visual/functional) associations

“kettle”

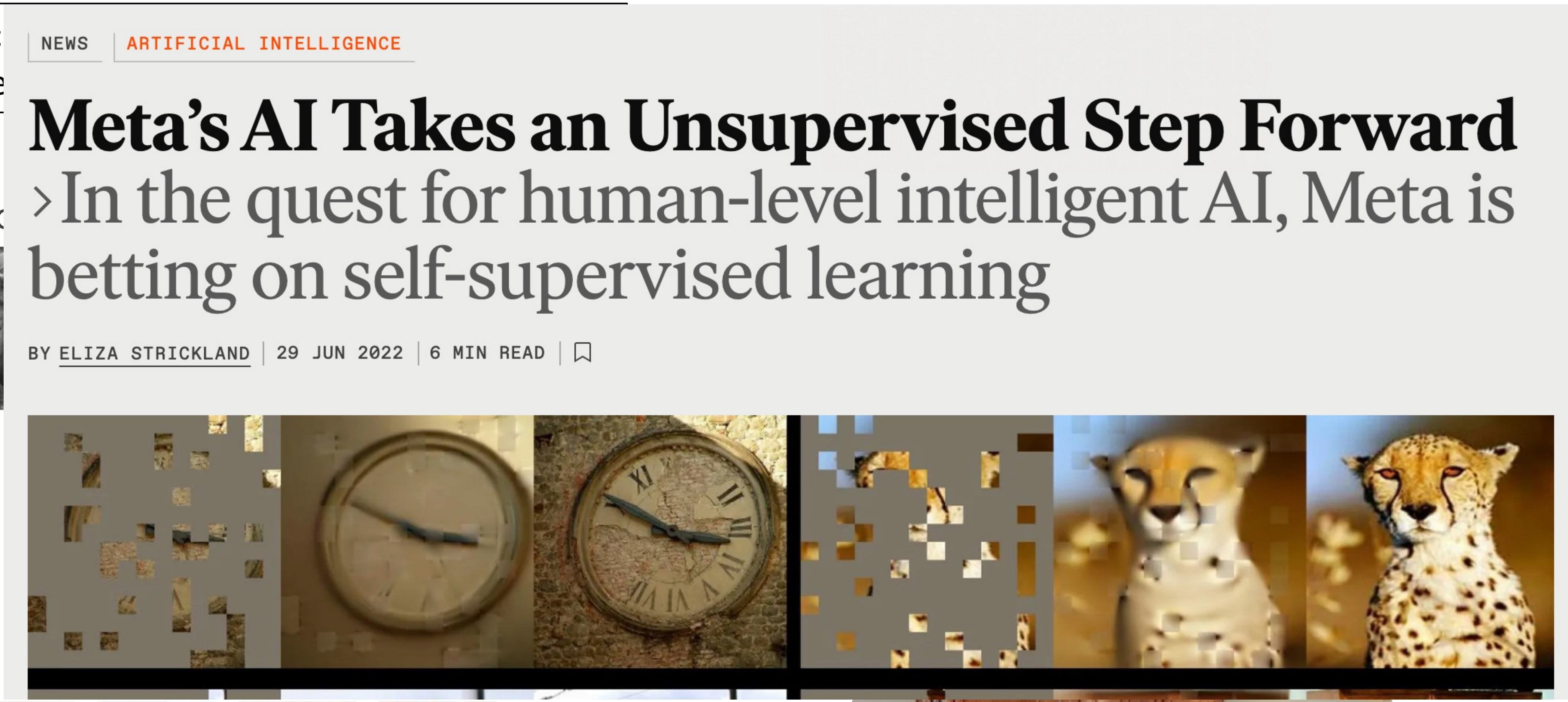
Internal Explorations: Self Supervised Learning

Agent uses downtimes to c
the absence of explicit labe

Fill in the blank



Co



NEWS | ARTIFICIAL INTELLIGENCE

Meta's AI Takes an Unsupervised Step Forward

›In the quest for human-level intelligent AI, Meta is betting on self-supervised learning

BY ELIZA STRICKLAND | 29 JUN 2022 | 6 MIN READ

Reference Image

Evaluating lifetime learning – APL

Standard ML datasets don't capture lifelong learning challenges. Richer datasets and environments are needed.

SRI

Modified StarCraft2* interface enables surprises to be injected into the game on-the-fly:

- Change terrain
- Alter unit capability
- Switch friends to foes
- Move goals
- Increase weapon range

* Blizzard Entertainment, 2010

Example simulation with injected surprises

How to Make Lifelong Learning more Robust?

What else can neuroscience tell us?

I. Robustness

1. Inherent abstraction - generalization
2. Reflexes save against surprises - safety
3. Temporal understanding of the environment – awareness

II. Ever-learning enablers:

4. Forward propagation algorithm
5. Parallel and distributed structures

III. Structural Efficiency

6. Fast adapting compact circuits: multifunctionality and neuromodulations
7. Rich multiscale neurons – computability
8. Asynchronous Computing

How to Make Lifelong Learning Effective?

What else can neuroscience tell us?

I. Robustness

1. Inherent abstraction - generalization
2. Reflexes save against surprises - safety
3. Temporal understanding of the environment – awareness

II. Ever-learning enablers:

4. Forward propagation algorithm
5. Parallel and distributed structures

III. Structural Efficiency

6. Fast adapting compact circuits: multifunctionality and neuromodulations
7. Rich multiscale neurons – computability
8. Asynchronous Computing

How to Make Lifelong Learning Effective?

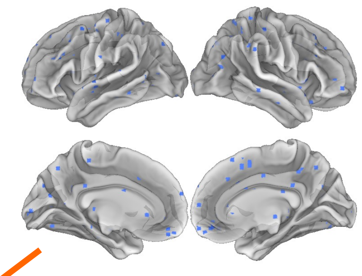
1. Inherent abstraction - generalization

*"The global landscape of cognition: hierarchical aggregation as an organizational principle of human cortical networks and functions," **Nature Scientific Reports** 2015.*

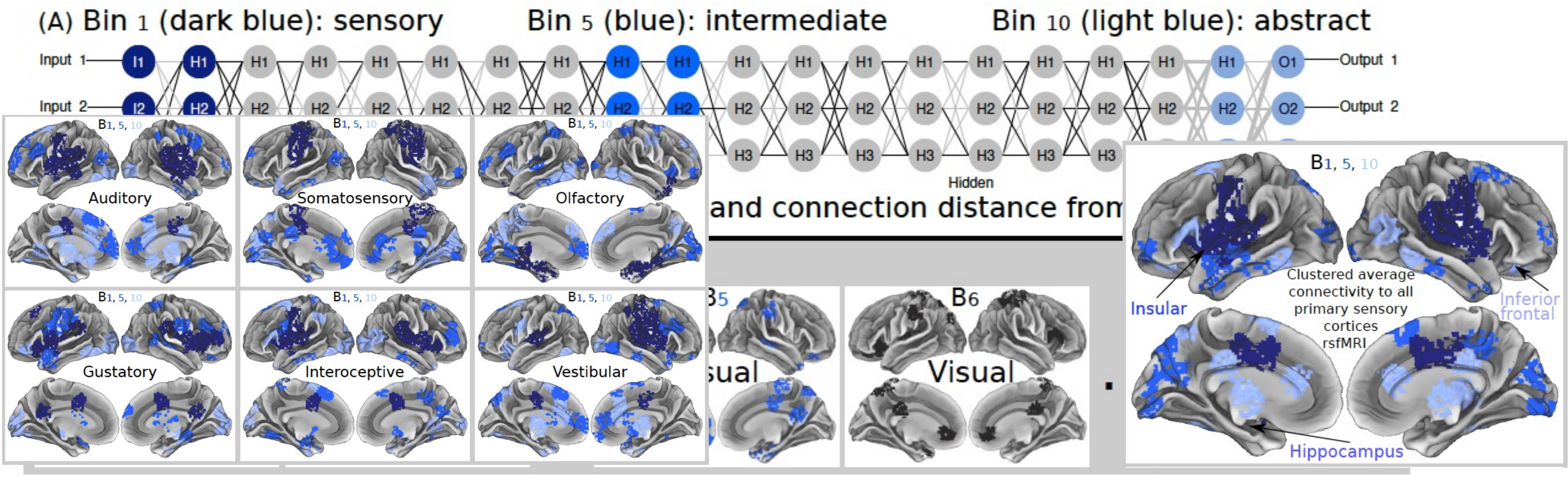
Is there a hierarchy (visual like) for cognition?

Connectome models are recursive, but we can define "distance" from inputs' cortices via time (rs-fmri) and highways (DTI)

Relating behavior with depth?



?

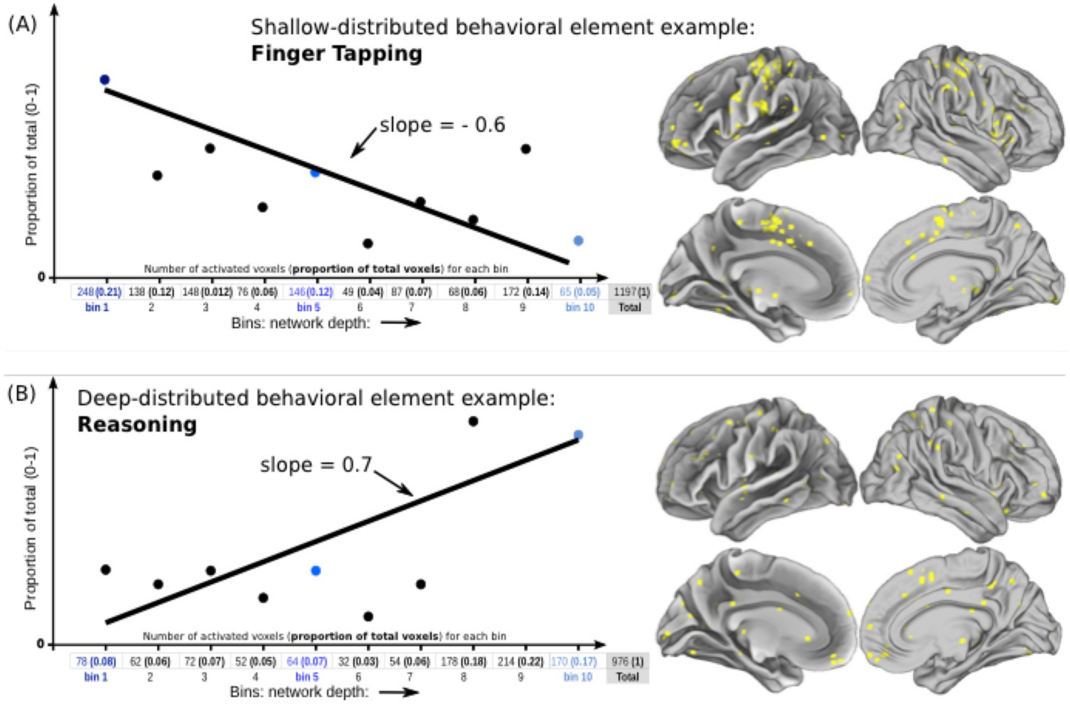


How to Make Lifelong Learning Effective?

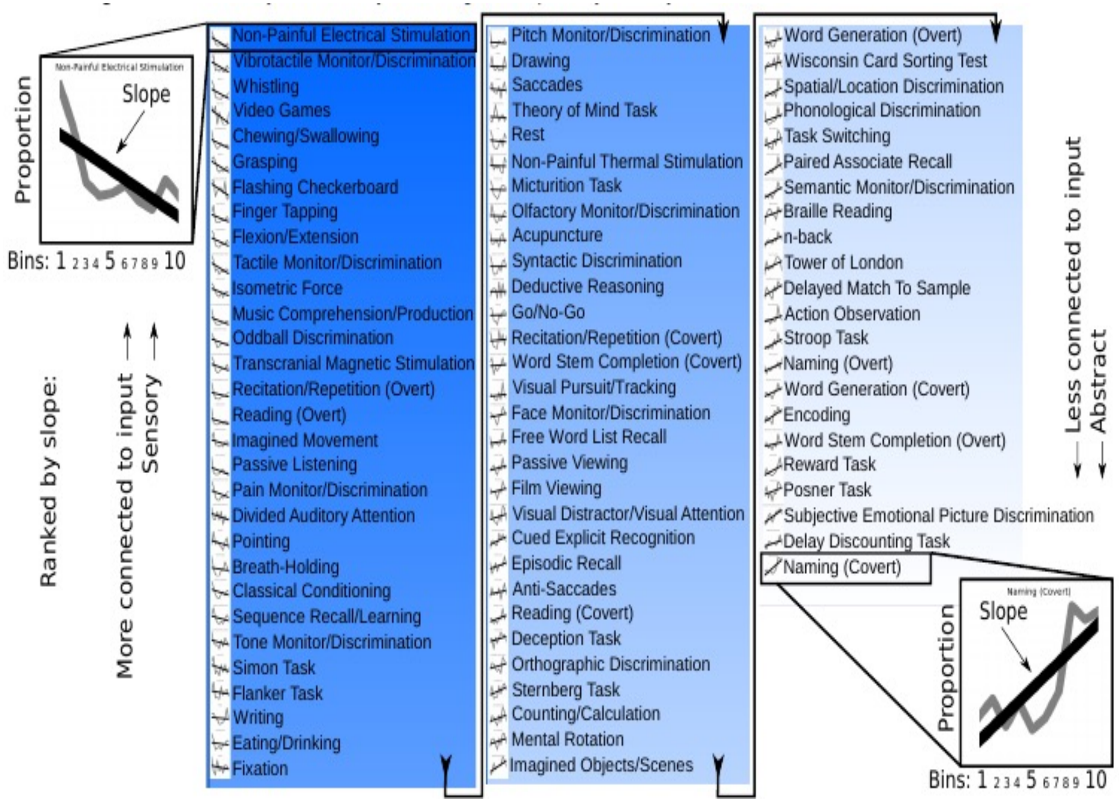
1. Inherent abstraction - generalization

*"The global landscape of cognition: hierarchical aggregation as an organizational principle of human cortical networks and functions," **Nature Scientific Reports** 2015.*

Neural activity hierarchy via fMRI ("sloped based")



Abstract thoughts: aggregation & abstractions

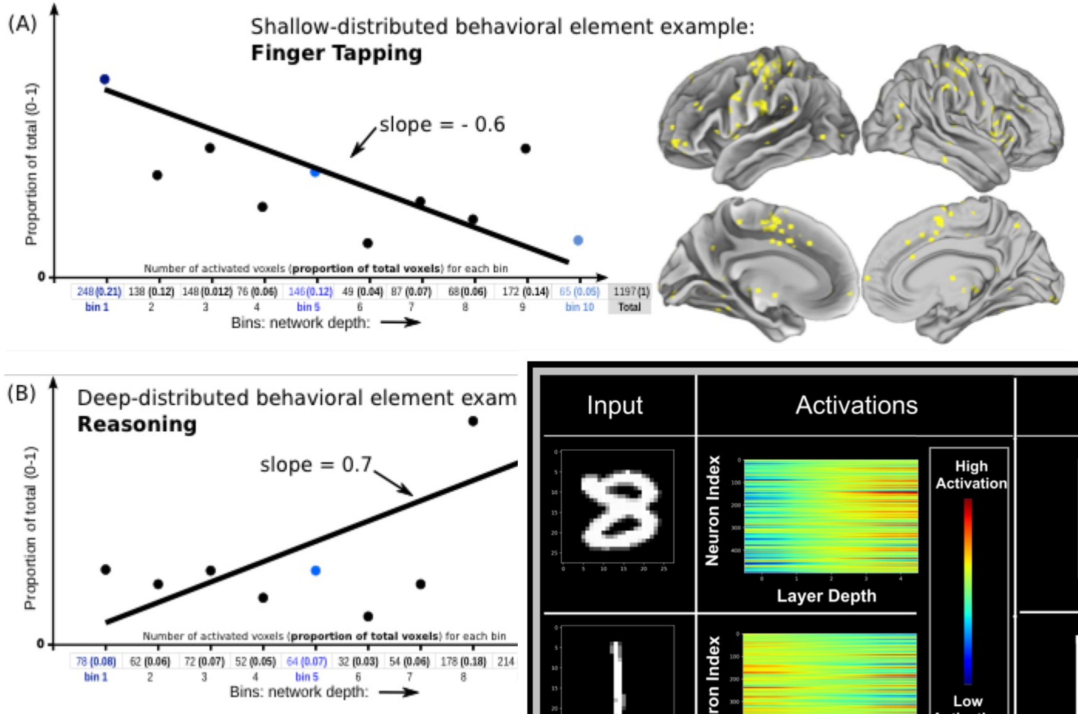


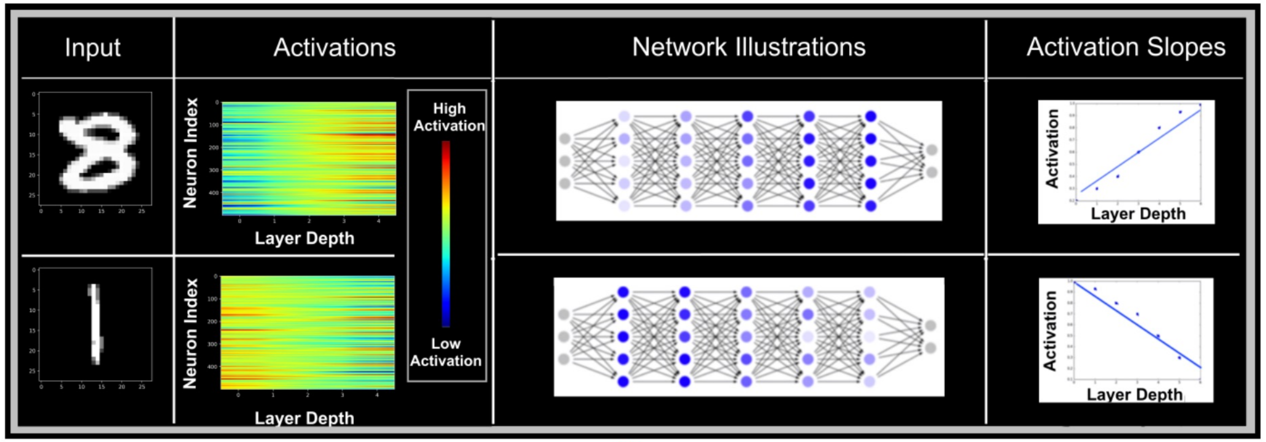
How to Make Lifelong Learning Effective?

1. Inherent abstraction - generalization

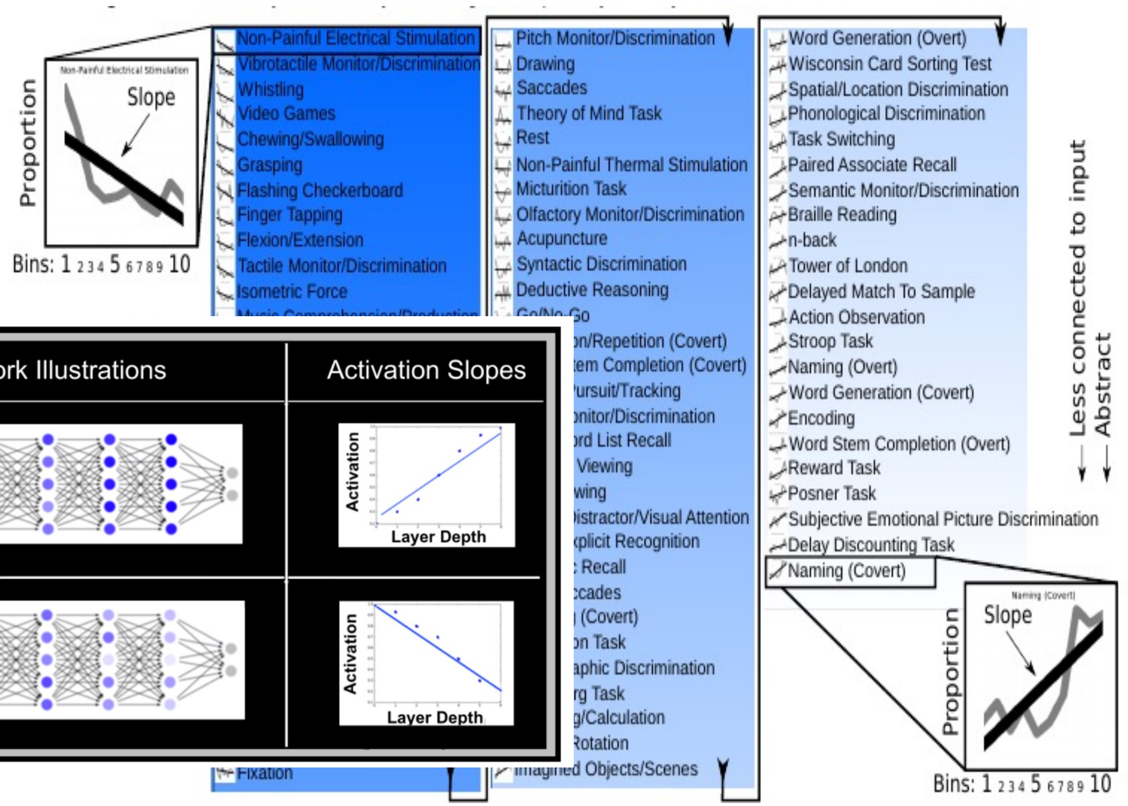
*"The global landscape of cognition: hierarchical aggregation as an organizational principle of human cortical networks and functions," **Nature Scientific Reports** 2015.*

Neural activity hierarchy via fMRI ("sloped based")





Abstract thoughts: aggregation & abstractions



How to Make Lifelong Learning Effective?

What else can neuroscience tell us?

I. Robustness

1. Inherent abstraction - generalization
2. Reflexes save against surprises - safety
3. Temporal understanding of the environment – awareness

II. Ever-learning enablers:

4. Forward propagation algorithm
5. Parallel and distributed structures

III. Structural Efficiency

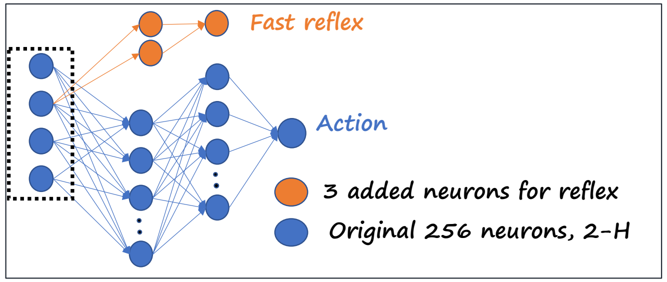
6. Fast adapting compact circuits: multifunctionality and neuromodulations
7. Rich multiscale neurons – computability
8. Asynchronous Computing

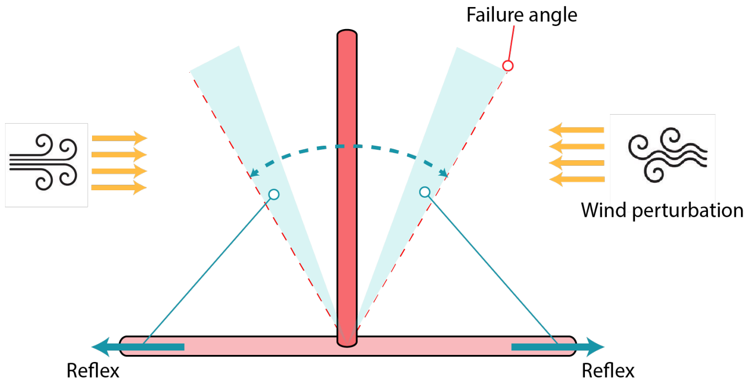
Saving from unknown disasters

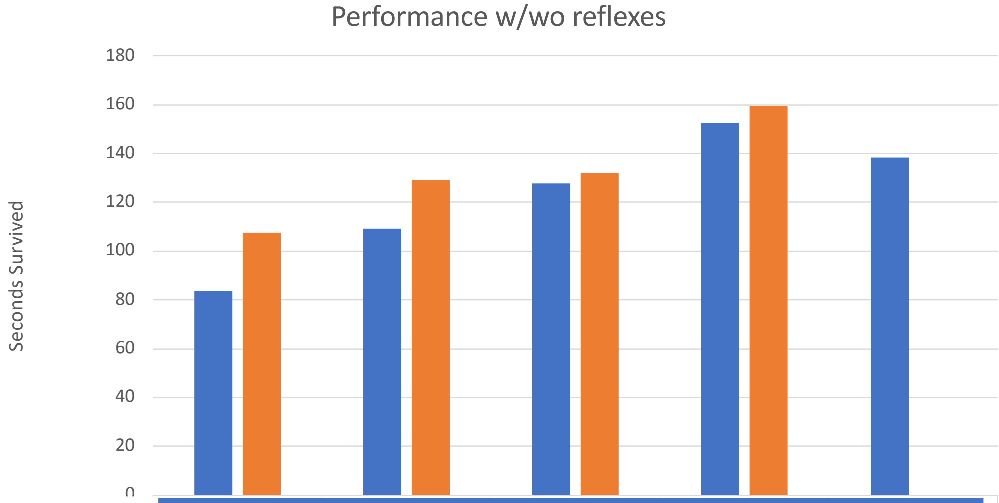
2. Reflexes save against surprises - safety

Devdhar Patel, Francesca Walsh, Terry Sejnowski

Protecting AI systems without increasing power requirements:
Fast reflexes help survival in adversarial environments







Seconds survived (Decisions made)

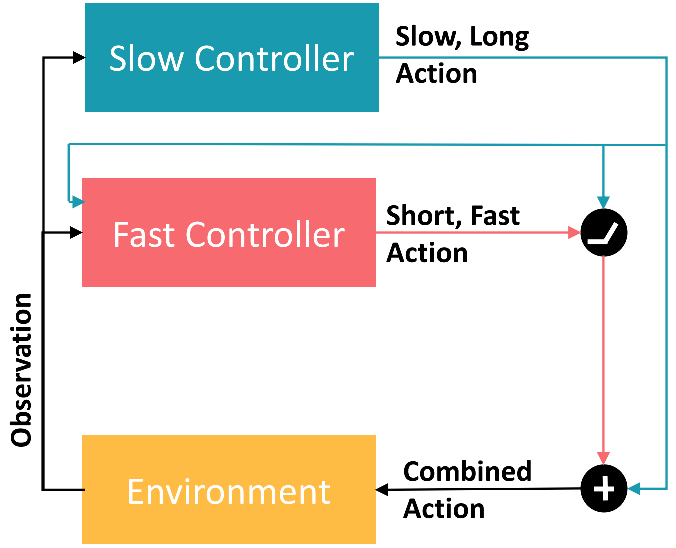
Res. speed	0.16s	0.08s	0.04	0.02	0.01
w/o Reflex	83.81 (534)	109.21 (1366)	127.8 (3195)	152.68 (7634)	138.4 (13840)
Reflex	107.72 (674)	129.14 (1618)	132.23 (3306)	159.63 (7982)	

Reflex activated at angles > 0.15
(agent fails at angles > 0.2)

Saving from unknown disasters

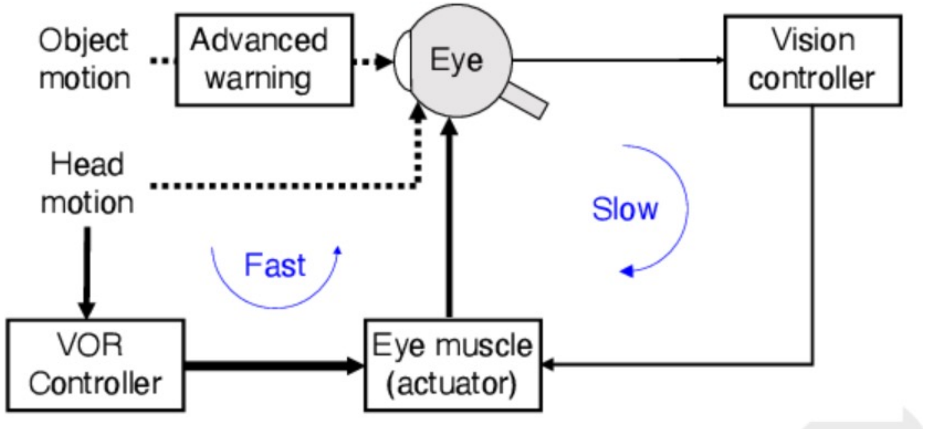
2. Reflexes save against surprises - safety

Devdhar Patel, Francesca Walsh, Terry Sejnowski – submitted



General solution of temporal layered architecture (TLA):

Brain may be using two layers together:



Picture taken from:
Nakahira, Yorie et al. "Diversity-enabled sweet spots in layered architectures and speed–accuracy trade-offs in sensorimotor control." *Proceedings of the National Academy of Sciences* 118 (2021): n. pag.

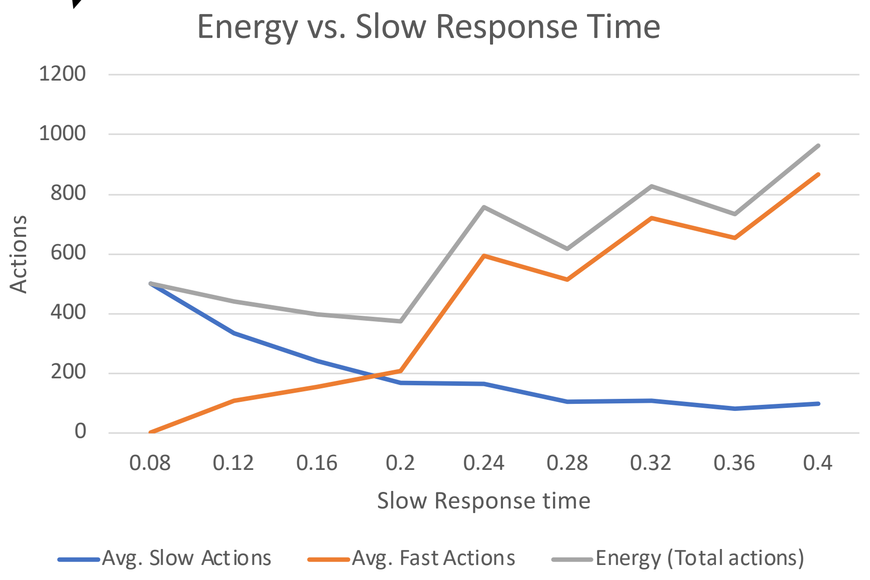
- Layer a fast controller over a slow controller. Gate fast action based on its value.
- Training slow controller first also allows faster exploration due to increase step sizes
- Improved training for fast controller and common activity

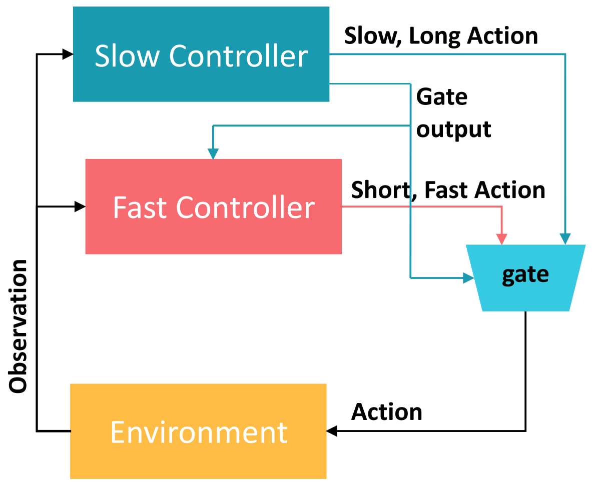
Also – towards distributed AI, different response times corresponds to communication and computation delays

Saving from unknown disasters

2. Reflexes save against surprises - safety

Devdhar Patel, Francesca Walsh, Terry Sejnowski – submitted





Return per Action RPA:
TLA best among RL variances for dynamic environments

Environment	TLA			TD3		
	Response Time	Avg. Return	RPA	Response Time	Avg. Return	RPA
InvertedPendulum-v2	0.04s, 0.08s	**1000 ± 0**	**1.9**	0.04s	1000 ± 0	1.00
InvertedDoublePendulum-v2	0.05s, 0.1s	**9358.94 ± 0.82**	**18.49**	0.05s	9358.48 ± 2.5	9.36
Hopper-v2	0.008s, 0.016s	**3443.21 ± 131.6**	**3.74**	0.008s	3032.25 ± 262.8	3.20
Walker2d-v2	0.008s, 0.016s	**3694.04 ± 128.58**	**3.96**	0.008s	3233.77 ± 895.3	3.23

How to Make Lifelong Learning Effective?

What else can neuroscience tell us?

I. Robustness

1. Inherent abstraction - generalization
2. Reflexes save against surprises - safety
3. Temporal understanding of the environment – awareness

II. Ever-learning enablers:

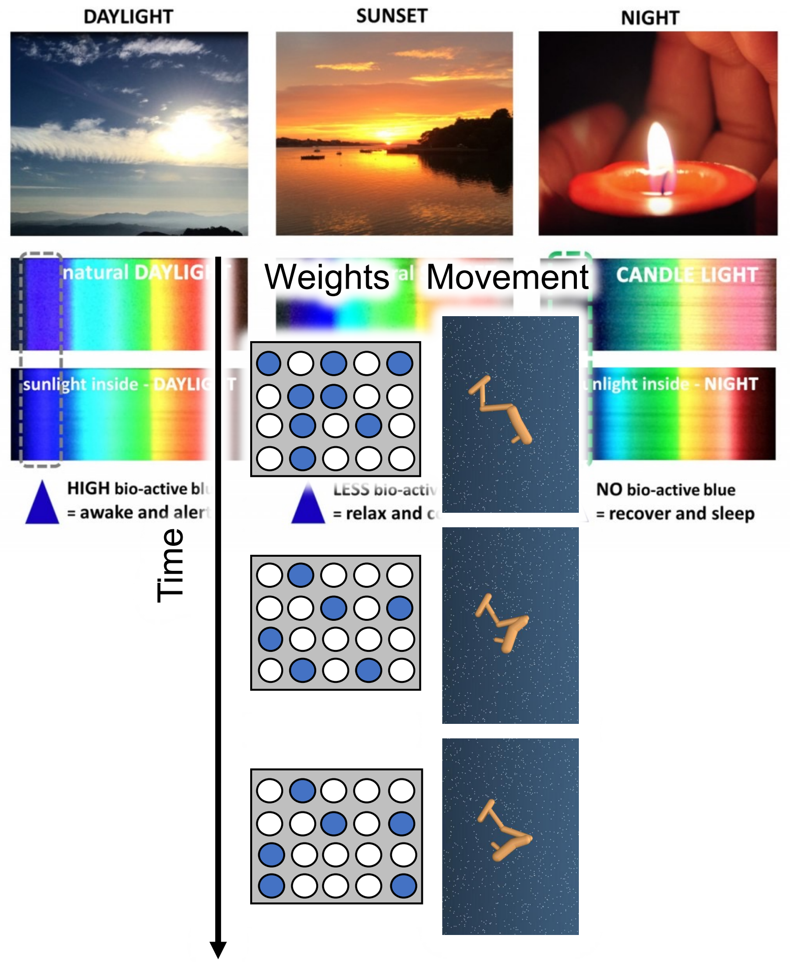
4. Forward propagation algorithm
5. Parallel and distributed structures

III. Structural Efficiency

6. Fast adapting compact circuits: multifunctionality and neuromodulations
7. Rich multiscale neurons – computability
8. Asynchronous Computing

How to Make Lifelong Learning Effective?

3. Temporal understanding of environment



Current AI, including L2 makes no use of repetitions and periodic behavior

Required for understanding of environment

How to Make Lifelong Learning Effective?

3. Temporal understanding of environment

Goal: Develop a new class of temporal aware neural networks:

a. High prediction accuracy

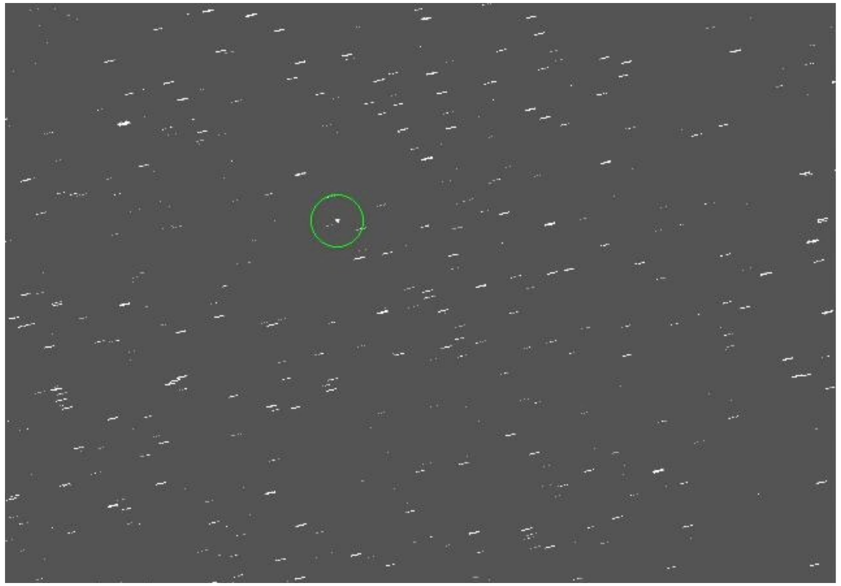
b. Irregular and missing data (realistic)

c. Fast learning and reaction

d. Get small networks first (rather than late pruning)

Health Monitoring: detect/predict anomalies to be reported or acted upon

Event Based Identification: improved tracking, partial irregular info

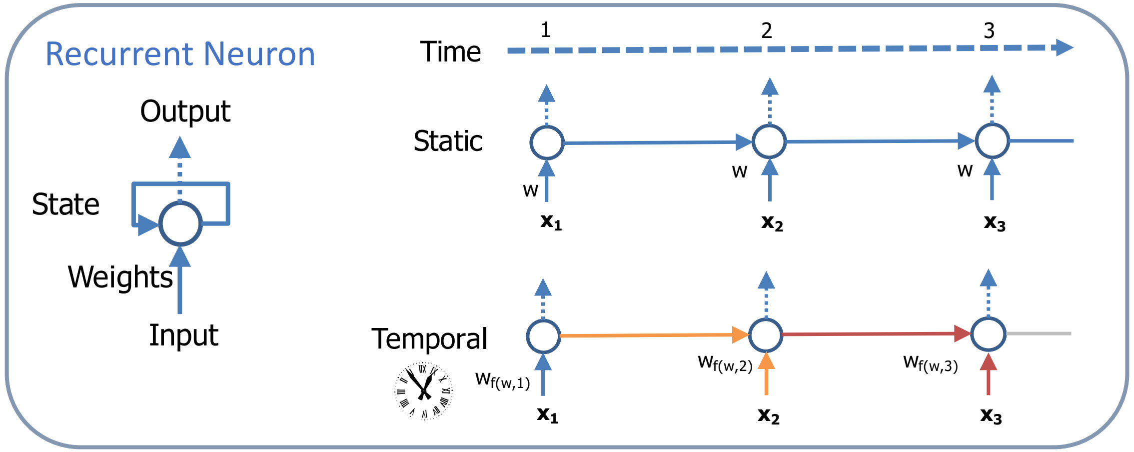


Solution: Neural Networks that Temporally Change

Goal: Develop a new class of temporal aware neural networks:

a. High prediction accuracy
b. Irregular and missing data (realistic)
c. Fast learning and reaction
d. Get small networks first (rather than late pruning)

Solution: Temporally changing weights



Inspiration: Turing 1948

NOTCH's Equations and Algorithm

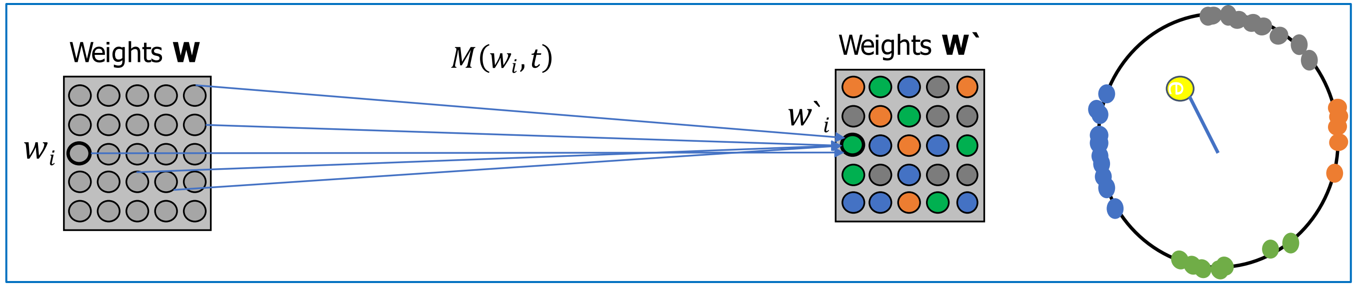
Calculate temporal weights:

$$M_i(W_l, t) = \frac{1}{\#W} \sum_{j=1}^{\#W} K_{\{ij\}} \sin\left[f(t) * \left(w_i - w_j\right) + \phi(t)\right]$$

Where:

- $f(t)$ time-varying frequency
- $\phi(t)$ time-varying phase shift
- K matrix of coupling coefficients
- W_l meta-weights of layer l

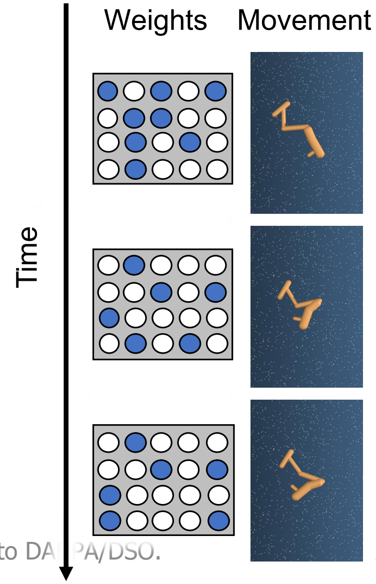
All learned simultaneously with the weights



Kuramoto Model: behavior of systems of chemical and biological oscillators

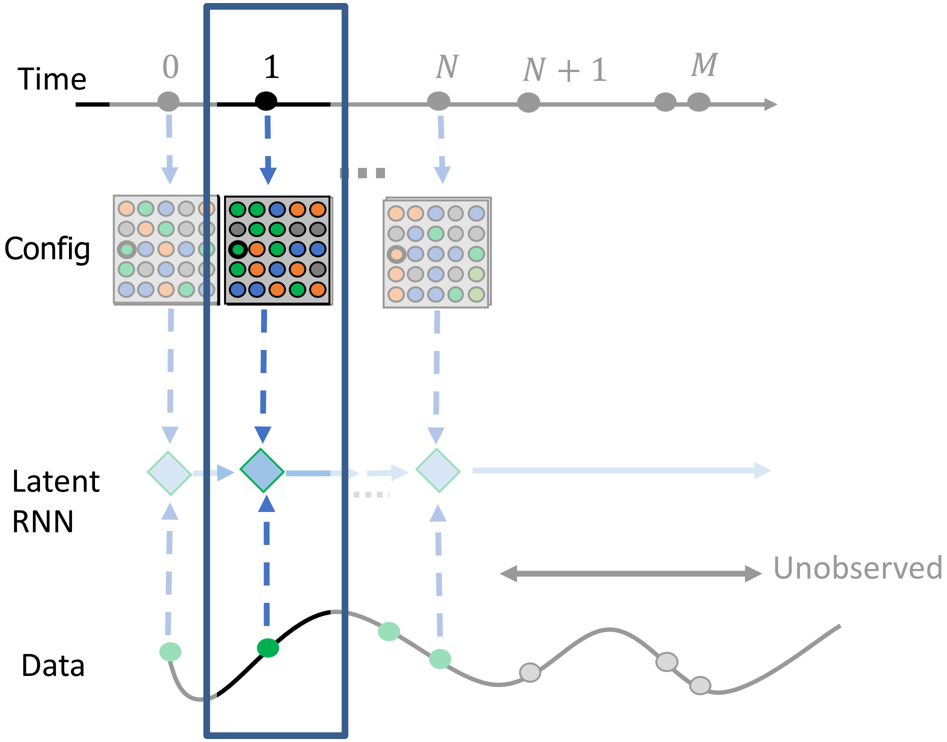
$$\frac{d\theta_i}{dt} = \omega_i + \frac{1}{N} \sum_{j=1}^{N} K_{ij} \sin(\theta_i - \theta_j)$$

- Composed of N limit-cycle oscillators, with phases θ_i, and coupling constant K
- Interactions depend sinusoidally on the phase difference between every two oscillators



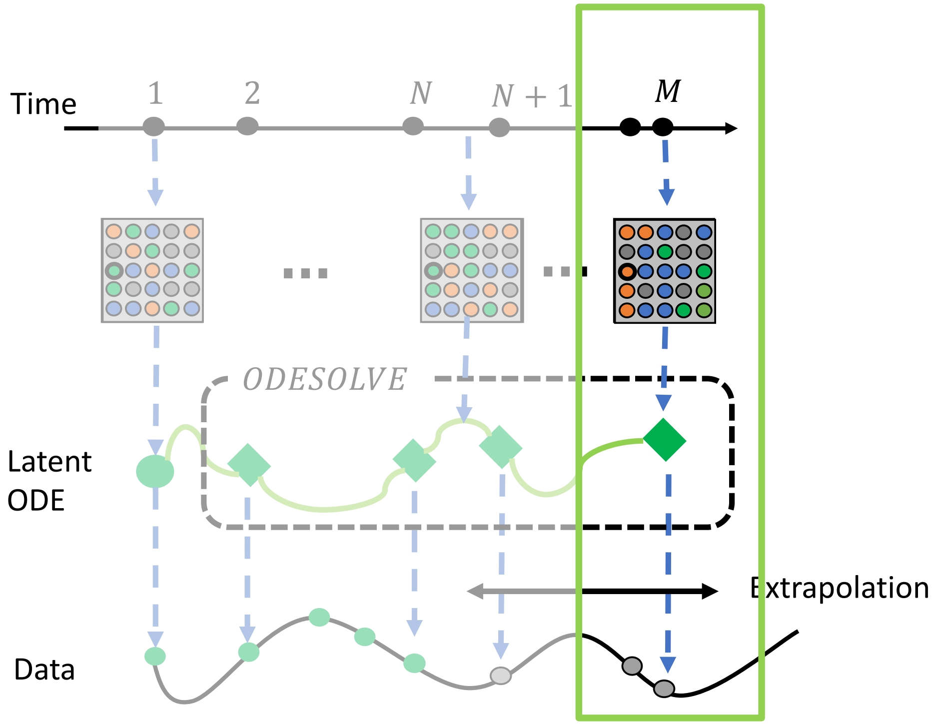
NOTCH with Discrete or Continuous Models

RNN/GRU generates weights with inputs

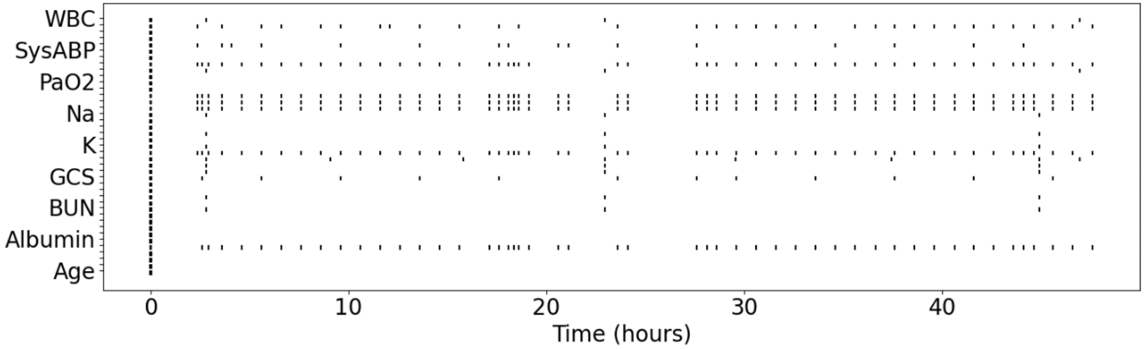


NeuralODE generates weights

1. with inputs
2. in between inputs
3. predicting for future points



Results: Slide 1: ICU Measurements Dataset



12,000 stays in the ICU
48 hours of a patient's admission to ICU
2,880 possible measurements (mostly lab) per time series
37 highly irregular measurements, 80% missing values
Released home ("1") or die in hospital ("0") - 86 vs. 14%

Discriminative Task

Task 1A: Predict in-hospital mortality

1. Observe first 48 hours of ICU measurements of patient
2. Predict in the hospital survival (1) or not (0)

Method	AUC	Epochs	# Params
LatentODE	0.829	> 100	163,972
LatentODE Scaling (Ours)	0.861	31	76,427

Generative Tasks

Task 1B: Predict the 80% missing points in time series

1. Observe ICU measurements (black points)
2. Given only an initial latent vector z, reconstruct all measurements

Method	Error (MSE)	Epochs	# Params
LatentODE	2.118×10^{-3}	83	67,071
LatentODE Scaling (Ours)	1.370×10^{-3}	49	52,016

Task 1C: Predict future points in time series

10% data similar results

1. Observe the first 24 hours of ICU measurements
2. Predict ICU measurements for next 24 hours

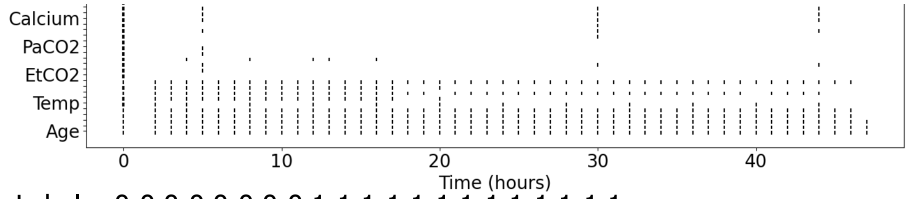
Method	Error (MSE)	Epochs	# Params
LatentODE	2.208×10^{-3}	98	67,071
LatentODE Scaling (Ours)	1.900×10^{-3}	16	52,016

NOTCH:
- Improved training efficiency
- Better performance
- Smaller Model

Results Slide 2: Online Sepsis Prediction

Discriminative Task

Predict at every hour whether the patient will have sepsis within the next 6 to 12 hours



Labels: 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 …..

Sepsis = 1
No Sepsis = 0

6 hr. window

Within times 1-6hr, should predict no sepsis

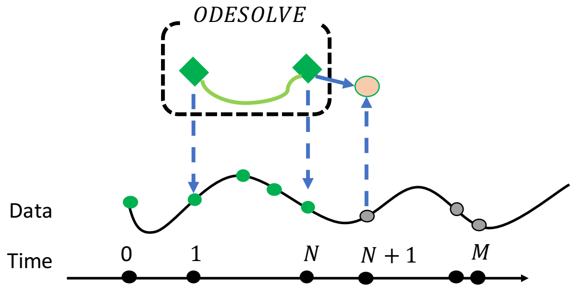
Within times 12-18hr, should predict sepsis

Task 2B: Continuous Models

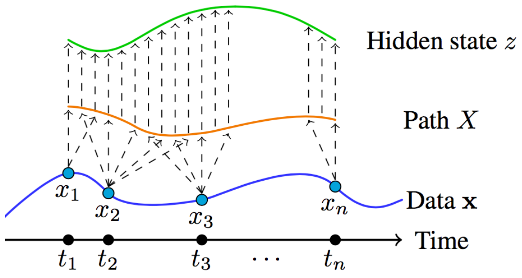
Method	Performance (AUC)	Epochs	# Params
ODERNN [1]	0.689	67	148,672
ODERNN [1]	0.765	86	680,462
w/ Scaling (Ours)	0.787	24	149,519
NCDE [2]	0.925	130	55,949
NCDE [2]	0.925	110	193,541
w/ Scaling (Ours)	0.931	180	58,553

Task 2A: Discrete Model

Method	Performance (AUC)	Epochs	# Params
GRU	0.781	> 100	1,132,544
w/ Scaling (Ours)	0.784	20	22,529
w/ Scaling (Ours)	0.774	65	4,998



ODERNN



NCDE

Similar results with 0.1% - 25 examples

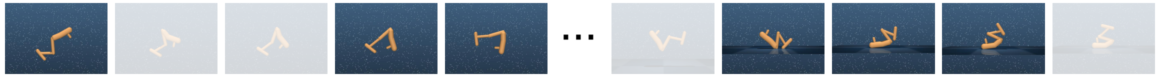
[1] Rubanova, Yulia, Ricky TQ Chen, and David K. Duvenaud. "Latent ordinary differential equations for irregularly-sampled time series." *Advances in neural information processing systems* 32 (2019).
[2] Kidger, Patrick, et al. "Neural controlled differential equations for irregular time series." *Advances in Neural Information Processing Systems* 33 (2020): 6696-6707.

Results Slide 3: MuJoCo Hopper and Human Activity Prediction

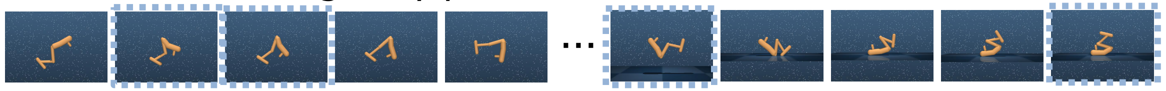
Generative Tasks: Hopper Physics

Task 3A: Predict missing points in Hopper time series

1. Observe irregular and sparse body positions



2. Generate missing body positions



Method	Error (MSE)	Epochs	# Params
LatentODE	3.60 x 10^{-3}	> 100	617,619
w/ Scaling (Ours)	2.40 x 10^{-3}	57	112,739

Task 3B: Generate next in Hopper body positions

Predict future points in time series: Observe 50, generate 50

Method	Error (MSE)	Epochs	# Params
LatentODE	14.41 x 10^{-3}	> 100	617,619
w/ Scaling (Ours)	12.72(11.91) x 10^{-3}	40(65)	112,739

Discriminative Task:

Human Activity Prediction

Task 3C: Classify each point to activity

Method	Performance (AUC)	Epochs	# Params
Latent ODE [1]	0.846	> 100	1,696,763
w/ Scaling (Ours)	0.870	10	141,023

NOTCH:
- Improved training efficiency
- Better performance
- Smaller Model

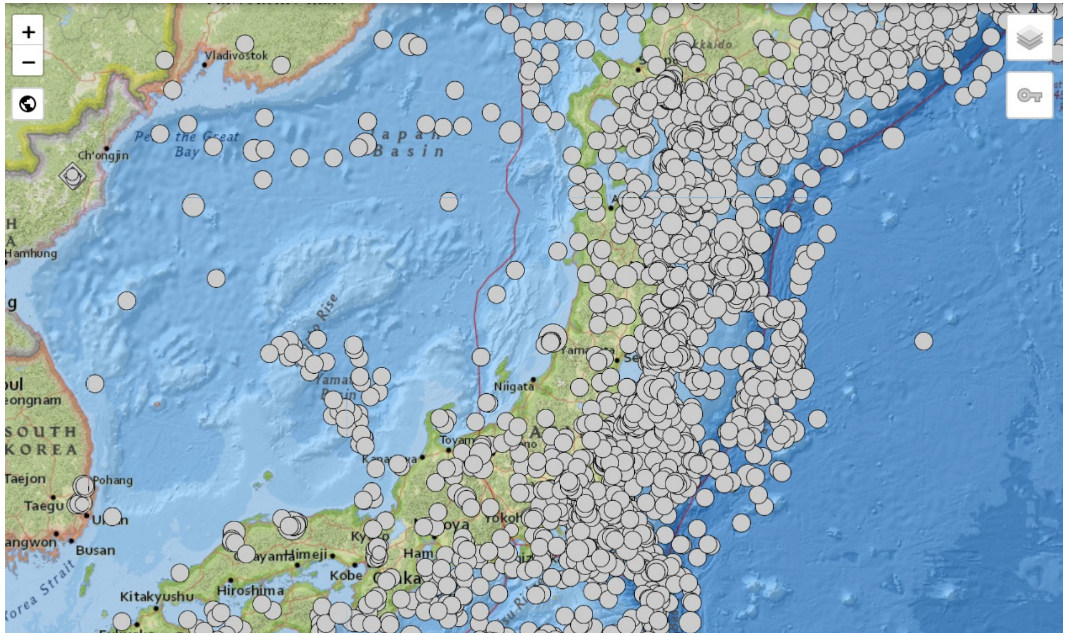
[1] Rubanova, Yulia, Ricky TQ Chen, and David K. Duvenaud. "Latent ordinary differential equations for irregularly-sampled time series." *Advances in neural information processing systems* 32 (2019).

Results Slide 4: SpatioTemporal

Generative Task

Task 4A: Earthquake events in Japan

Method	Error (NLL)	Epochs	# Params
Time-var. CNF [2]	1.510 (1.955)	(497)	13,064
w/ Scaling (Ours)	1.481 (1.552)	(350)	6,669

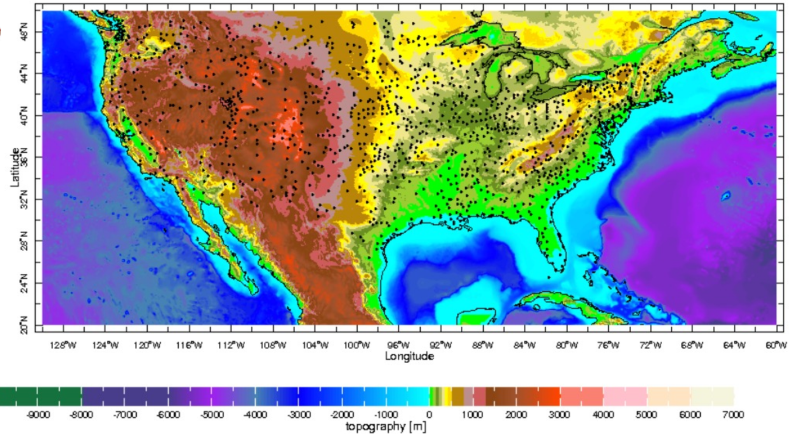


[1] De Brouwer, Edward, et al. "Gru-ode-bayes: Continuous modeling of sporadically-observed time series." *Advances in neural information processing systems* 32 (2019).
[2] Chen, Ricky TQ, Brandon Amos, and Maximilian Nickel. "Neural Spatio-Temporal Point Processes." *International Conference on Learning Representations*. 2020.

Discriminative Task: Climate Prediction

Task 4B: Given 3 years of observations, predict the next 3 measurements

Method	Error (MSE)	Epochs	# Params
GRUODE [1]	0.43	74	42,640
w/ Scaling (Ours)	0.40	68	9,105



NOTCH:
- Improved training efficiency
- Better performance
- Smaller Model

Results Slide 5: SpatioTemporal

Generative Tasks

Task 4C: Covid cases for the state of New Jersey

Method	Error (NLL)	Epochs	# Params
Time-var. CNF [1]	1.952 (2.182)	(231)	13,064
w/ Scaling (Ours)	1.916 (2.151)	(197)	6,669

Task 4D: Location of Bike Rentals

Method	Error (NLL)	# Params
Time-var. CNF [1]	2.315	13,064
w/ Scaling (Ours)	1.480	6,669

Task 4E: Timestamps of page edits in Wiki

Pseudo Spatial - pages have different linking distances to each-other

Method	Error (NLL)	Epochs	# Params
Jump ODE [2]	1.550 (1.170)	(6)	576,769
w/ Scaling (Ours)	0.083 (-0.394)	(159)	280,229

NOTCH:
- Improved training efficiency
- Better performance
- Smaller Model

[1] Chen, Ricky TQ, Brandon Amos, and Maximilian Nickel. "Neural Spatio-Temporal Point Processes." *International Conference on Learning Representations*. 2020.
[2] Jia, Junteng, and Austin R. Benson. "Neural jump stochastic differential equations." *Advances in Neural Information Processing Systems* 32 (2019).

How to Make Lifelong Learning Effective?

What else can neuroscience tell us?

I. Robustness

1. Inherent abstraction - generalization
2. Reflexes save against surprises - safety
3. Temporal understanding of the environment – awareness

II. Ever-learning enablers:

4. Forward propagation algorithm
5. Parallel and distributed structures

III. Structural Efficiency

6. Fast adapting compact circuits: multifunctionality and neuromodulations
7. Rich multiscale neurons – computability
8. Asynchronous Computing

How to Make Lifelong Learning Effective?

6. Forward propagation algorithm

Adam Kohan and Ed Rietman – under review
https://arxiv.org/abs/2204.01723

Backprop:

a. Freezes: forward, backward passes errors
b. Neurons need to remember their output
c. Need to process one I/O at a time

- Hardware excessive
- Problem with online activity
- Not biological

Forward Learning:

Forward pass
Local update follows local compute
Pipeline possible

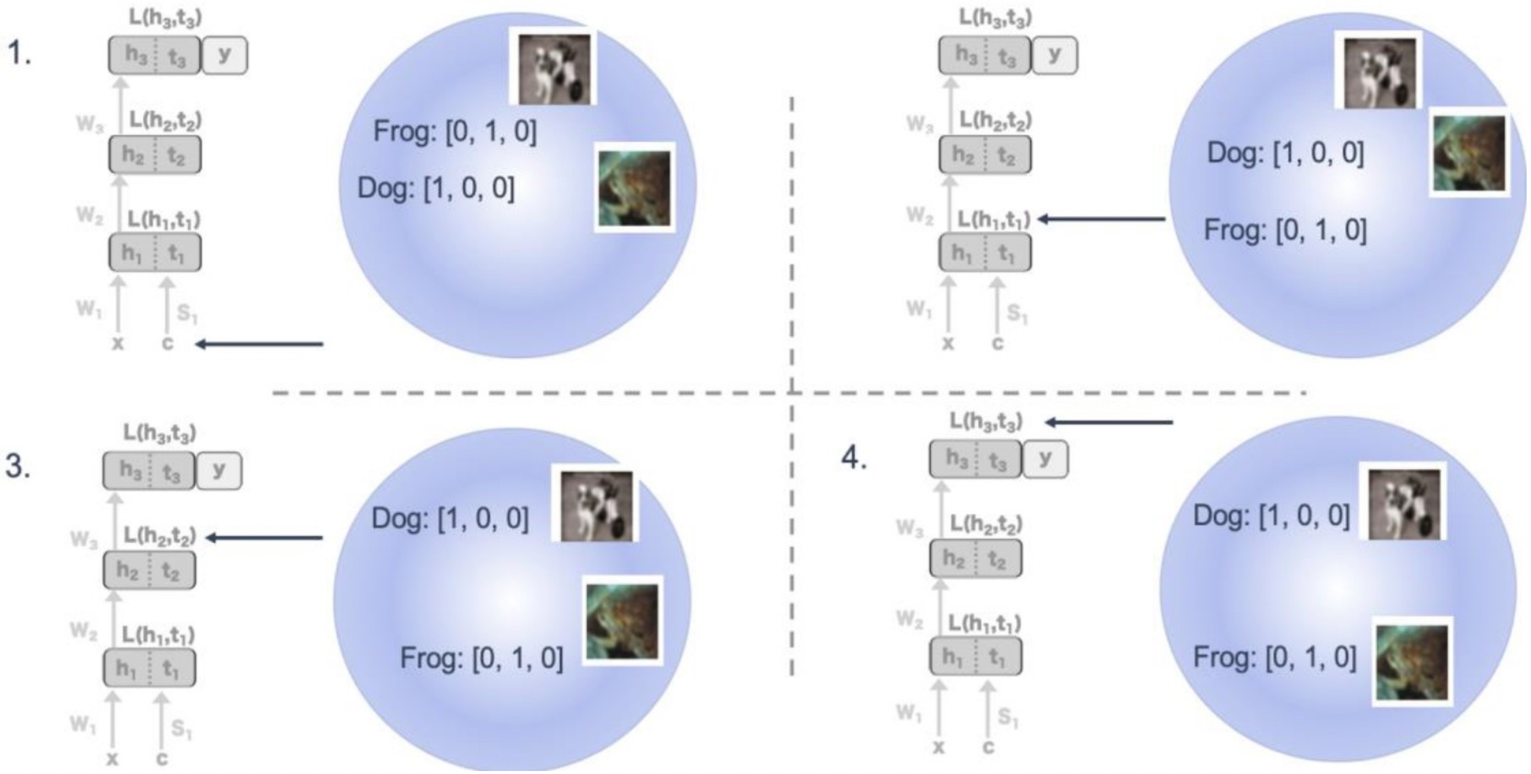
- No feedback connectivity
- Online addition of classes
- New idea for learning signals

How to Make Lifelong Learning Effective?

6. Forward propagation algorithm

- ✓ Add classes on the go
- ✓ Local update, targeted learning
- ✓ No need for backward pass

ForProp Algorithm in training mode, receives input and target/label. It first morphs them to a common state space, and only then updates the weights to assure that at the output layer, the input and label are co-transformed to match



How to Make Lifelong Learning Effective?

What else can neuroscience tell us?

I. Robustness

1. Inherent abstraction - generalization
2. Reflexes save against surprises - safety
3. Temporal understanding of the environment – awareness

II. Ever-learning enablers:

4. Forward propagation algorithm
5. Parallel and distributed structures

III. Structural Efficiency

6. Fast adapting compact circuits: multifunctionality and neuromodulations
7. Rich multiscale neurons – computability
8. Asynchronous Computing

How to Make Lifelong Learning Effective?

What else can neuroscience tell us?

I. Robustness

1. Inherent abstraction - generalization
2. Reflexes save against surprises - safety
3. Temporal understanding of the environment – awareness

II. Ever-learning enablers:

4. Forward propagation algorithm
5. Parallel and distributed structures

III. Structural Efficiency

6. Fast adapting compact circuits: multifunctionality and neuromodulations
7. Rich multiscale neurons – computability
8. Asynchronous Computing

How to Make Lifelong Learning Effective?

8. Reliable asynchronous computing:

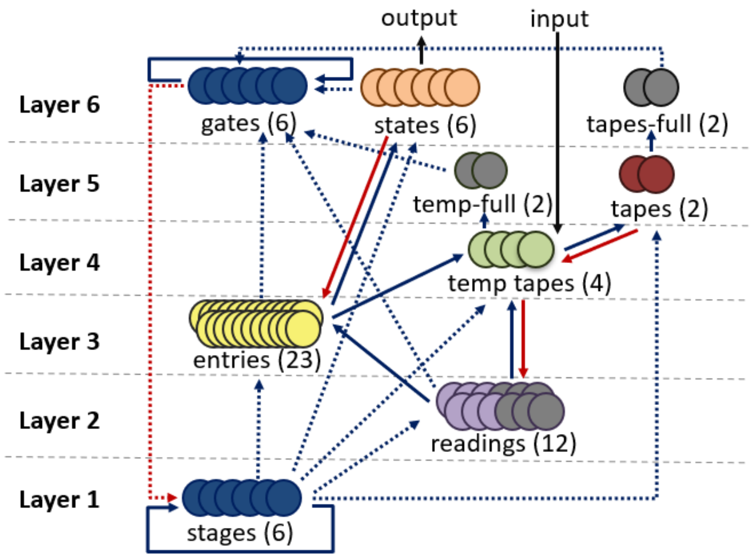
Steven Chung and Hava Siegelmann, NerIPS2021: No need of large neural precision for Turing universality

The brain is made of asynchronous neural networks, but SOTA computational complexity neural models is synchronous, having a global clock which assures that the neurons update at the same times.

Asynchronicity is a key feature of biological intelligence: associations among inputs and modalities for correlations and perhaps causality. Time sensitive adaptivity (STDP), great energy saving

Circuits with 80-100 neurons, bear a striking similarity to cortical minicolumns – the building blocks of the mammalian brain - in both size and architectural details.

Hypotheses: the brain can run close to billion algorithms in parallel and that it can tune to be stronger than today's computers.



Lifelong Learning Robust AI

Perhaps, in a few years, much of what we call AI won't be considered intelligent without lifelong brain-inspired robust learning

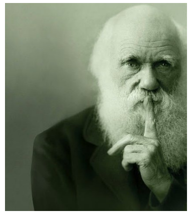
BINDS Lab

Bioinspired Neural & Dynamical Systems

Work with:
Terrance Sejnowski (UCSD)
Ed Rietman
Adam Kohan
Devdhar Patel
Francesca Walsh
Joshua Russel
Peter Delmastro
Ignacio Gavier
Cooper Sigrist
Rushiv Arora
Arjun Karuvally

"…it is not the strongest that survives; but…the one that is able best to adapt…to the changing environment…."

L.C. Megginson, re
"On the Origin of Species"



Thank you !

For-Propagation Algorithm Results

	CIFAR-10			CIFAR-100			SVHN		
	LL-BP	LL-FA	LAMA	LL-BP	LL-FA	LAMA	LL-BP	LL-FA	LAMA
Accuracy (%)	94.42	90.98	91.66	70.69	61.59	65.70	98.23	97.45	97.85
Time (s)	8.11	8.50	5.91	10.20	9.44	6.25	95.51	89.32	69.74
Memory (MiB)	8.85	13.03	6.19	11.45	5.51	5.19	11.41	5.43	4.91

Local targeted learning allows true online without missing important environmental changes

The Label Morphing Algorithm (LAMA) vs. SOTA local backpropagation (LL-BP) and local feedback alignment (LL-FA). LL-BP computes local errors by using additional hardware - local auxiliary networks at each layer - and training them with backprop. LAMA neither waits (as in the other local algorithms) nor adds hardware (in LL-BP), keeping low memory and time. It works in lifelong learning setups where new classes are formed online .

How to be ready for fast changes?

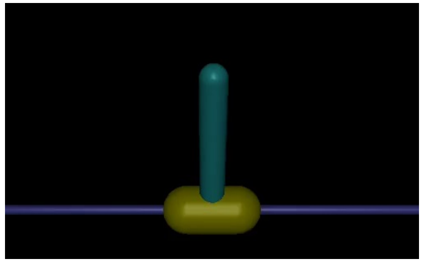
Sweet Spot of Response Rate

- *Not too fast*: An agent that responds too fast will need to handle a larger number of states, making it harder to learn
- *Not too slow*: An agent that is too slow to respond will fail when fast perturbation is applied

Avg. seconds in test environment / Min. Perturbation force required for failure

Max Pertubation during training \ Response Time	0.01s	0.02s	0.04s	0.08s	0.16s	0.32s	0.64s
20g	103.34s / 5.17g	116.86s / 5.84g	107.67s / 5.38g	104.97s / 5.25g	60.48s / 3.02g	0.56s / 0.03g	0.64s / 0.03g
19g	81.35s / 4.07g	131.7s / 6.58g	123.36s / 6.17g	123.0s / 6.15g	52.02s / 2.6g	0.56s / 0.03g	0.64s / 0.03g
18g	82.17s / 4.11g	106.42s / 5.32g	93.58s / 4.68g	125.72s / 6.29g	47.96s / 2.4g	0.56s / 0.03g	0.64s / 0.03g
17g	112.39s / 5.62g	122.07s / 6.1g	77.68s / 3.88g	124.9s / 6.24g	50.23s / 2.51g	0.56s / 0.03g	0.64s / 0.03g
16g	112.84s / 5.64g	117.17s / 5.86g	109.81s / 5.49g	120.75s / 6.04g	51.14s / 2.56g	0.56s / 0.03g	0.64s / 0.03g
15g	126.54s / 6.33g	135.1s / 6.76g	129.63s / 6.48g	144.4s / 7.22g	46.53s / 2.33g	0.56s / 0.03g	0.64s / 0.03g
14g	110.41s / 5.52g	103.62s / 5.18g	122.26s / 6.11g	125.4s / 6.27g	49.68s / 2.48g	0.56s / 0.03g	0.64s / 0.03g
13g	130.7s / 6.53g	135.36s / 6.77g	125.59s / 6.28g	100.0s / 5.0g	48.68s / 2.43g	0.56s / 0.03g	0.64s / 0.03g
12g	145.28s / 7.26g	150.51s / 7.53g	129.8s / 6.49g	126.94s / 6.35g	45.32s / 2.27g	0.56s / 0.03g	0.64s / 0.03g
11g	146.35s / 7.32g	136.8s / 6.84g	143.97s / 7.2g	125.95s / 6.3g	42.22s / 2.11g	0.56s / 0.03g	0.64s / 0.03g
10g	137.53s / 6.88g	150.76s / 7.54g	151.17s / 7.56g	134.74s / 6.74g	72.61s / 3.63g	0.56s / 0.03g	0.64s / 0.03g
9g	148.79s / 7.44g	169.98s / 8.5g	140.11s / 7.01g	115.18s / 5.76g	61.23s / 3.06g	0.56s / 0.03g	0.64s / 0.03g
8g	151.71s / 7.59g	163.66s / 8.18g	153.66s / 7.68g	143.22s / 7.16g	65.5s / 3.28g	0.56s / 0.03g	0.64s / 0.03g
7g	153.33s / 7.67g	164.35s / 8.22g	152.18s / 7.61g	130.3s / 6.52g	50.96s / 2.55g	0.56s / 0.03g	0.64s / 0.03g
6g	145.05s / 7.25g	160.05s / 8.0g	147.24s / 7.36g	130.72s / 6.54g	66.36s / 3.32g	0.56s / 0.03g	0.64s / 0.03g
5g	138.48s / 6.92g	137.42s / 6.87g	134.25s / 6.71g	114.41s / 5.72g	84.97s / 4.25g	0.56s / 0.03g	0.64s / 0.03g
4g	116.12s / 5.81g	124.57s / 6.23g	111.5s / 5.58g	106.21s / 5.31g	75.74s / 3.79g	0.56s / 0.03g	0.64s / 0.03g
3g	112.22s / 5.61g	97.75s / 4.89g	89.57s / 4.48g	97.09s / 4.85g	56.64s / 2.83g	0.56s / 0.03g	0.64s / 0.03g
2g	83.76s / 4.19g	77.34s / 3.87g	85.87s / 4.29g	80.13s / 4.01g	51.26s / 2.56g	0.56s / 0.03g	0.64s / 0.03g
1g	67.44s / 3.37g	54.61s / 2.73g	82.35s / 4.12g	76.75s / 3.84g	53.08s / 2.65g	0.56s / 0.03g	0.64s / 0.03g
0g	46.34s / 2.32g	61.85s / 3.09g	93.46s / 4.67g	81.26s / 4.06g	50.21s / 2.51g	0.56s / 0.03g	0.64s / 0.03g

Color scale: 20, 40, 60, 80, 100, 120, 140, 160



(Avg. over 5 trials)

Human in the Loop – But Smartly (work with Patrick Taylor)

Assistive technology that empowers controller to manage many tasks, work efficiently, reduce biases & errors, limit cognitive load (Taylor & Siege, USP 2020. Taylor et al *Frontiers in ICT* 2:15, 2015)

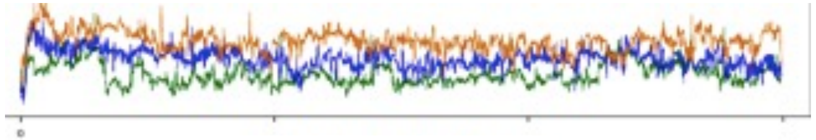
Computerized externalization of short-term working memory greatly aids real-time interaction

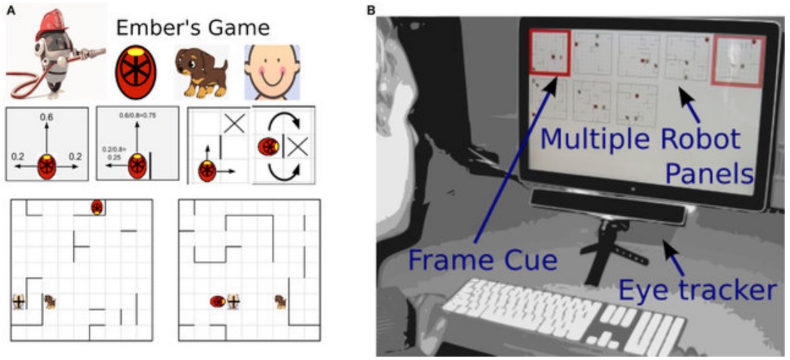
1. Operator defines "panels" - (frame/filled, transparent, color)
2. Can be overlayed on application
3. Provides visual cues to assist operators in determining the appropriate time to look at a specific location on the screen

Demonstration in semi-autonomous robots

Evaluation: 3 groups: frames on, off, random frames

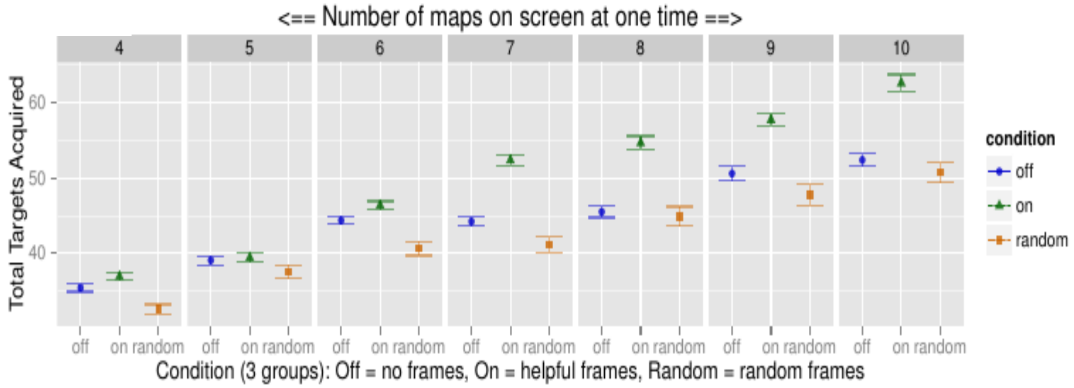
a. Cognitive load reduced





b. Improved # total targets

Participants manage better: (A) Faster to set paths to targets, (B) Robots wait less time at targets, (C) Users acquire points faster, and (D) Reduced bias



c. Micromanaging reduced: Utilizing automated system to fullest: Gaze and mouse moves optimized: (A) Macro-level gaze –more search (less fixation)

(B) mouse total mileage bigger, but more clustered locally

(C) Gaze and mouse meet rarely (Hand faster than eye, 33ms)