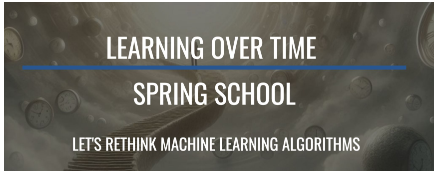
UNIVERSITÀ DI SIENA 1240

LOT Spring School

https://cai.diism.unisi.it/school



Collectionless AI: The World of NARNIAN

Inspired by **Collectionless AI**
https://cai.diism.unisi.it

Stefano Melacci
DIISM, University of Siena
stefano.melacci@unisi.it

March 24, 2025

It's a world of Interactions, and Time does matter!

Progressive development

Deepening knowledge

Interacting with a teacher

Taking exams

Interacting with other people

Effective interactions

Overview

https://cai.diism.unisi.it

1. **Learning Over Time**
   - Main challenges
   - Models & learning algorithms of the demos (see below)

2. **Collectionless AI**
   - Motivations & perspective

3. **Collectionless AI: The NARNIAN Project**
   - (a) What is NARNIAN
   - (b) Live demos of 3 different NARNIAN environments (sandboxes) + brief tutorial on how to set up and use NARNIAN

Collectionless AI Team

Jul 20, 2024 · 1 min read · People

Stefano Melacci

Associate Professor, University of Siena

Jul 21, 2024 · 1 min read · People

Marco Gori

Full Professor, University of Siena

Jul 22, 2024 · 1 min read · People

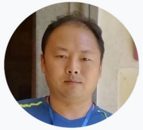
Matteo Tiezzi

PostDoctoral Researcher, Italian Institute of Technology

Jul 26, 2024 · 1 min read · People

Christian Di Maio

PhD Student, University of Pisa



Jul 27, 2024 · 1 min read · People

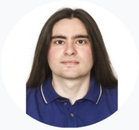
Jinwei Zhao

Associate Professor, Xi'an University of Technology, China

Jul 28, 2024 · 1 min read · People

Marco Lippi

Associate Professor, University of Florence



Jul 23, 2024 · 1 min read · People

Alessandro Betti

Assistant Professor, Scuola IMT Alti Studi Lucca

Jul 24, 2024 · 1 min read · People

Michele Casoni

PhD Student, University of Siena

Jul 25, 2024 · 1 min read · People

Tommaso Guidi

PhD Student, Università degli Studi di Firenze

Jul 29, 2024 · 1 min read · People

Luca Salvatore Lorello

PhD Student, University of Pisa - University of Modena & Reggio Emilia

Jul 30, 2024 · 1 min read · People

Stefan Knerr

Serial Entrepreneur

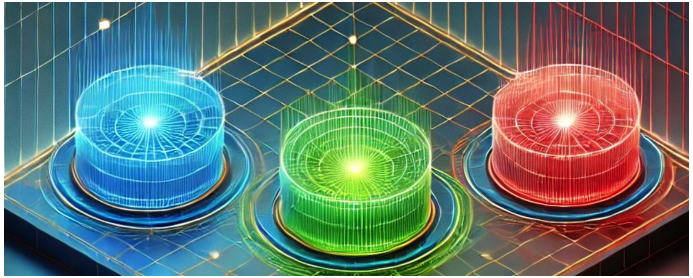
1. Learning Over Time

…more problems than solutions

Challenges

Three axes in learning over time with neural networks

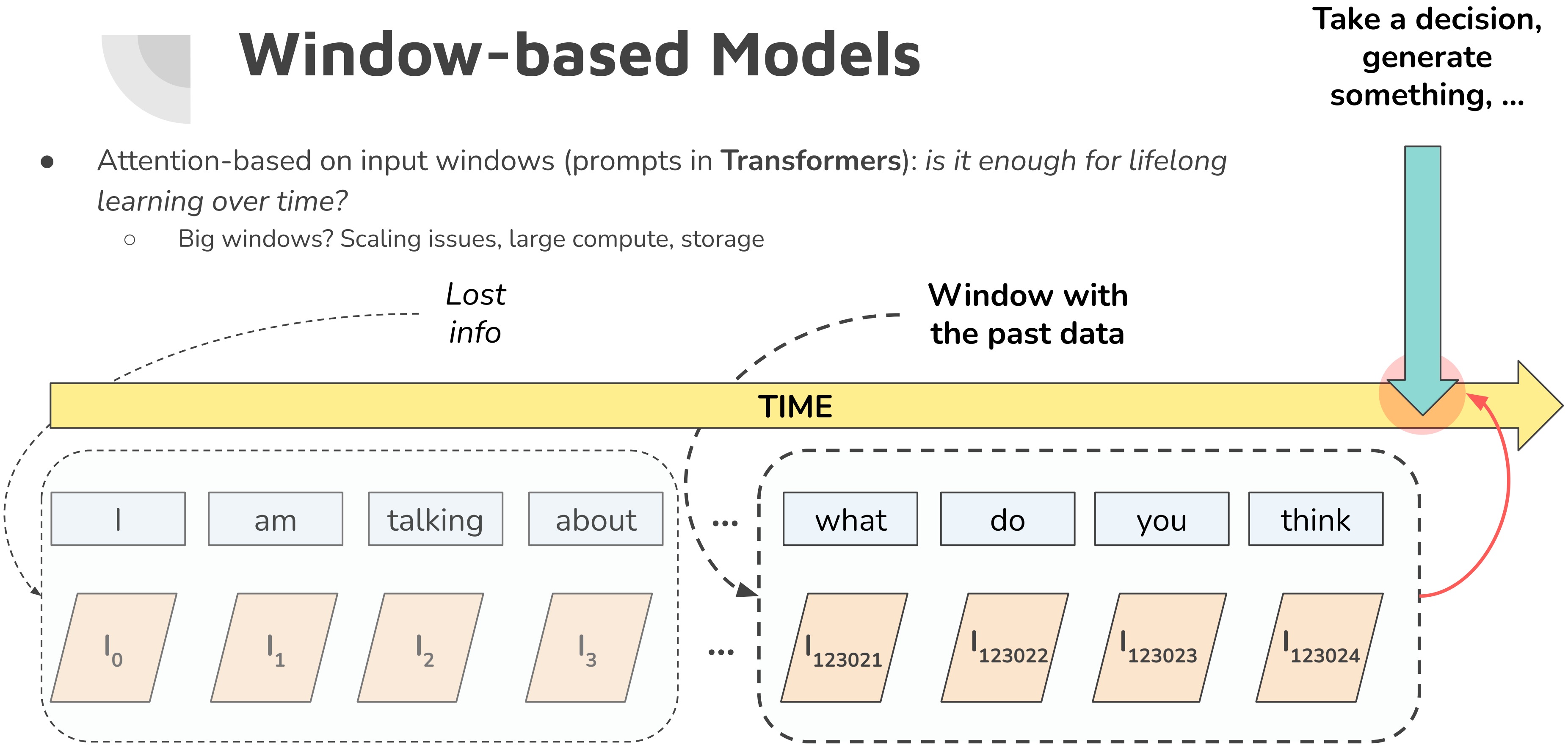
- **A. Learning strategies**
  - Learn in a "local" manner, without requiring the whole "past"
  - Define valid learning dynamics "over time"
- **B. Neural architectures**
  - Avoid forgetting the learned concepts
  - Be plastic enough to learn on-the-fly, still being able to generalize
- **C. Out-of-network Knowledge (*not covered here*)**
  - Exploit information not stored in the weights of the network
  - Example: symbolic knowledge bases (different from collections of examples of perceptual stimuli)
  - Example: rules, facts, ...



A. Learning strategies B. Neural architectures C. Out-of-network Knowledge

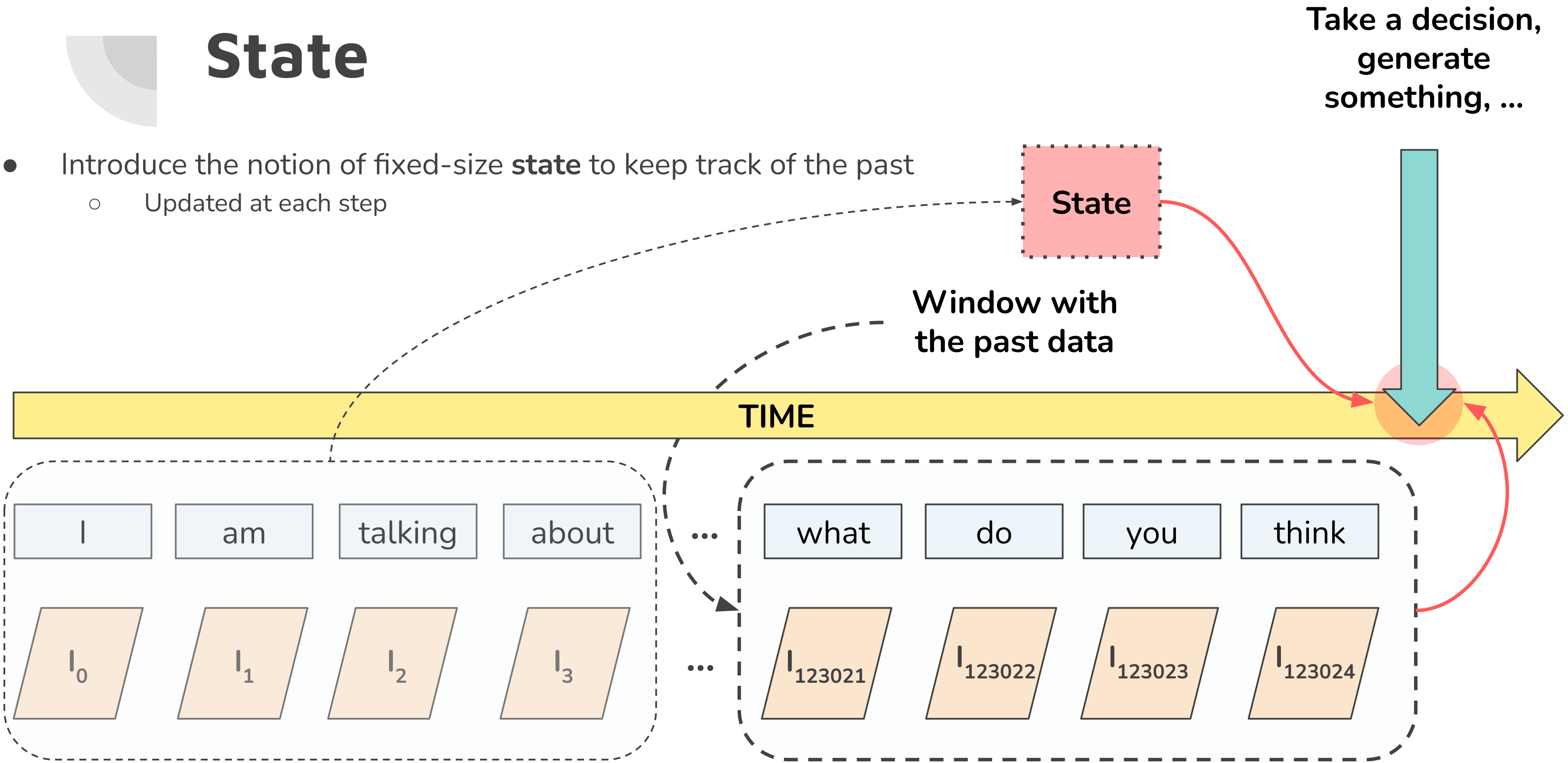
Window-based Models

- Attention-based on input windows (prompts in **Transformers**): *is it enough for lifelong learning over time?*
    - Big windows? Scaling issues, large compute, storage



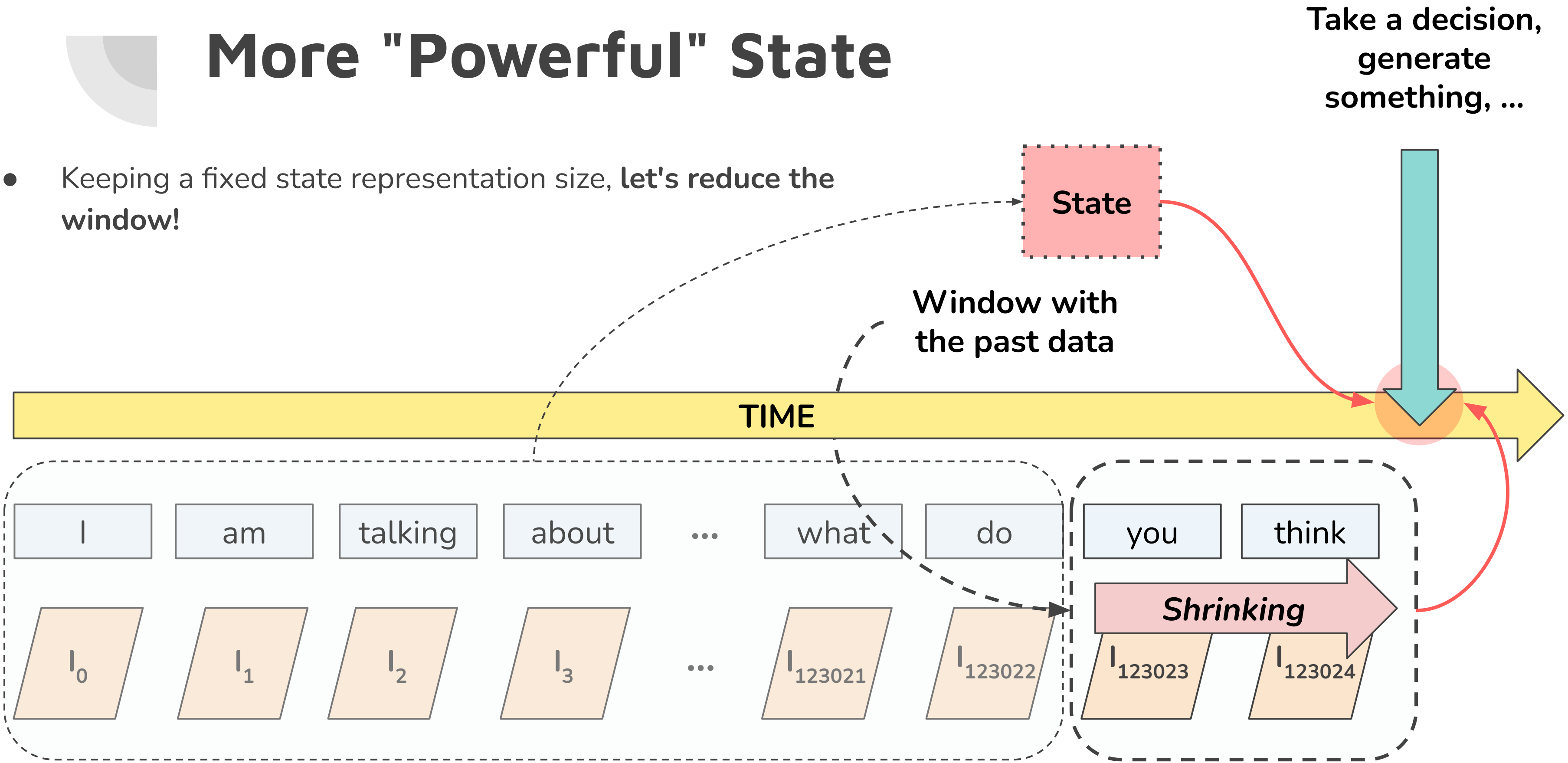
State

- Introduce the notion of fixed-size **state** to keep track of the past
  - Updated at each step



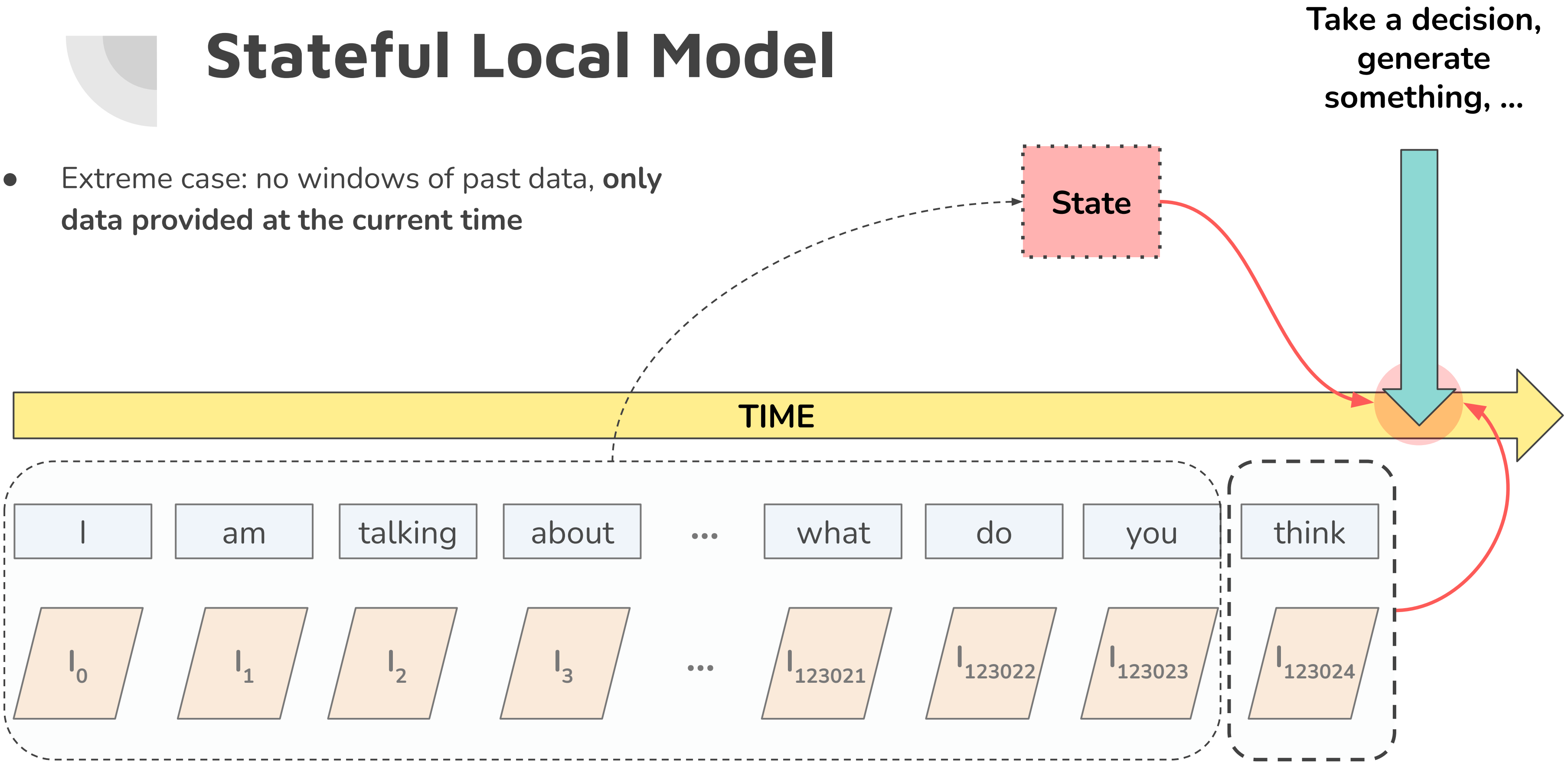
More "Powerful" State

- Keeping a fixed state representation size, **let's reduce the window!**



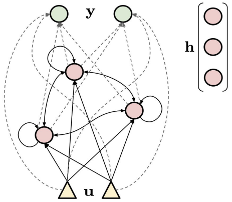
Stateful Local Model

- Extreme case: no windows of past data, **only data provided at the current time**

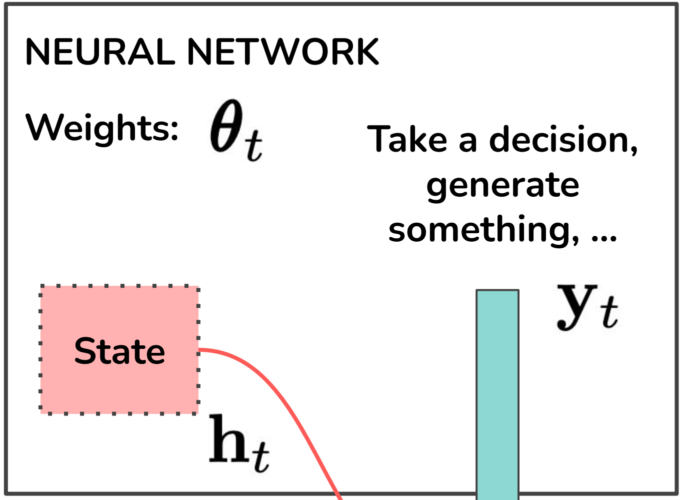


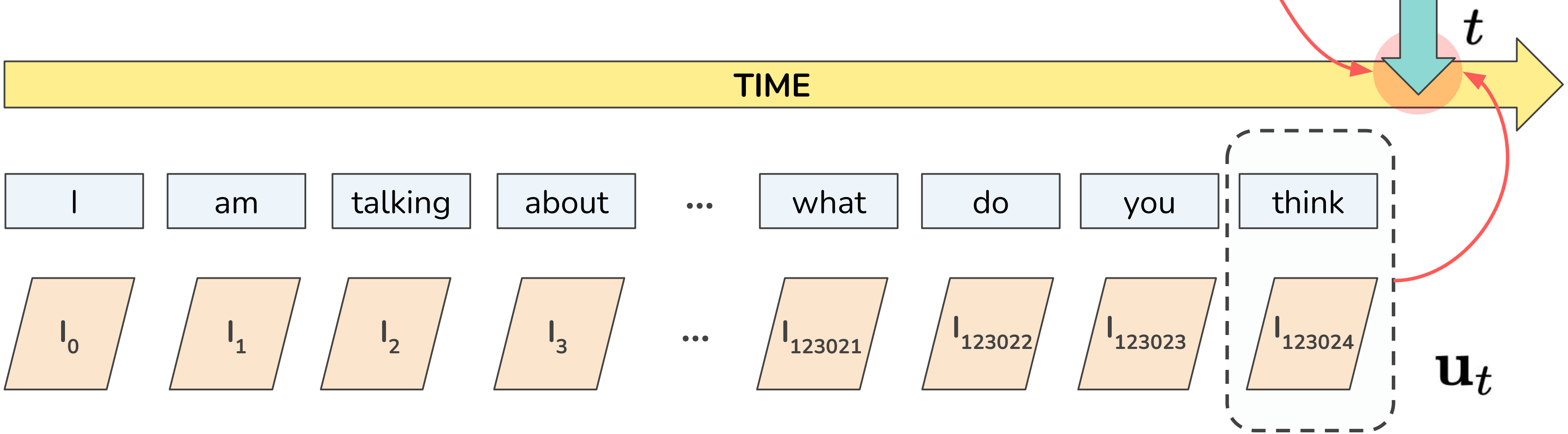
Notation: here and in the rest of this presentation, using the **subscript *t*** is a shorthand notation for ***(t)***, e.g.,, $y_t := y(t)$, *while* ***subscript k***, is used to indicate a discrete step index

Neural Networks

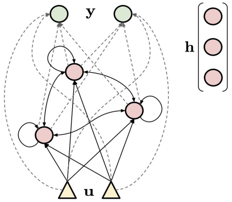


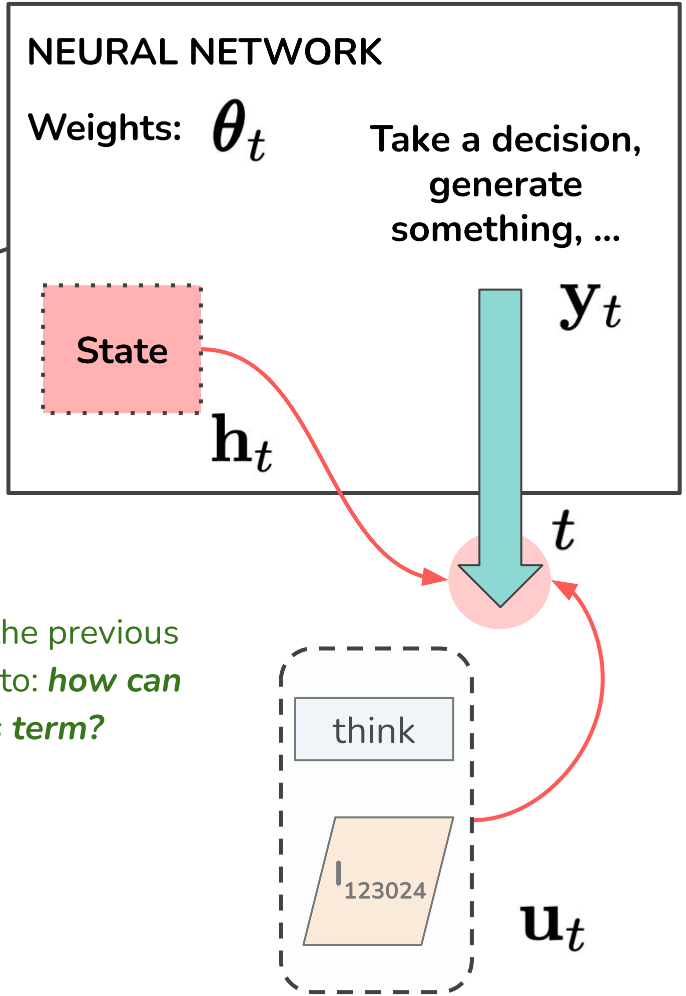
- Consider the case of Neural Nets, with **weights** and **biases** in θ_t
- **What do we need to learn them?**
  - A window of past states? We are back to the original problem (e.g., **BackPropagation Through Time** to train Recurrent Neural Nets)





State-Space Model





Let's "connect" these symbols

NEURAL NETWORK

$$\dot{\mathbf{h}}_t = f^{\mathbf{h}}(\mathbf{u}_t, \mathbf{h}_t, \boldsymbol{\theta}_t^{\mathbf{h}})$$

$$\mathbf{y}_t = f^{\mathbf{y}}(\mathbf{u}_t, \mathbf{h}_t, \boldsymbol{\theta}_t^{\mathbf{y}})$$

How does the model evolve?

(recall that $\boldsymbol{\theta}_t$ is the set of all the weights)

Update Step

$$\mathbf{h}_{t+\tau} = \mathbf{h}_t + \tau \dot{\mathbf{h}}_t$$

$$\boldsymbol{\theta}_{t+\tau} = \boldsymbol{\theta}_t + \tau \dot{\boldsymbol{\theta}}_t$$

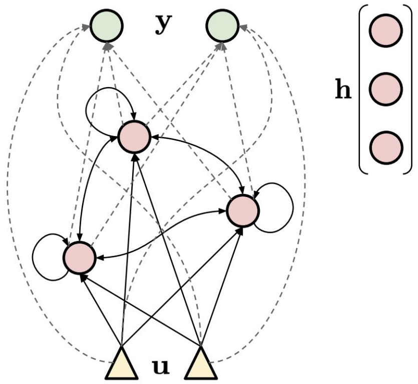
The question of the previous slide boils down to: ***how can we compute this term?***

Learning Strategies

We will consider two answers to the previous question

- "Vanilla" Gradient Descent (GD)
  - Gradient of a loss function
  - Watch out! **Not "stochastic" GD**, the data is provided over time and it is processed in the order in which it is streamed
- Hamiltonian Learning (HL)
  - The loss function is wrapped into an **optimal control** problem
  - **Fully local learning**: no layer-wise sequential operations both in inference and learning

Disclaimer: Here and in the following, I am not going into the specific differences between the continuous and discrete formulation



NEURAL NETWORK

$$\dot{\mathbf{h}}_t = f^{\mathbf{h}}(\mathbf{u}_t, \mathbf{h}_t, \boldsymbol{\theta}^{\mathbf{h}}_t)$$
$$\mathbf{y}_t = f^{\mathbf{y}}(\mathbf{u}_t, \mathbf{h}_t, \boldsymbol{\theta}^{\mathbf{y}}_t)$$

Neural Architectures

In the demo section, we will focus on **three neural models**, all instances of what has been presented so far

(will go back to them)

1. Convolutional Network

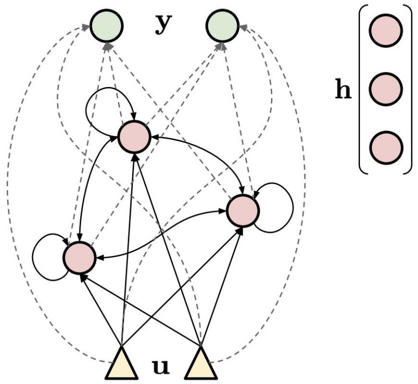
$$\mathbf{h}_k = \mathrm{CNN}(\mathbf{u}_k, \cdot, \theta^{\mathbf{h}})$$
$$\mathbf{y}_k = C\mathbf{h}_k$$

2. Autoregressive-like Network

$$\mathbf{h}_k = \sigma(A\mathbf{h}_{k-1} + B\mathbf{u}_k)$$
$$\mathbf{y}_k = C\mathbf{h}_k$$
$$\text{with } \mathbf{u}_0 = \mathbf{0} \text{ and } \mathbf{u}_{k>0} = \mathbf{y}_{k-1}$$

3. Continuous-Time Linear State Space Model

$$\dot{\mathbf{h}}_t = A\mathbf{h}_t + B\mathbf{u}_t$$
$$\mathbf{y}_t = C\mathbf{h}_t$$
$$\text{with } \mathbf{h}_0 = \mathbf{0} \text{ and } \mathbf{u}_{t>0} = \mathbf{0}$$



NEURAL NETWORK

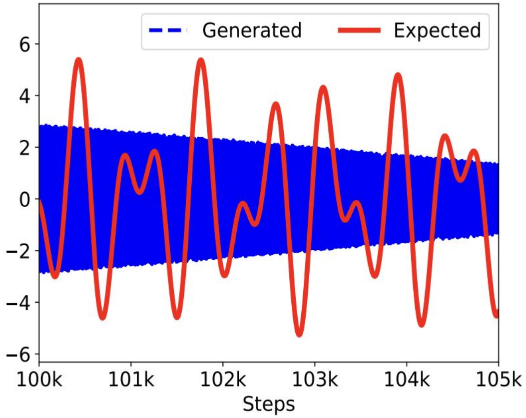
$$\dot{\mathbf{h}}_t = f^{\mathbf{h}}(\mathbf{u}_t, \mathbf{h}_t, \boldsymbol{\theta}_t^{\mathbf{h}})$$
$$\mathbf{y}_t = f^{\mathbf{y}}(\mathbf{u}_t, \mathbf{h}_t, \boldsymbol{\theta}_t^{\mathbf{y}})$$

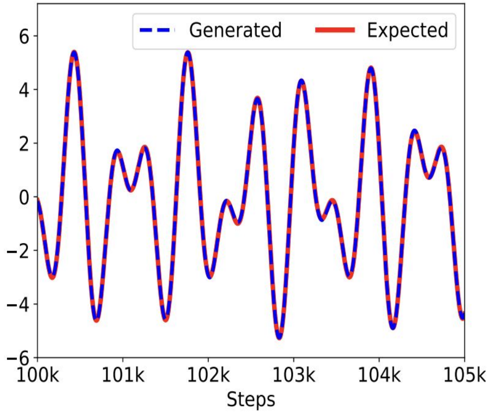
Perpetual Generation

y_t with t →∞, no-learning

- We need models that can autonomously generate data for a very long time (talking, video data, signals, ...)

It might also explode





As it is (no control on *A*)

3. Continuous-Time Linear State Space Model

$$\dot{\mathbf{h}}_t = A\mathbf{h}_t + B\mathbf{u}_t$$

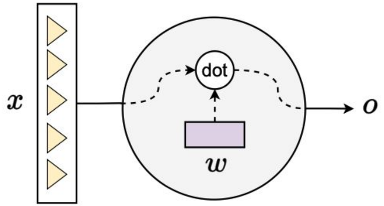
$$\mathbf{y}_t = C\mathbf{h}_t$$

$$\text{with } \mathbf{h}_0 = \mathbf{0} \text{ and } \mathbf{u}_{t>0} = \mathbf{0}$$

Controlling the spectrum of *A* (set of the eigenvalues): zero imaginary part

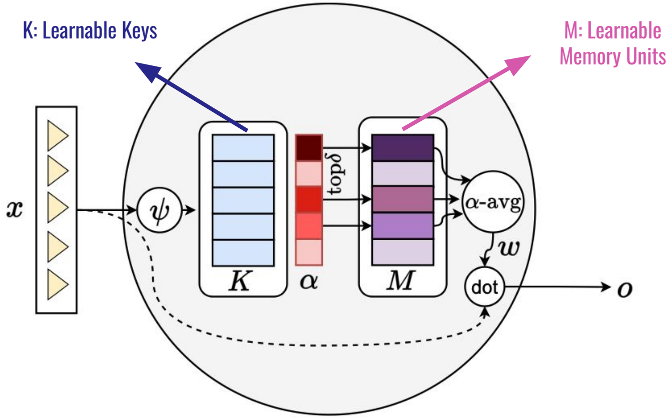
Neuron Model

- We reconsider the **neuron model**, to make it easier to find a good trade-off between plasticity and stability (no replays), reducing **forgetting**
- **Continual Neural Units (CNUs)** - generalization of vanilla neurons
- In the implementations of the output functions which yields y_t in **cases *1*** and ***3*** of the presented neural architectures, we will consider CNUs



Neuron Output

$$f(x, w) = w'x$$



Neuron Output

$$f(x, K, M) = \hat{w}(x, K, M)'x$$

2. Collectionless AI

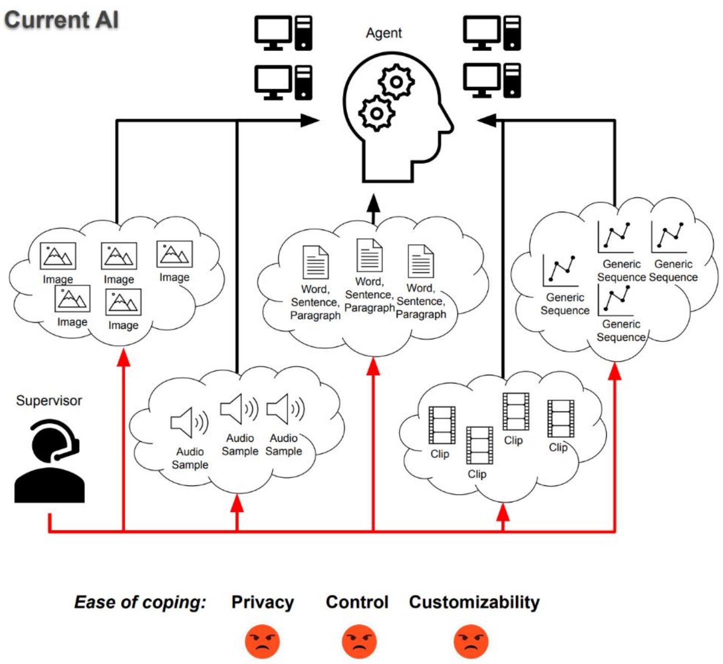
Beyond Offline Learning from Datasets

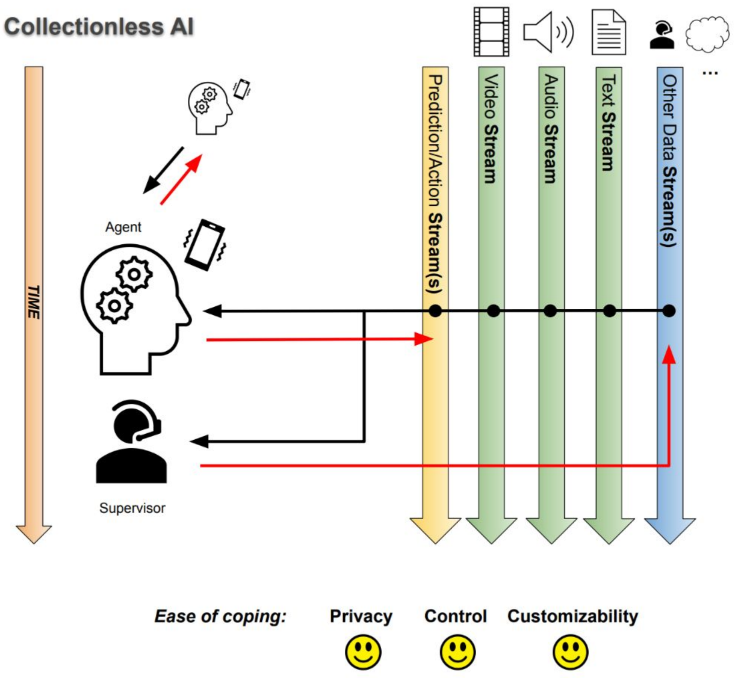
Collectionless AI: Beyond Mainstream AI

*"Learning from huge data collections introduces risks related to **data centralization, privacy, energy efficiency, limited customizability, and control**. Collectionless AI focuses on the perspective in which artificial agents are **progressively developed over time by online learning** from potentially **lifelong streams** of sensory data. This is achieved without storing the sensory information and without building datasets for offline-learning purposes while pushing towards **interactions with the environment**, including **humans and other artificial agents**."*

Marco Gori & **Stefano Melacci**, from the Research Summary at the Montreal AI Ethics Institute (MAIEI)

Streams, Interactions, Learning Over Time





Decentralized Real-time Computations

Huge server(s), cloud computing

LLM-based Agents

A lot of work in progress to make inference more scalable, but still trained on these servers

vs.

On-the-edge devices

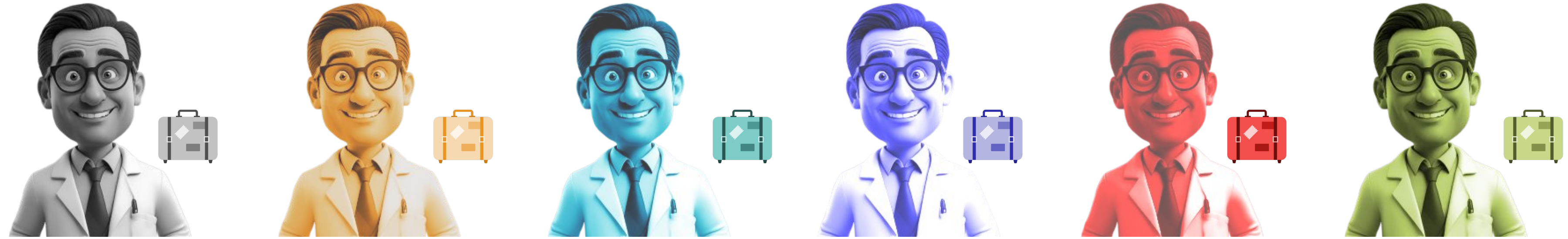
Collectionless AI Agents

Privacy! It promotes **efficient ways of transferring information by a few interactions**, instead of relying on collections of (sometimes redundant) data

Today's ML/AI: Keywords, Sub-Topics, ...

Deep Learning	Reinforcement Learning	Active Learning	(Large) Language Models	Large Vision Models	Object Detection

... ...



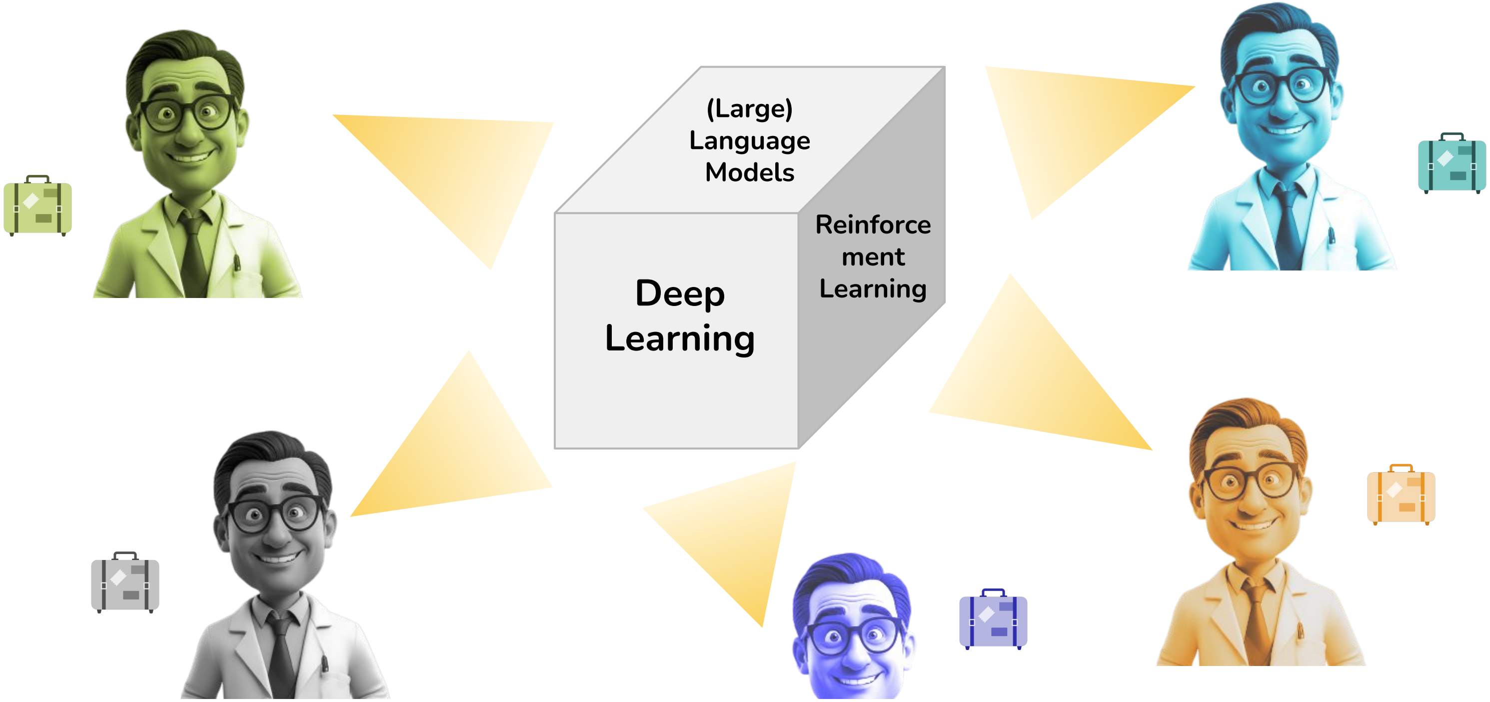
Today's ML/AI: Static Benchmarks

Deep Learning

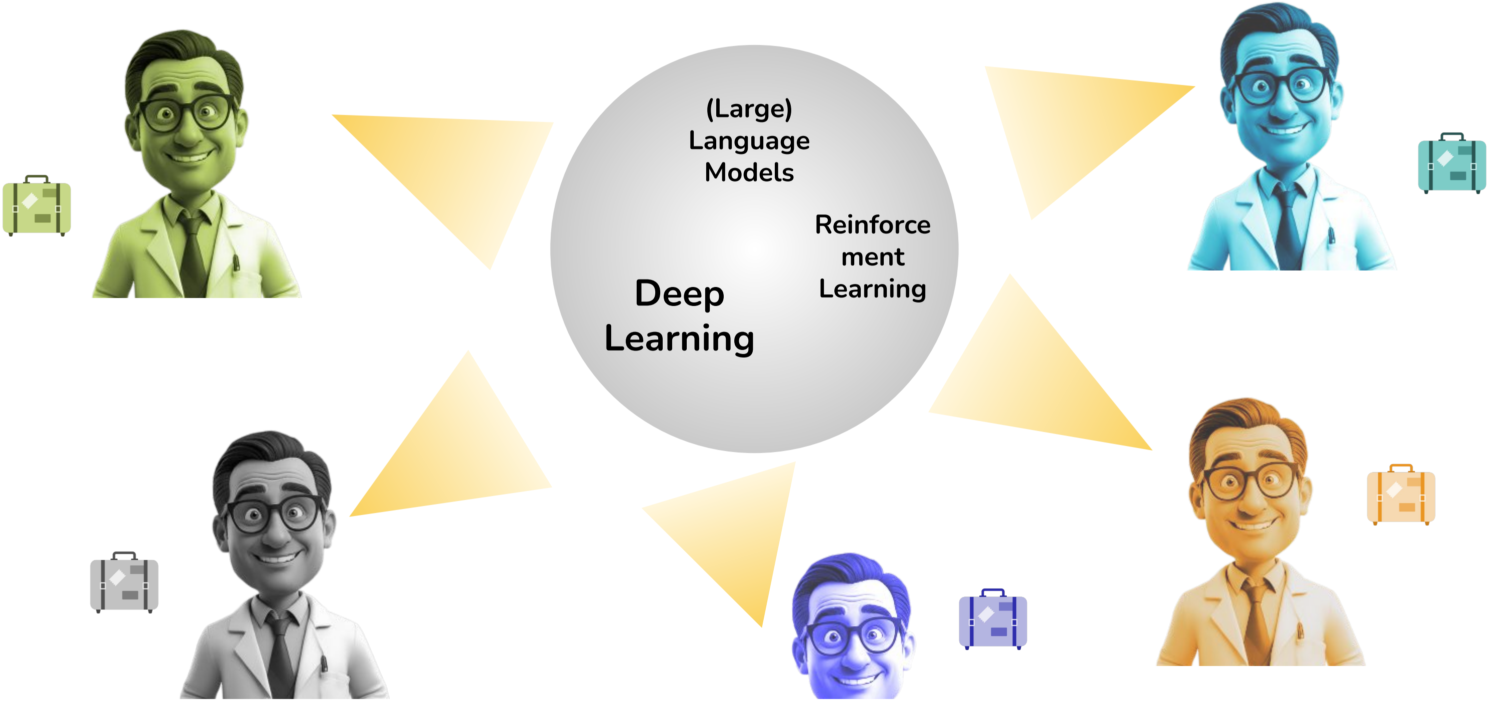
Object Detection

(Big) Datasets & Static Benchmarks

Different Facets of the Same Hypercube

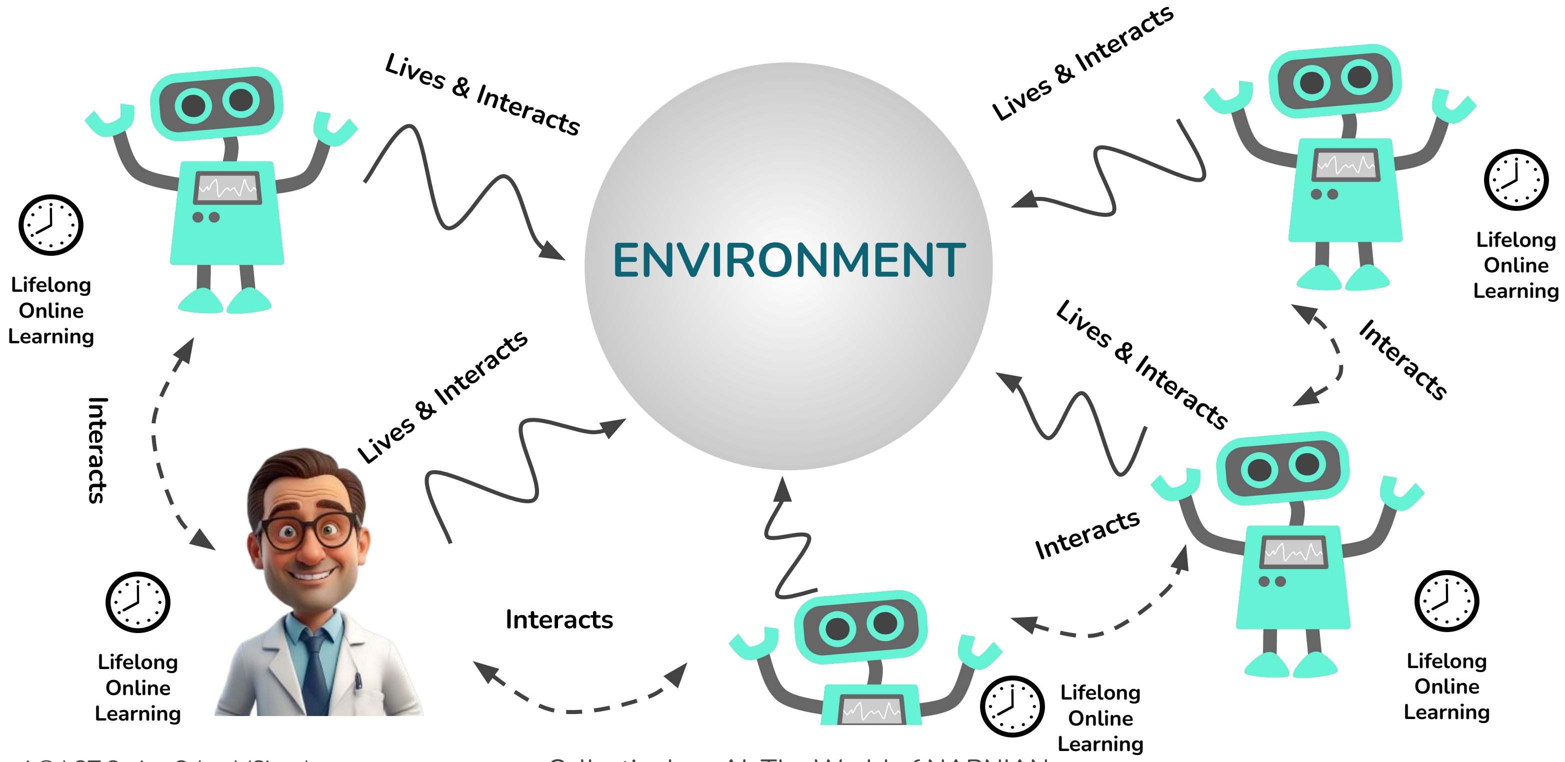


Step 1/3: Relax the Boundaries

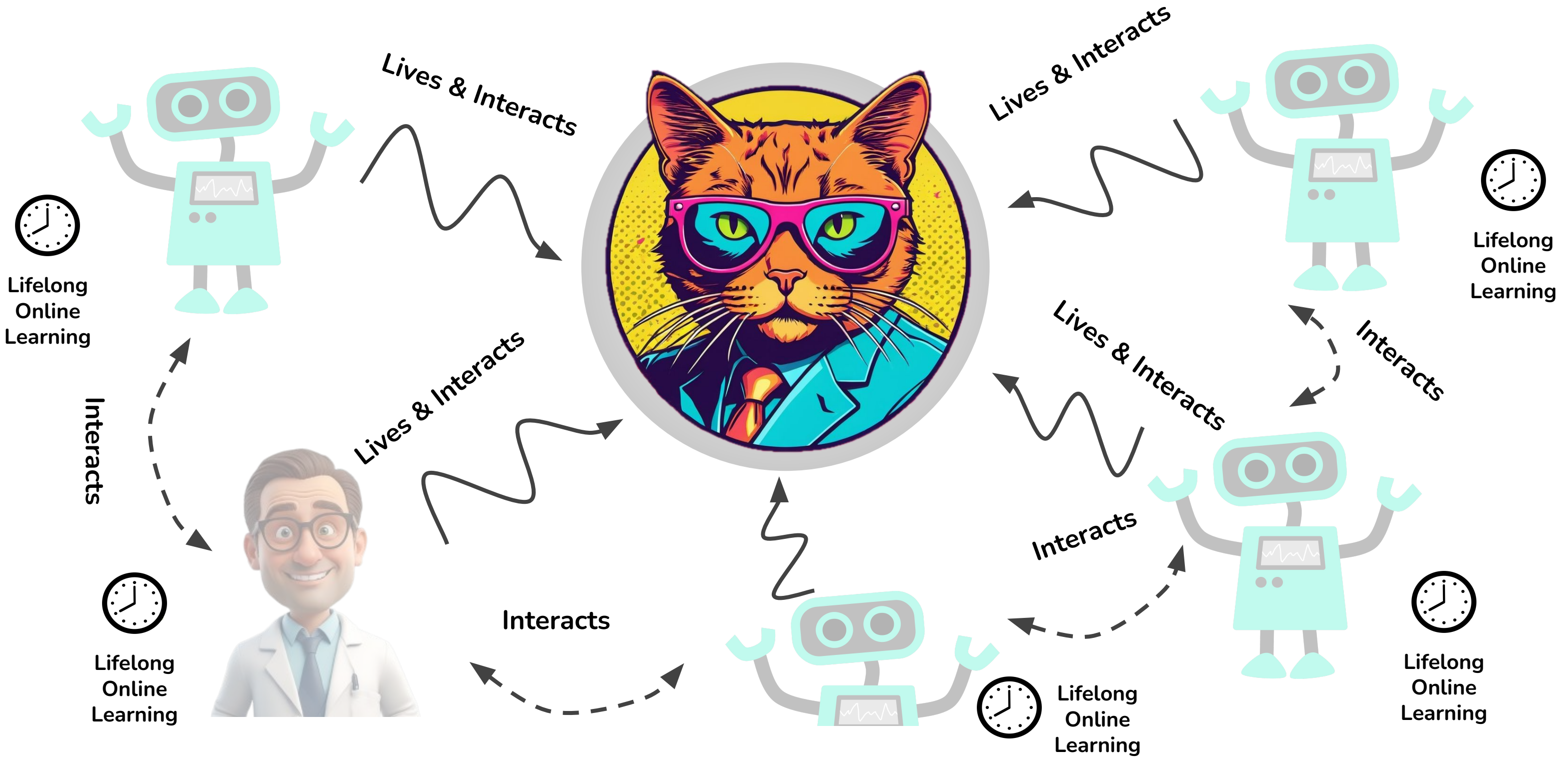


Step 2/3: Embrace Time

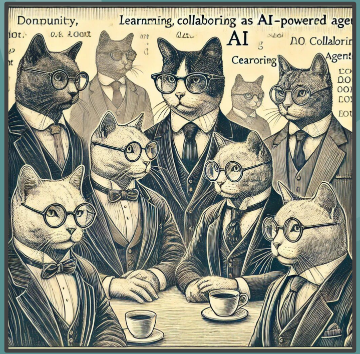
Step 3/3: Promote Interactions



<Welcome to Collectionless AI>

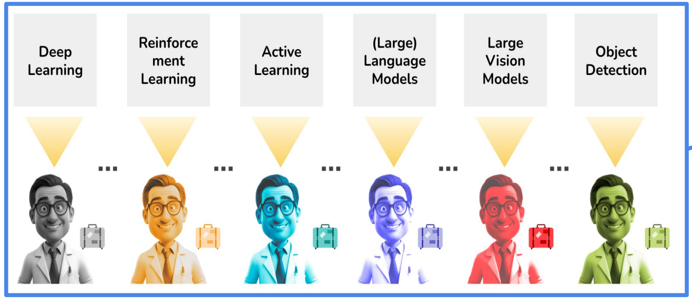


3a. Toward building Collectionless AI agents: NARNIAN



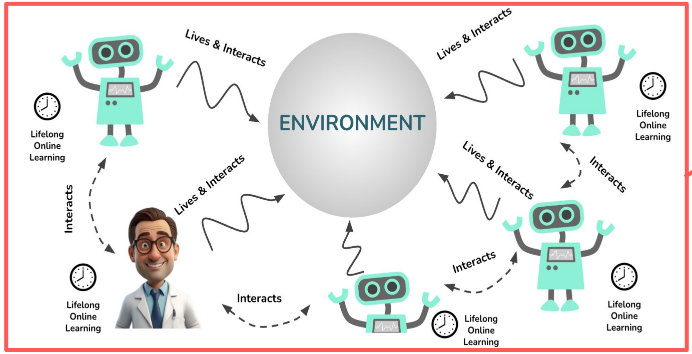
The NARNIAN Project

Today's AI



Static benchmarks with offline experiments for improving existing technologies (**today**)

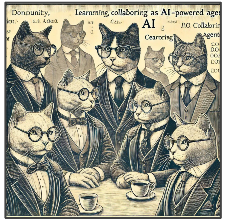
Collectionless AI



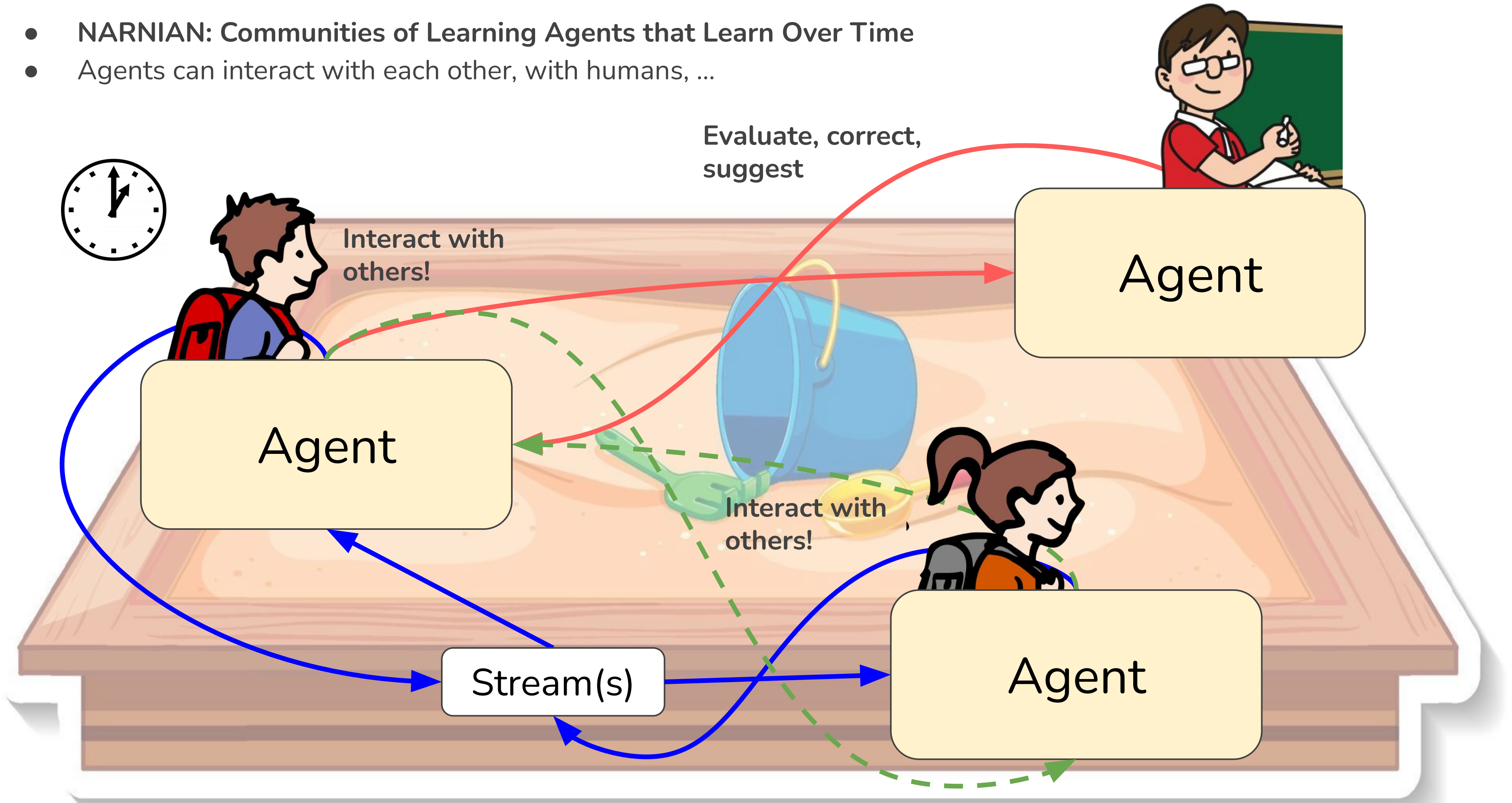
"Communities of Agents that Learn Over Time"

NARNIAN

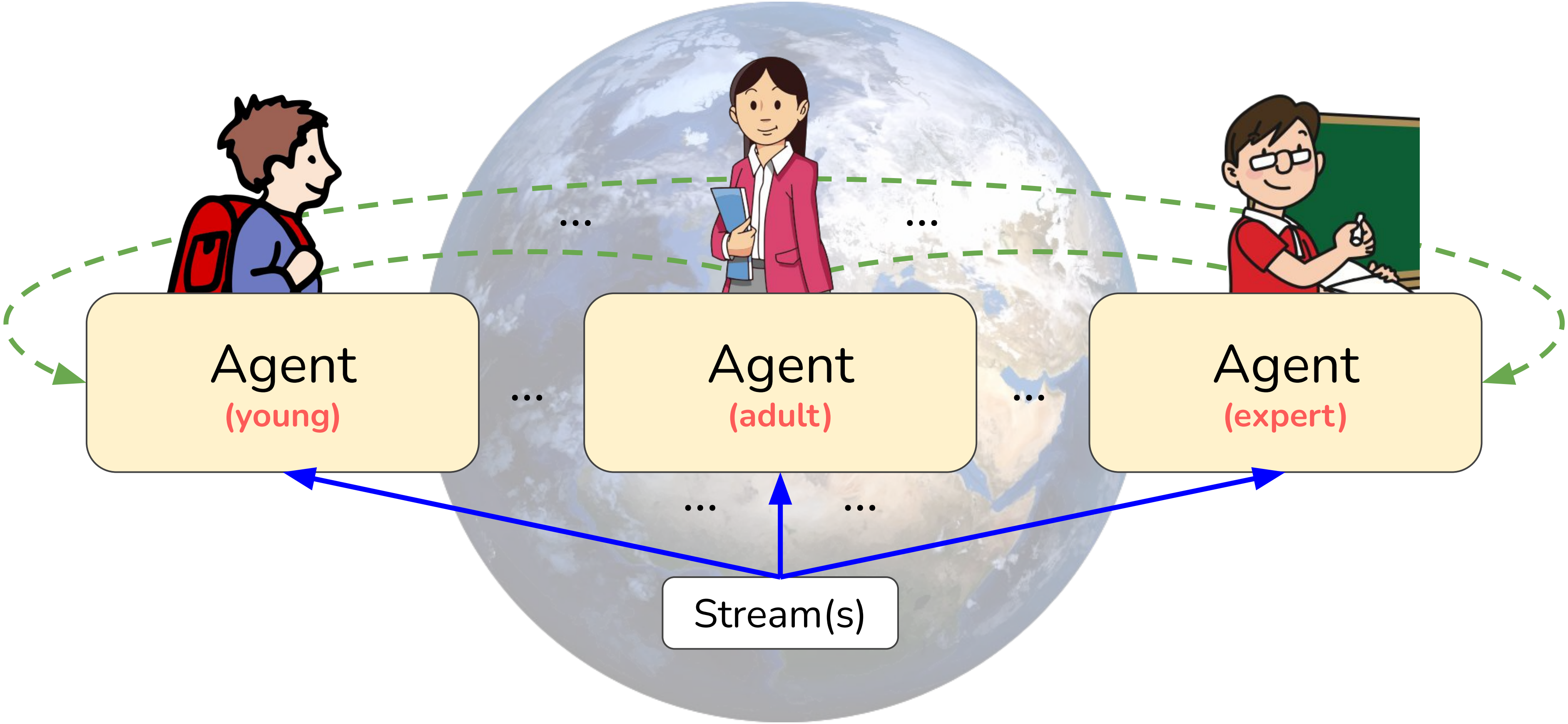
Platform modeling dynamic environments where to study and progressively develop novel types of agents (**tomorrow**)



- **NARNIAN: Communities of Learning Agents that Learn Over Time**
- Agents can interact with each other, with humans, ...



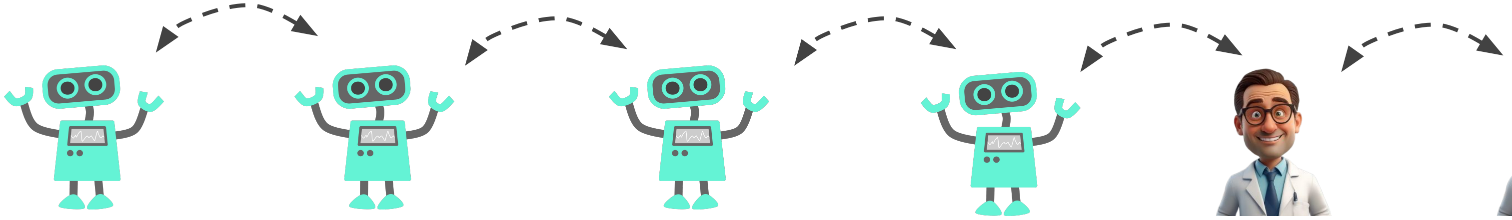
- Agents "**grow**" by means of interactions, gaining different levels of expertise ("***the pupil has become the master***"), and being **transferred to the real-world... or the real-world is already part of NARNIAN?**

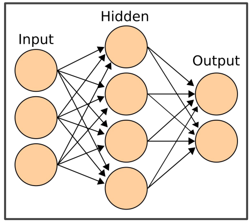


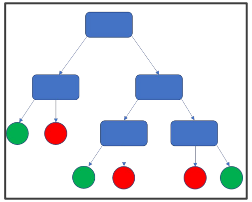
NARNIAN: Communities of YOUR Agents

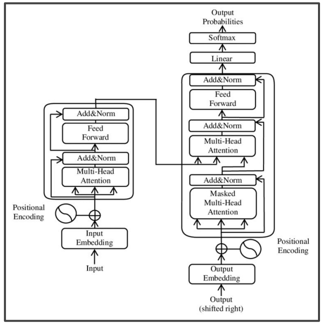
NARNIAN allows the **cross-over of multiple types of models**, joined by the perspective of **learning over time** and **interacting: never seen before scenario?**

Interacts Interacts Interacts Interacts Interacts









ChatGPT, Gemini, Llama, ... every "Large Something Model"

You

Somebody Else

Learn how to interact! (RL or other ideas)

...pretrained, fine-tuned, WHATEVER YOU LIKE + LEARNING THE WAY YOU LIKE!

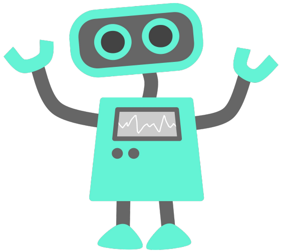
Components of NARNIAN

Main ingredients:

1. **Streams**
   a. Raw Information
   b. Descriptors
2. **Agent(s)**
   a. Behavior
   b. Model
3. **Environment**
   a. Behavior

1. Streams

Data streamed by **known sources** or **generated by agents**



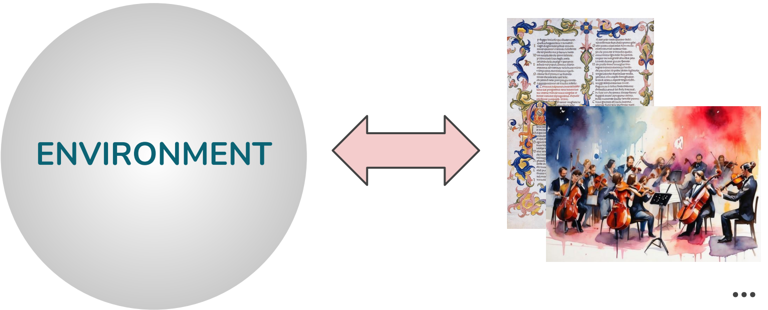
2. Agent

Citizen of NARNIAN

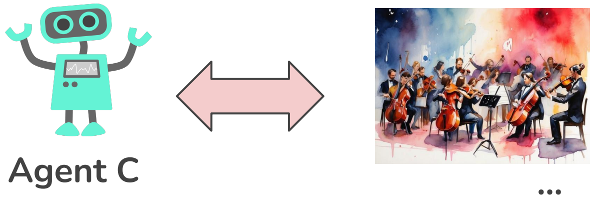
3. Environment

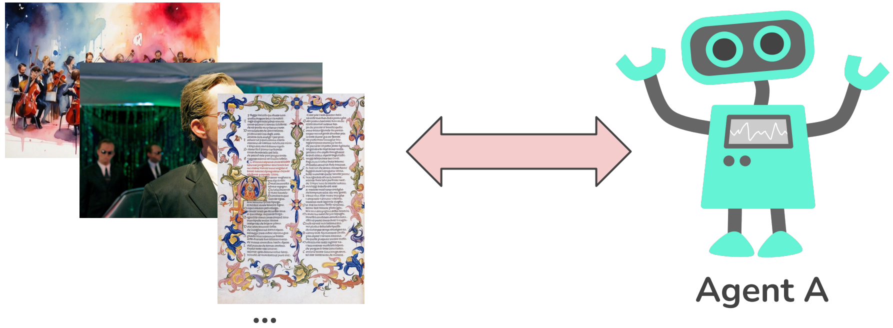
A special "agent" that manages the world

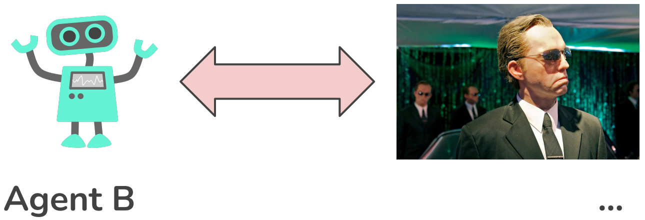
Streams



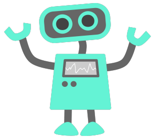
- Each stream is composed **2 elements**:
  - the main **signal** with "raw" information (function of time)
  - another signal that describes such raw information: **descriptor**
    - E.g., video (**signal**), actions in each frame (**descriptor**)
    - E.g., images (**signal**), categories of each image (**descriptor**)
    - E.g., data from a sensor (**signal**), normal vs. anomaly label (**descriptor**)
    - ...





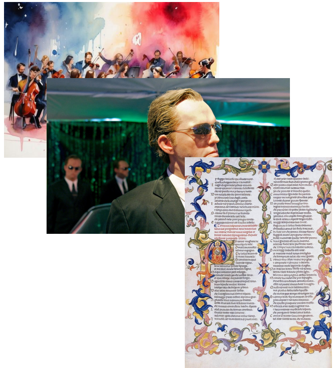


Agent

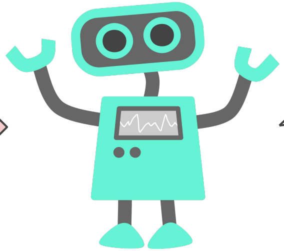


do this,
get this,
learn this,
share those,
...

- How does the agent handle interactions with others?
- What can the agent do or not do at each time instant?



...



Behavior

Model

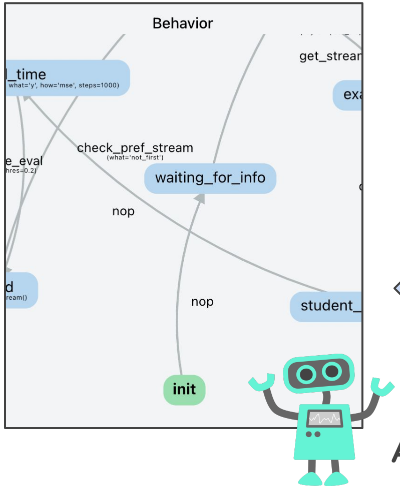
generate,
describe (a.k.a. *predict*)
learn to generate,
learn to describe

- How does the agent generate data or descriptions?
- How does the agent handle the learning-over-time process?

Agent: Behavior

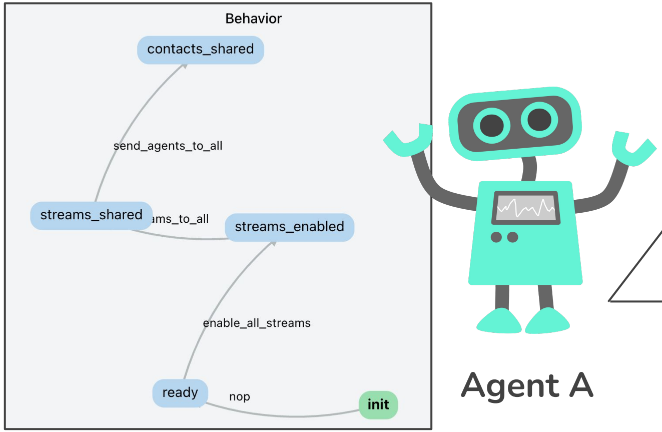
- In the most naive way, it can taught as a **Finite State Machine (FSM)**
- FSMs of different agents **interact with each other**

- How does the agent handle interactions with others?
- What can the agent do or not do at each time instant?

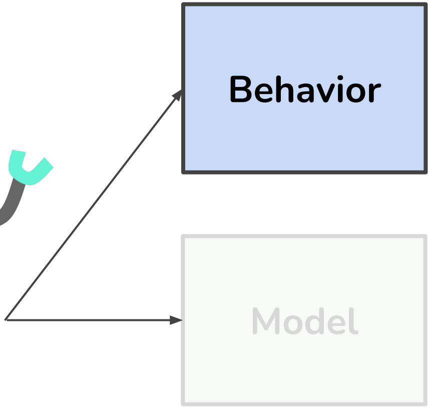


Agent B

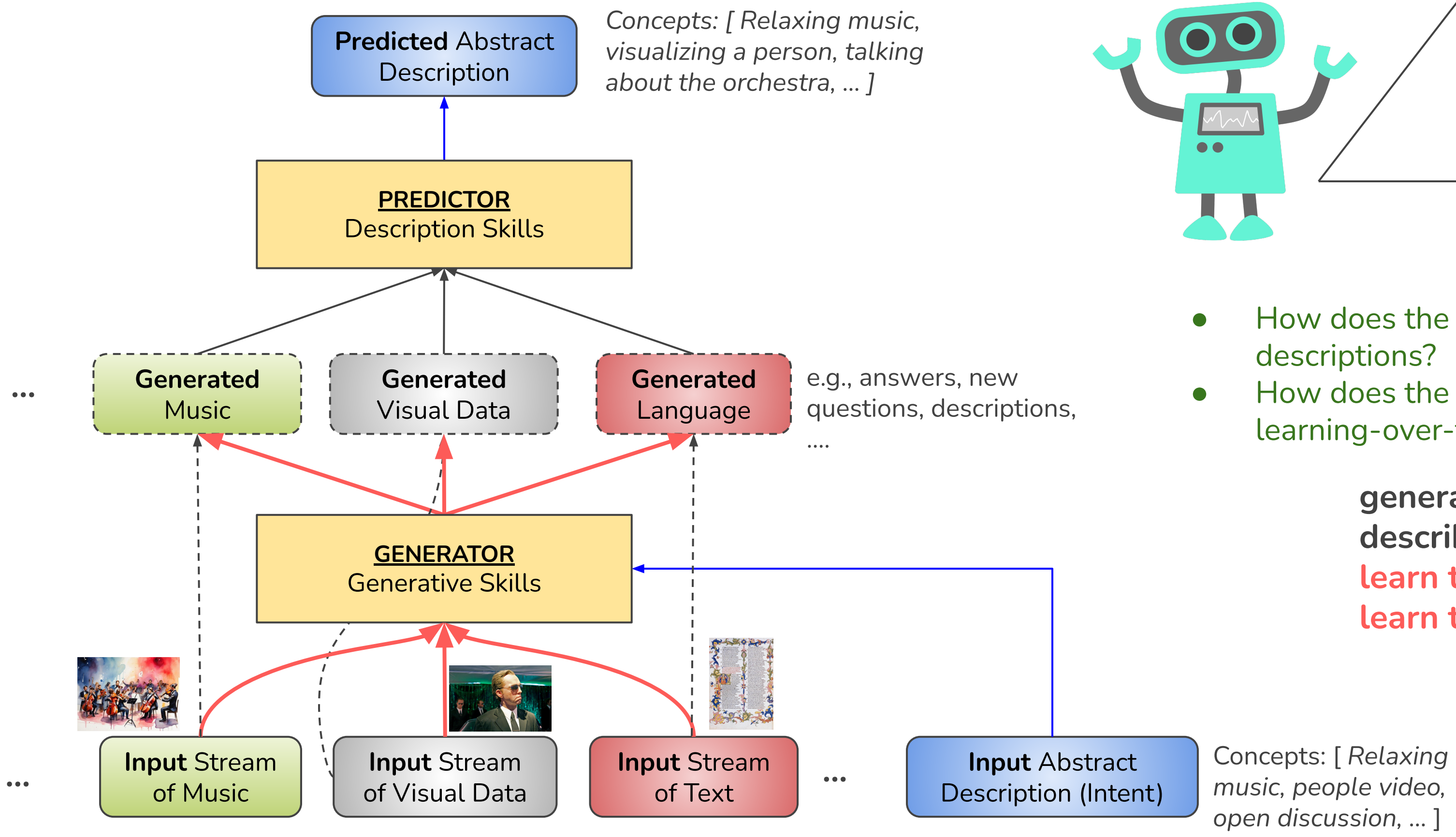
do this,
get this,
learn this,
share those,
...

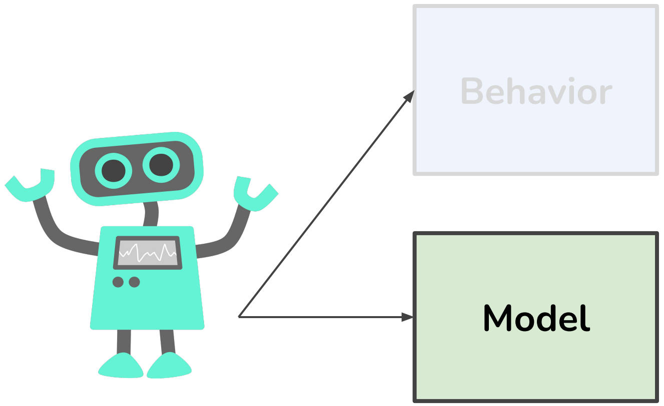


Agent A



Agent: Model

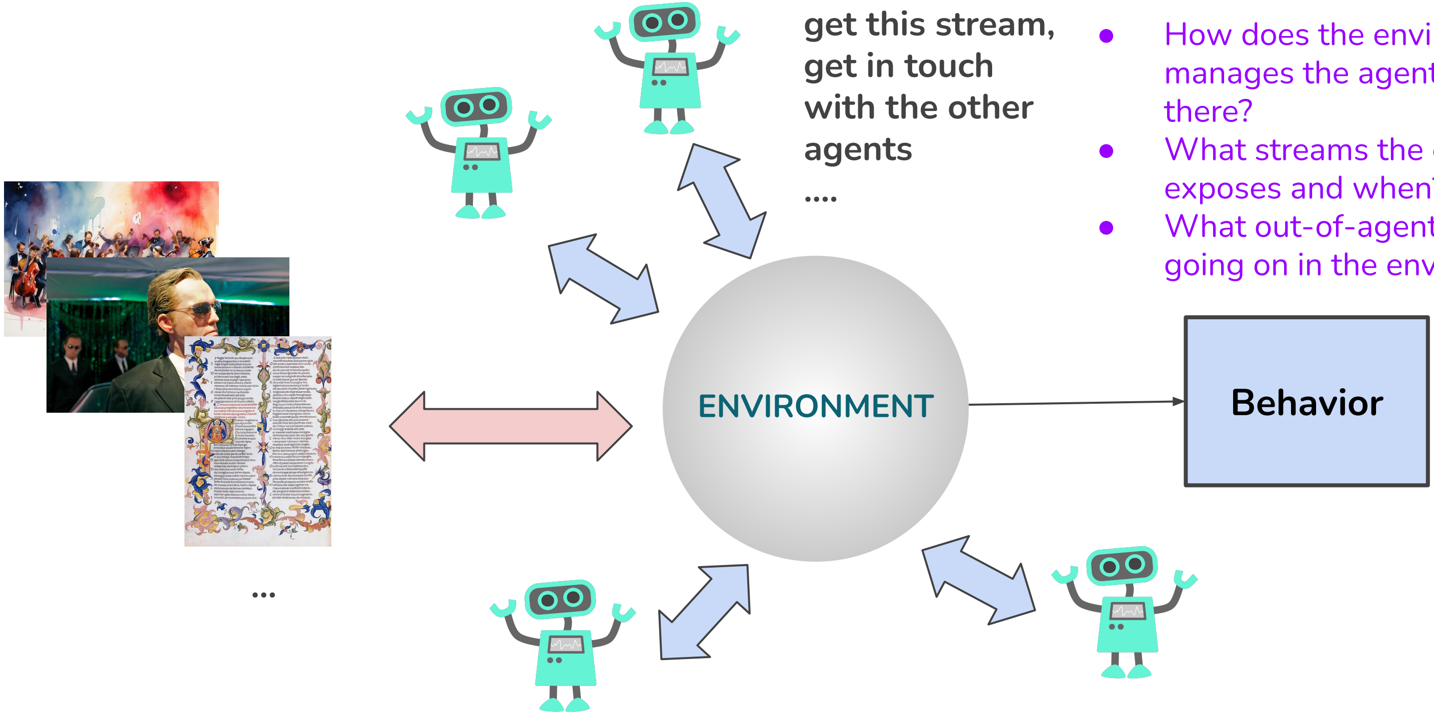




- How does the agent generate data or descriptions?
- How does the agent handle the learning-over-time process?

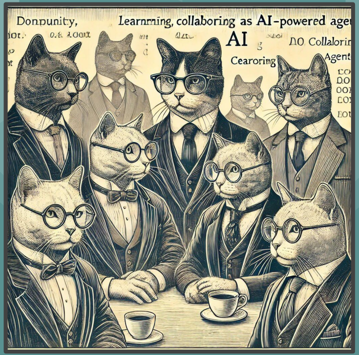
generate,
describe (a.k.a. *predict*)
learn to generate,
learn to describe

Environment: A Special "Agent"



- How does the environment manages the agents that live there?
- What streams the environment exposes and when?
- What out-of-agent dynamics are going on in the environment?

3b. Demos of NARNIAN Sandboxes



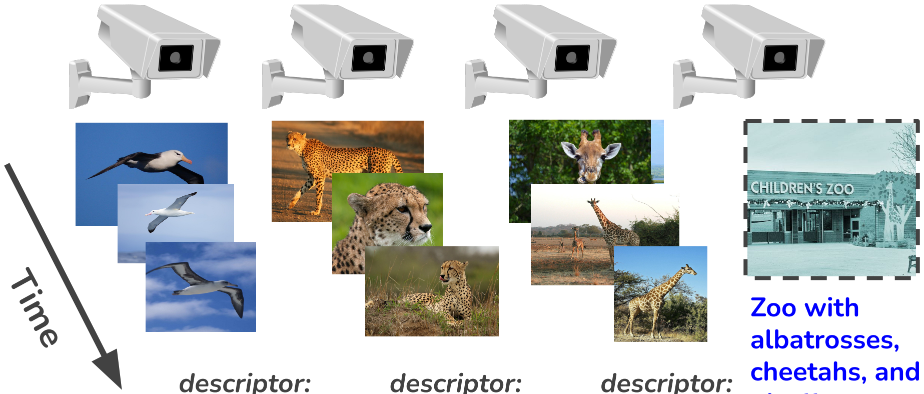
The NARNIAN Project

Challenge: Learning over time (class incremental image classification), no data storage

School of Animals

- **Dr. Green** teaches **Mario** and **Luigi** about 3 animals, showing pictures of each of them, and then evaluates the recognition capabilities of the both the students by showing them all the animals in a zoo
- 1st lecture: albatrosses; 2nd: cheetahs; 3rd: giraffes

- The student who succeeds, will become a new teacher, and will help the other students improve



descriptor: "albatross"

descriptor: "cheetah"

descriptor: "giraffe"

Zoo with albatrosses, cheetahs, and giraffes

- **Gradient Descent (GD)**

MODEL

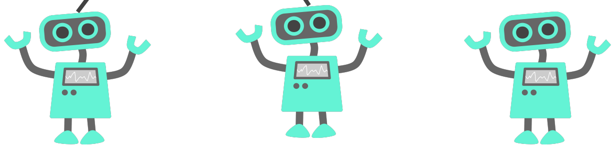
Neural Network: PREDICTOR
Description Skills

1. Convolutional Network

$$\mathbf{h}_k = \mathrm{CNN}(\mathbf{u}_k, \cdot, \theta^{\mathbf{h}})$$

$$\mathbf{y}_k = C\mathbf{h}_k$$

Watch out: Mario uses Continual Neural Units!



Mario (student)

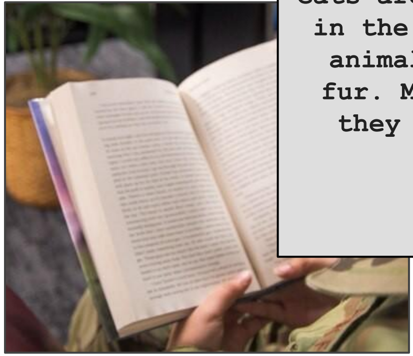
Luigi (student)

Dr. Green (teacher)

Challenge: Learning over time (next word prediction), **local in time** (no BackPropagation Through Time), no data storage

Cat Library

- **Dr. Green** prepares a book that talks about cats and asks **Mario** to memorize it
- **Mario** listens **Dr. Green** reading the book multiple times, and then tries to repeat it, word-by-word (learning embeddings as well)



```
Cats are one of the most popular pets
in the world. They are small, furry
animals with sharp claws and soft
fur. Many people love cats because
they are cute and independent...
...
...
```

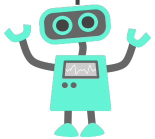
- **Gradient Descent (GD)**

Neural Network: GENERATOR
Generative Skills

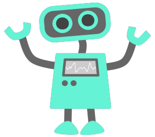
MODEL

2. Autoregressive-like Network

$$\mathbf{h}_k = \sigma(A\mathbf{h}_{k-1} + B\mathbf{u}_k)$$
$$\mathbf{y}_k = C\mathbf{h}_k$$
$$\text{with } \mathbf{u}_0 = \mathbf{0} \text{ and } \mathbf{u}_{k>0} = \mathbf{y}_{k-1}$$



Mario
(student)



Dr. Green
(teacher)

Challenge: Learning over time (input-free generation), **fully local** (no BackPropagation Through Time, parallel computation of gradients over the neurons), no data storage

Signal School

- **Dr. Green** shows **Mario** data coming from multiple sensors, consisting of scalar signals, paired with a multi-label descriptors
- **Mario** is asked to learn to re-generate each of the signals given an input a query descriptor

- **Dr. Green** evaluates how good he is, providing him a descriptor and checking the generated signal
- **Dr. Green** will also present **Mario** a query descriptor that he ***never saw before***, and Mario will have to generate a signal which is coherent with it

- Hamiltonian Learning (HL)

Neural Network: GENERATOR
Generative Skills

MODEL

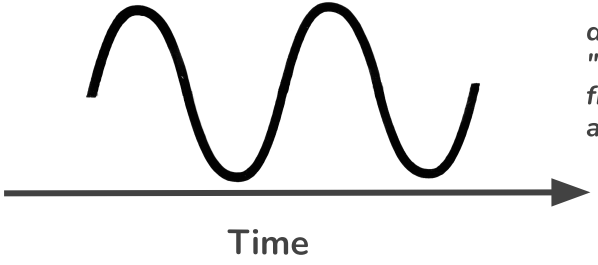
3. Continuous-Time Linear State Space Model

$$\dot{\mathbf{h}}_t = A\mathbf{h}_t + B\mathbf{u}_t$$

$$\mathbf{y}_t = C\mathbf{h}_t$$

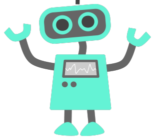
$$\text{with } \mathbf{h}_0 = \mathbf{0} \text{ and } \mathbf{u}_{t>0} = \mathbf{0}$$

*Watch out: Mario uses **Continual Neural Units!***

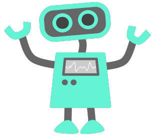


descriptor: "sinusoidal, low frequency, low amplitude"

Overall, we have 7 signals, with descriptors composed of combinations of attributes as the ones of this example (low/high frequency, low/high amplitude, ...)



Mario (student)



Dr. Green (teacher)

How To Use NARNIAN?

https://github.com/mela64/narnian

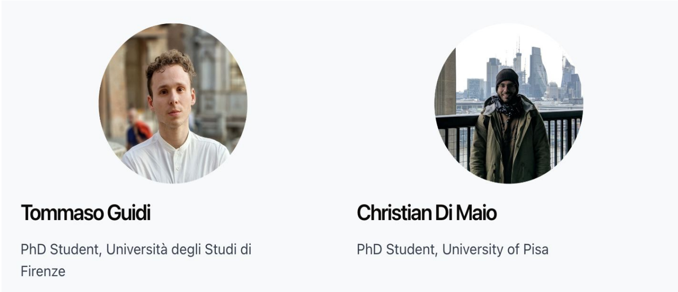
Setup your system:

- Get the code
- Follow the instructions in **README.md**

Learning NARNIAN:

- Check the notebook **sandbox_example_tutorial.ipynb** (**it runs on Colab as well**)
- ...or check the corresponding non-notebook example: **sandbox_example.py**
- Then, the three sandboxes of the previous slides are on the root of the code repository, easy to spot

Tommaso and Christian will show you how...



References

1. S. Melacci, A. Betti, M. Casoni, T. Guidi, M. Tiezzi, M. Gori. A Unified Framework for Neural Computation and Learning Over Time. arXiv:2409.12038, 2024
2. A. Betti, M. Gori. Nature-Inspired Local Propagation. NeurIPS 2024
3. M. Casoni, T. Guidi, M. Tiezzi, A. Betti, M. Gori, S. Melacci. Pitfalls in Processing Infinite-Length Sequences with Popular Approaches for Sequential Data. ANNPR 2024.
4. M. Tiezzi, M. Casoni, A. Betti, T. Guidi, M. Gori, S. Melacci.Back to Recurrent Processing at the Crossroad of Transformers and State Space Models
5. M. Tiezzi, M. Casoni, A. Betti, M. Gori, S. Melacci. State-Space Modeling in Long Sequence Processing: A Survey on Recurrence in the Transformer Era. arXiv:2406.09062, 2024
6. S. Guidi, A. Casoni, M. Betti, T. Melacci, M. Gori. Perpetual Generation: Online Learning of Linear State-Space Models from a Single Stream, Technical Report
7. M. Tiezzi, S. Marullo, F. Becattini, S. Melacci. Continual Neural Computation. ECML-PKDD 2024
8. M. Tiezzi, F. Becattini, S. Marullo, S. Melacci. Memory Head for Pre-Trained Backbones in Continual Learning. Conference on Lifelong Learning Agents (CoLLAs) 2024
9. C. Di Maio, A. Zugarini, F. Giannini, M. Maggini, M. Melacci. Tomorrow Brings Greater Knowledge: Large Language Models Join Dynamic Temporal Knowledge Graphs. Conference on Lifelong Learning Agents (CoLLAs) 2024.
10. L. S. Lorello, M. Lippi, S. Melacci. Continual Learning for Unsupervised Concept Bottleneck Discovery. Conference on Lifelong Learning Agents (CoLLAs) 2024.
11. L. S. Lorello, M. Lippi, S. Melacci. The KANDY Benchmark: Incremental Neuro-Symbolic Learning and Reasoning with Kandinsky Patterns. arXiv 2402.17431, 2024

Thanks for your attention!

https://cai.diism.unisi.it

Stefano Melacci, stefano.melacci@unisi.it