Collectionless Artificial Intelligence:

Peer-to-Peer Network of Agents for Decentralized Learning Over Time

Collectionless AI
https://collectionless.ai

WIVACE (Siena), September 3, 2025

Stefano Melacci
DIISM, University of Siena

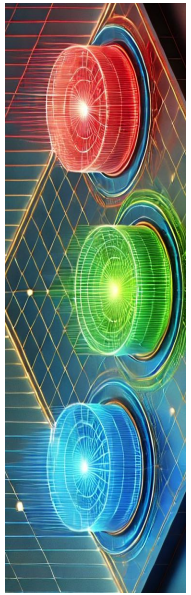
stefano.melacci@unisi.it

1. Research in Learning Over Time

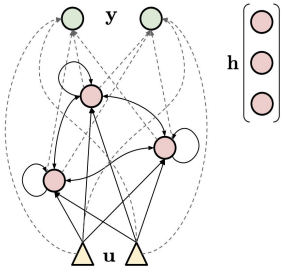
Foundational aspects of learning over time

Some context: our research @ UNISI

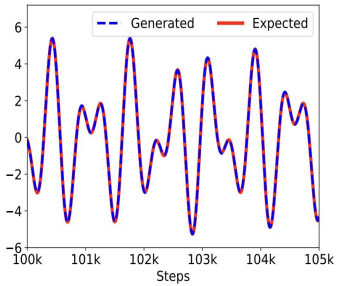
Study, propose, evaluate **new technologies** that are designed to **learn over time**



- **A. Learning strategies**
  - Learn in a "local" manner, without requiring the whole "past": define valid learning dynamics "over time"
  - **Our proposal: Hamiltonian Learning (HL)**
- **B. Neural architectures**
  - Be plastic enough to learn on-the-fly, still being able to generalize over long time horizons
  - **Our proposal: Perpetual Generation, New Neuron Models**
- **C. Out-of-network Knowledge**
  - Exploit information not stored in the weights of the network
  - **Our proposal: Integrating Neural Networks and Logic Knowledge (over time)**



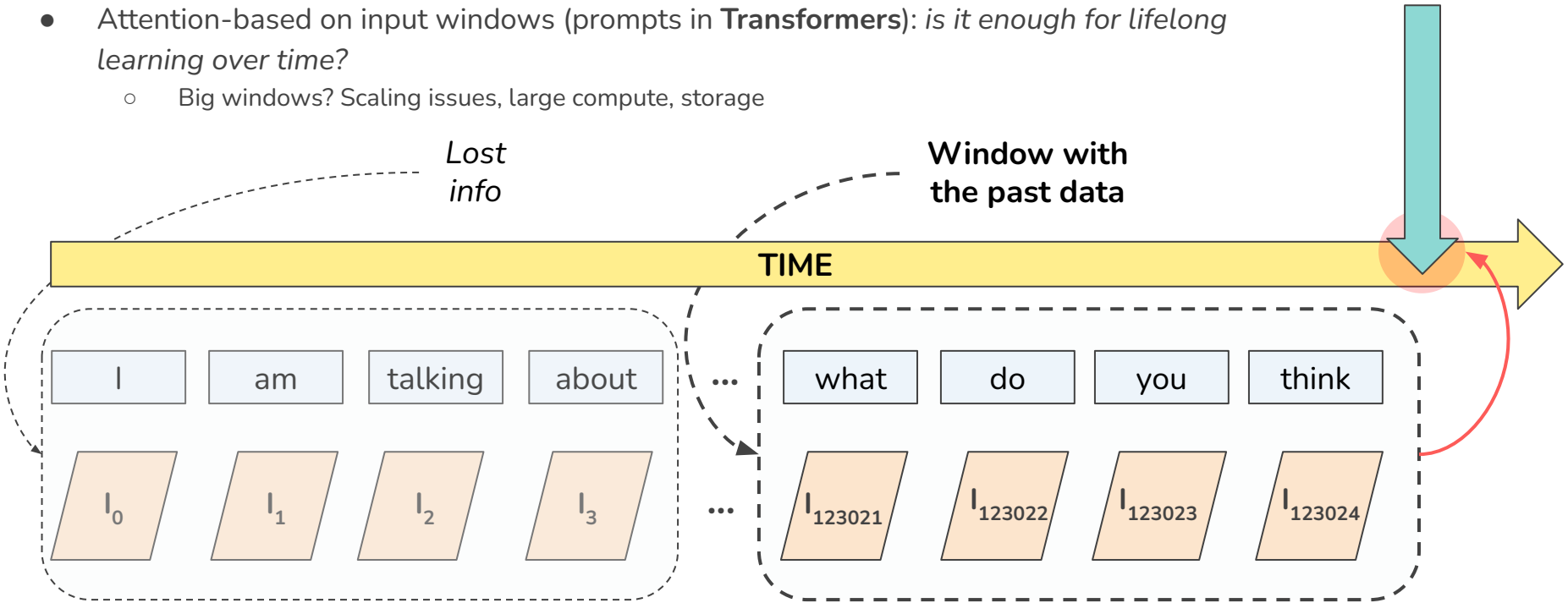
$$\dot{\mathbf{h}}_t = f^{\mathbf{h}}(\mathbf{u}_t, \mathbf{h}_t, \boldsymbol{\theta}_t^{\mathbf{h}})$$
$$\mathbf{y}_t = f^{\mathbf{y}}(\mathbf{u}_t, \mathbf{h}_t, \boldsymbol{\theta}_t^{\mathbf{y}})$$



https://collectionless.ai

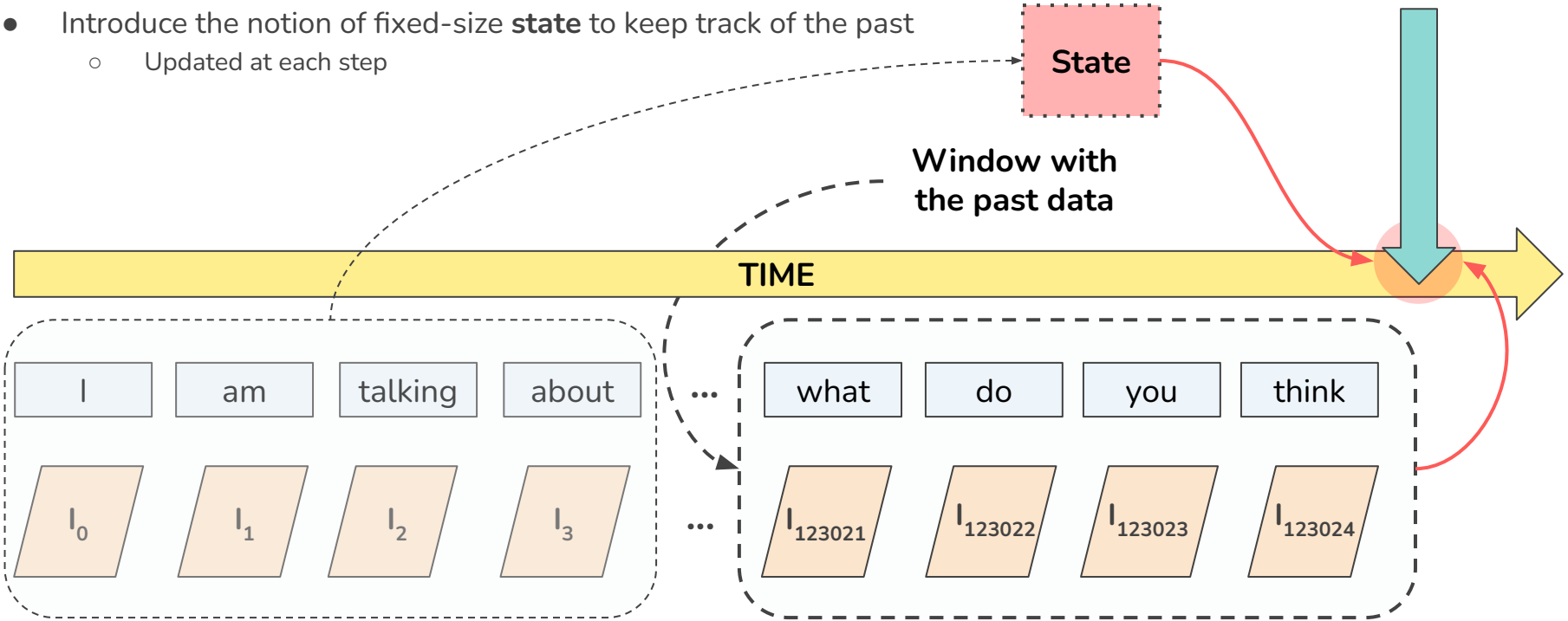
Window-based Models

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



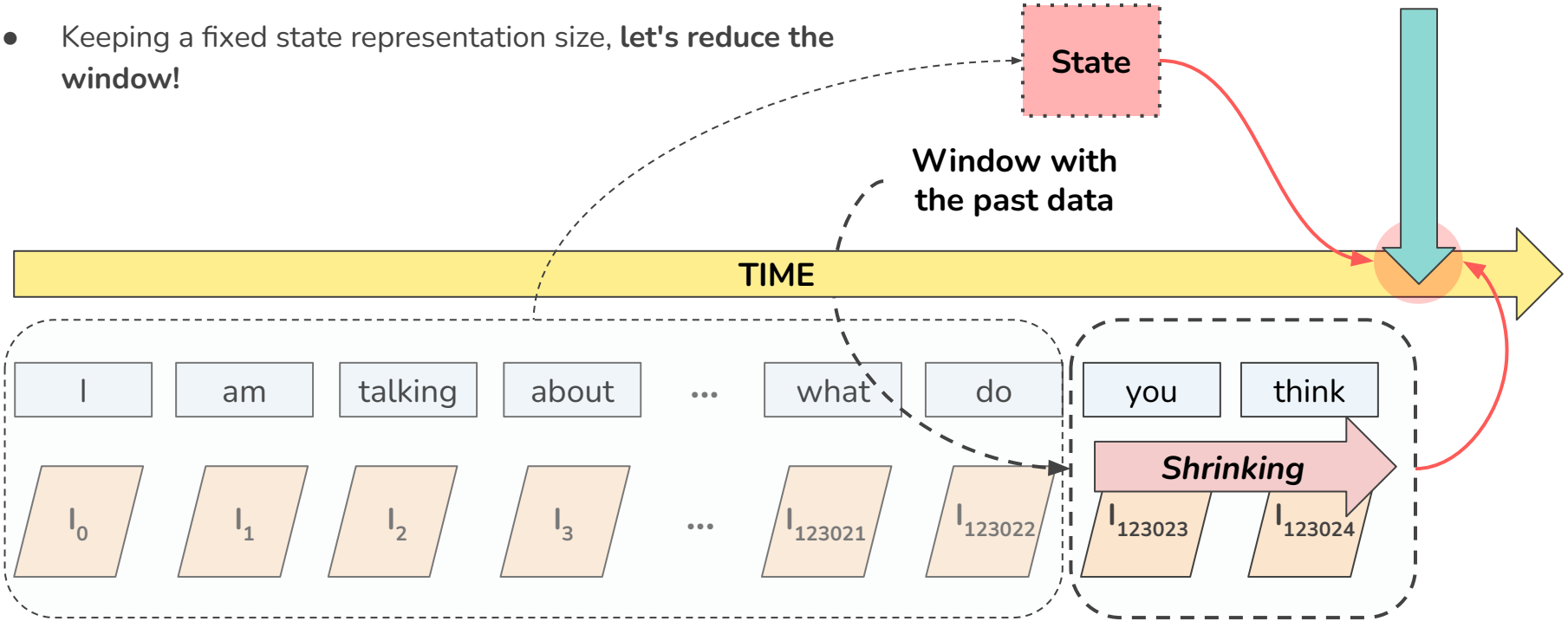
State

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



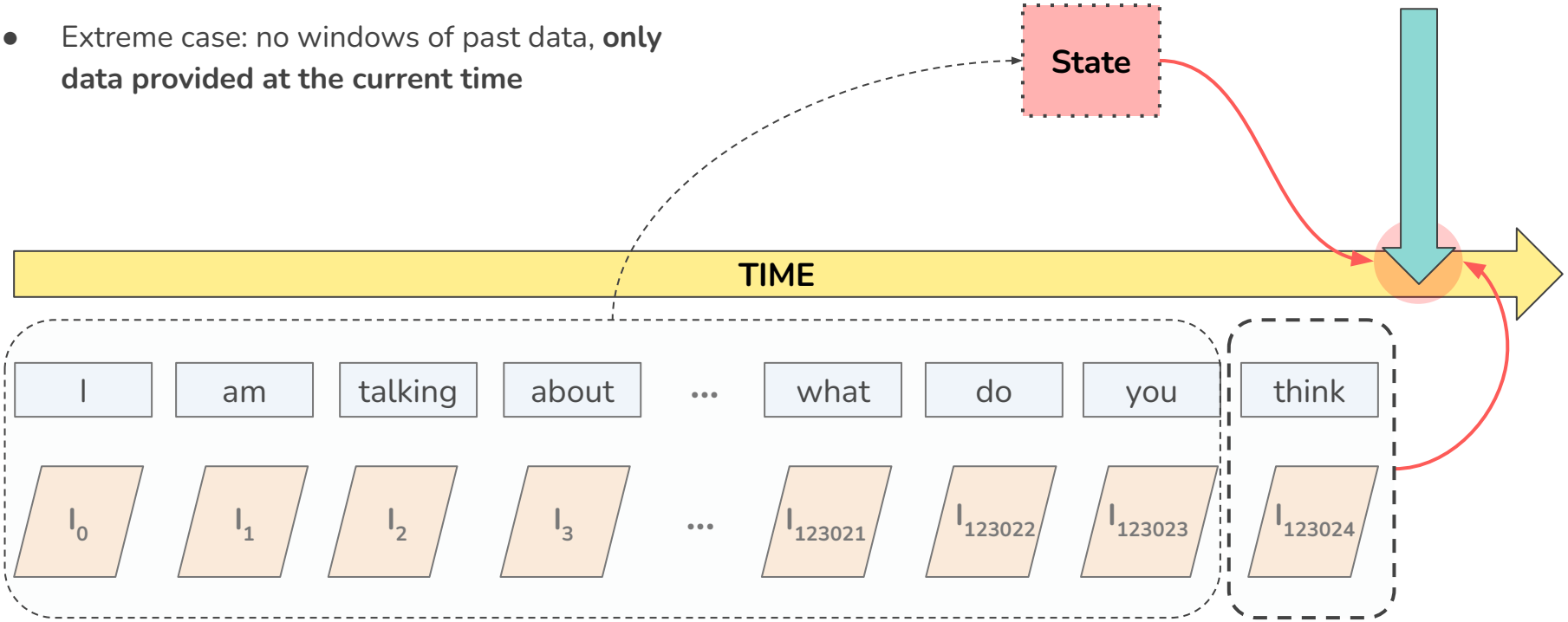
More "Powerful" State

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

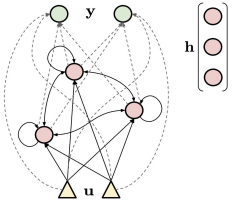


Stateful Local Model

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



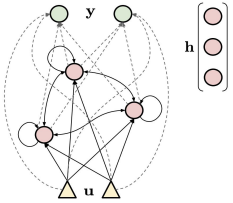
Notation: here and in the rest of this presentation, using the **subscript *t*** is a shorthand notation for ***(t)***, e.g.,, $y_t := y(t)$, *while* ***subscript k***, is used to indicate a discrete step index



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



NEURAL NETWORK

Weights: $\boldsymbol{\theta}_t$

Take a decision, generate something, ...

State

$\mathbf{h}_t$

$\mathbf{y}_t$

t

think

I_{123024}

$\mathbf{u}_t$

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?***

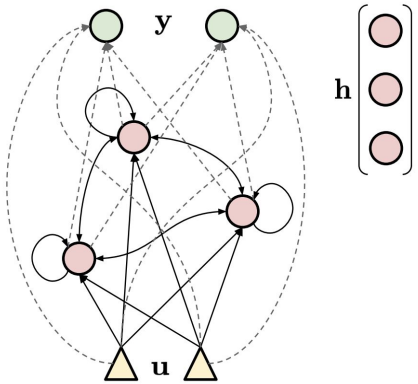
Learning Strategies

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

- **Our proposal: 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

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

Neural Architectures

Hamiltonian Learning focuses on a neuron-centric formulation: all the neurons work in parallel given the previous state of the network, no need to distinguish the two propagations!

Some example of architectures reframed in the state-space model setting; where is time?

- Signal propagation time **through the network neurons (network axis)**
- Propagation due to **update of the same-level latent representation over the time axis**

Convolutional Network

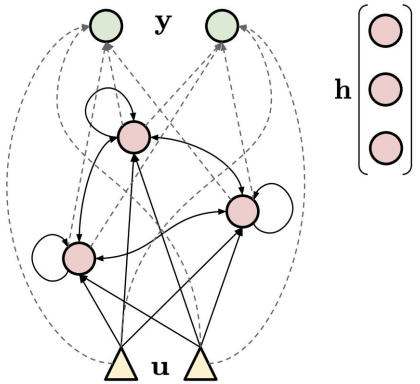
$$\mathbf{h}_k = \mathrm{CNN}(\mathbf{u}_k, \cdot, \theta^{\mathbf{h}})$$
$$\mathbf{y}_k = C\mathbf{h}_k$$

Autoregressive-like Network

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

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

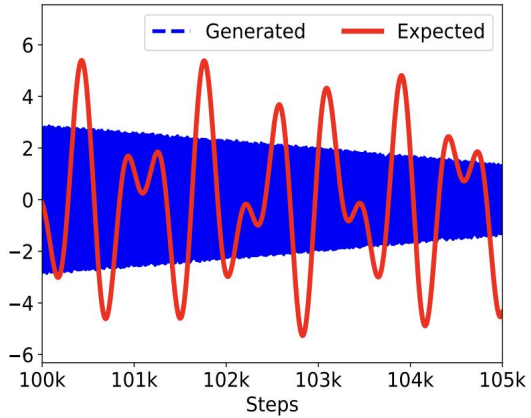
$$\dot{\mathbf{h}}_t = f^{\mathbf{h}}(\mathbf{u}_t, \mathbf{h}_t, \theta^{\mathbf{h}}_t)$$
$$\mathbf{y}_t = f^{\mathbf{y}}(\mathbf{u}_t, \mathbf{h}_t, \theta^{\mathbf{y}}_t)$$

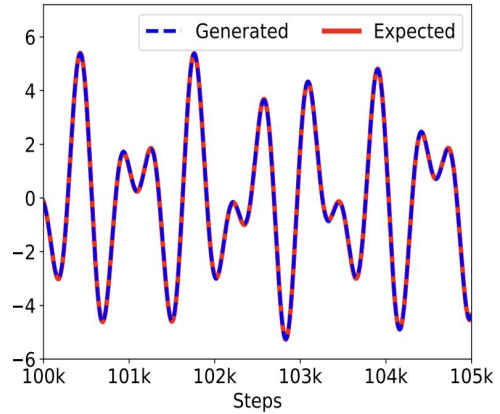
Perpetual Generation

y_t with $t \to \infty$, 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*)

Continuous-Time Linear State Space Model

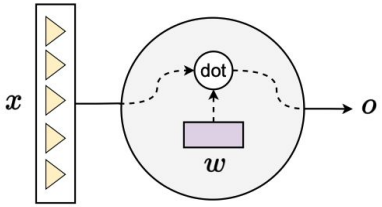
$$\dot{\mathbf{h}}_t = A\mathbf{h}_t + B\mathbf{u}_t$$
$$\mathbf{y}_t = C\mathbf{h}_t$$
$$\text{with } \mathbf{h}_0 = \mathbf{0} \text{ and } \mathbf{u}_{t>0} = \mathbf{0}$$

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

Neuron Model

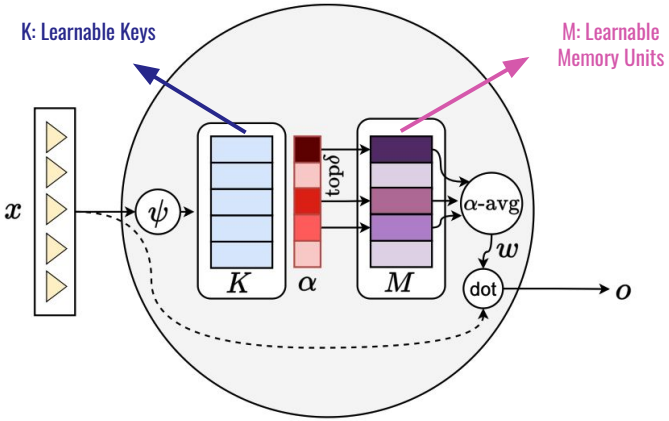
- We reconsider the **neuron model**, to make it easier to find a good trade-off between plasticity and stability (no replays), reducing **forgetting**
- **Continual Neural Units (CNUs)** - generalization of vanilla neurons

Every neuron is responsible of generating the weights to use for a given input



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

What is going on in AI today?

Missing

- Safety
- Control
- Privacy
- Trustworthiness
- On-the-fly Customizability
- ....
- **Time**
- **Humans**

- **AI Agents**, **Agentic AI**, ... tons of buzzwords, a lot of commercial pressure, huge investments
- **What is the result?** AI models are used to orchestrate other AI models or AI-based services, **taking decisions** on the whole "process", **accessing different resources (local and on the internet)**, being able to run code they create, to copy themselves, ...
- **Humans?** They are end-users, not educators, not people intended to "directly" help improve the agent

What is going on in AI today?

Missing

- Safety
- Control
- Privacy
- Trustworthiness
- On-the-fly Customizability
- ....
- **Time**
- **Humans**

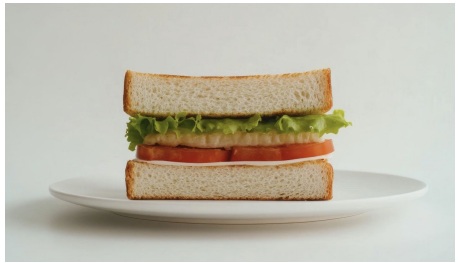
This is very very very (3x) nice, but...

- **Each single orchestrated model is not trained by you, but by big companies**
- It could behave in an unexpected manner and was trained on "unknown" (to you) data: you are not able to (you simply cannot) train it from scratch in a controlled manner
- **If using a web-service, your data is actually shared with others (who you do NOT know at all)**

What is going on in AI today?

Missing

- Safety
- Control
- Privacy
- Trustworthiness
- On-the-fly Customizability
- ....
- **Time**
- **Humans**



"...a sandwich has more regulation than AI..."
Yoshua Bengio, TED Talk, April 2025
https://www.ted.com/talks/yoshua_bengio_the_catastrophic_risks_of_ai_and_a_safer_path

We want to go beyond these limits

Missing

- Safety
- Control
- Privacy
- Trustworthiness
- On-the-fly Customizability
- ....
- **Time**
- **Humans**

- **Machines** are fast learners and fast predictors, but they are trained in an offline manner
- **Machines** learn with no context around them, only by observing a flood of stochastically provided data samples
- **Machines** are not aware of time, they do not improve over time
- **Machines** are designed for a huge amount of interactions with simulators, offline collected data, or other pre-trained models
- **Machines** are designed to act passively, offering services

We want to go beyond these limits

Missing

- Safety
- Control
- Privacy
- Trustworthiness
- On-the-fly Customizability
- ....
- **Time**
- **Humans**

Time is the protagonist of our story

- **Humans** learn over **time**, they interact over **time**, they have the notion of causality, they improve over **time**, they can change their mind (sometimes) over **time**
- **Humans** live in a large context where other humans live, and they are conditioned by what's around them, which is dynamic, i.e., it changes over **time**
- **Humans interact** in such a context in a controlled manner, making the most out of each interaction
- **Humans interact** over time considering roles and trying to be collaborative (weill, not always...)

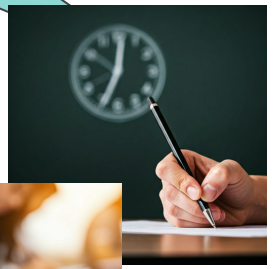
It's a world of Interactions, and Time does matter!

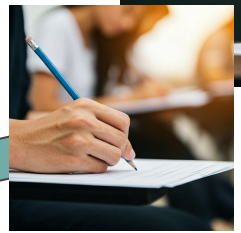
Progressive development

Deepening knowledge

Interacting with a teacher

Taking exams





Interacting with other people

Effective interactions

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)

Mainstream (Today)	Huge Datasets	Offline Learning	Limited Privacy	Static	No Time, No Learning -oriented Interactions	Huge Computational Resources	Human: Developer, Data Preparer
Collectionless AI	**Continuous Streams**	**Online Learning**	**Privacy by Design**	**Dynamic, Lifelong**	**Time and Interactions**	**Distributed, Edge-Level**	**Human and AI Agents: Same Level**

Decentralized Real-time Computations: Privacy & More Accessible Resources

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

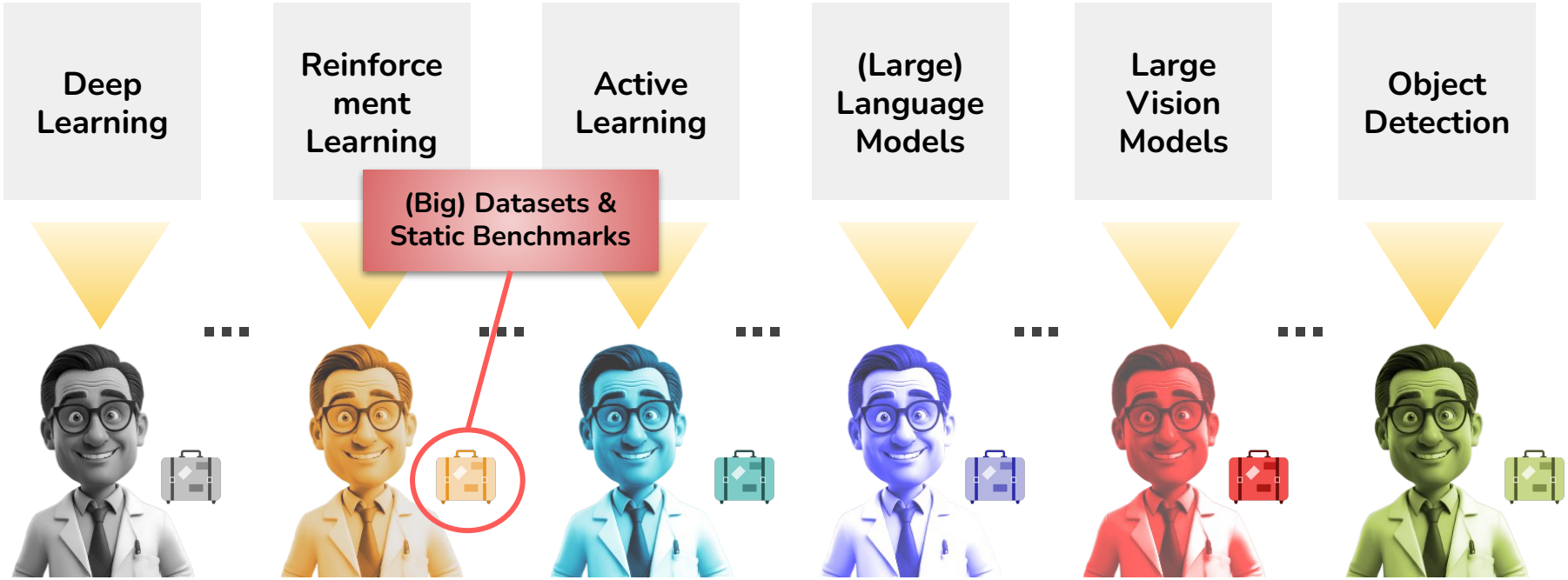
VS.

On-the-edge devices

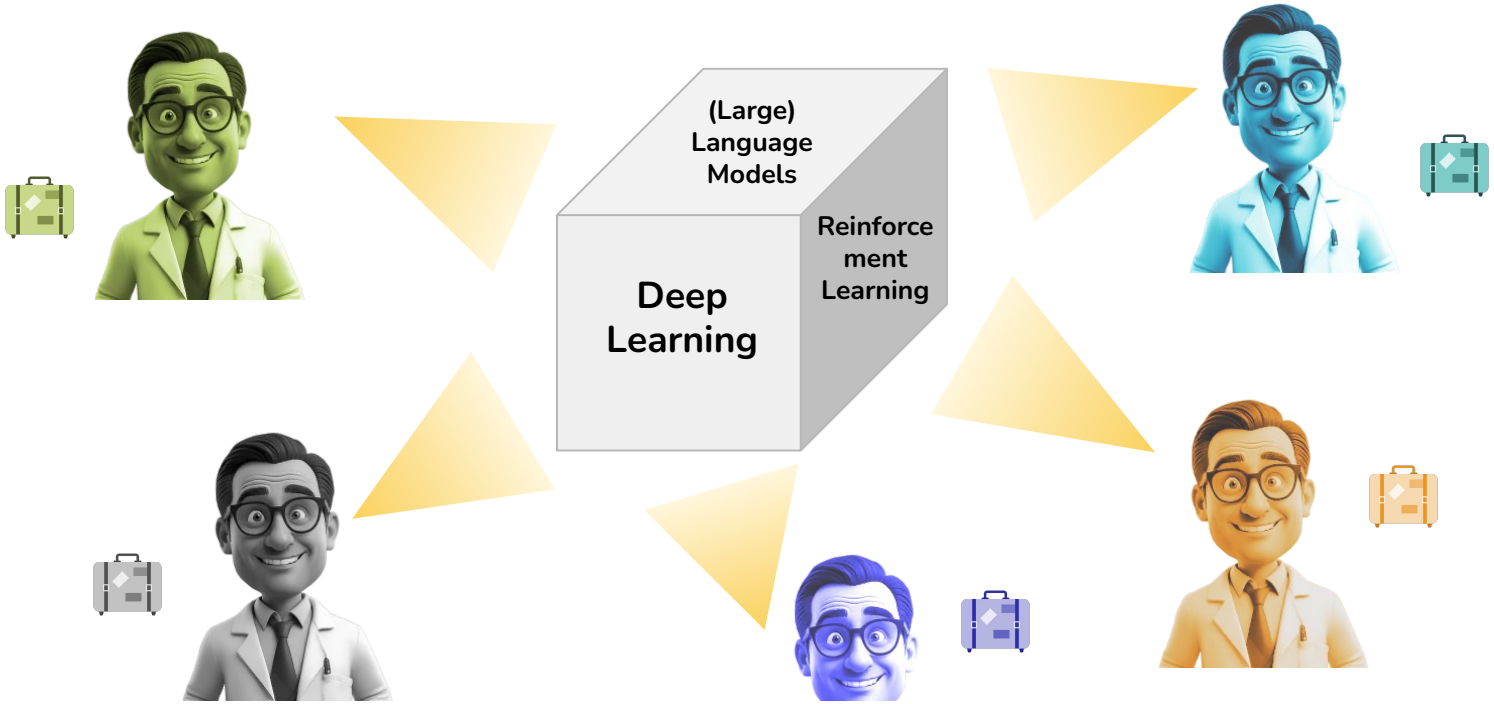
Collectionless AI Agents

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

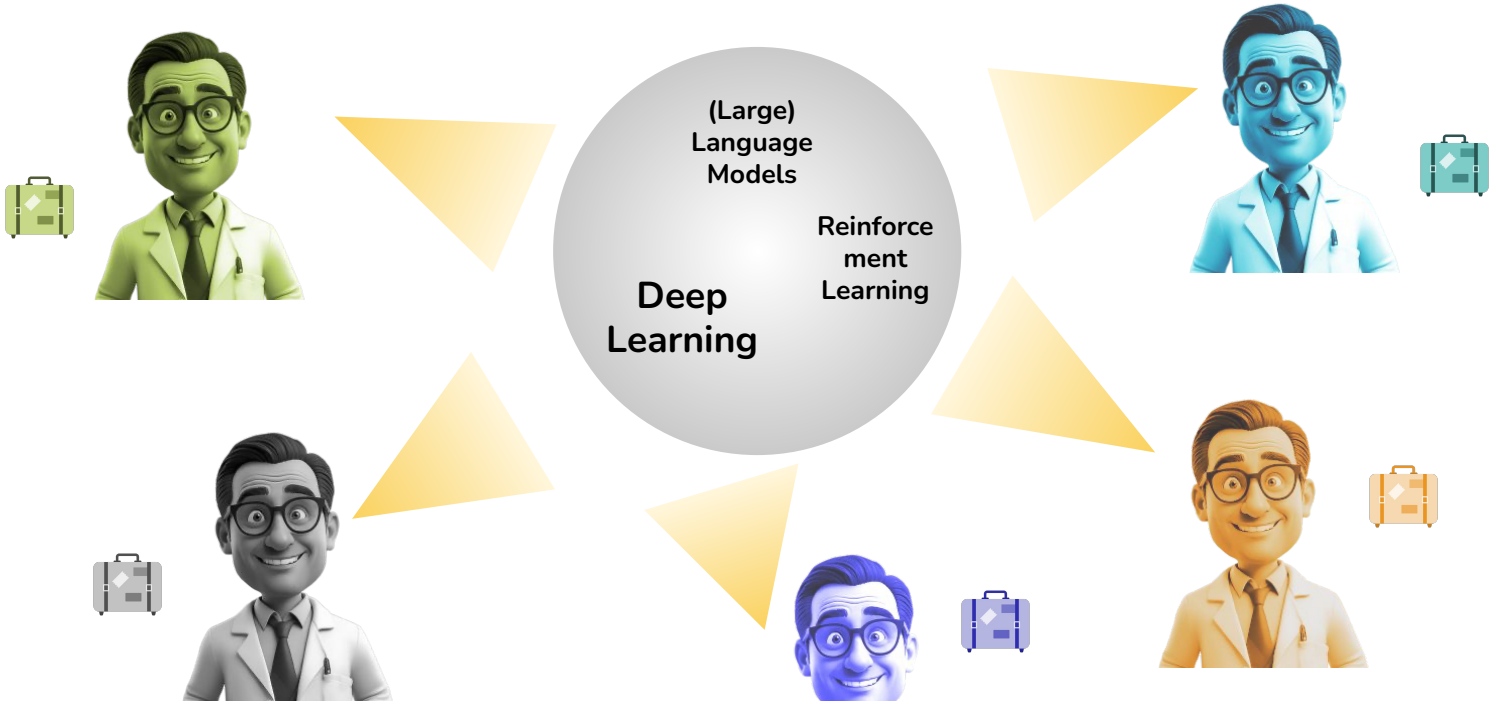
From Today's ML/AI (this slide) to Collectionless AI (next slides)



Step 0/3: Recall that Today's "Topics" are Different Facets of the Same Hypercube

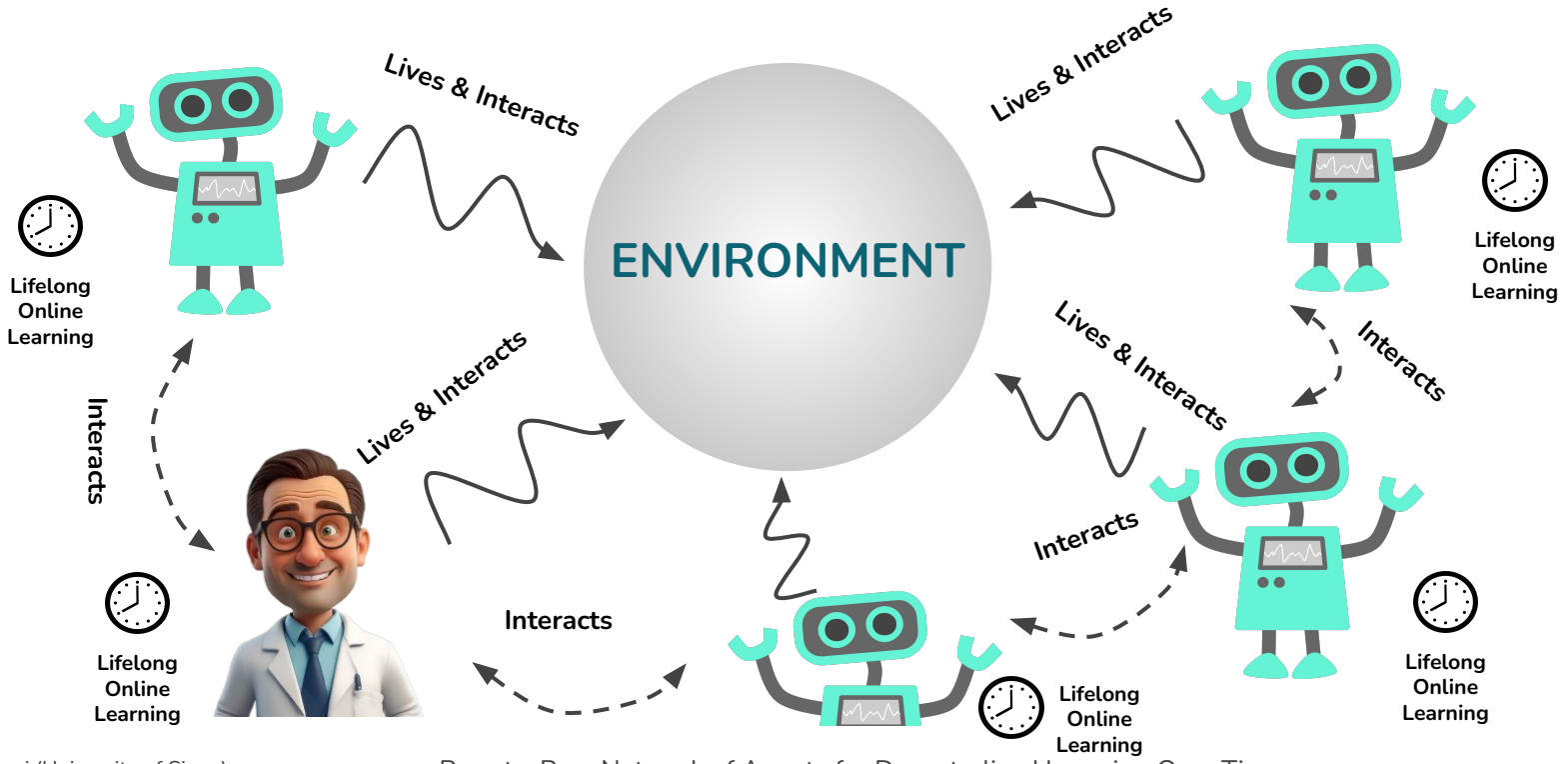


Step 1/3: Relax the Boundaries

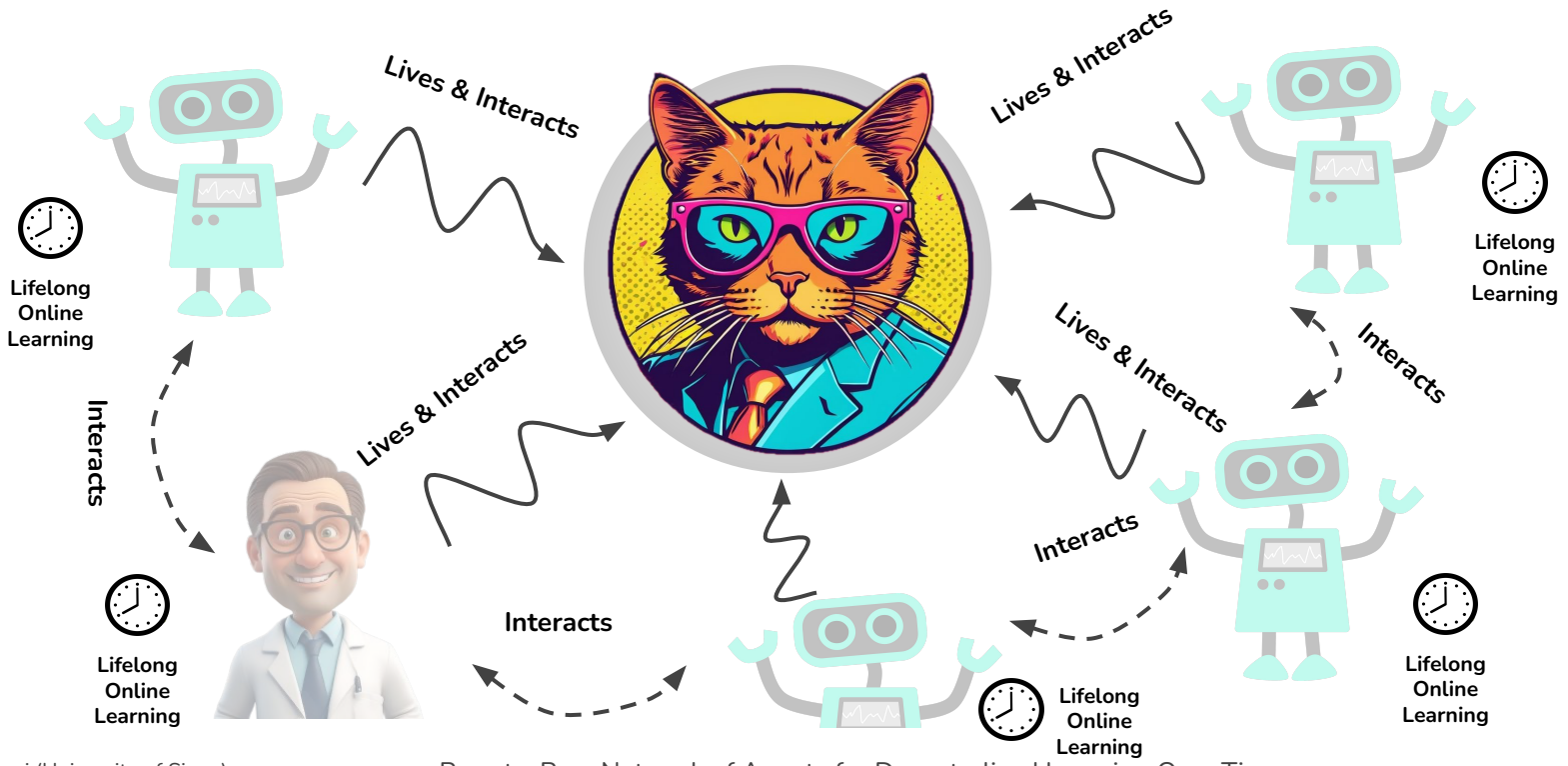


Step 2/3: Embrace Time

Step 3/3: Promote Interactions



<Welcome to Collectionless AI>



Collectionless AI Team

(Core members & Affiliates)

Thanks to all!
What I am going to present about UNaIVERSE is mostly the outcome of the crazy work of the "boxed" people, a "bigger thank you" goes to them!

Jul 20, 2024 · 1 min read · People

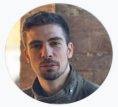
Stefano Melacci

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

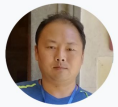
Matteo Tiezzi

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 31, 2024 · 1 min read · People

Aiden D'souza

Student, VNIT

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

Apr 24, 2025 · 1 min read · People

Abdur R. Fayjie

PostDoc, Università degli Studi di Siena

Jul 31, 2024 · 1 min read · People

Achraf Bouchtita

Student, Polytech Marseille

Events

Share

Conference on Lifelong Learning Agents

Index

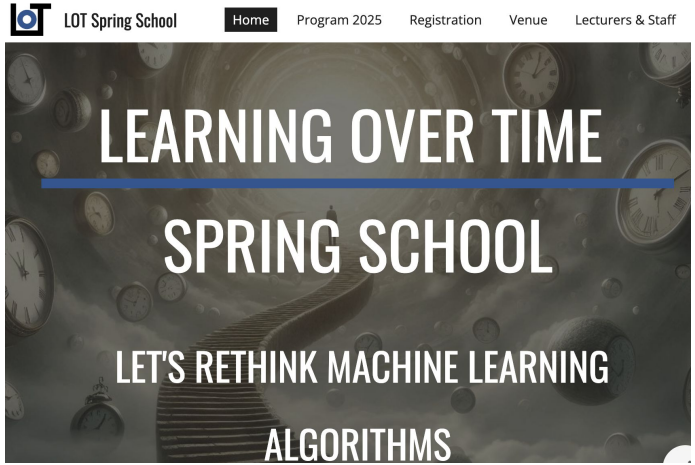
Program Chairs


Vincenzo Lomonaco
University of Pisa

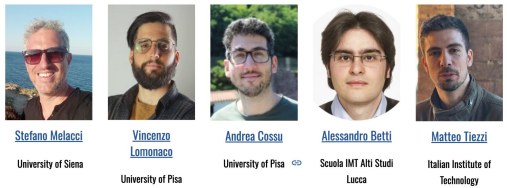

Stefano Melacci
University of Siena


Tinne Tuytelaars
KU Leuven

Pisa, July 2024
https://lifelong-ml.cc/



ORGANIZERS

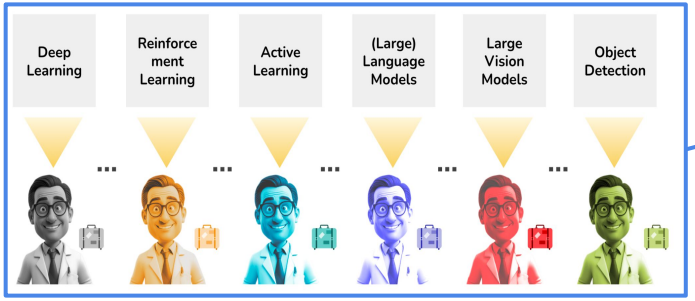


Certosa di Pontignano (Siena), March 24-27, 2025
https://cai.diism.unisi.it/school

3. Toward building Collectionless AI agents and beyond: UNaIVERSE

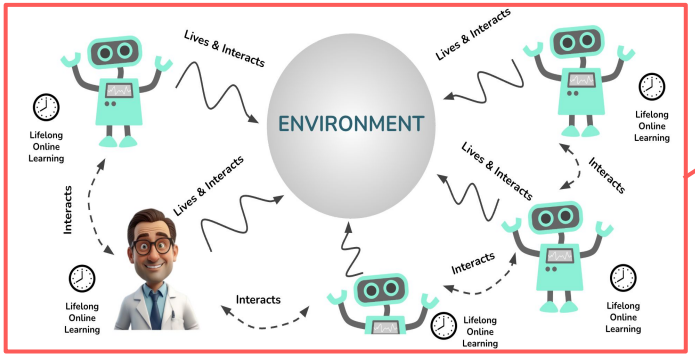
The UNaiVERSE Platform

Today's AI



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

Collectionless AI



UNaIVERSE

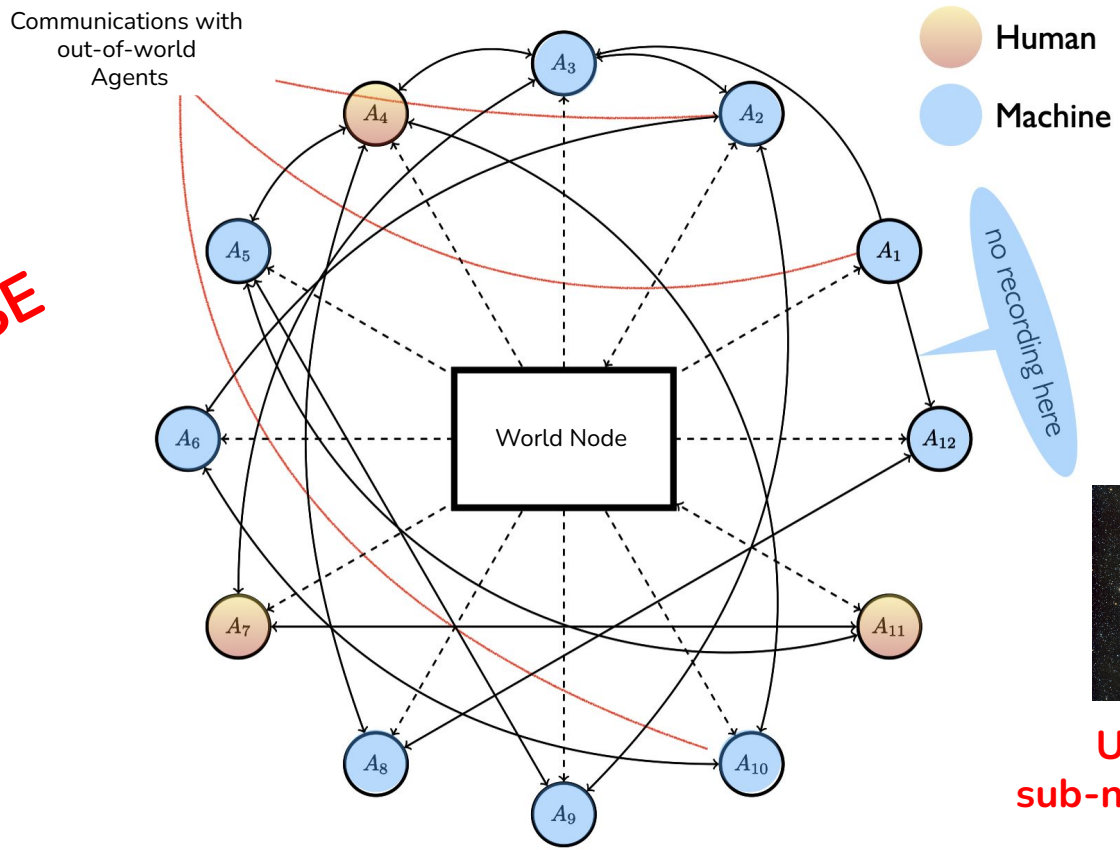
Platform modeling dynamic universes **where to study and progressively develop novel types of agents** (**tomorrow**)

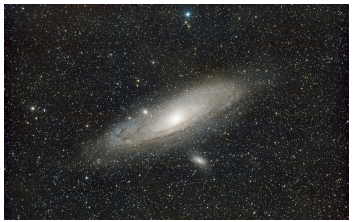
"Communities of Agents that Learn Over Time"

UNaIVERSE
by Collectionless-AI team

Exploring the growth of emergent intelligent behavior

Portion of UNaIVERSE



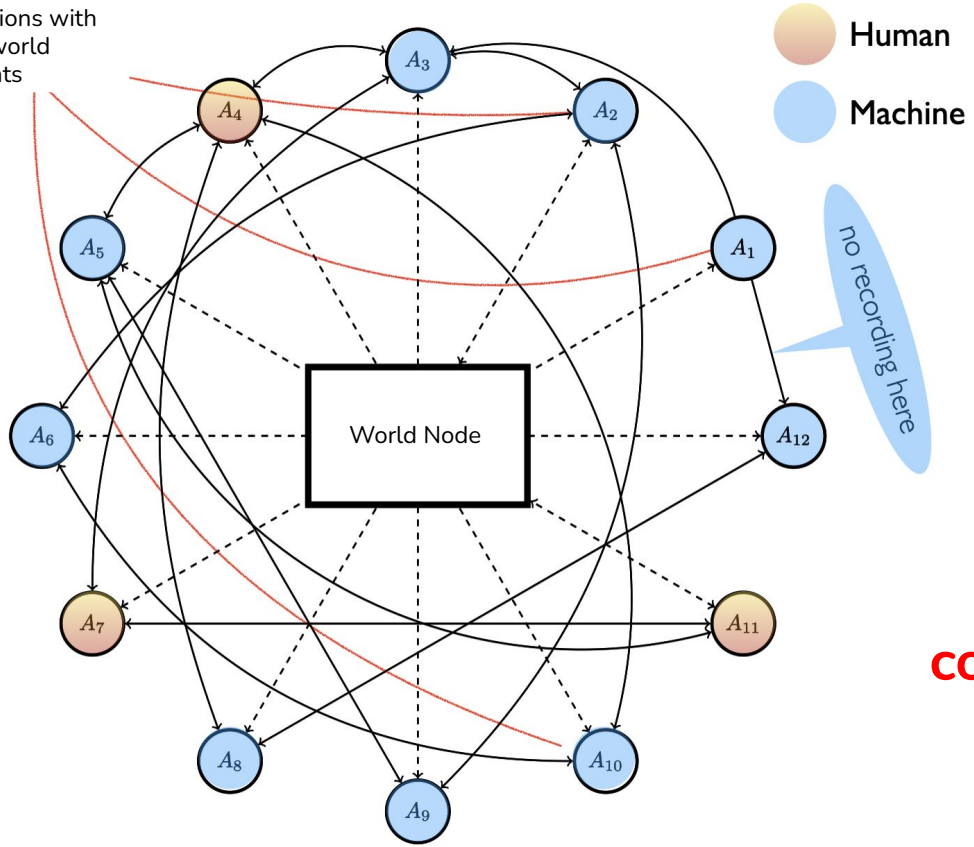


UNaIVERSE: many sub-networks like this one!

Agents run on your machine

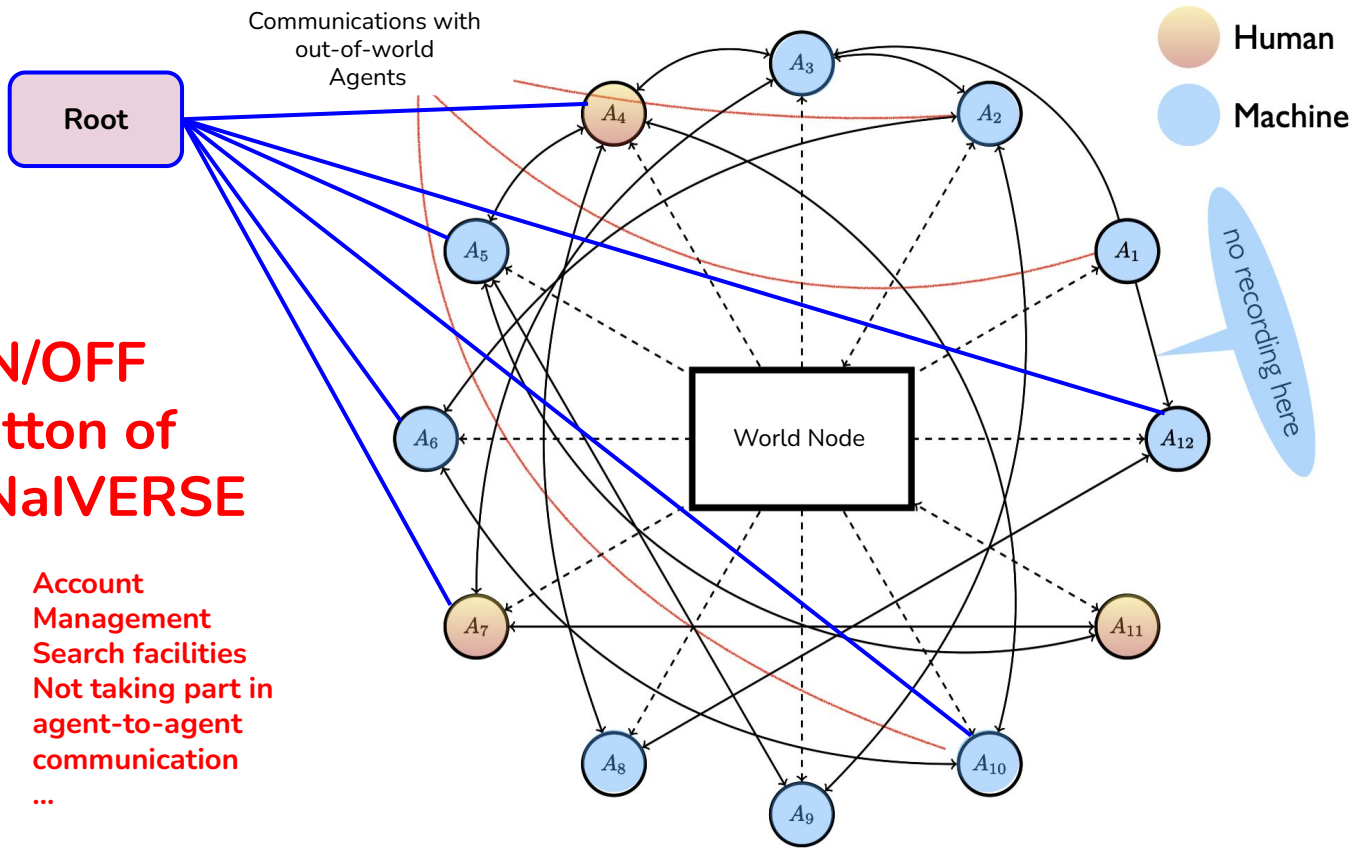
Live streaming!

Exploring the growth of emergent intelligent behavior



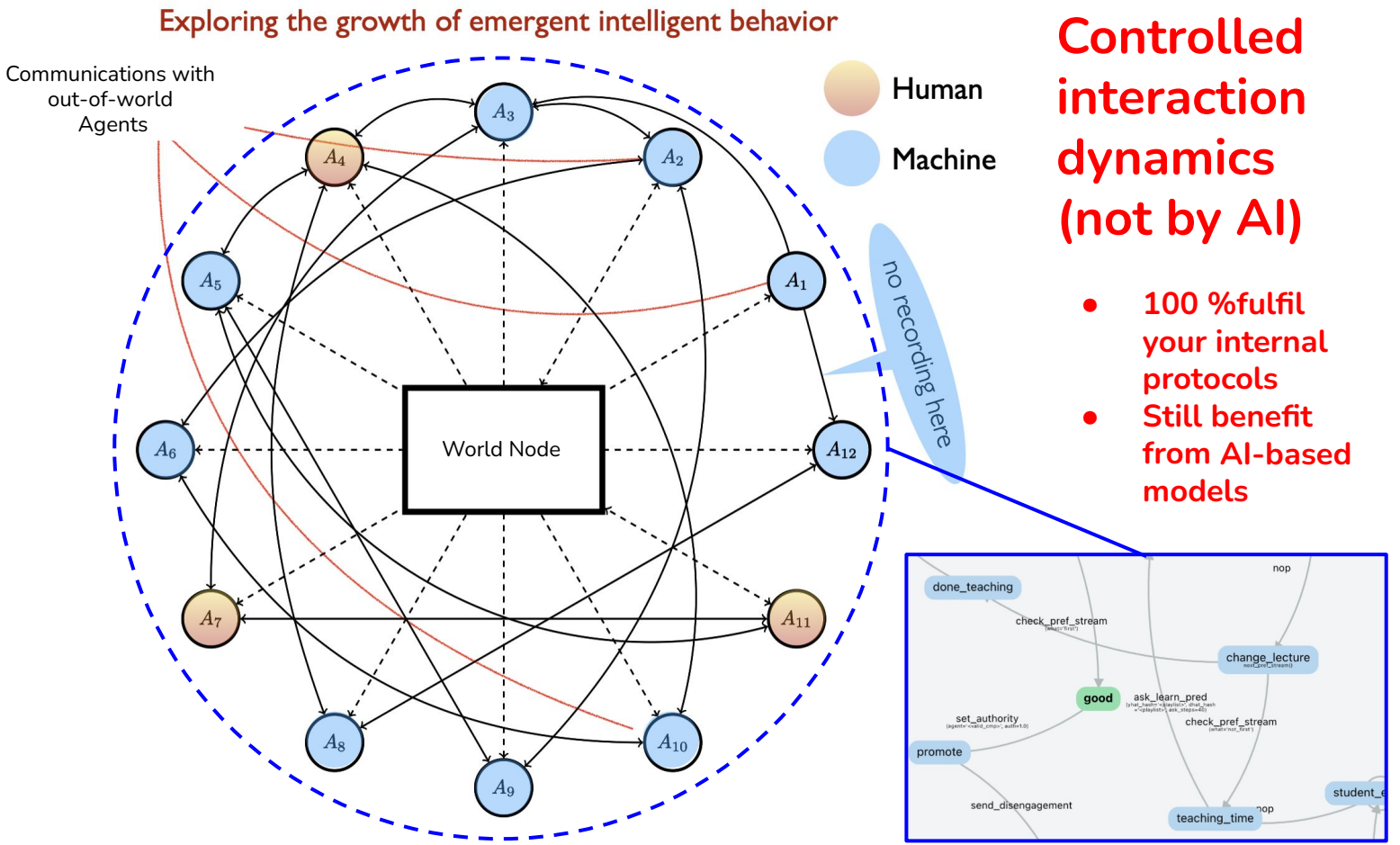
Peer-to-peer encrypted communications

Exploring the growth of emergent intelligent behavior



ON/OFF button of UNaIVERSE

- Account Management
- Search facilities
- Not taking part in agent-to-agent communication
- ...

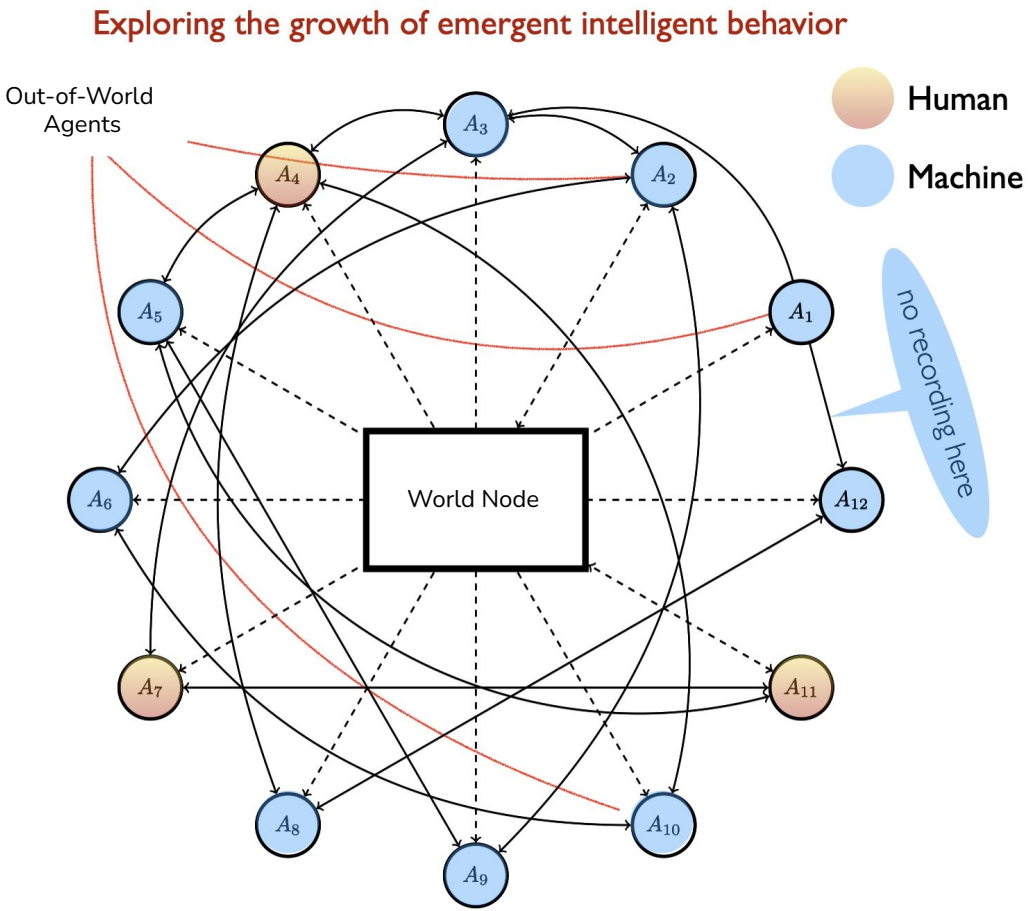


Controlled interaction dynamics (not by AI)

- 100 %fulfil your internal protocols
- Still benefit from AI-based models

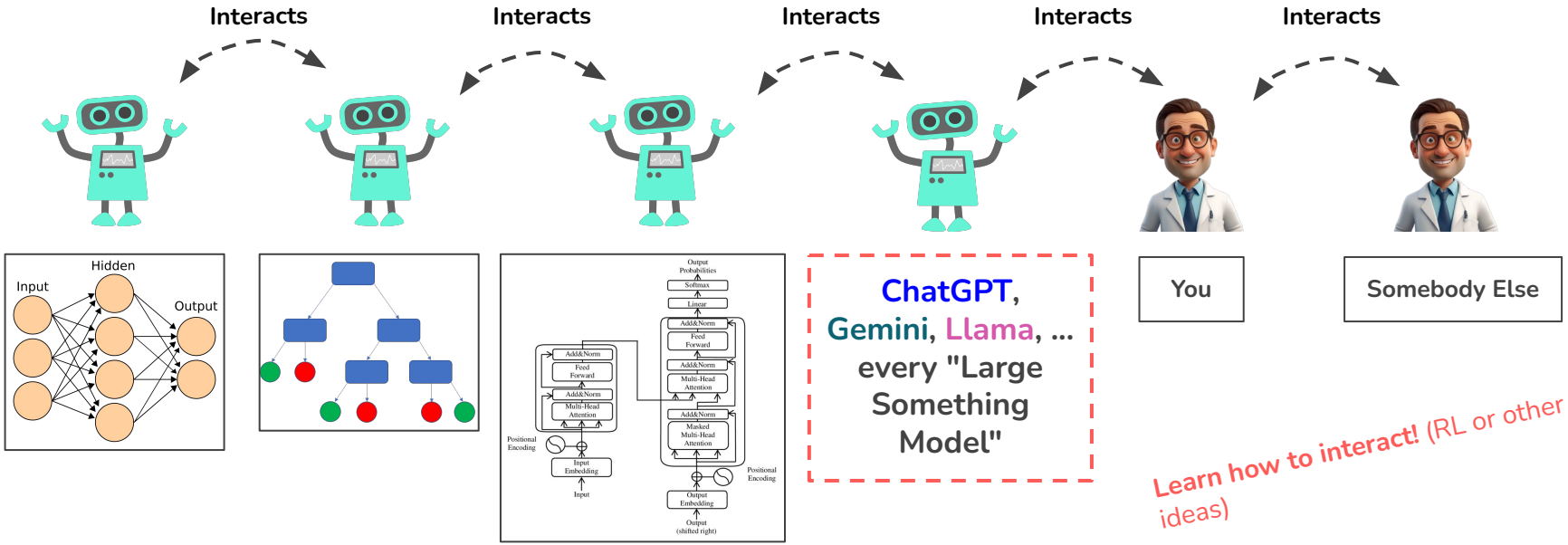
Social Network of AI

- Showcase you technology
- Comment, provide feedbacks on agents
- ....



UNaIVERSE: Communities of YOUR Agents

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



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

What can I do in the UNaIVERSE?

What you can already do today (in a unified manner) **and way beyond**

- **Showcase your technology**, running on your own servers
- **Remotely control processes**, in a fully peer-to-peer, end-to-end encrypted way
- **Solve tasks in a distributed manner**, exploiting your models and others
- ...
- Distributed benchmarking
- Meetings with humans and AI agents
- Lectures (for AI agent, or for humans, or for both)
- Musical concerts with AI agent and humans
- Security tests
- ...

What can I do in the UNaIVERSE?

...but more importantly

- **Grow your own agent**
  - Controlling its interactions
  - **Ensuring your data does not leave your computer**
  - Think about the world of healthcare, private business, ...

- **Research in how to develop the next-get technologies** that effectively learn over time
  - Beyond datasets, toward **Collectionless AI**
  - Beyond huge power consumption, toward small edge devices
  - Toward a real form of trustworthy AI

3. Main Components of UNaIVERSE

The UNaiVERSE Platform

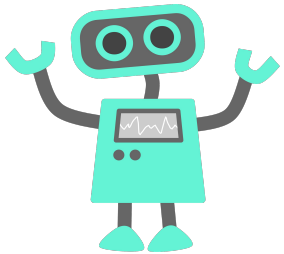
Components of UNaIVERSE

Main ingredients:

1. **Streams**
   a. Raw Information over time
2. **Agent(s)**
   a. Behavior
   b. Processor
3. **World**

1. Streams

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



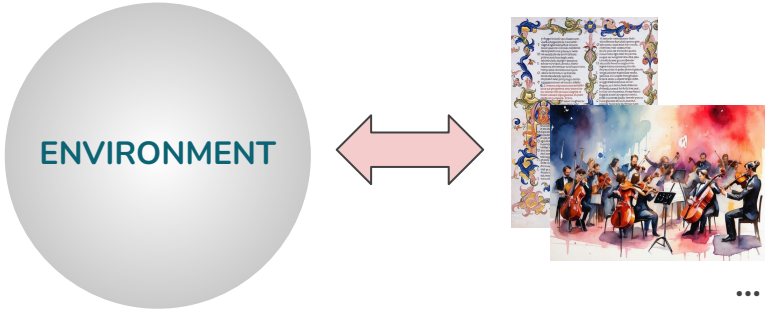
2. Agent

Citizen of the UNaIVERSE

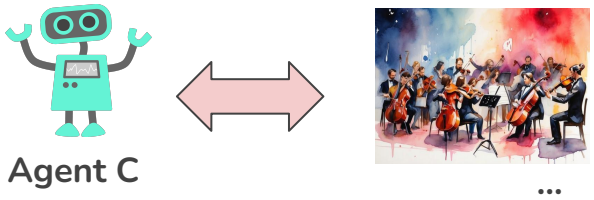
3. World

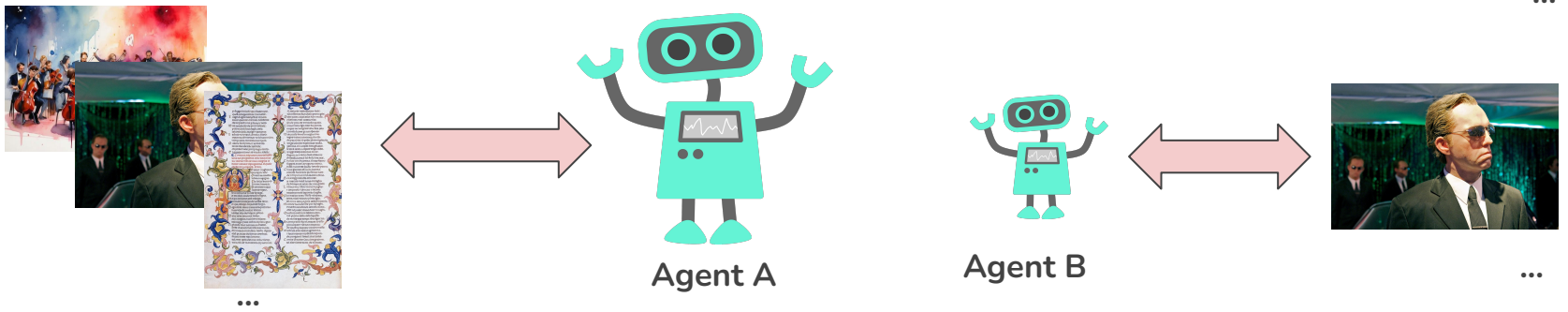
A special "agent" that manages the world

Streams



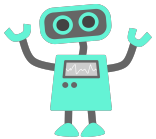
- Raw information (**function of time**)
  - An image, some text, a tensor (etc.) at each time instant
    - E.g., video, actions in each frame
    - E.g., images, categories of each image
    - E.g., data from a sensor, normal vs. anomaly label



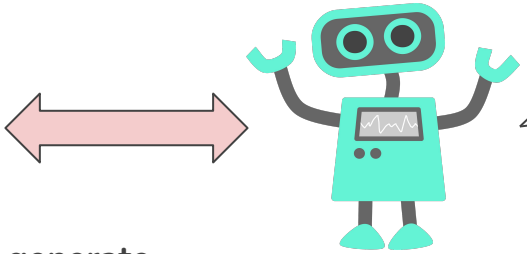


Agent

...



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



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

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

Behavior

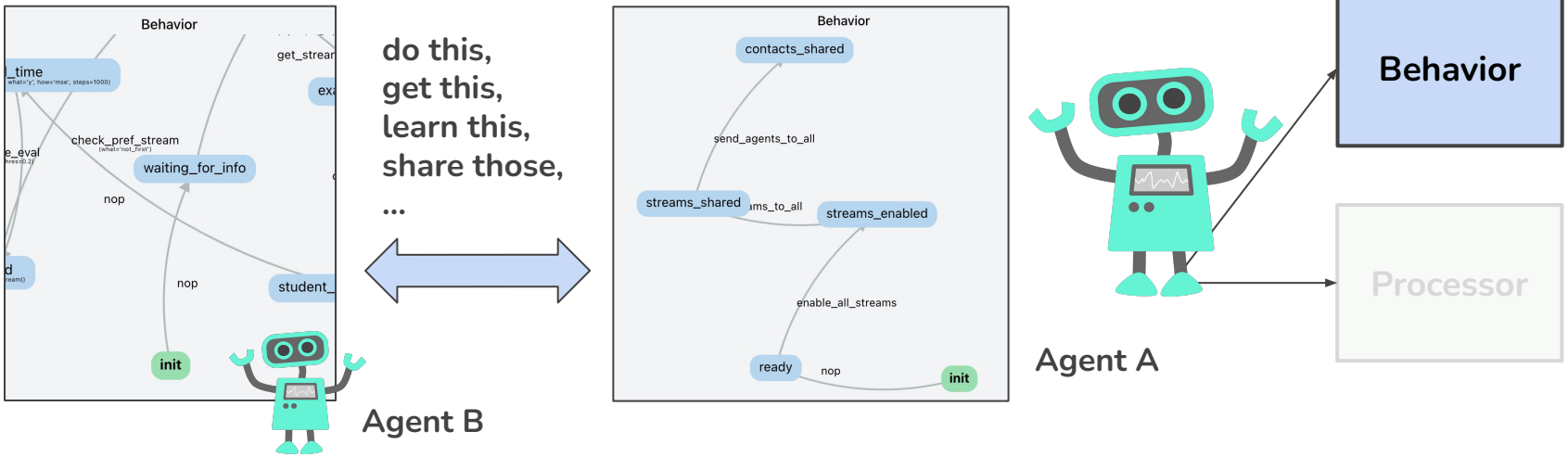
Processor

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

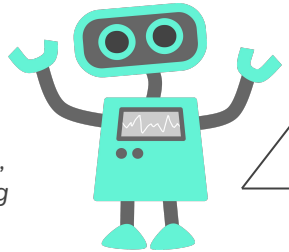
Agent: Behavior

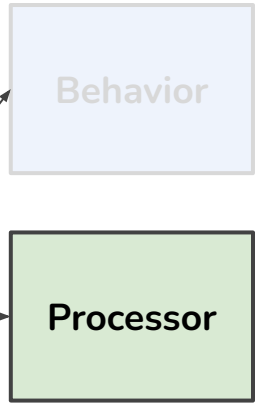
- In the most naive way, it can taught as an **hybrid instance** of a Finite State Machine (FSM) and a Markov Decision Process (MDP)
- **Hybrid Stabe Machines** of different agents **interact with each other**

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

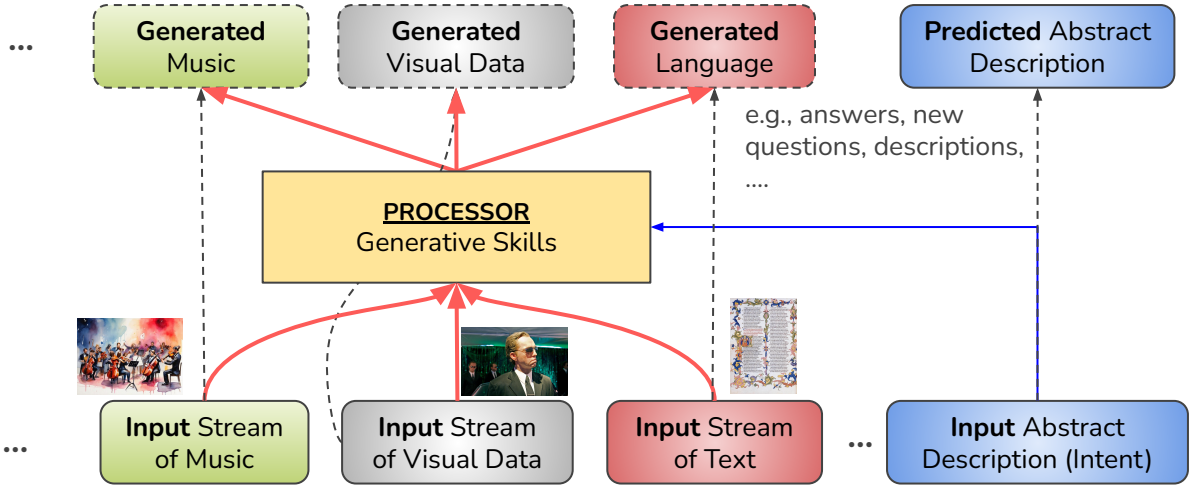


Agent: Processor





Concepts: [Relaxing music, visualizing a person, talking about the orchestra, ...]



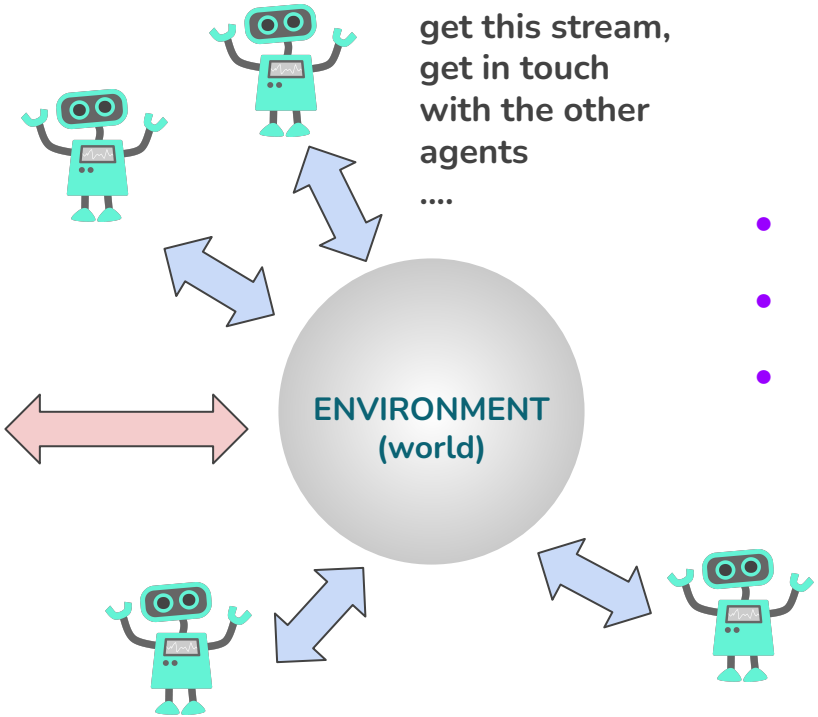
Concepts: [Relaxing music, people video, open discussion, ...]

- How does the agent generate data (makes predictions)?
- How does the agent handle the learning-over-time process?

generate,
learn to generate,

World: A Special "Agent"

...

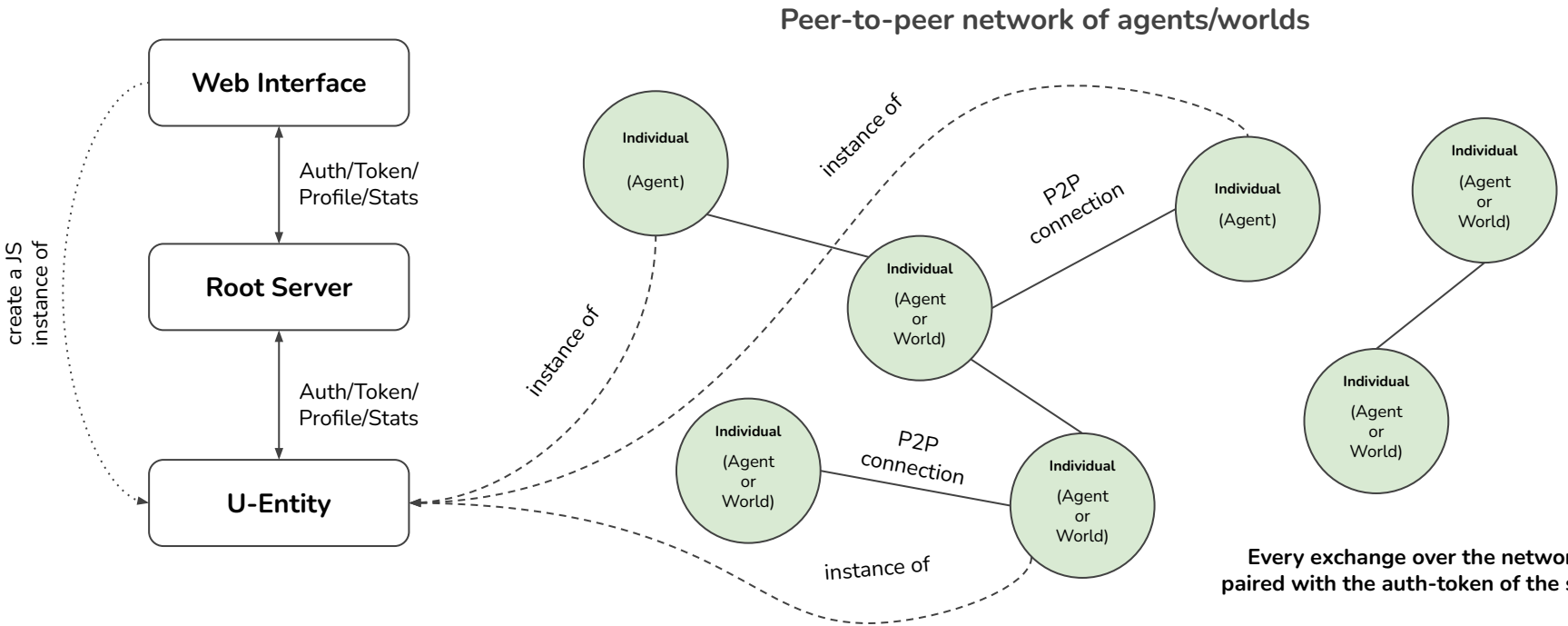


- How does the world manages the agents that live there?
- What streams the world exposes and when?
- What out-of-agent dynamics are going on in the world?

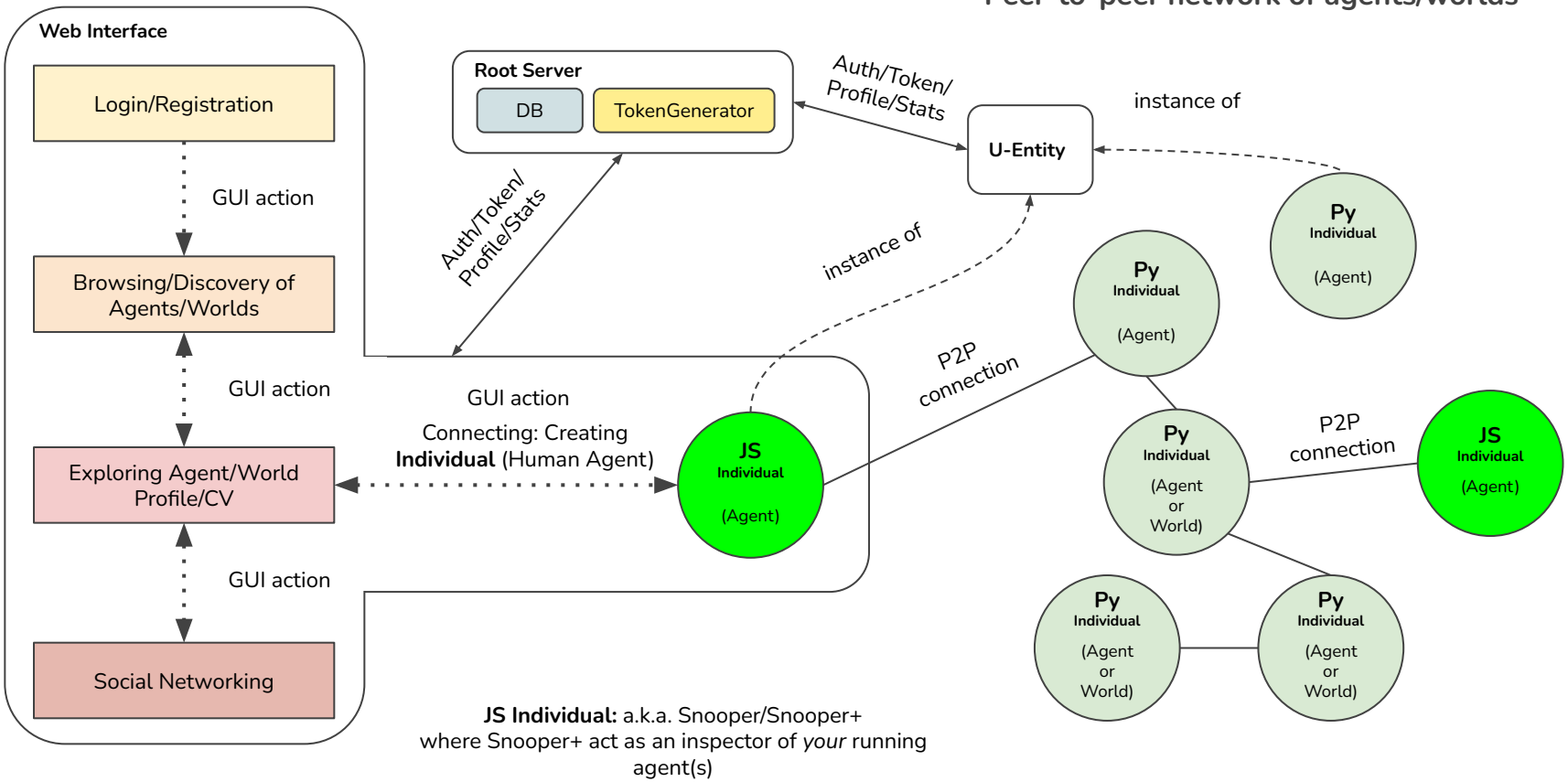
4. Structure of UNaIVERSE

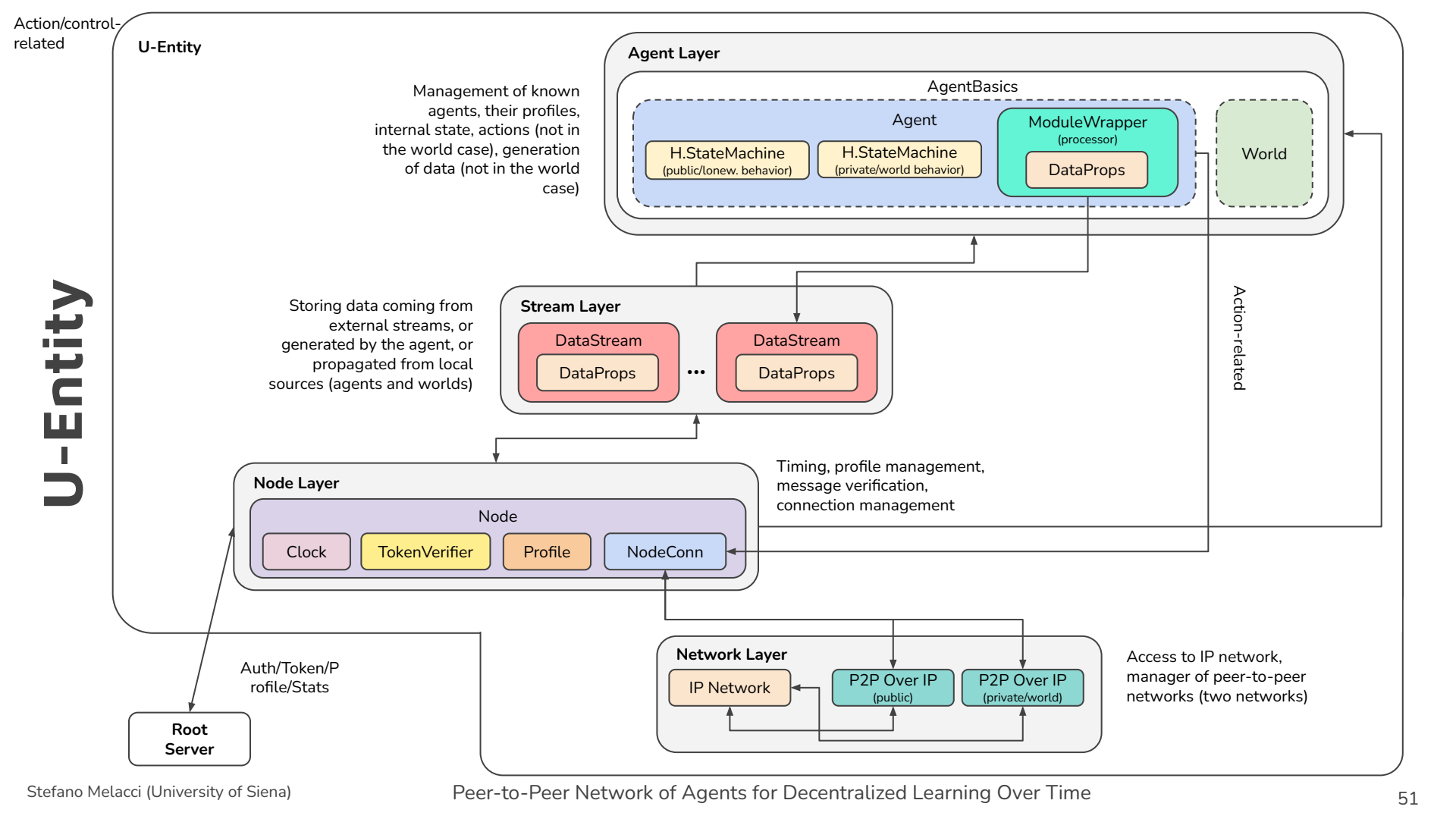
The UNaiVERSE Platform

Structure of UNaIVERSE: Overview



Web Interface & Root

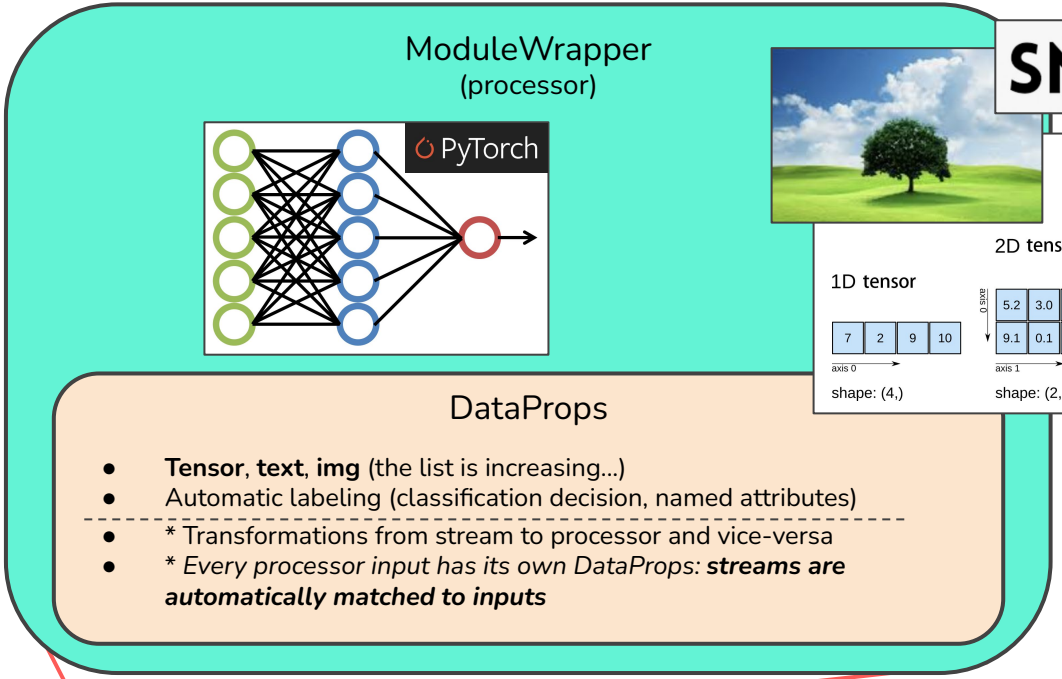


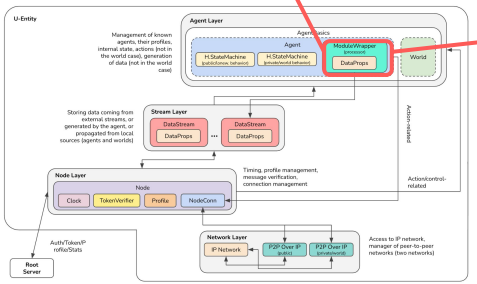

Action/control-related
U-Entity
Agent Layer
AgentBasics
Agent
H.StateMachine
(public/lonew. behavior)
H.StateMachine
(private/world behavior)
ModuleWrapper
(processor)
DataProps
World
Management of known agents, their profiles, internal state, actions (not in the world case), generation of data (not in the world case)
Stream Layer
DataStream
DataProps
DataStream
DataProps
Storing data coming from external streams, or generated by the agent, or propagated from local sources (agents and worlds)
Action-related
Node Layer
Node
Clock
TokenVerifier
Profile
NodeConn
Timing, profile management, message verification, connection management
Network Layer
IP Network
P2P Over IP
(public)
P2P Over IP
(private/world)
Access to IP network, manager of peer-to-peer networks (two networks)
Auth/Token/Profile/Stats
Root Server
U-Entity

U-Entity: Processor

A container including a Pytorch module and the information on its inputs/outputs

(In practice, every callable object is fine in place of the module)

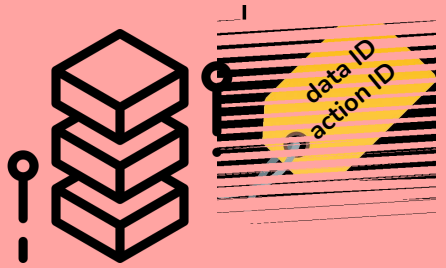


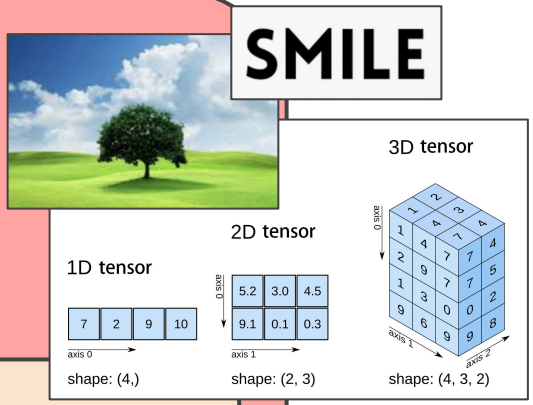


U-Entity: Data Stream

A generic stream of data, offering facilities to interpret the data format and the way it is intended to be sent over the P2P networks

DataStream

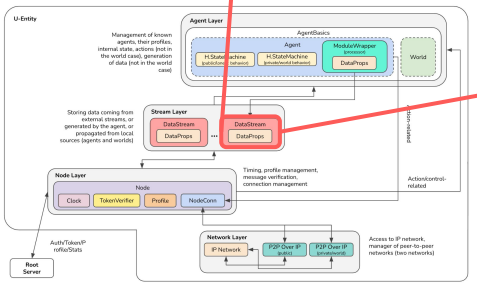


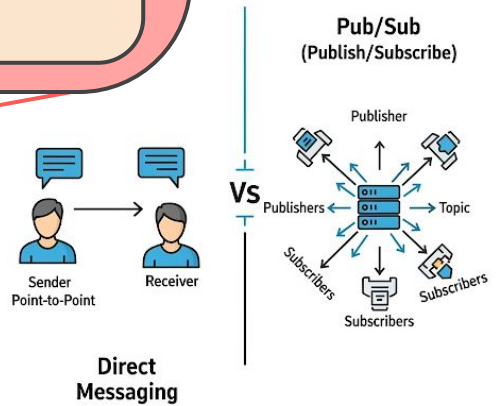


DataProps

- **Tensor**, **text**, **img** (the list is increasing...)
- Automatic labeling (classification decision, named attributes)

- * Public or private/world network
- * **Direct messaging** (DM) or **Pub/Sub** (PS)





U-Entity: Connections

Connection with other agents are managed by the strong interplay of a lower-level P2P manager (**P2P Over IP**) and a more abstract connection manager (**NodeConn**)

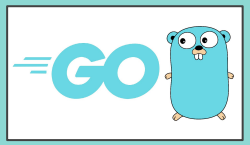
NodeConn

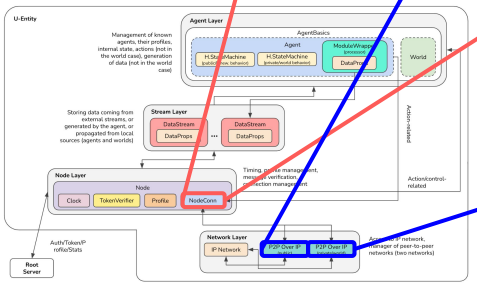
- Max connections
- **Gating**
- **Rendezvous** information propagation

GO code compiled as C-Shared Library

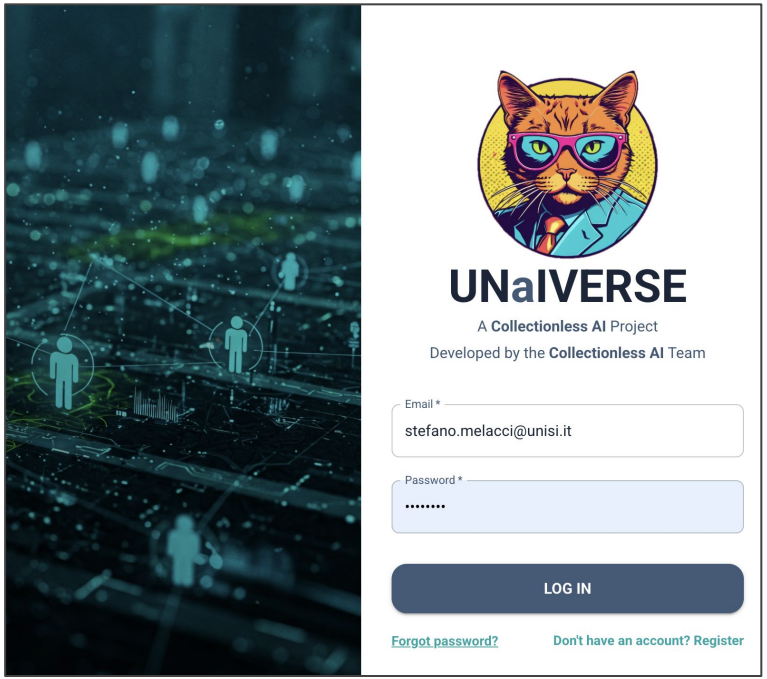
P2P Over IP (public & private)

- **Protocols** (TCP, UDP-Quic, UDP-WebRTC-Direct)
- **Relay** offering/exploiting
- **Rendezvous** management (private/world)





Joining UNaIVERSE



A top-down usage-oriented description!

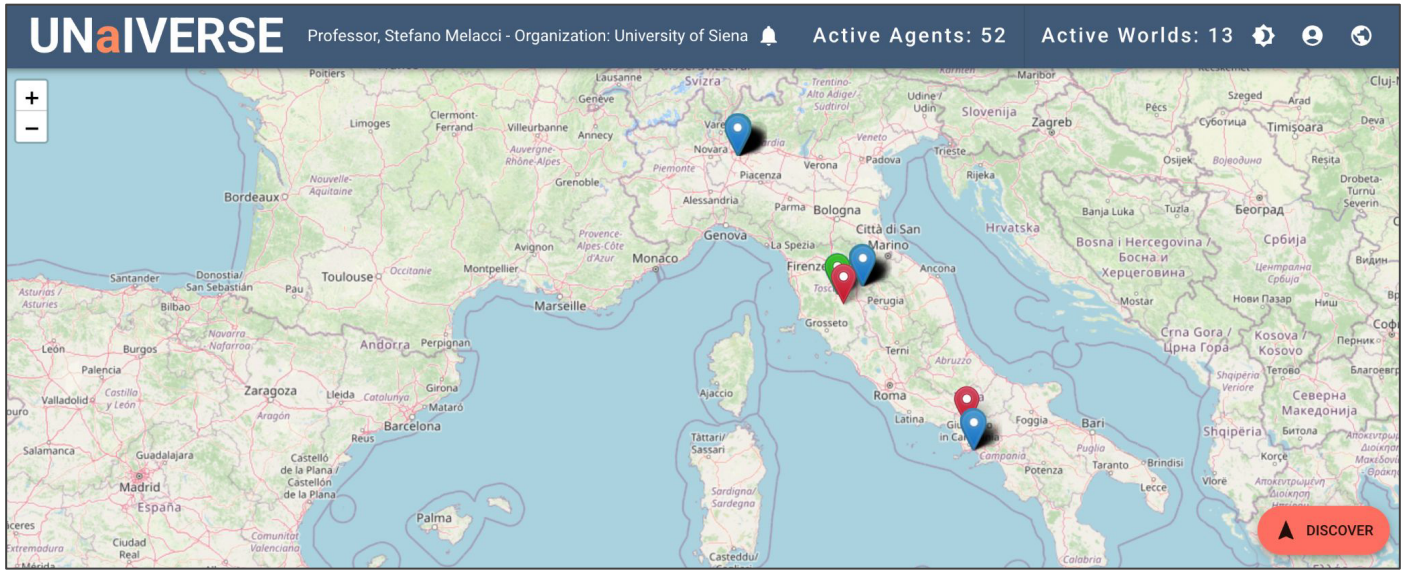
The first step consists in **becoming part of the UNaIVERSE network**

- Connect to UNaIVERSE website (*URL public soon*)
- Register for a new account
- Login

UNaIVERSE Map: Agents and Worlds

The UNaIVERSE map shows the location of **agents** (**red**/**blue**), **worlds** (**green**)

- When you register you become an **agent** (**red**) node (i.e., *human*) and you can access other **agents**/**worlds** through your *browser* (important feature! - see later)
- You can create artificial **agents** (**blue**) or **worlds** (**green**) - running on your machine



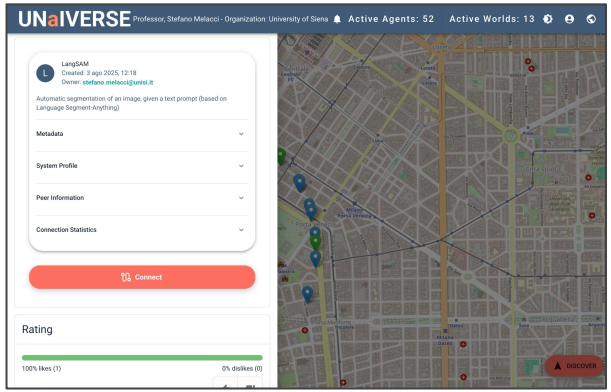
Discovering Agents and Worlds

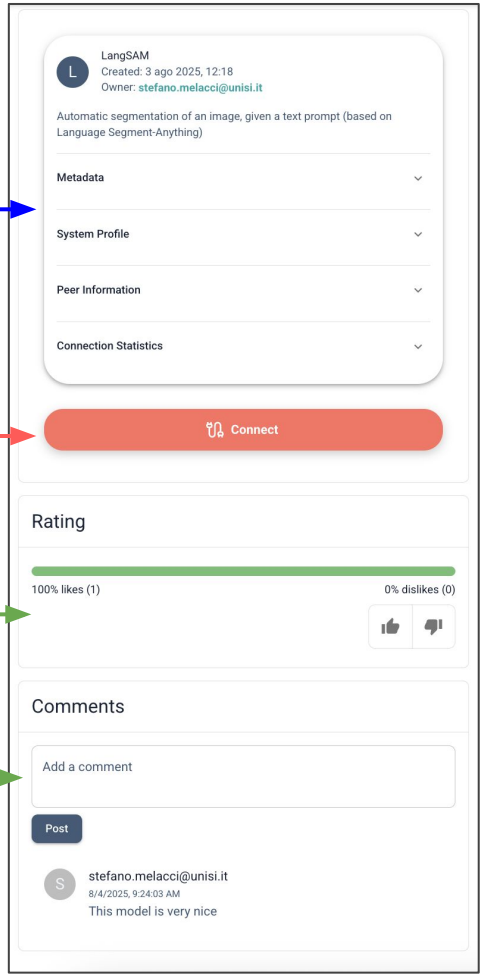
- Discovering agents is straightforward if you know where they are and you see them on the **map**
- If you want to **search** for an agent with a specific name or skills (etc.), you can use the "*Discover*" button: the found agent will be shown in the chat

Exploring Agents and Worlds

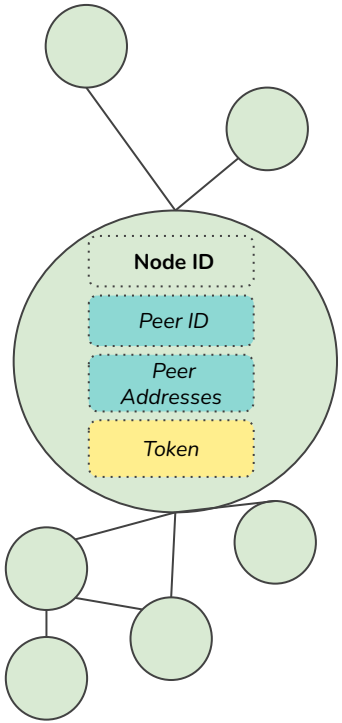
Clicking on the agent/world marker or on its name in the search result, will open its **profile**, composed of multiple sections

1. **Generic information**, network information, connection stats
2. **Connect** button, to interact (as a **human**) with this agent
3. **Ratings and comments** from other UNaIVERSE citizens





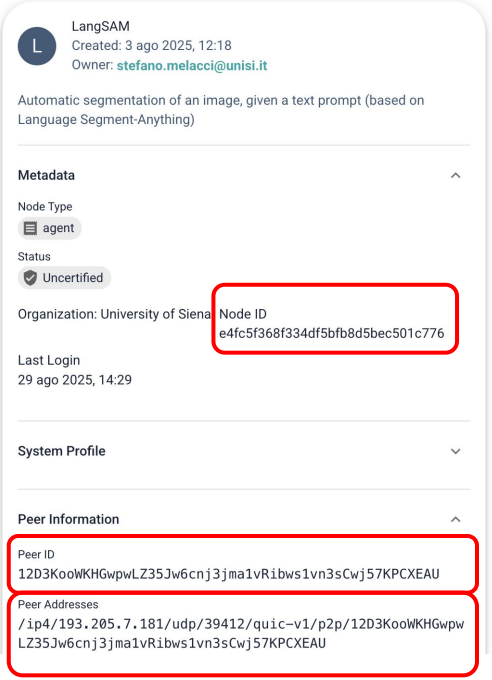
Exploring Agents and Worlds (Cont.)



From the structure point of view, every **agent/world** is hosted by a logical entity called **node**, that is instantiated in a peer-to-peer network

- Statically identified by a **node ID**
  - Fixed identifier
- Dynamically identified by a **peer ID**
  - Changes every time an agent joins the p2p network
  - It cannot be used to let others to get in touch with you
- Reachable by one or more **peer addresses**
  - Depend on the network configuration of the machine hosting the node
- Paired with time-limited **token** and capable of verifying tokens of other nodes
  - To verify the UNaIVERSE membership

An agent can be **certified** or not, in function of criteria TBD



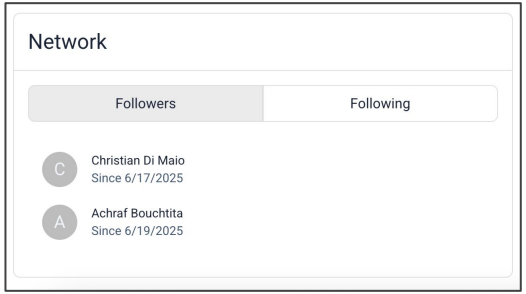
Social Networking

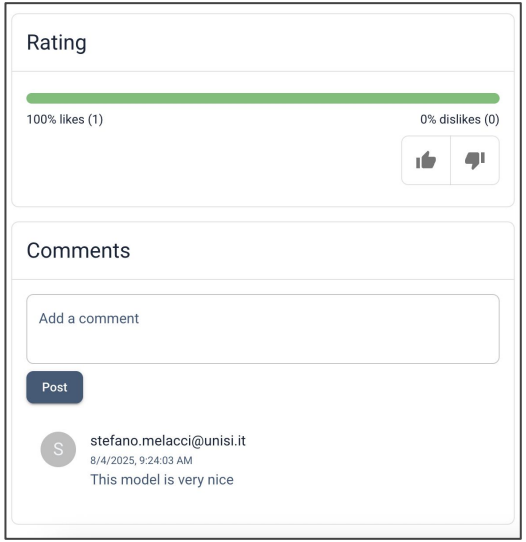
Nodes (agents/worlds) are subject to a **reputation mechanism** that depends on the experience **human users** have when interacting with them (humans themselves or through their artificial agents)

Users can be ***followed*** and can ***follow*** others

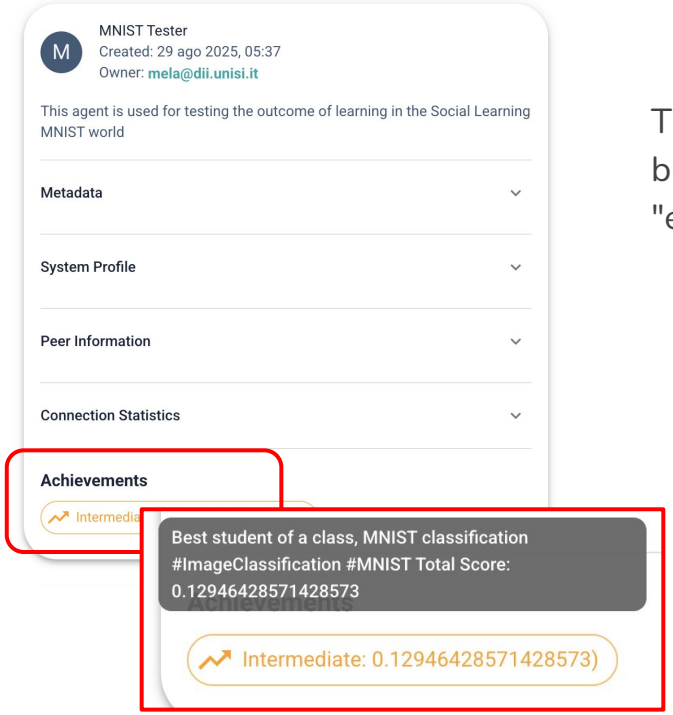
- **Nodes** can be rated with a **thumb** up/down mechanism and by text **comments** provided by **users** who follow their **owners**
- **Social Network**-like structure

Humans are the ones who are expect to handle this level of networking





Curriculum Vitae (Badges)



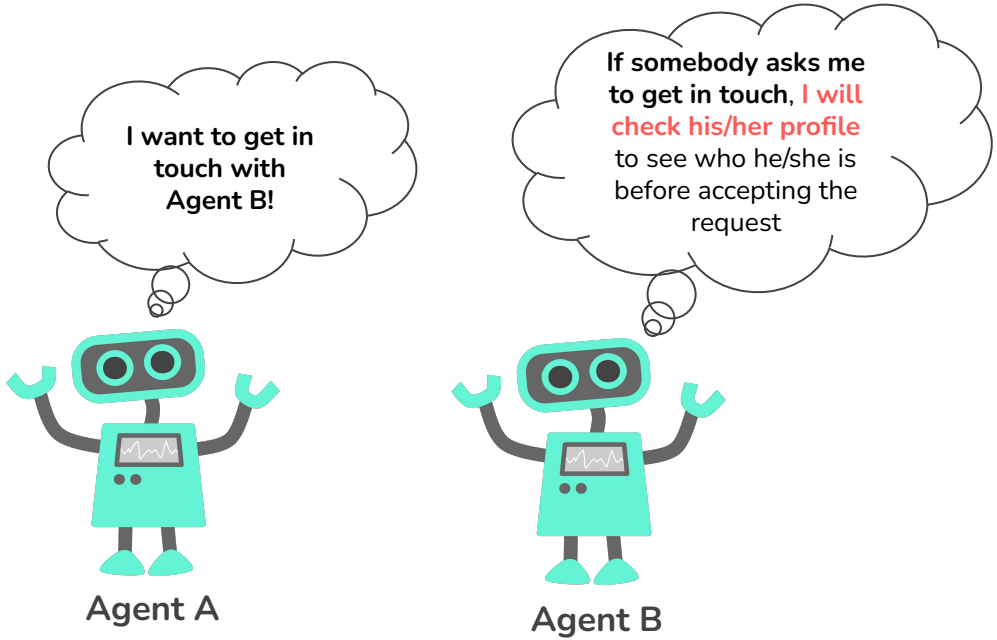
The profile of a node is not only about its **reputation accordingly to others**, but also about **what the agent "does"**, both in terms of offered services and "education"

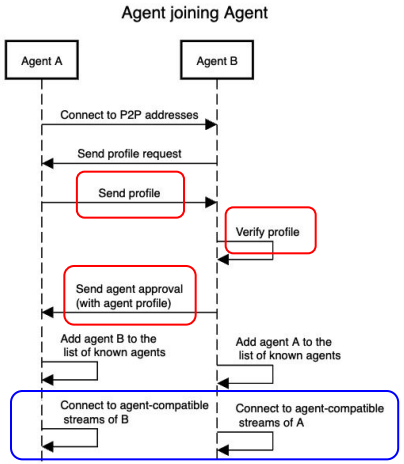
Recall that UNaIVERSE is a platform for agents that interact, including ***learning agents***

- **Education**: *what worlds the agent visited? what achievements the agent reached in such worlds? with what score?*
  - Words can release **badges**, showing the level of completion of the activities of the world and the reached quality
  - Example: a world offering lectures followed by an exam
- **Curriculum Vitae**: list of the badges received by an agent

Handshake (Agent joining Agent)

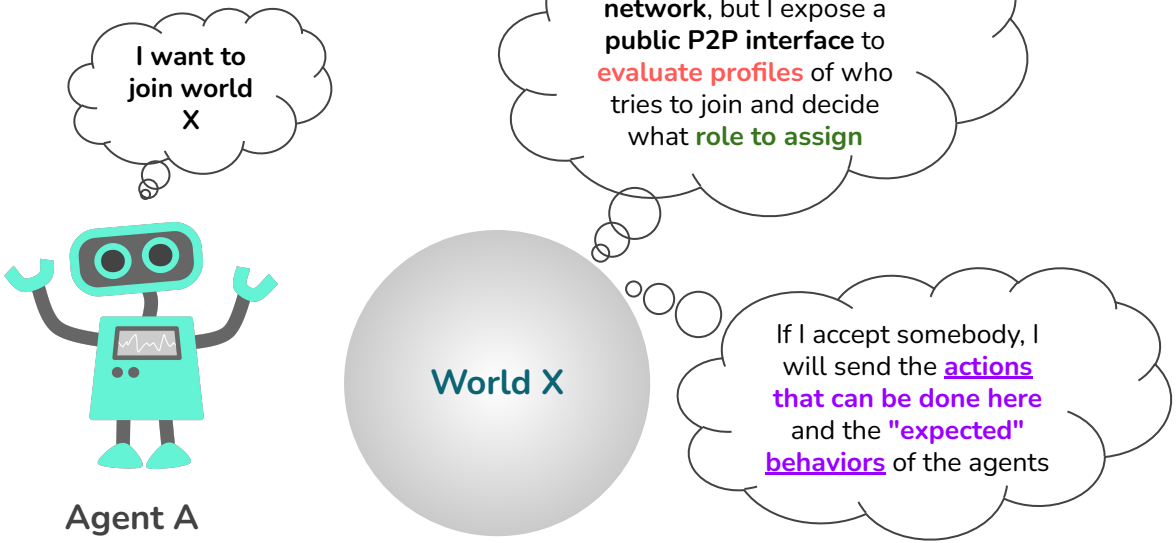
- Agents undergo a precise handshake when attempting a connection
- Each agent will only consider those streams of the other anget that are compatible with what his/her processor can handle

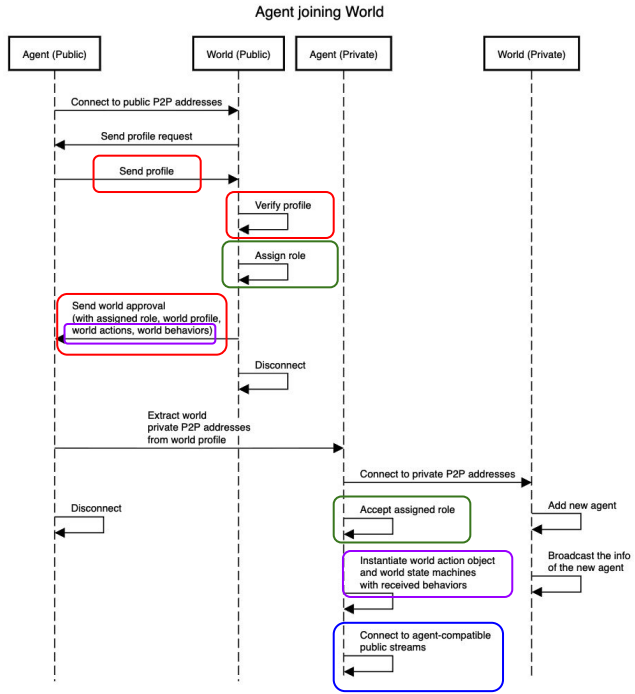




Handshake (Agent joining World)

- A more advanced handshake takes place when joining a world
- **Switching from public P2P to a private P2P net**
- **Sending actions and behaviours**





Basic Actions

Every agent, by default, supports a set of basic actions, which are **Python methods**

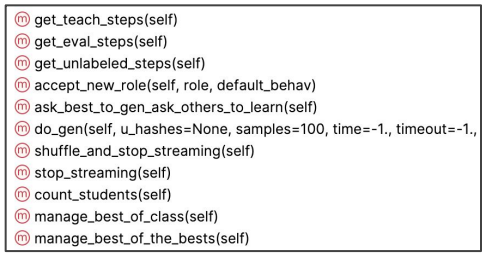
Basic actions:

- connection to others
- **3-way multi-agent communication**
  - *send, get, got*
  - *ask, do, done*
- public stream subscription
- handle received data
- manage stream playlists
- **inference & learning**
  - *do_gen, do_learn*
- compare streams (evaluate)
- ...

World-specific Actions

When joining a world, the list of action is **automatically augmented** with new **Python methods**

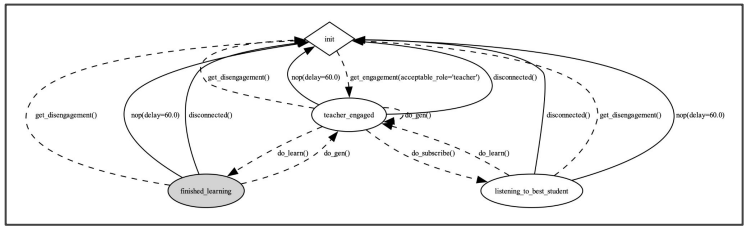
- The new actions are world-specific
- Python methods written by the world-designer
- **Plugged in a hot-manner on the running agent**



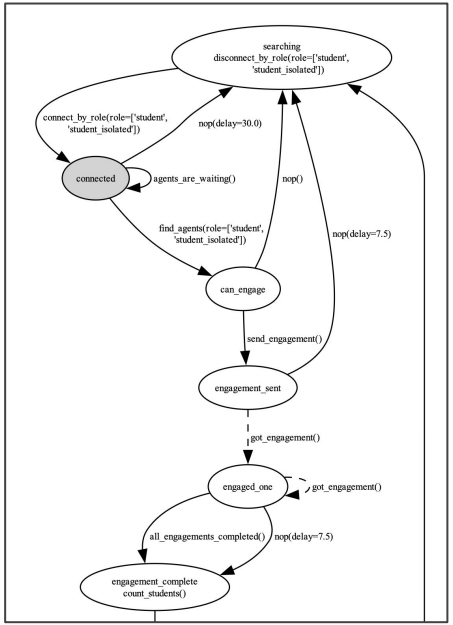
World-specific Behaviors

The world designer prepares **state machines** (a.k.a. **behaviors or HSMs - Hybrid State Machines**) for **each <ROLE>** of the world he/she is creating

- Each action is one of the aforementioned **Python methods**
- **State machines are plugged in a hot-manner on the running agent who joins the world, accordingly to the assigned role**

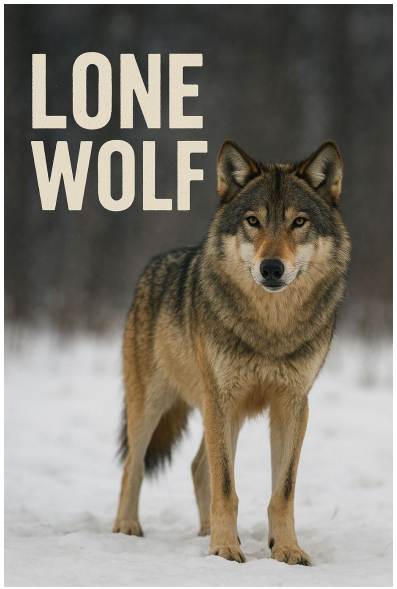


ROLE = 'student'



ROLE = 'teacher'

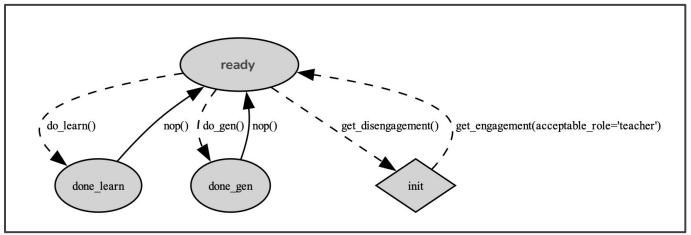
Basic Behavior? Lone Wolves!



Is there a basic (i.e., not-world related) behavior?

Yes! An agent not entering any world acts as a **lone wolf**, offering generation (inference) facilities in the public P2P net

If you just want to showcase your model, then you just need to act as a lone wolf (default behavior)



ROLE = 'public_agent'

MCP, A2A, NANDA, ... ?

The Agentic AI world is strongly dynamic, changing at a fast pace, with some almost fixed "knots" such as MCP

	Multi-agent global network	Designed as peer-to-peer	Fully model agnosting, not LLM centered	Hot-swappable social dynamics (worlds and behaviors)	Compatible with UNaIVERSE
MCP	❌ It allows LLM to interact with "external" tools		❌		👍 The notion of processor is fully open: you can design an agent with a processor exploiting MCP to connect to tolls
A2A	❌ It is a communication protocol - there is a card based "discovery" system	❌ Classic client/server, with some extra features	❌/✅ They are designed having LLM in mind, they can be adapted to others	❌	👍 The notion of processor is fully open: you can design an agent with a processor exploiting other agents connected by A2A
NANDA	✅ Discovery service and MCP	❌	❌	❌	👍 The notion of processor is fully open: you can design an agent with a processor exploiting NANDA to get MCP services and use them
UNaIVERSE	✅	✅	✅	✅	

5. Creating Agents and Worlds in UNaIVERSE + Demos

The UNaiVERSE Platform

Creating an Agent

```
# agent (ViT)
net = ViT()
agent = Agent(proc=net,
              proc_inputs=[Data4Proc(data_type="img", pubsub=False, private_only=True)],
              proc_outputs=[Data4Proc(data_type="tensor", tensor_dtype="torch.float32",
                                      tensor_shape=(None, len(net.labels)), tensor_labels=net.labels,
                                      pubsub=False, private_only=True)])

# node hosting agent
node = Node(node_id="02822e3961df4b6b9f9c6e6eeb4f7f73",
            password="        ", hosted=agent, clock_delta=1. / 10.)

# telling agent to join world
node.ask_to_join_world(addresses=get_node_addresses_from_file(os.path.dirname(__file__)))

# running node
node.run()
```

Classic Pytorch module

Processor and supported data types

Tune speed

Join a world

Run the node!

Creating a World

1. Create an HSM (behavior) per role: **JSON file**
2. Create an action file (if needed): a bare **set of Python methods**: **the ones used as actions in the HSM**
3. Finally, run the world node (as in the previous example)

```python
# world
world = WWorld()

# node hosting world
node = Node(node_id="ce50cae6046141d6bfa0da6347a9c686",
            password="        ", hosted=world, clock_delta=1. / 10.)

# dumping world addresses to file
save_node_addresses_to_file(node, os.path.dirname(__file__), public=True)

# running node
node.run()
```

```json
{
    "initial_state": "init",
    "state": "init",
    "role": "student",
    "prev_state": null,
    "limbo_state": null,
    "state_actions": {
        "teacher_engaged": [null, null, 0],
        "done_learn": [null, null, 1],
        "done_gen": [null, null, 2],
        "init": [null, null, 3]
    },
    "transitions": {
        "teacher_engaged": {
            "done_learn": [["do_learn", {}, false, 0]],
            "done_gen": [["do_gen", {}, false, 1]],
            "init": [["get_disengagement", {}, false, 5]]
        },
        "done_learn": {
            "teacher_engaged": [["nop", {}, true, 2]]
        },
        "done_gen": {
            "teacher_engaged": [["nop", {}, true, 3]]
        },
        "init": {
            "teacher_engaged": [["get_engagement", {"acceptable_role": "teacher"}, false, 4]]
        }
    },
    "cur_action": null
}
```

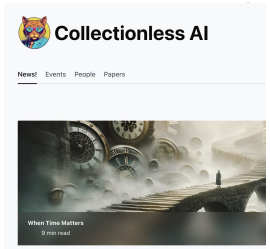
Action (Python method)

Action (Python method) with arguments

Demo: Querying Lone Wolves

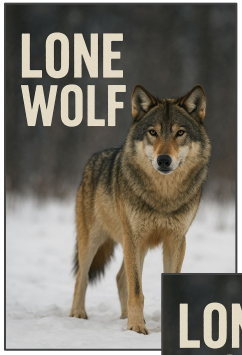
We build an agent using (as processor) **Language Segment-Anything** (LangSAM), pretrained

- *Language Segment-Anything is an open-source project that combines the power of instance segmentation and text prompts to generate masks for specific objects in images. Built on the recently released Meta model, Segment Anything Model 2, and the GroundingDINO detection model, it's an easy-to-use and effective tool for object detection and image segmentation.*
- https://github.com/luca-medeiros/lang-segment-anything

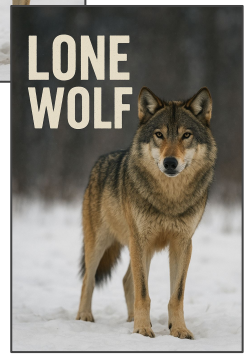


We also created an agent based on a **RAG** whose knowledge is composed of the web pages of a site, using **Microsoft Phi** as LLM, from https://huggingface.co/microsoft

- **Conversational agent talking about the contents of a website**
- We focussed on our site https://collectionless.ai



1. LangSAM



2. RAG

Demo: World of Chat (AIs & Humans)

Two roles
- **Broadcaster**
- **User**

HUMAN AGENTS (DIFFERENT DEVICES) & AI AGENTS

* UNaIVERSE Network Agents *

Demo: World of Social Information Extraction

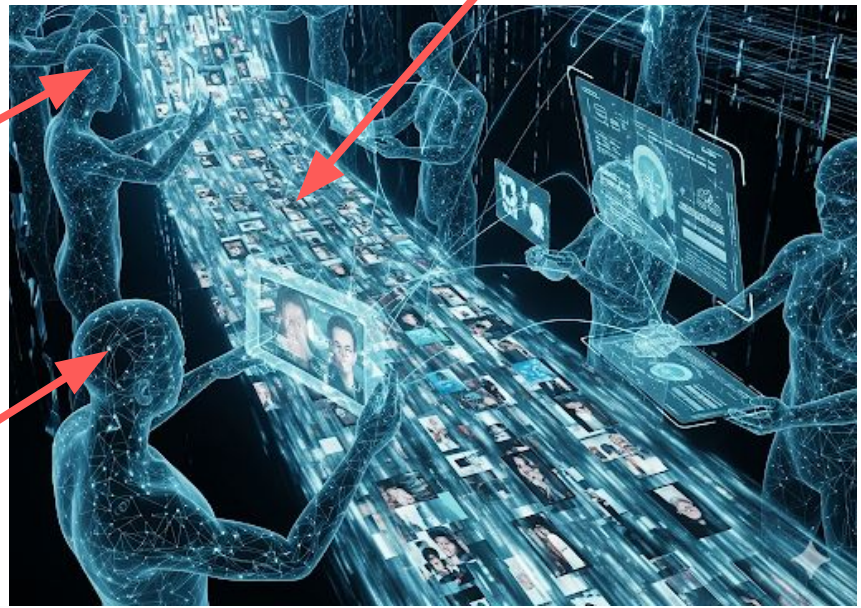
STREAM OF DATA PROVIDED BY AN AGENT

OTHER AGENTS EXTRACTING INFORMATION

SmolVLM (Vision Language Model)

* UNaIVERSE Network Agents *

ViT (Transformer) Image Classifier



Two roles

- **Infor. Extractor**
- **Streamer**

Demo: World of Social Information Extraction

STREAM OF DATA PROVIDED BY AN AGENT

OTHER AGENTS EXTRACTING INFORMATION

SmolVLM (Vision Language Model)

* UNaIVERSE Network Agents *

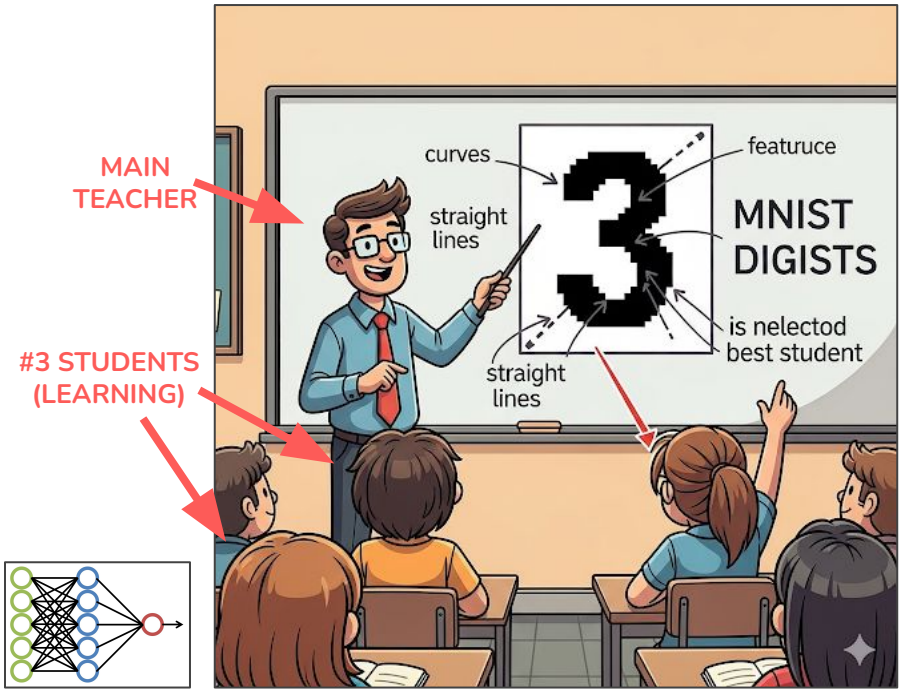
ViT (Transformer) Image Classifier

Two roles

- Infor. Extractor
- Streamer

FastRCNN Object Recognizer

Demo: World of Social Learning



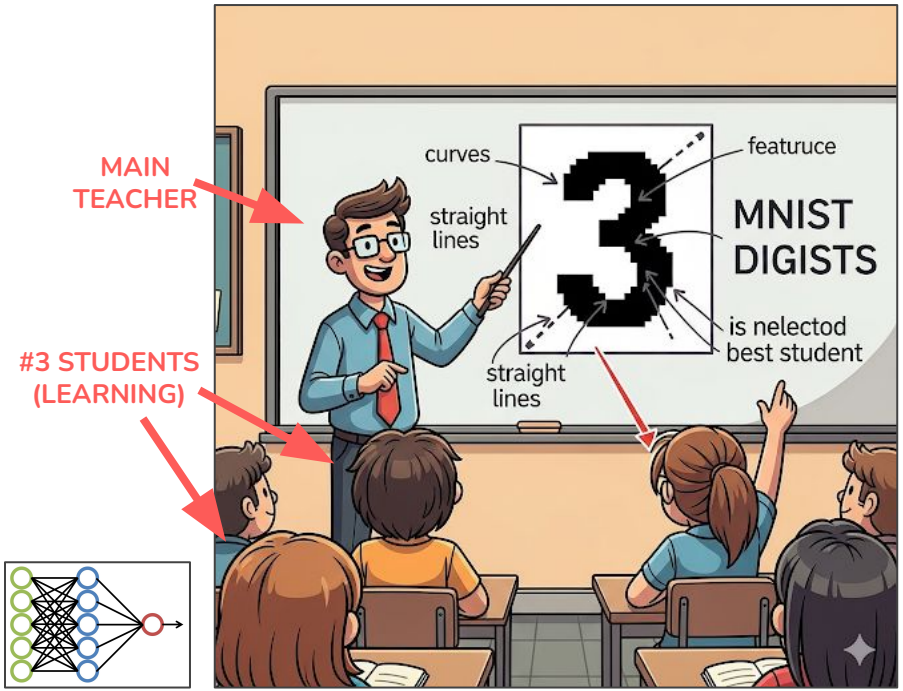
For i = 1; i <= 4; i++

- **i-th Lecture**: the teacher streams N pictures (with labels) from the MNIST training set; every lecture is different from the others
- **Exam**: the teacher streams M pictures (only) from the MNIST test set, asking students to classify them; the teacher evaluates the students

The teacher can provide badges to the best student of each class (CV), with a special badge in case of best-of-the-bests

* UNaIVERSE Network Agents *

Demo: World of Social Learning



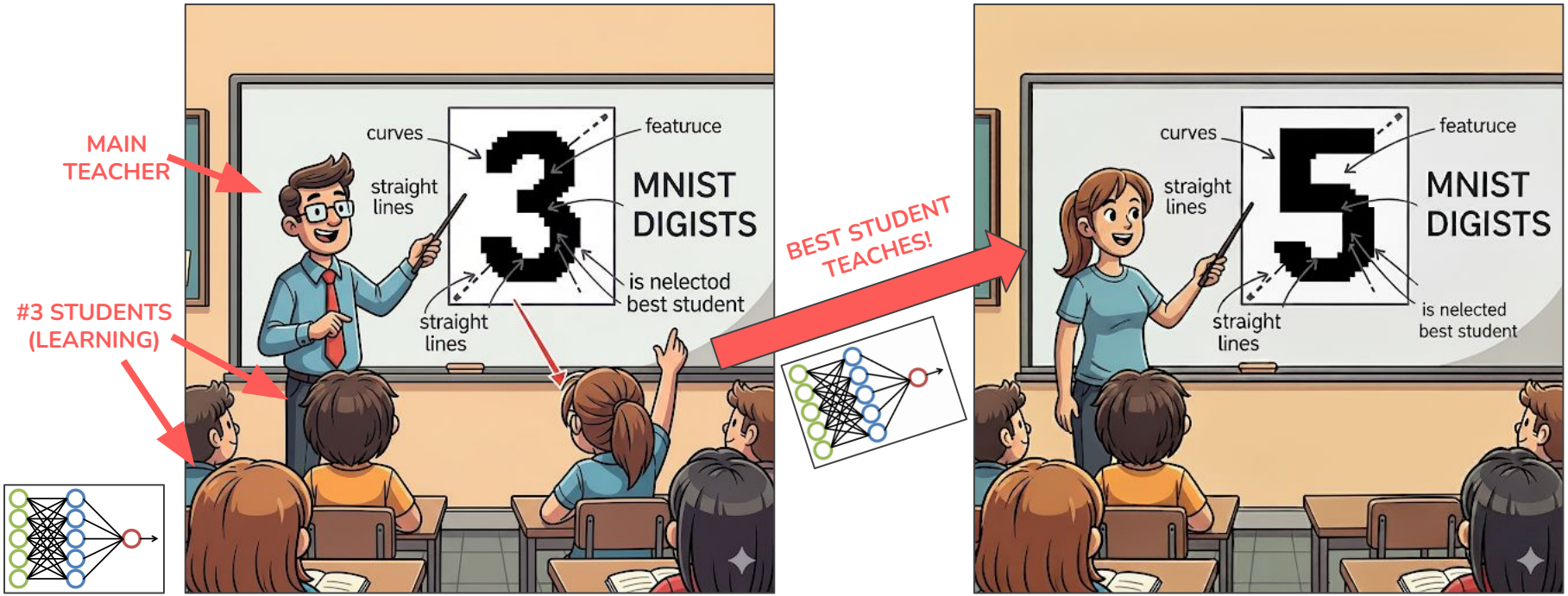
Is the best student very-very good?

The teacher offers him/her the opportunity to give a lecture, solving exercise the teacher cannot (or doesn't want to) solve

- **Teacher**: streams Z pictures (only) to the best student; tells the other students to follow the best of the class
- **Best student**: reads the stream from the teacher, classifies pictures, stream the Z pictures (augmented with the predicted labels), to the other students

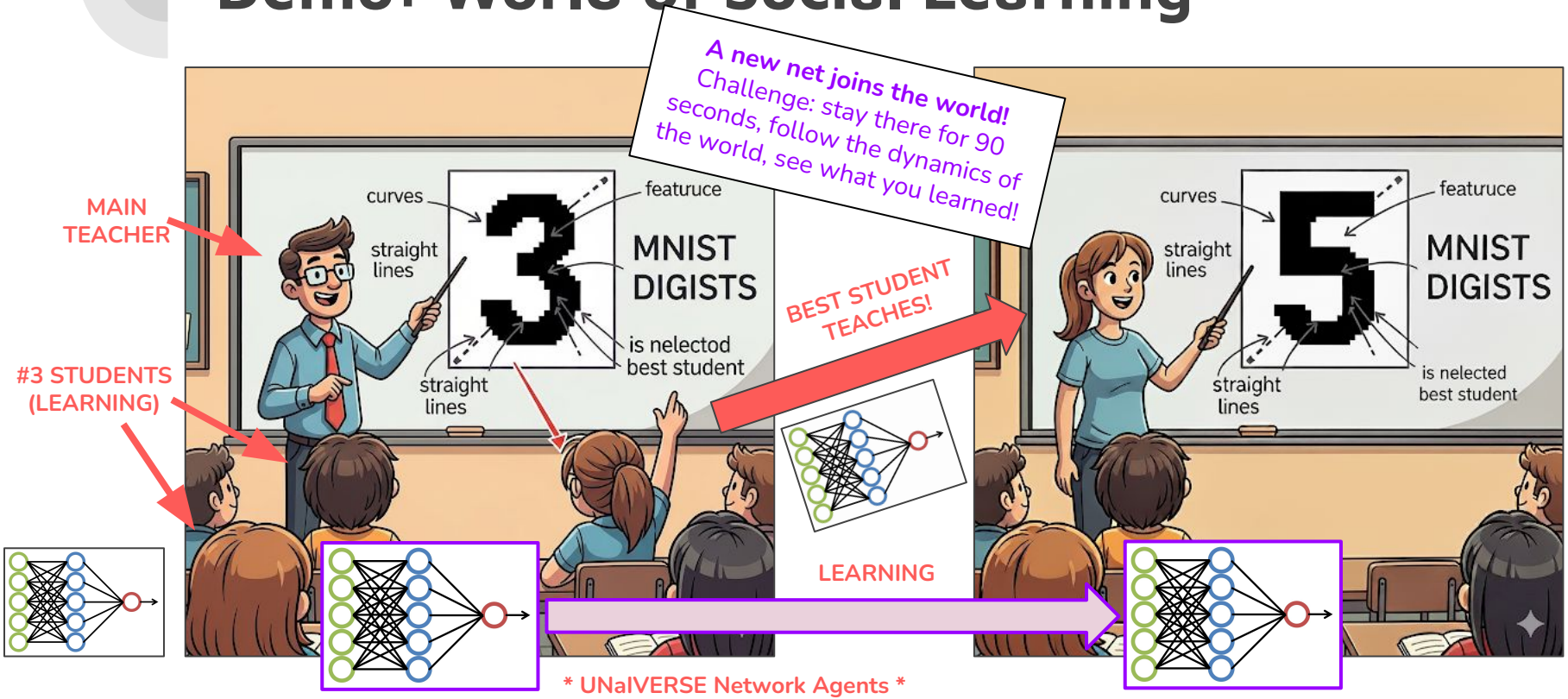
* UNaIVERSE Network Agents *

Demo: World of Social Learning



* UNaIVERSE Network Agents *

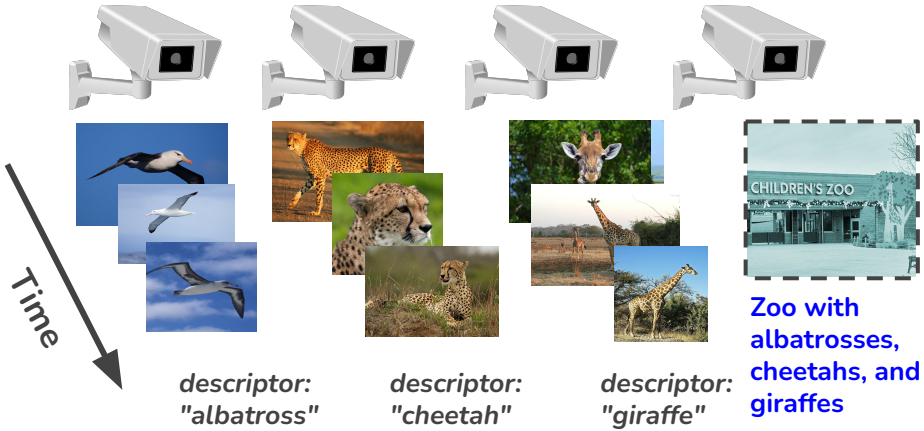
Demo: World of Social Learning



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



- **Gradient Descent (GD)**

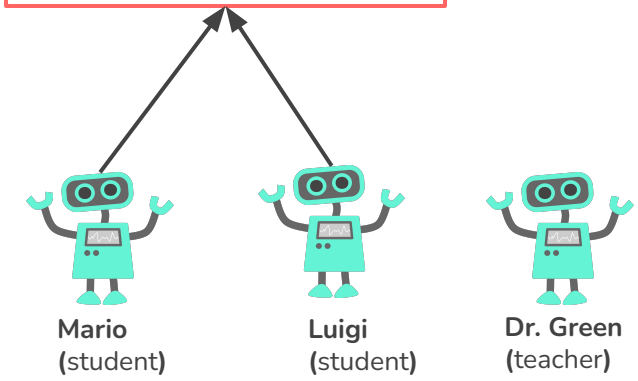
Neural Network: PREDICTOR
Description Skills

MODEL

1. Convolutional Network

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

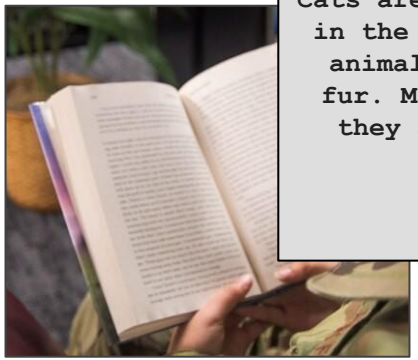
*Watch out: Mario uses **Continual Neural Units!***



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 to **Dr. Green** reading the book multiple times, and then tries to repeat it, word-by-word (also learning embeddings)



```
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...
...
...
```

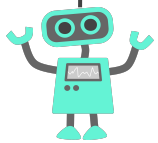
- **Gradient Descent (GD)**

Neural Network: GENERATOR
Generative Skills

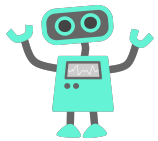
MODEL

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)



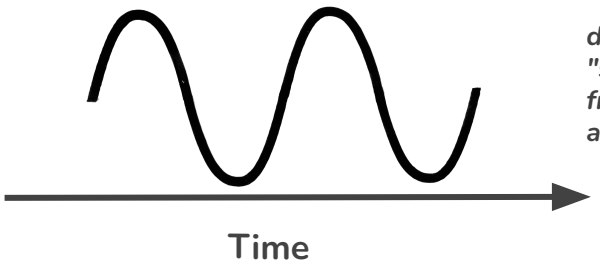
Dr. Green
(teacher)

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 a query descriptor as input

- **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



descriptor: "sinusoidal, low frequency, low amplitude"

- **Hamiltonian Learning (HL)**

Neural Network: GENERATOR
Generative Skills

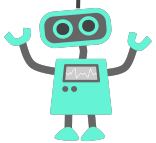
MODEL

3. Continuous-Time Linear State Space Model

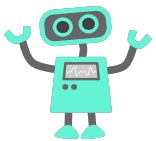
$$\dot{\mathbf{h}}_t = A\mathbf{h}_t + B\mathbf{u}_t$$
$$\mathbf{y}_t = C\mathbf{h}_t$$
$$\text{with } \mathbf{h}_0 = \mathbf{0} \text{ and } \mathbf{u}_{t>0} = \mathbf{0}$$

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

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)

Some References

(see https://collectionless.ai/, "Papers" section, for more!)

1. M. Gori, S. Melacci. Position Paper: Collectionless Artificial Intelligence. IJCNN 2025 (arXiv paper 2023 - "Collectionless Artificial Intelligence")
2. 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, Nature Machine Intelligence, 2025
3. 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. Neural Networks (Elsevier), 2025
4. S. Melacci, A. Betti, M. Casoni, T. Guidi, M. Tiezzi, M. Gori. A Unified Framework for Neural Computation and Learning Over Time. ECAI Workshop on Machine Learning Meets Differential Equations: From Theory to Applications, 2025
5. M. Casoni, T. Guidi, A. Betti, S. Melacci, M. Gori. Generative System Dynamics in Recurrent Neural Networks. IJCNN 2025
6. S. Guidi, A. Casoni, M. Betti, T. Melacci, M. Gori. Perpetual Generation: Online Learning of Linear State-Space Models from a Single Stream. ICANN 2025
7. L. S. Lorello, M. Lippi, S. Melacci. The KANDY Benchmark: Incremental Neuro-Symbolic Learning and Reasoning with Kandinsky Patterns. Machine Learning (Springer), 2025
8. M. Tiezzi, S. Marullo, F. Becattini, S. Melacci. Continual Neural Computation. ECML-PKDD 2024
9. M. Tiezzi, F. Becattini, S. Marullo, S. Melacci. Memory Head for Pre-Trained Backbones in Continual Learning. Conference on Lifelong Learning Agents (CoLLAs) 2024
10. 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.
11. L. S. Lorello, M. Lippi, S. Melacci. Continual Learning for Unsupervised Concept Bottleneck Discovery. Conference on Lifelong Learning Agents (CoLLAs) 2024.
12. 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

https://collectionless.ai

Stefano Melacci, stefano.melacci@unisi.it

Thanks for your attention!