

A Language-based Software Engineering Approach for Cyber-Physical Swarms

Gianluca Aguzzi gianluca.aguzzi@unibo.it
Supervisor: Mirko Viroli mirko.viroli@unibo.it

ALMA MATER STUDIORUM – Università di Bologna

PhD defense

23/06/2024



Cyber-Physical Swarms

Definition

Systems composed of **large** sets of **cyber-physical** components executing **collective** tasks, strongly relying on local component **interactions** (neighborhood-based), and showing **inherent adaptivity, scalability, and robustness** properties.

Applications

- Swarm robotics
- COMMON-WEARS (PRIN)
- Large-scale IoT systems
- Smart cities



Challenges

- (Unwanted) emergents
- Scale
- Failures
- Distributed control
- Complex & layered architectures

Thesis Goal

Thesis Goal

Develop a software engineering framework for CPSWs, incorporating *manual* and *automatic* techniques to enable the design, synthesis, and deployment of *self-organizing* behaviors with *scalable* and *robust* outcomes.

Multi-Faceted Scientific Problem

- **How** to express collective behaviors (algorithms & methodologies)
- **How** to execute specific behaviors (execution models & middleware dynamics)
- **How** to deploy the computation *smartly* (deployment)



State-of-the-art

Manual Design

- **Macroprogramming:** languages with collective abstraction as first citizen of the language.
- **Aggregate computing:** a language for programming collective behaviors based on computational fields.

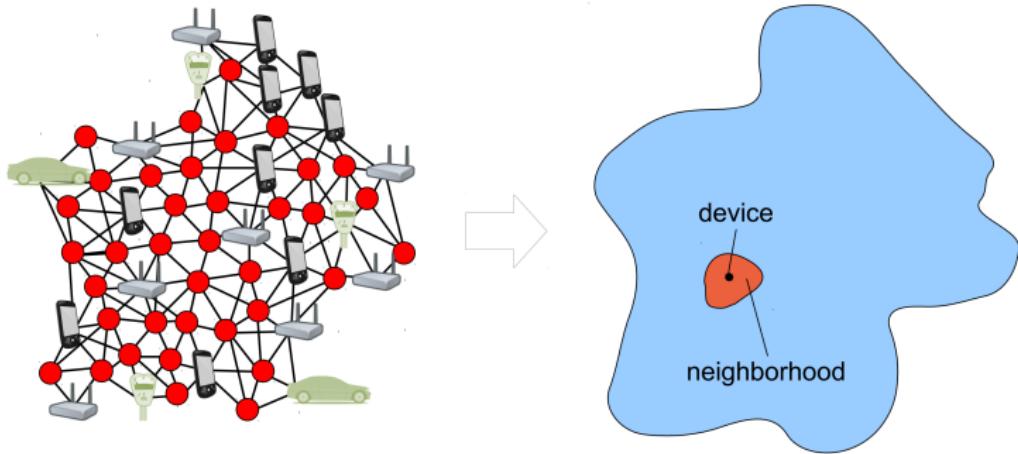
Automatic Design

- **Machine learning:** techniques to learn collective behaviors from data and experiences.
- **Many-agent reinforcement learning:** a multi-agent learning environment in which a large group of agents learn to coordinate their actions to achieve a collective goal.



Aggregate Computing¹

Program the aggregate, not the individual



¹Jacob Beal, Danilo Pianini, and Mirko Viroli. "Aggregate Programming for the Internet of Things". In: *IEEE Computer* 48.9 (2015), pp. 22–30. DOI: 10.1109/MC.2015.261

Aggregate Computing – One slide

Collective Abstraction

computational fields ($dev / evt \mapsto \forall$)

examples: **temperature fields, actuation fields**



Jacob Beal, Danilo Pianini, and Mirko Viroli. "Aggregate Programming for the Internet of Things". In: *IEEE Computer* 48.9 (2015), pp. 22–30. DOI: 10.1109/MC.2015.261

Self-org-like computational model

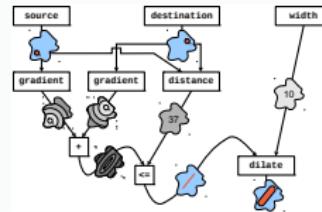
interaction: **repeated** msg exchange with **neighbors**

behavior: **repeated** execution of **async rounds** of **sense – compute – (inter)act**

Programming Model

formal core language: **field calculus**

paradigm: **functional, macro-programming**



Why?

- 👍 Decouple the collective specification from the IT networks
- 👍 Scale naturally with nodes in the systems
- 👍 High-level abstractions to devise applications

Many-agent Reinforcement Learning²³

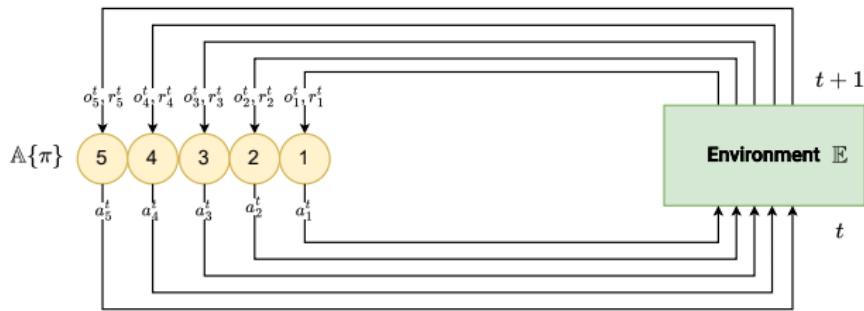
A multi-agent learning environment in which a **large** group agents learn to **coordinate** their actions to achieve a collective goal through a sequential interaction with the environment.

²Yaodong Yang. "Many-agent reinforcement learning". PhD thesis. UCL (University College London), 2021

³Kai Cui et al. "A Survey on Large-Population Systems and Scalable Multi-Agent Reinforcement Learning". In: *CoRR* abs/2209.03859 (2022). DOI: 10.48550/ARXIV.2209.03859. arXiv: 2209.03859. URL: <https://doi.org/10.48550/arXiv.2209.03859>

Many-agent Reinforcement Learning

SwarMDP⁴ is a model composed of a *swarming agent* \mathbb{A} and an *environment* \mathbb{E}



Learning in SwarMDP

- Goal: find a policy π that maximizes the long term collective **return**
- SwarMDP → large-scale handling via a **homogeneous** policy:
 - The same policy is shared within the population.
- Training: **centralized**, but execution is **decentralized** (typically with **shared experiences**)
- Algorithm: **Swarm RL**⁵: A way to learn the Q-function in a swarm of agents.

⁴Adrian Šošić et al. "Inverse reinforcement learning in swarm systems". In: *arXiv preprint arXiv:1602.05450* (2016)

⁵Hitoshi Iima and Yasuaki Kuroe. "Swarm reinforcement learning algorithms based on Sarsa method". In: *2008 SICE Annual Conference*. IEEE. 2008, pp. 2045–2049

Going Beyond the State-of-the-Art

Engineering Perspective

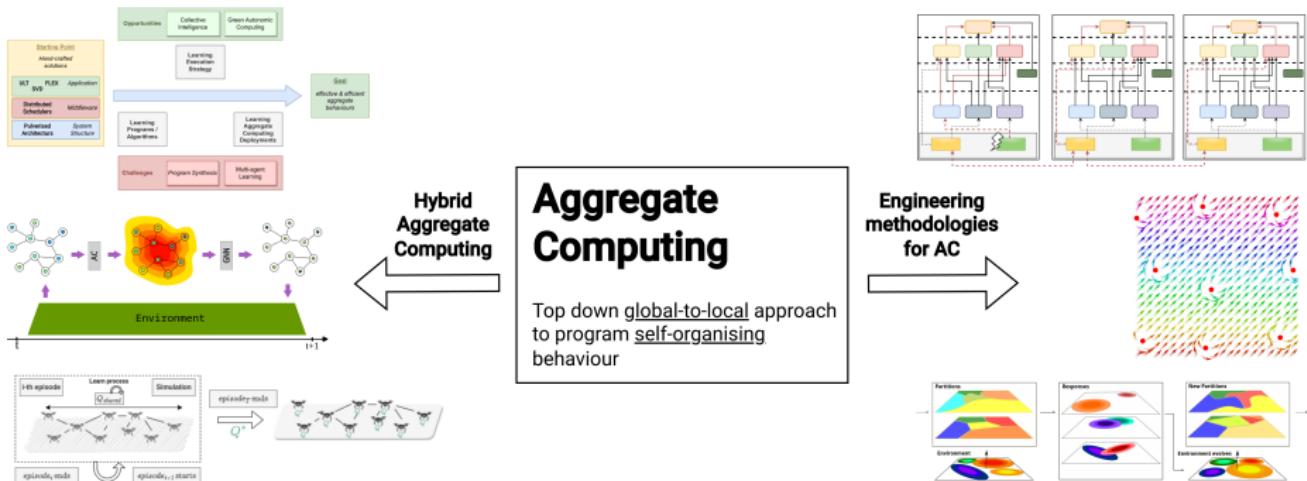
- **Current limitations:** lack of refined abstractions and tools to effectively engineer the complex world of CPSWs.
- **API Design:** bridge the gap between macroprogramming application and real-world applications.
- **Programming Model:** develop an intuitive way to express collective behaviors.
- **Intelligent Deployment:** optimize the deployment of macroprograms across diverse infrastructures, enhancing adaptability and efficiency.
- **Practical Tooling:** create user-friendly tools for prototyping and deploying macroprograms, fostering broader adoption.

Learning Perspective

- **Current limitations:** learning and macroprogramming are often treated as separate entities, limiting the potential of both.
- **Macroprogramming for Learning:** utilize macroprogramming to build robust and resilient learning infrastructures, e.g., for federated learning or multi-agent RL.
- **Learning for macroprogramming:** leverage machine learning techniques to dynamically optimize and adapt the macroprogramming model itself.

Contribution – One slide

Language-based software engineering for CPSWs



Hybrid Aggregate Computing ⁹

Areas of Improvement:

① Execution Policies for Efficiency:

- **Adaptive Collective Schedulers⁶:** *shared-experience* Q-learning to dynamically optimize execution for convergence speed or resource consumption.
- **Self-Stabilizing Systems:** focus on reducing consumption while maintaining the self-healing properties of collectives.

② Collective Program Synthesis:

- **Program Sketching⁷:** Q Learning for completing partial programs, tailoring coordination blocks to specific environmental dynamics (e.g., mobility, failures).
- **Goal:** automate the generation of adaptable blocks, overcoming limitations of heuristic-based approaches.

③ Many-Agent RL through AC:

- **Field-Informed RL⁸:** aggregated information (fields) guides policy learning in complex CPSW environments.
- **Graph Neural Networks:** leverage graph neural networks for policy approximation, achieving faster and more scalable learning.

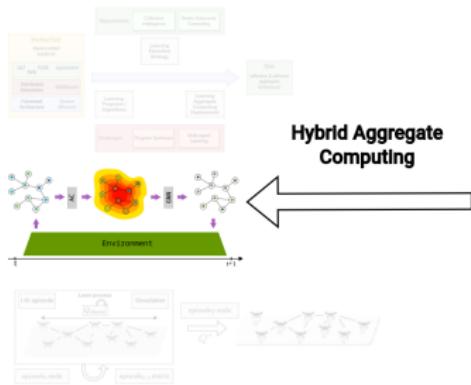
⁶ Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. “Addressing Collective Computations Efficiency: Towards a Platform-level Reinforcement Learning Approach”. In: *ACSOS 2022*. Ed. by Roberto Casadei et al. IEEE, 2022, pp. 11–20

⁷ Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. “Towards Reinforcement Learning-based Aggregate Computing”. In: *COORDINATION 2022, Proceedings*. Ed. by Maurice H. ter Beek and Marjan Sirjani. Vol. 13271. Lecture Notes in Computer Science. Springer, 2022, pp. 72–91

⁸ Gianluca Aguzzi, Mirko Viroli, and Lukas Esterle. “Field-informed Reinforcement Learning of Collective Tasks with Graph Neural Networks”. In: *ACSOS 2023*. 2023

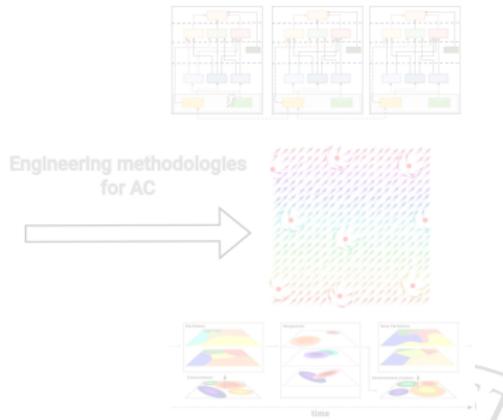
⁹ Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. “Machine Learning for Aggregate Computing: a Research Roadmap”. In: *DISCOLI 2022, July 10, 2022*. IEEE, 2022, pp. 119–124

AC for Learning: Field-Informed Reinforcement Learning



Aggregate Computing

Top down global-to-local approach
to program self-organising
behaviour



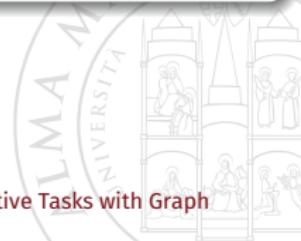
AC for Learning: Field-Informed Reinforcement Learning

Problem

- **Collective aspect:** the state typically lacks a global representation → partial observability.
 - It is built through communication with neighbors.
- **Variable neighborhood:** the need for a policy that accounts for neighborhood variability.

Contribution: Field-Informed Reinforcement Learning¹⁰

- **Fields:** used to produce a stigmergic representation of the environment.
- **GNN:** policy function approximator that handles neighborhood variability.
- **Graph-Deep Q Learning:** the learning algorithm.

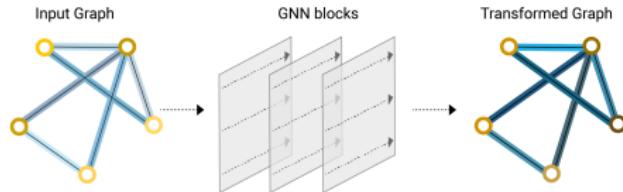


¹⁰ Gianluca Aguzzi, Mirko Viroli, and Lukas Esterle. "Field-informed Reinforcement Learning of Collective Tasks with Graph Neural Networks". In: ACSOS 2023. 2023

FIRL: Graph Neural Networks

What

A neural network architecture designed to work with **graph-structured** data.



How

- **Message passing:** update node representations based on their neighborhood.
- **Aggregation:** combine information from neighboring nodes.
- **Update:** modify node representations based on the aggregated information.

Why

- **Dual Interpretation:** offers both global and local perspectives on graph data.
- **Handles Variability (Topology):** effectively addresses the varying structure and size of neighborhoods in graphs.

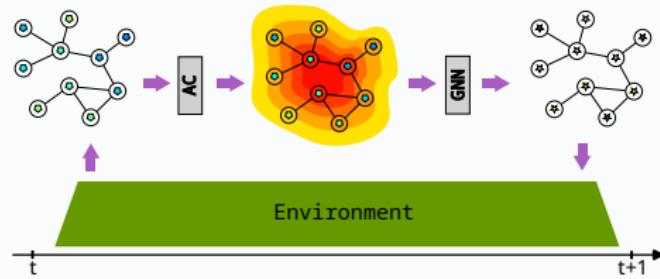
FIRL: Graph-Deep Q Learning

What

A learning algorithm which uses GNN as a Q function approximator

How

- **Graph Experience replay:** store the experience in a replay buffer under the form of graphs.
- **Learning step:** follow the swarMDP model.



Why

- Handle many-agent through shared experience pattern
- Fast and scalable learning
- Follow a centralized learning approach but with a decentralized execution (thanks to GNN)

FIRL: Algorithm

Input: Environment \mathbb{E} , Graph Replay \mathcal{D} , Target Network θ^- , Current Network θ , Exploration ϵ

Initialize $\mathcal{D}, \theta, \theta^- \leftarrow \theta$;

while not done **do**

 Observe G_o ;

$$a \leftarrow \begin{cases} \text{random action} & \text{if random } < \epsilon \\ \operatorname{argmax}_{a_v} Q(G_o, \theta)[v](a_v) & \text{otherwise} \end{cases};$$

 Execute a , observe G_r, G'_o ;

 Store (G_o, a, G_r, G'_o) in \mathcal{D} ;

 Sample batch $(G_o^b, a^b, G_r^b, G'_o)$ from \mathcal{D} ;

$$y_v \leftarrow G_r^b[v] + \gamma \max_{a'} Q(G_o'^b[v], a'; \theta^-);$$

$$y_v^* \leftarrow Q(G_o^b[v], a^b[v]; \theta);$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \frac{1}{|G^b|} (y - y^*)^2;$$

if C steps **then**

$$|\quad \theta^- \leftarrow \theta$$

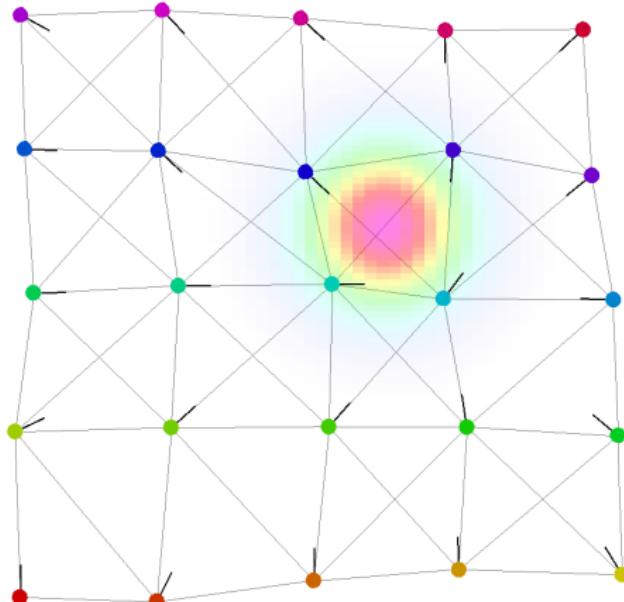
end

end



FIRL: Simulation Setup

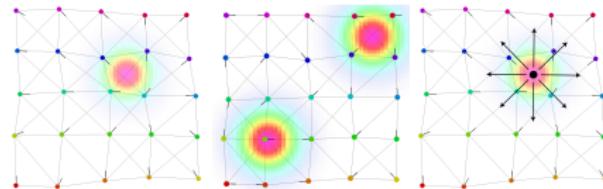
- **Setup:** group of nodes in a 2D space, neighborhood based on spatial proximity.
- **Goal:** maximize the coverage of a spatial phenomenon (fire, pollution, etc.).
- **Rewards:**
 - *Inside Phenomenon:* maximize the number of nodes inside the phenomenon.
 - *Cohesion:* minimize the number of isolated nodes.
 - *Coverage:* maximize the intersection of covered areas.
- **States:** relative position of nodes and potential field of the phenomenon.



FIRL: Evaluation

Three different scenarios:

- **Static:** the phenomenon is static.
- **Multiple:** there are multiple phenomena.
- **Dynamic:** the phenomenon is moving.

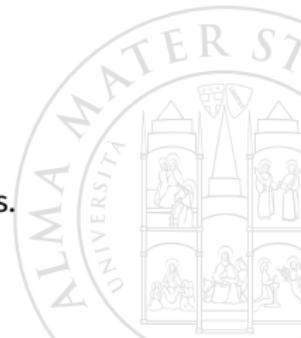


Under three different solutions:

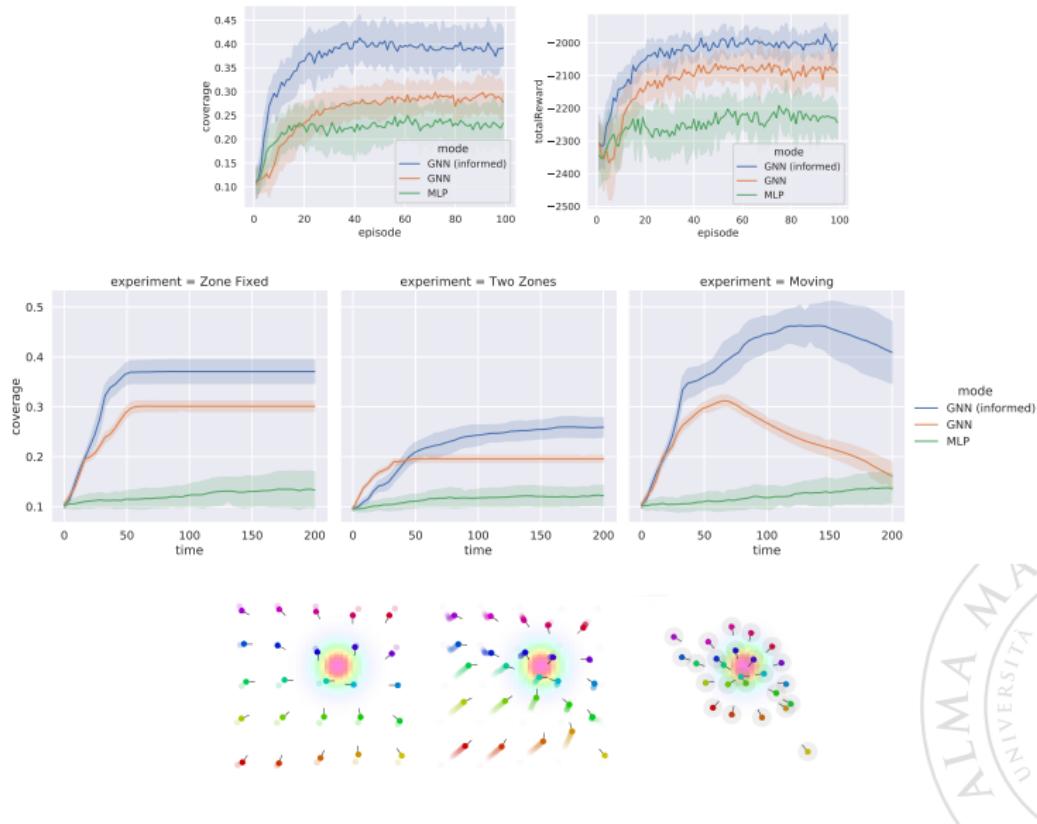
- **FIRL:** the proposed solution.
- **Graph Q-learning:** GNN without AC.
- **Deep Q Learning + AC:** MLP with AC.

Evaluation divided into:

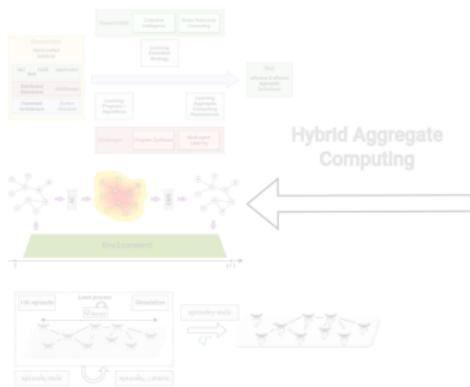
- ① **Training Phase:** performed in the fixed scenario for a total of 2000 steps.
- ② **Evaluation Phase:** used all scenarios for 64 different runs.



FIRL: Results I



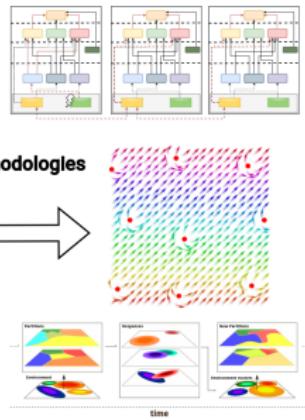
Engineering Methodologies



Aggregate Computing

Top down global-to-local approach
to program self-organising
behaviour

Engineering methodologies
for AC



Engineering Methodologies I

Areas of Improvement:

① Programming Models & APIs:

- **FRASP:**¹¹ functional reactive approach for self-organization programming (distributed, inspired by functional reactive programming).
- **MacroSwarm:**¹² API for expressing and composing swarm behaviors (extensive, covers various behaviors like aggregation and flocking).

② Swarm Distributed Sensing

- **Swarm Sensing Clustering**¹³: distributed clustering based on spatial-phenomena.
- **Dynamic Decentralization Domains**¹⁴: distributed and fault-tolerant pattern for collective sensing and acting.

¹¹Roberto Casadei et al. "Self-Organisation Programming: A Functional Reactive Macro Approach". In: *4th IEEE International Conference on Autonomic Computing and Self-Organizing Systems - ACSOS 2023, Toronto*. 2023

¹²Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. "MacroSwarm: A Field-Based Compositional Framework for Swarm Programming". In: *Coordination Models and Languages*. Cham: Springer Nature Switzerland, 2023

¹³Gianluca Aguzzi et al. "A field-based computing approach to sensing-driven clustering in robot swarms". In: *Swarm Intelligence* (Sept. 2022)

¹⁴Gianluca Aguzzi et al. "Dynamic decentralization domains for the internet of things". In: *IEEE Internet Computing* 26.6 (2022), pp. 16–23

Engineering Methodologies II

Areas of Improvement:

③ DevOps; Deployments:

- **DevOps for CPSW**¹⁵: continuous integration of software and physical parts of the system.
- **Pulverized Architecture**¹⁶ opportunistic deployment of aggregate programs.

④ Tools:

- **ScaRLib**¹⁷: a for cooperative many-agent reinforcement learning.
- **ScaFi**¹⁸-**Web**¹⁹: web-based simulator for aggregate computing (Teaching and prototyping).

¹⁵ Roberto Casadei, Danilo Pianini, and Gianluca Aguzzi. "Towards Automated Engineering for Collective Adaptive Systems: Vision and Research Directions". In: *PiCom 2022*. IEEE. 2022, pp. 1–6

¹⁶ Gianluca Aguzzi et al. "Towards Pulverised Architectures for Collective Adaptive Systems through Multi-Tier Programming". In: *ECAS 2021*. IEEE. 2021, pp. 99–104

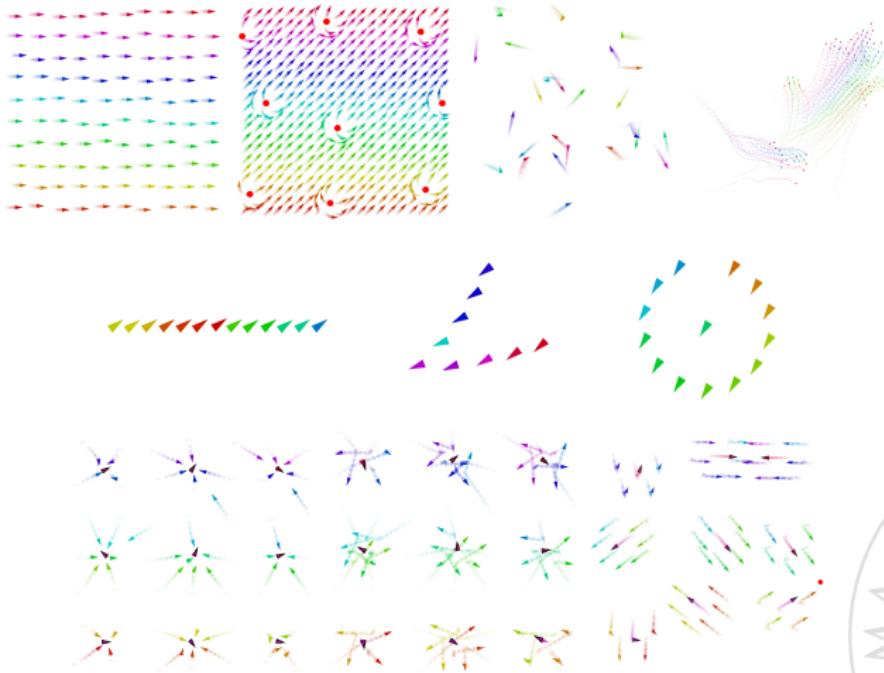
¹⁷ Davide Domini et al. "ScaRLib: A Framework for Cooperative Many Agent Deep Reinforcement Learning in Scala". In: *International Conference on Coordination Languages and Models*. Springer Nature Switzerland Cham. 2023, pp. 52–70

¹⁸ Roberto Casadei et al. "ScaFi: A Scala DSL and toolkit for aggregate programming". In: *SoftwareX* 20 (2022), p. 101248

¹⁹ Gianluca Aguzzi et al. "ScaFi-Web: a web-based application for field-based coordination programming". In: *COORDINATION 2021*. Springer International Publishing. 2021, pp. 285–299

MacroSwarm – Overview

A high-level²⁰ compositional framework for expressing and composing swarm behaviors



²⁰ Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. "MacroSwarm: A Field-Based Compositional Framework for Swarm Programming". In: *Coordination Models and Languages*. Cham: Springer Nature Switzerland, 2023

Conclusions and Future Works

Conclusions

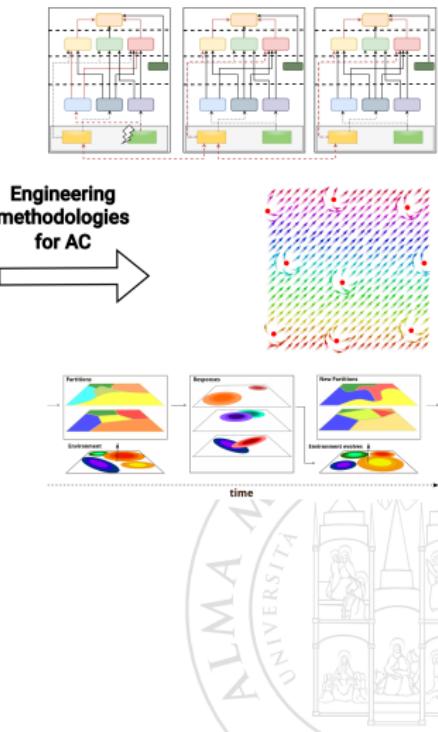
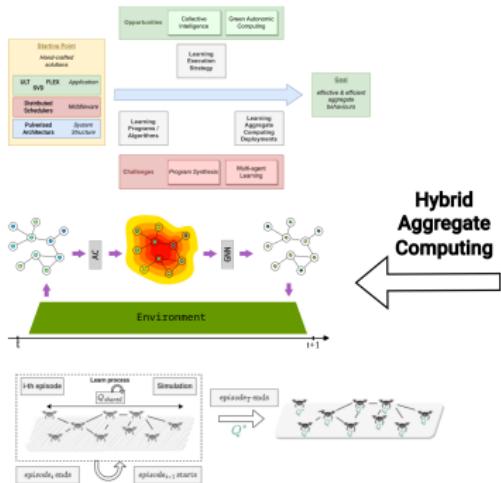
- This thesis discusses novel approaches to the engineering CPSW applications
- This poses the basis for a *language-based* approach to CPSW, putting together:
 - *Learning*: field-informed reinforcement learning, collective program sketching, distributed schedulers
 - *Methodologies*: FRASP, MacroSwarm, Swarm-Sensing API
 - *Tools*: ScaFi, ScaRLib, ScaFi-Web

Future Works

- *Close the reality gap*: deploy the aggregate programs on real CPSW, improve the toolchain, leverage realistic simulators (Gazebo, etc.)
- *Improve the learning algorithms*: learn to deploy, learn to adjust runtime execution, etc.
- *AC for learning*: aggregate computing as a way to create resilient infrastructure for learning (e.g., federated learning)



Thank you!



References I

- [ACV22a] Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. "Addressing Collective Computations Efficiency: Towards a Platform-level Reinforcement Learning Approach". In: *ACSOS 2022*. Ed. by Roberto Casadei et al. IEEE, 2022, pp. 11–20.
- [ACV22b] Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. "Machine Learning for Aggregate Computing: a Research Roadmap". In: *DISCOLI 2022, July 10, 2022*. IEEE, 2022, pp. 119–124.
- [ACV22c] Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. "Towards Reinforcement Learning-based Aggregate Computing". In: *COORDINATION 2022, Proceedings*. Ed. by Maurice H. ter Beek and Marjan Sirjani. Vol. 13271. Lecture Notes in Computer Science. Springer, 2022, pp. 72–91.
- [ACV23] Gianluca Aguzzi, Roberto Casadei, and Mirko Viroli. "MacroSwarm: A Field-Based Compositional Framework for Swarm Programming". In: *Coordination Models and Languages*. Cham: Springer Nature Switzerland, 2023.
- [Agu+21a] Gianluca Aguzzi et al. "ScaFi-Web: a web-based application for field-based coordination programming". In: *COORDINATION 2021*. Springer International Publishing, 2021, pp. 285–299.
- [Agu+21b] Gianluca Aguzzi et al. "Towards Pulverised Architectures for Collective Adaptive Systems through Multi-Tier Programming". In: *ECAS 2021*. IEEE, 2021, pp. 99–104.
- [Agu+22a] Gianluca Aguzzi et al. "A field-based computing approach to sensing-driven clustering in robot swarms". In: *Swarm Intelligence* (Sept. 2022).
- [Agu+22b] Gianluca Aguzzi et al. "Dynamic decentralization domains for the internet of things". In: *IEEE Internet Computing* 26.6 (2022), pp. 16–23.
- [AVE23] Gianluca Aguzzi, Mirko Viroli, and Lukas Esterle. "Field-informed Reinforcement Learning of Collective Tasks with Graph Neural Networks". In: *ACSOS 2023*. 2023.
- [BPV15] Jacob Beal, Danilo Pianini, and Mirko Viroli. "Aggregate Programming for the Internet of Things". In: *IEEE Computer* 48.9 (2015), pp. 22–30. DOI: 10.1109/MC.2015.261.
- [Cas+22] Roberto Casadei et al. "ScaFi: A Scala DSL and toolkit for aggregate programming". In: *SoftwareX* 20 (2022), p. 101248.

References II

- [Cas+23] Roberto Casadei et al. "Self-Organisation Programming: A Functional Reactive Macro Approach". In: *4th IEEE International Conference on Autonomic Computing and Self-Organizing Systems - ACSOS 2023, Toronto*. 2023.
- [CPA22] Roberto Casadei, Danilo Pianini, and Gianluca Aguzzi. "Towards Automated Engineering for Collective Adaptive Systems: Vision and Research Directions". In: *PiCom 2022*. IEEE. 2022, pp. 1–6.
- [Cui+22] Kai Cui et al. "A Survey on Large-Population Systems and Scalable Multi-Agent Reinforcement Learning". In: *CoRR abs/2209.03859* (2022). DOI: 10.48550/ARXIV.2209.03859. arXiv: 2209.03859. URL: <https://doi.org/10.48550/arXiv.2209.03859>.
- [Dom+23] Davide Domini et al. "ScaRLib: A Framework for Cooperative Many Agent Deep Reinforcement Learning in Scala". In: *International Conference on Coordination Languages and Models*. Springer Nature Switzerland Cham. 2023, pp. 52–70.
- [IK08] Hitoshi Iima and Yasuaki Kuroe. "Swarm reinforcement learning algorithms based on Sarsa method". In: *2008 SICE Annual Conference*. IEEE. 2008, pp. 2045–2049.
- [Šoš+16] Adrian Šošić et al. "Inverse reinforcement learning in swarm systems". In: *arXiv preprint arXiv:1602.05450* (2016).
- [Yan21] Yaodong Yang. "Many-agent reinforcement learning". PhD thesis. UCL (University College London), 2021.

