



SwarmBox:

A Plug-and-Play Drone Swarm Framework for Streamlined Development and Comprehensive Analysis

Minki LEE¹, Seojin LEE², Seulbae KIM¹

¹Pohang University of Science and Technology (POSTECH)

²Daegu Gyeongbuk Institute of Science and Technology (DGIST)



Drone Swarm: breaks the limit.

Drone Swarm: A decentralized, collaborative multi-agent system tackling shared objectives.



Drone Swarm: breaks the limit?

Drone swarms are expected to break the boundaries of human activities via **efficient, scalable, flexible, and resilient** operations.



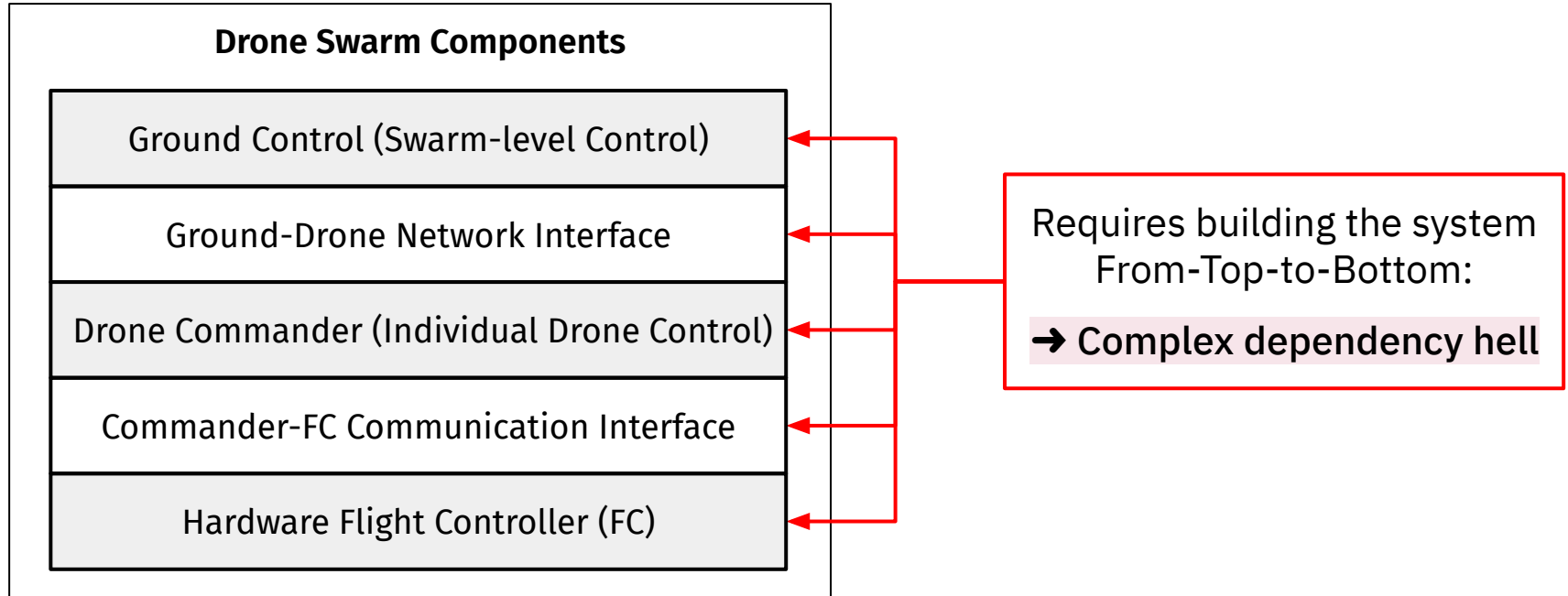
Disaster Response



Infrastructure Surveillance

AI generated images used.

Complexity of Drone Swarm System



Fragmentation of Swarm Ecosystem

Researchers had to constantly “reinvent the wheel.”

Flocking Behavior



Obstacle Avoidance



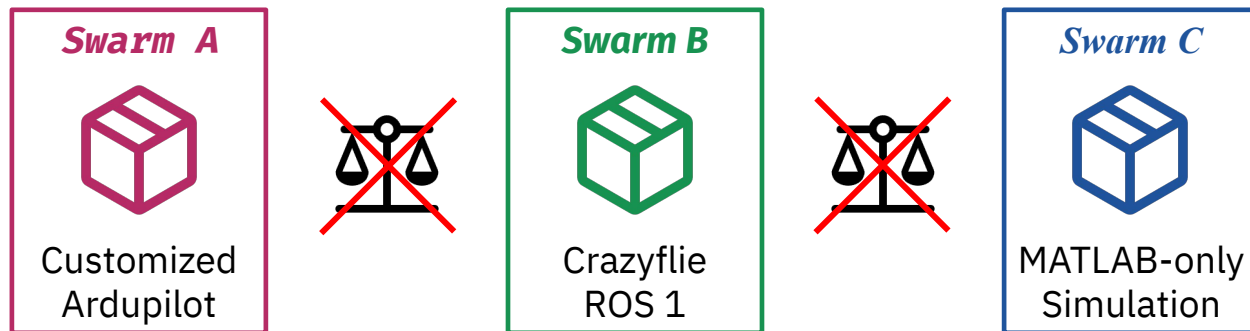
Swarm Simulator



→ Fragmented ecosystem; existing works are rarely maintained.

Incomparability of existing works

Bespoke testbeds of existing works have architectural differences.

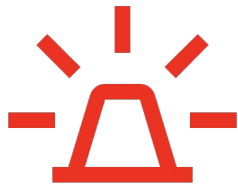


→ Hindering reproducibility, making fair comparison impossible.

Lack of Swarm-level Observability

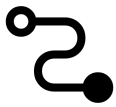
What if there's a failure in the swarm?

Researchers have to manually debug each layer upon failure.



Swarm System
Failure (Crash)



	Control log	Sensor log	Trajectory
Drone #1			
Drone #2			
Drone #3			
...	...	...	...



→ Hard to observe and understand swarm-emergent behaviors.

Challenges in Drone Swarm Development

Fragmentation

Lack of standardized interfaces forces bespoke testbeds.



Complexity

Tight coupling of flight control, middleware, and swarm logic.



Observability

Absence of time-synchronized, swarm-level analysis tools.



Incomparability

Architectural differences make results difficult to reproduce.



SwarmBox:

A plug-and-play drone swarm foundation
for reproducible research & development.

From Challenges to Design Goals

Fragmentation

Lack of standardized interfaces forces bespoke testbeds.



Reusable and Effortless Integration

Complexity

Tight coupling of flight control, middleware, and swarm logic.



Abstraction of Multi-layered System Complexity

Observability

Absence of time-synchronized, swarm-level analysis tools.



Comprehensive Swarm-Level Analysis Support

Incomparability

Architectural differences make results difficult to reproduce.

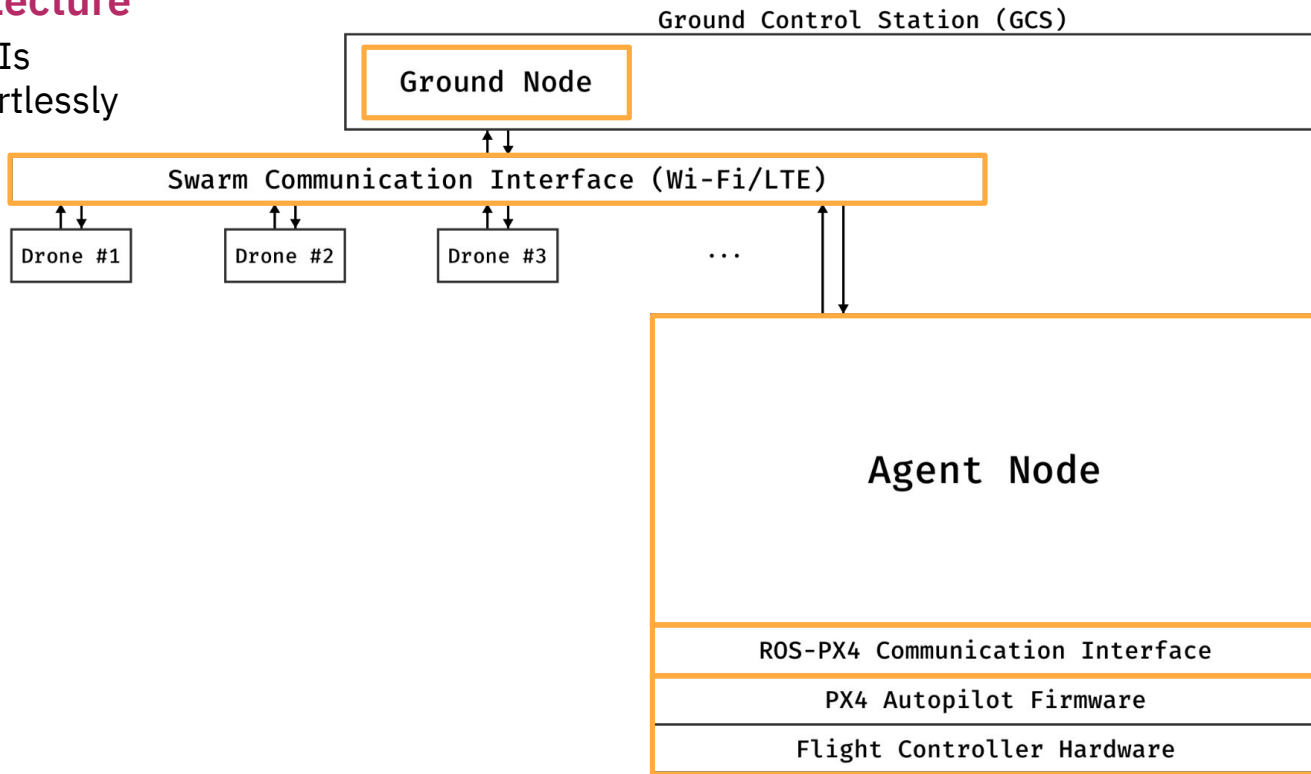


Reproducible and Standardized Execution

Reusable and Effortless Integration

Generalized Architecture

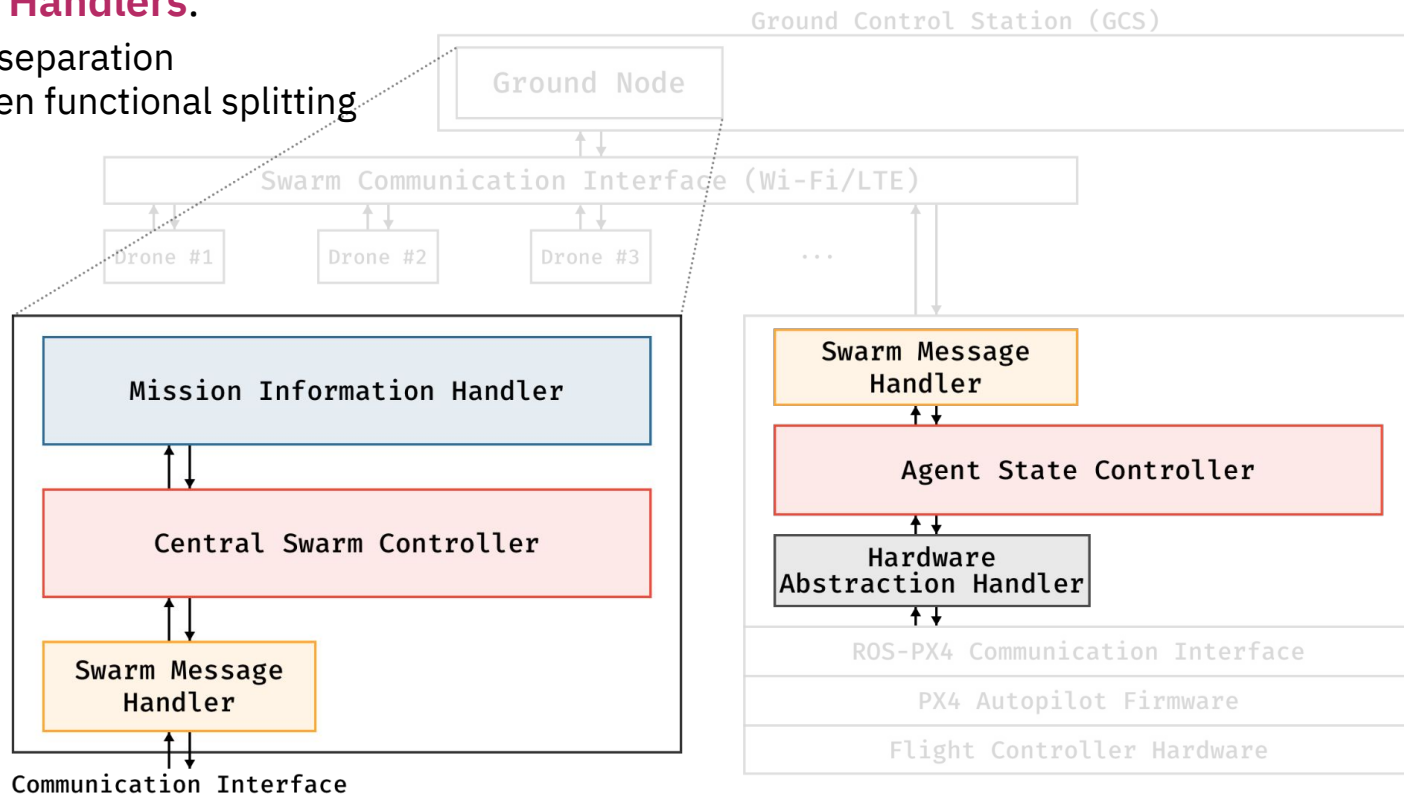
- lifecycle-based APIs
- plug-and-play effortlessly



Abstraction of Multi-layered System Complexity

4 specialized Handlers:

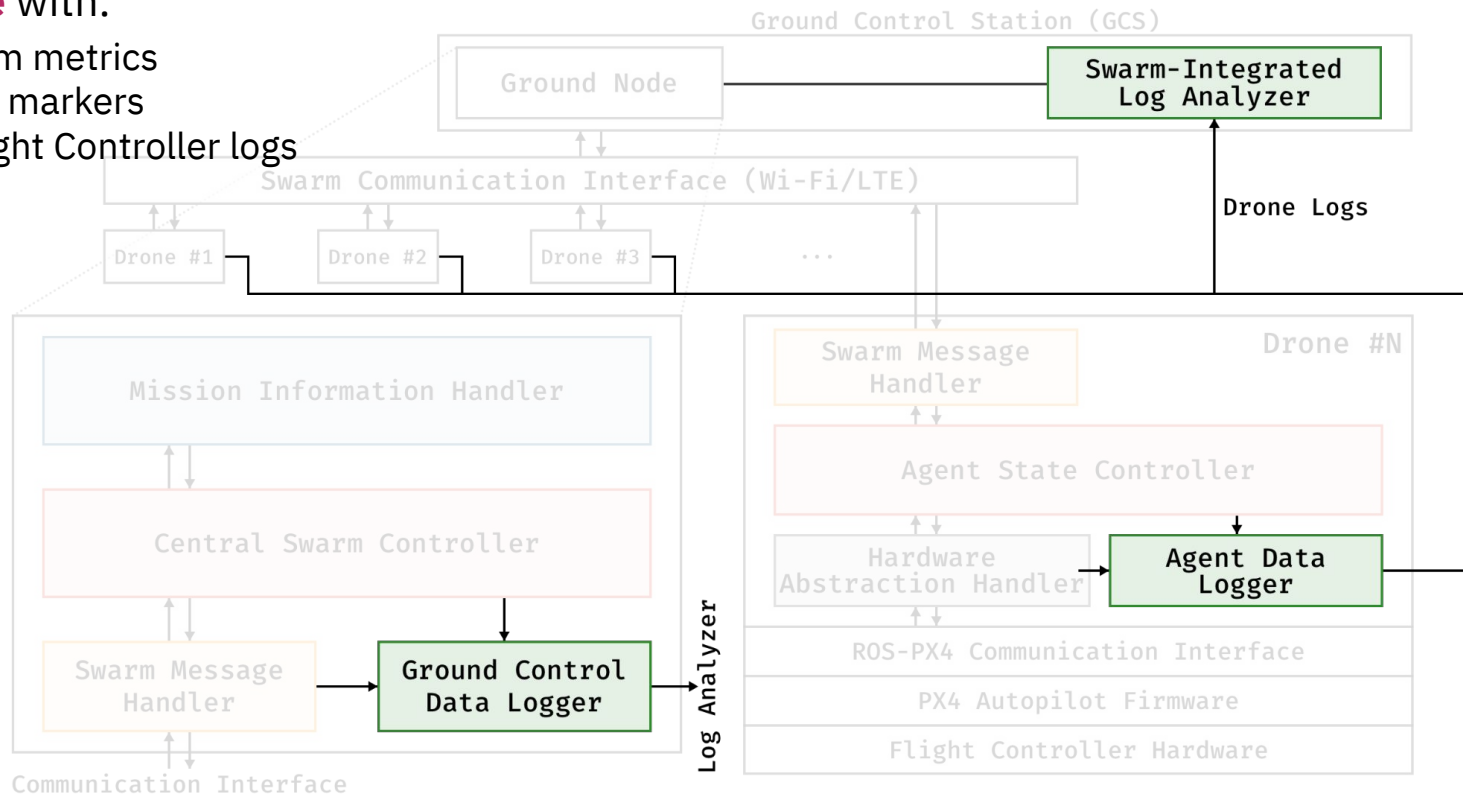
- Hierarchical separation
- Context-driven functional splitting



Comprehensive Swarm-Level Analysis Support

Log & Analyze with:

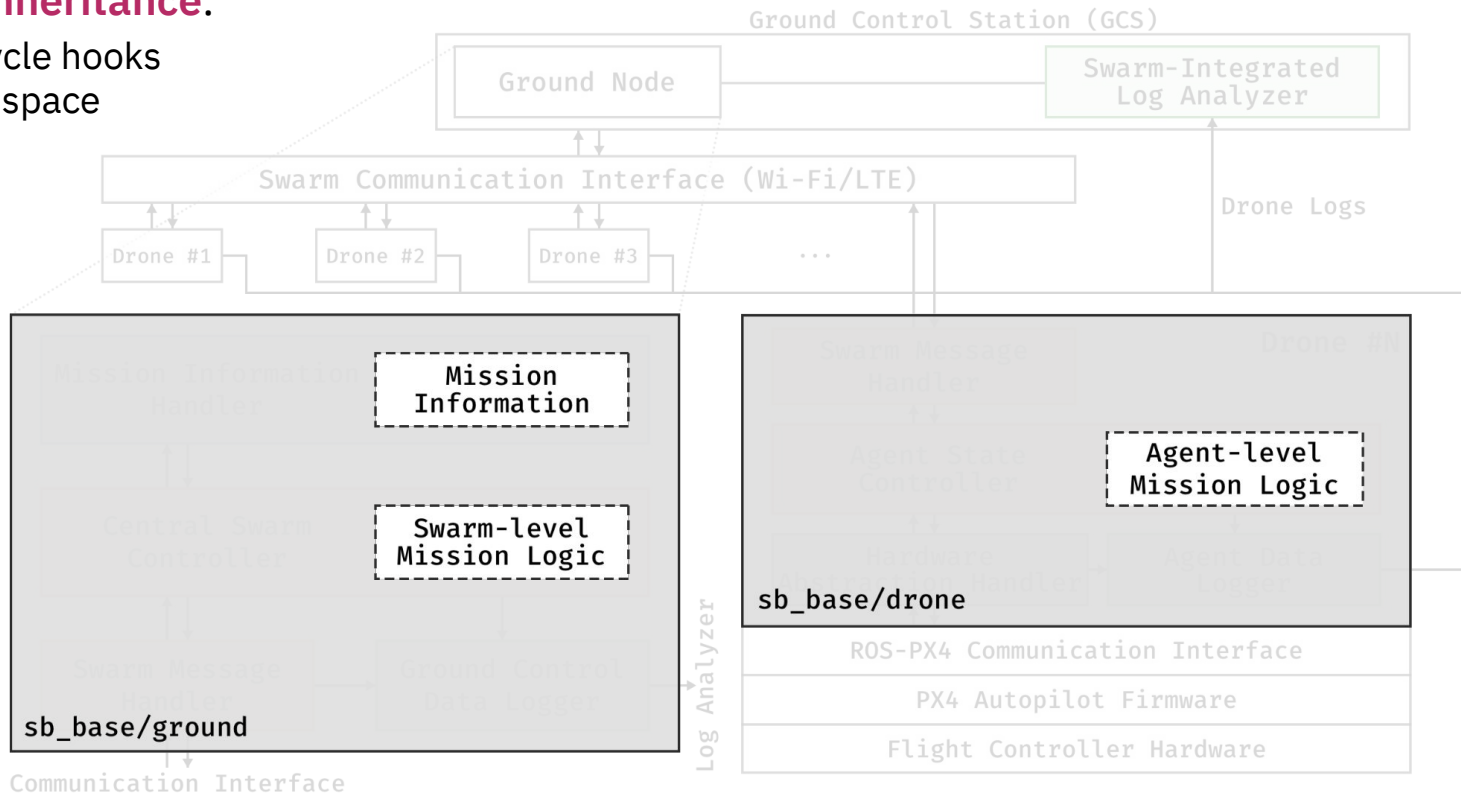
- Built-in swarm metrics
- User-defined markers
- Low-level Flight Controller logs



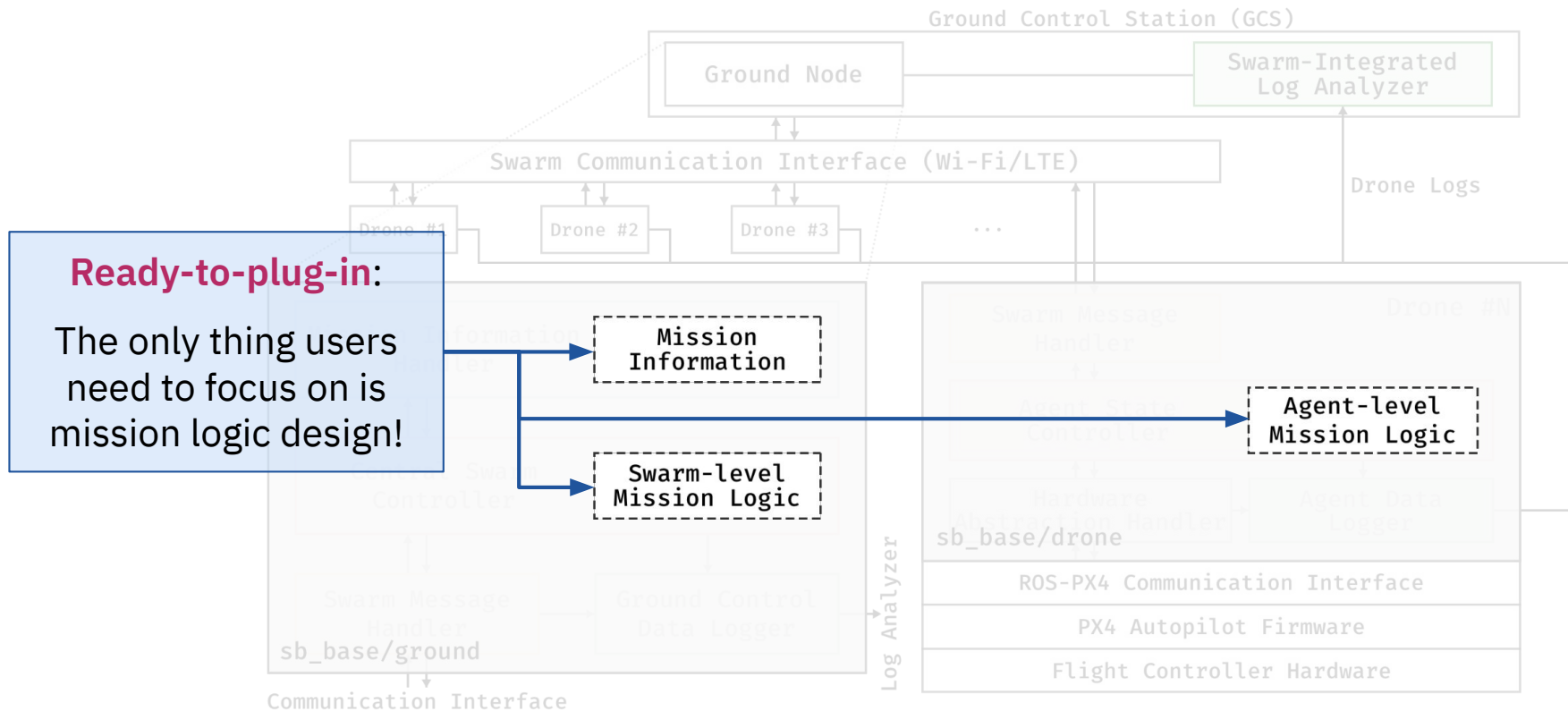
Reproducible and Standardized Execution

Hierarchical inheritance:

- Drop-in lifecycle hooks
- Minimal user space



Reproducible and Standardized Execution



Evaluation Setup

Software Stack

PX4 Autopilot

v1.15

ROS 2

Humble

Ubuntu

22.04.5

Execution Environments

SITL (Simulation-In-The-Loop):

Fully simulated on Workstation

SIH (Simulation-In-Hardware):

Simulated physics, Computation on FC

Physical:

Real-World Flight, Computation on FC

Hardware Setup

in-silico (SITL/SIH):

CPU: AMD 9800X3D

RAM: 64GB DDR5

Gazebo Simulator

in-situ (SIH/Physical):

Holybro X500 V2

Pixhawk 6C

Raspberry Pi 3 Model B

Evaluation Criteria

Evaluated 5 core capabilities required for drone swarms

Engineering Effort Reduction



How effectively does SwarmBox reduce the engineering effort?

Sim-to-Real Fidelity



Does SwarmBox maintain high fidelity across sim-to-real environments?

Swarm-Level Observability



Can SwarmBox help diagnosing emergent, swarm-level faults?

Benchmarking Capability



Can we utilize SwarmBox for a fair-comparison benchmarking?

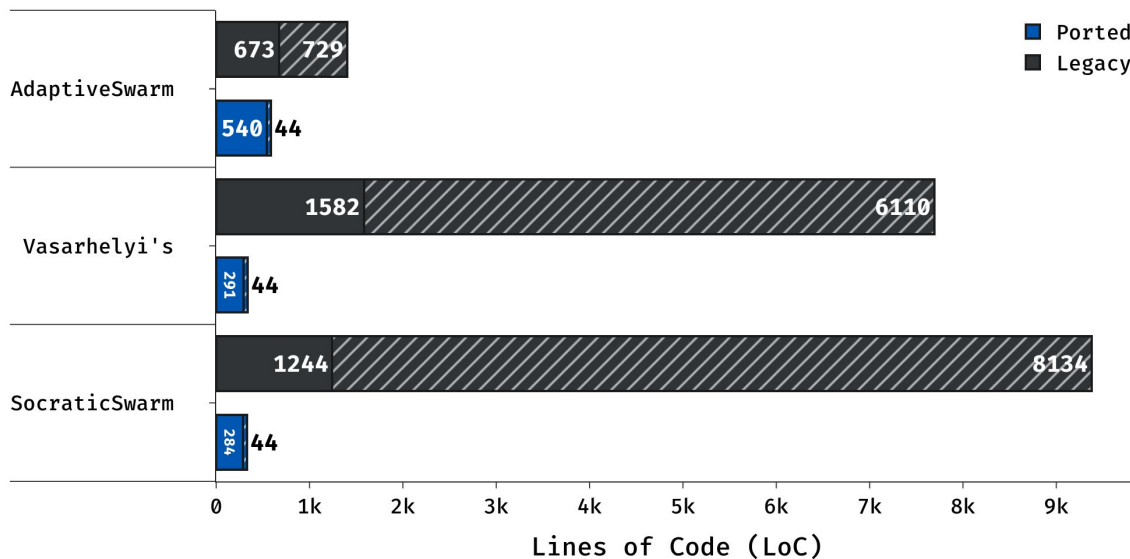
Generality & Scalability: *Is SwarmBox suitable for various missions and large-scale usages?*



RQ1. Engineering Effort Reduction

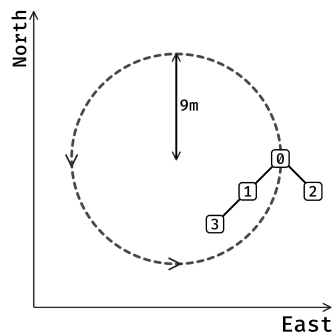


Up to **99.5%** reduction of boilerplate implementation

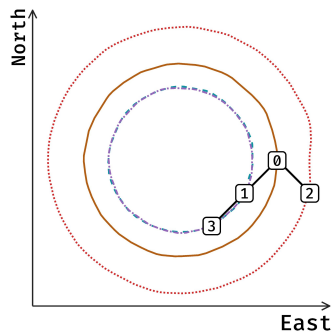


SwarmBox removes boilerplate burdens via Logic-Environment Isolation, empowering users to focus exclusively on scientific validation.

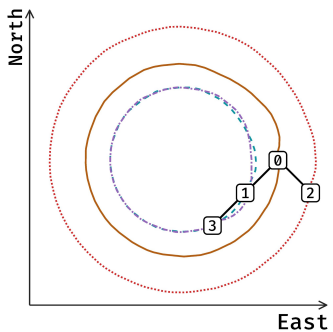
RQ2. Sim-to-Real Fidelity



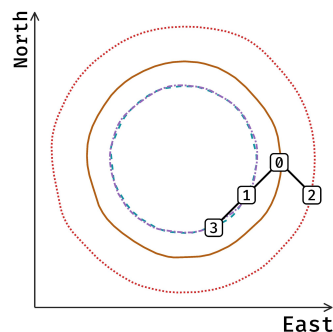
(a) Trajectory Plan



(b) SITL



(c) SIH



(d) Physical



SwarmBox bridges the sim-to-real gap via Hardware-Fidelity Abstraction, empowering users to achieve rapid deployment.

RQ3. Swarm-Level Observability



- (1) Internal state uncertainty of each agent
 - sensor fault (GPS Drift)
- (2) Inter-agent interaction faults
 - network latency / packet loss

RQ3. Swarm-Level Observability



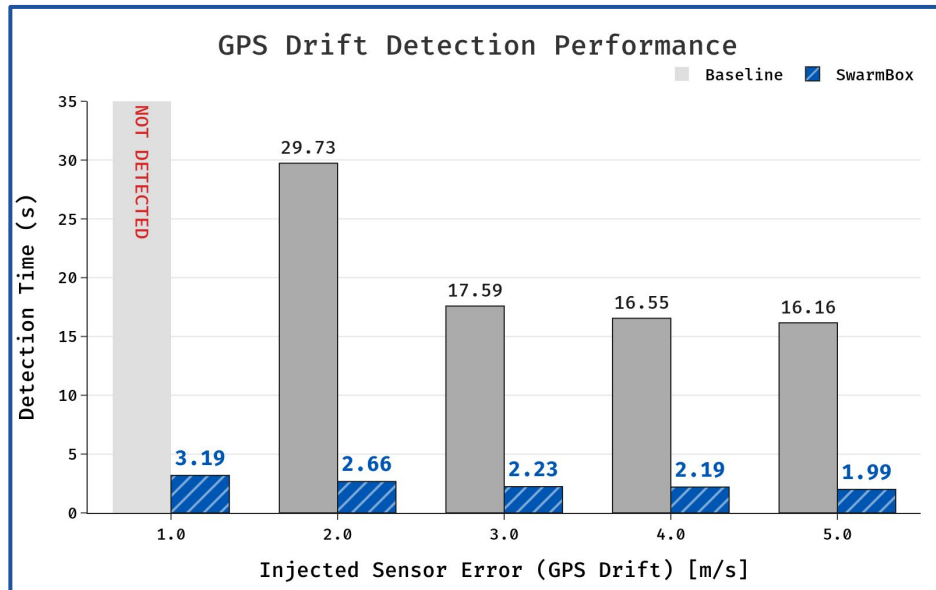
(1) Internal state uncertainty of each agent

- sensor fault (GPS Drift)

(2) Inter-agent interaction faults

- network latency / packet loss

- Faster sensor fault detection
- Enables diagnosis of inter-agent interaction faults



RQ3. Swarm-Level Observability



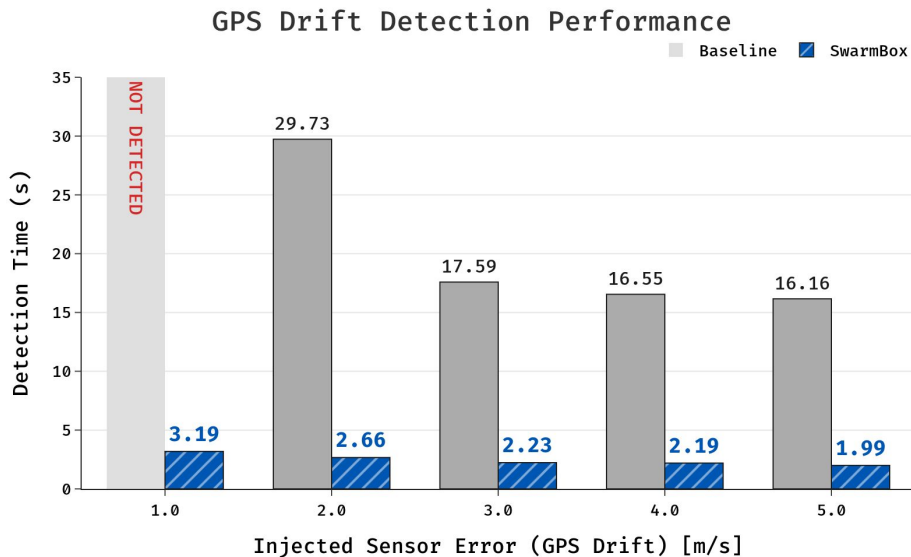
(1) Internal state uncertainty of each agent

- sensor fault (GPS Drift)

(2) Inter-agent interaction faults

- network latency / packet loss

- Faster sensor fault detection
- Enables diagnosis of inter-agent interaction faults



SwarmBox exposes complex inter-agent faults via Unified Behavior Analysis, empowering users with holistic diagnostic viewpoints.

RQ4. Benchmarking Capability



Evaluating SwarmBox's capability for thorough algorithmic comparison.

Scenario: Turnaround trip to 100 random targets with 5 drones (delivery)

Combinations of package assignment methods (3 sorting rules, 3 assignment methods)

Category	Metric	Unit
Cost Efficiency (General-Purpose)	Total flight distance	<i>m</i>
	Total mission duration	<i>s</i>
Resource Utilization (General-Purpose)	Idle time	<i>s</i>
	Utilization	%
	Workload deviation: distance	<i>m</i>
	Workload deviation: time	<i>s</i>
Delivery Quality (Mission-Specific)	Average delivery wait	<i>s</i>
	Maximum delivery wait	<i>s</i>

6 General-Purpose metrics

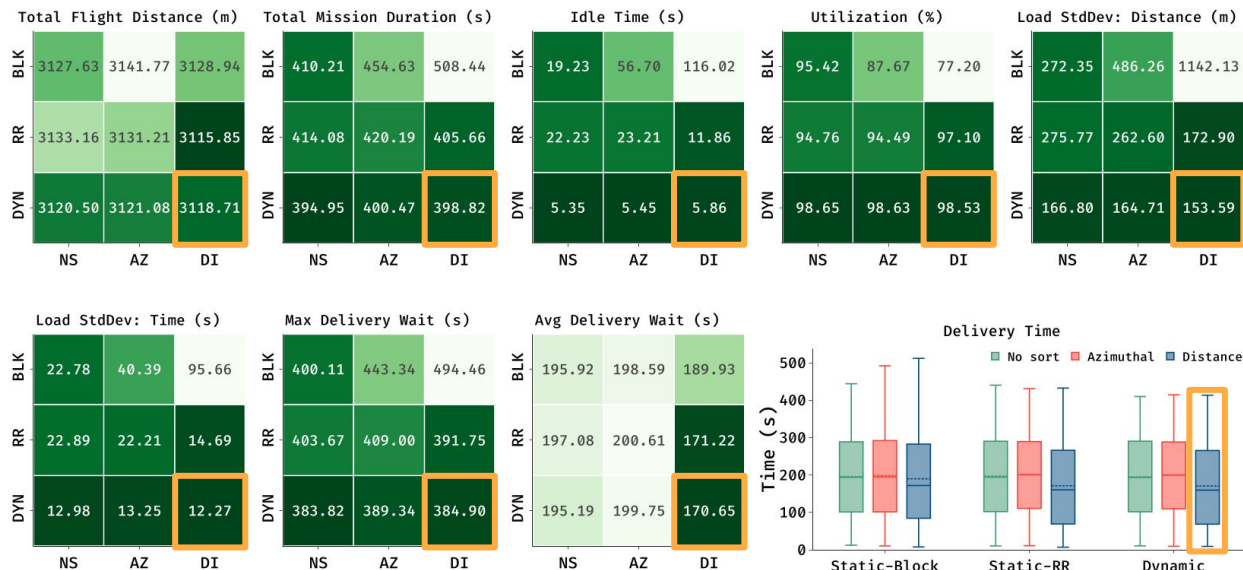
2 Mission-Specific metrics

RQ4. Benchmarking Capability



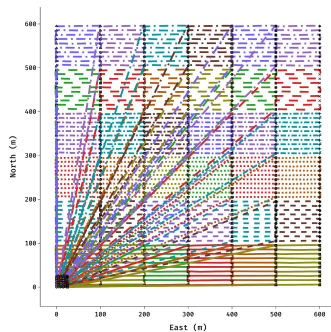
Benchmark-optimal:
Dynamic assignment &
Distance-ordered sort

darker cells, better performance.

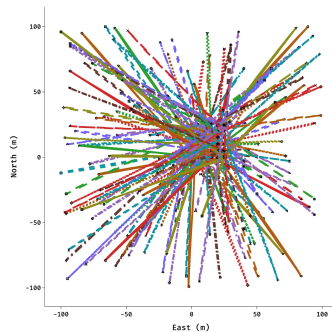


SwarmBox provides fair algorithmic comparison via Multi-Faceted Benchmarking, empowering users with a standardized evaluation platform.

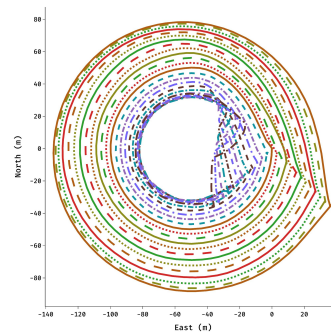
RQ5. Generality & Scalability



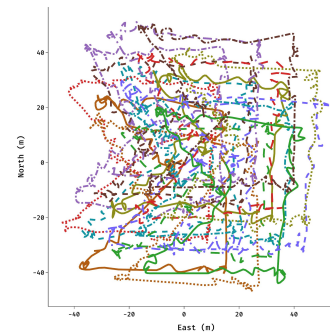
(a) airview



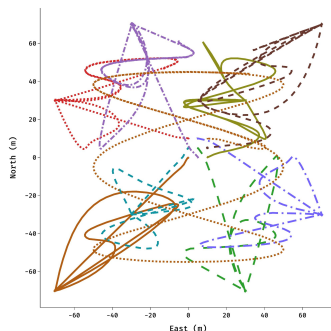
(b) delivery



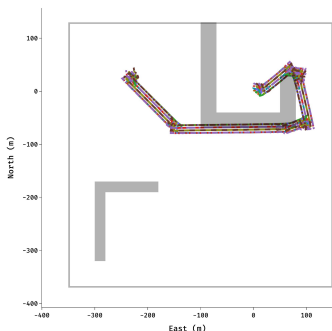
(c) formation



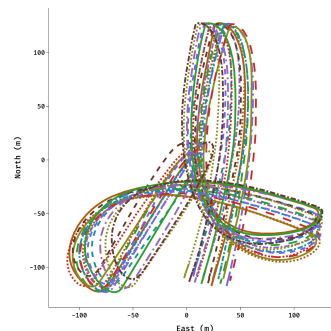
(d) network



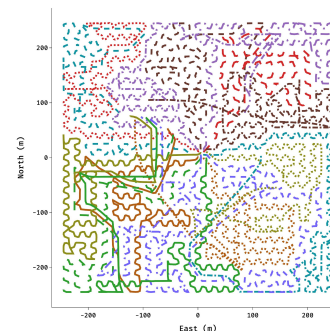
(e) tracking



(f) adaptive_swarm

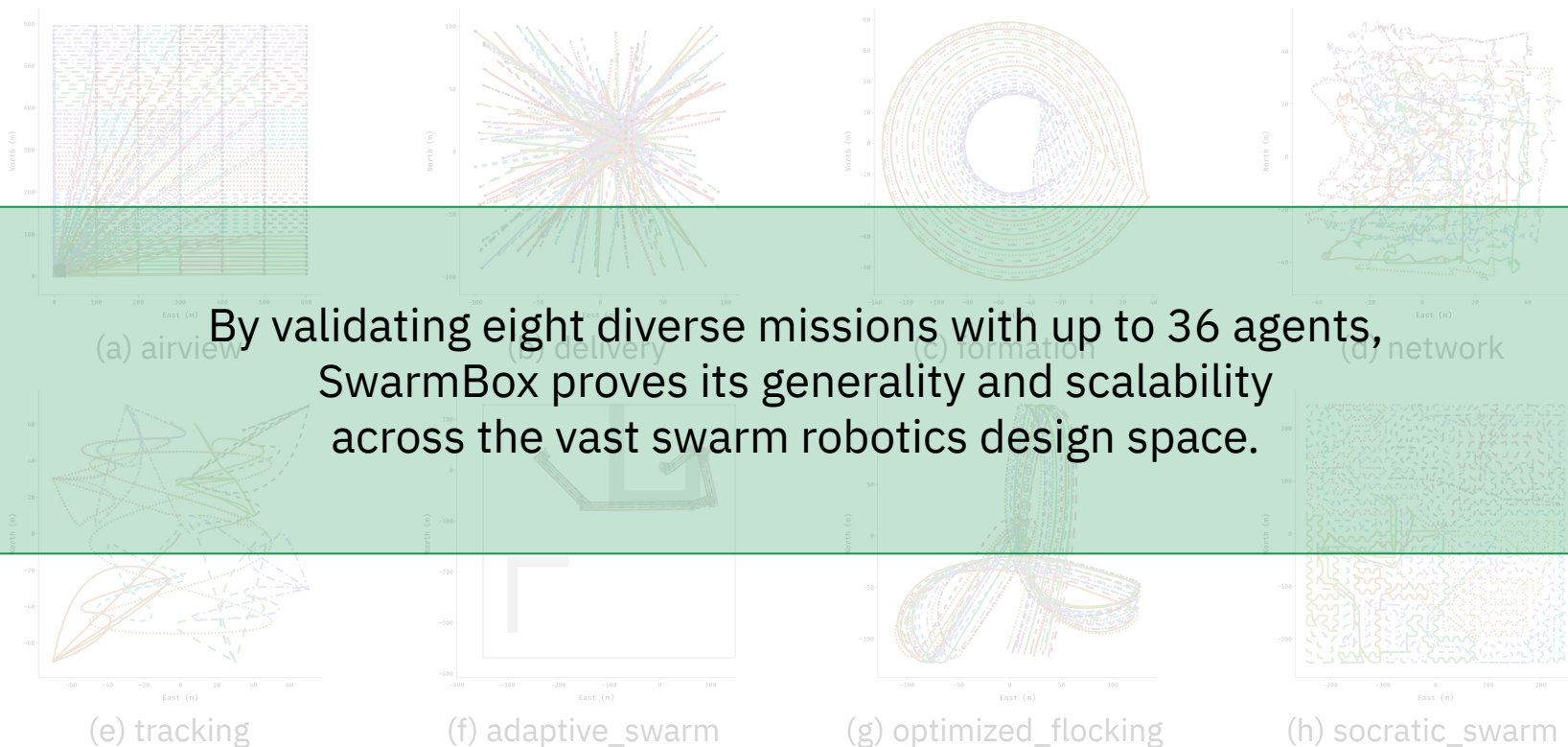


(g) optimized_flocking



(h) socratic_swarm

RQ5. Generality & Scalability



Open-Source, Active and Alive



To facilitate open innovation on the drone swarm ecosystem, we released and are maintaining SwarmBox as an **open-source software**.

- **Artifacts Evaluated:**
Available, **Functional** and **Reusable**.
- **Supporting modern stack:**
Ubuntu 22.04 → 24.04,
ROS 2 Humble → Jazzy,
PX4 v1.15.4 → v1.16.1
- **Upcoming features:**
 - Support for Multilayered DDS() , No-DDS, and Ardupilot

Summary



SwarmBox provides a unified foundation that breaks down engineering barriers to accelerate swarm research.

Effortless Integration



Removes boilerplate implementation effort up to 99.5%

Abstraction of Architecture



High sim-to-real fidelity: positioning errors are within GPS error range

Swarm-Level Analysis



Can detect swarm-level emergent faulty behaviors

Benchmarking Capability



Provides a standardized comparison platform

General & Scalable: Tested for 8 different missions, up to 36 drones



Open-Science: Open-sourced, actively maintained at <https://compsec.postech.ac.kr/swarmbox>



Q & A



SwarmBox provides a unified foundation that breaks down engineering barriers to accelerate swarm research.



Contact: Minki LEE, leeminki@postech.ac.kr
<https://compsec.postech.ac.kr/swarmbox>