Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

Università di Pisa

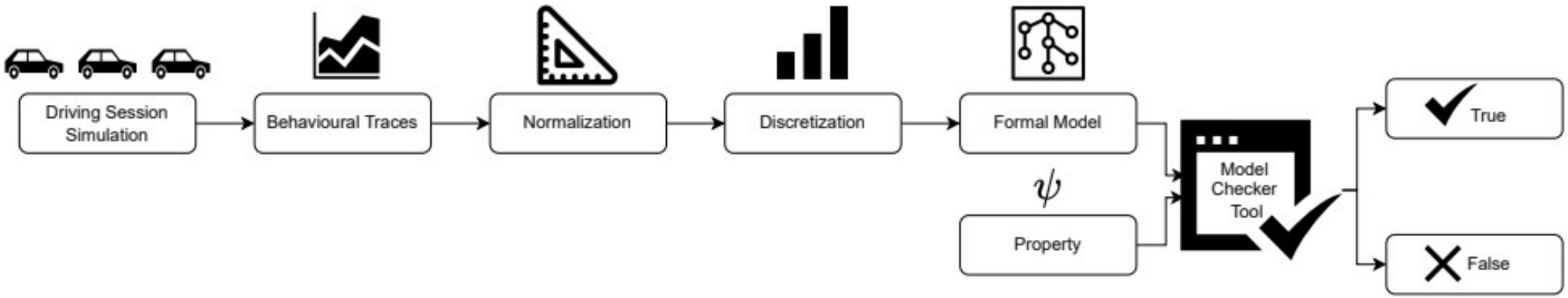
From Execution Traces to Formal Models

Team UNIMOL

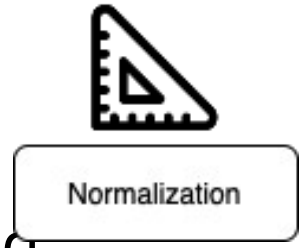
April 16th, 2025

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

Workflow of our Approach



Data Normalization: Preparing Traces for Discretization



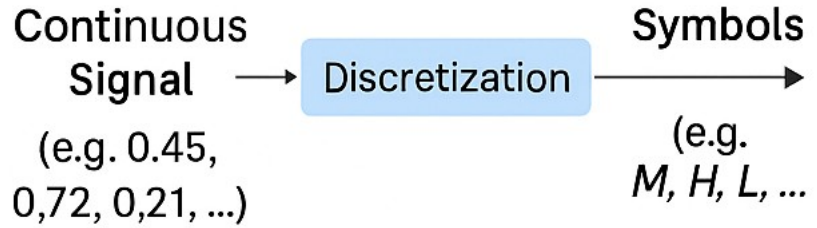
- Raw traces are time-series of velocity and acceleration (collected every 0.01s)
- These values vary in range and unit (e.g., m/s, m/s^2) → not directly usable
- We apply Min-Max normalization to map all values into the [0, 1] interval
- Why?
  - To standardize the data before applying symbolic discretization

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

From Continuous Traces to Symbolic Models

Discretization

- Our input traces are continuous time-series (velocity, acceleration)
- Formal verification requires symbolic models with a finite state space
- Discretization transforms continuous signals into a finite set of symbols
  - "Low", "Medium", "High"
  - "Increasing", "Stable", "Decreasing"
- This makes it possible to:
  - Model system behaviour algebraically
  - Check properties using model checking tools



Discretization is the bridge between **numeric simulation data** and **formal symbolic modelling**

[0.10] → L [0.55] → M [0.88] → H

Discretization (1/2): Fixed Intervals over Normalized Data

Discretization

- After normalization, we convert continuous values into symbolic categories
- First method: cut the [0–1] range into 3 intervals, applied independently to velocity and acceleration
  - Low (0–0.33) → L
  - Medium (0.34–0.66) → M
  - High (0.67–1) → H

Time	Velocity (norm)	Discrete	Acceleration (norm)	Discrete
T1	0.15	**L**	0.74	**H**
T2	0.60	**M**	0.30	**L**

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

Discretization (2/2): Window-Based Trend Detection

Discretization

- Second method: sliding window on each signal (e.g., 50 samples)
- Use ADF test to check if the window is stationary
  - If stationary → assign symbolic label (e.g., “stable-low”, “stable-high”)
  - If not → compute slope → classify as increasing (↑) or decreasing (↓)
- Captures more nuanced temporal dynamics

|----50 samples----|----50 samples----|----50 samples----|
stable low | increasing | stable high

This method enables a more dynamic discretization, better reflecting driving behaviour changes over time.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

Comparing Discretization Strategies: Instant vs Window-Based

Discretization

Fixed Binning

Time	Acc (norm)	Acc (disc)
t0	0.10	**L**
t1	0.56	**M**
t2	0.87	**H**

Trend-Based

Window (samples)	ADF result	Slope	Acc (disc)
t0–t49	Stationary	—	**stable-low**
t50–t99	Non-stationary	> 0	**increasing**
t100–t149	Stationary	—	**stable-high**

The trend-based approach uses statistical testing + slope to abstract behaviour over time intervals, not single points.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

Two Levels of Modelling: Individual and Collective Behaviour

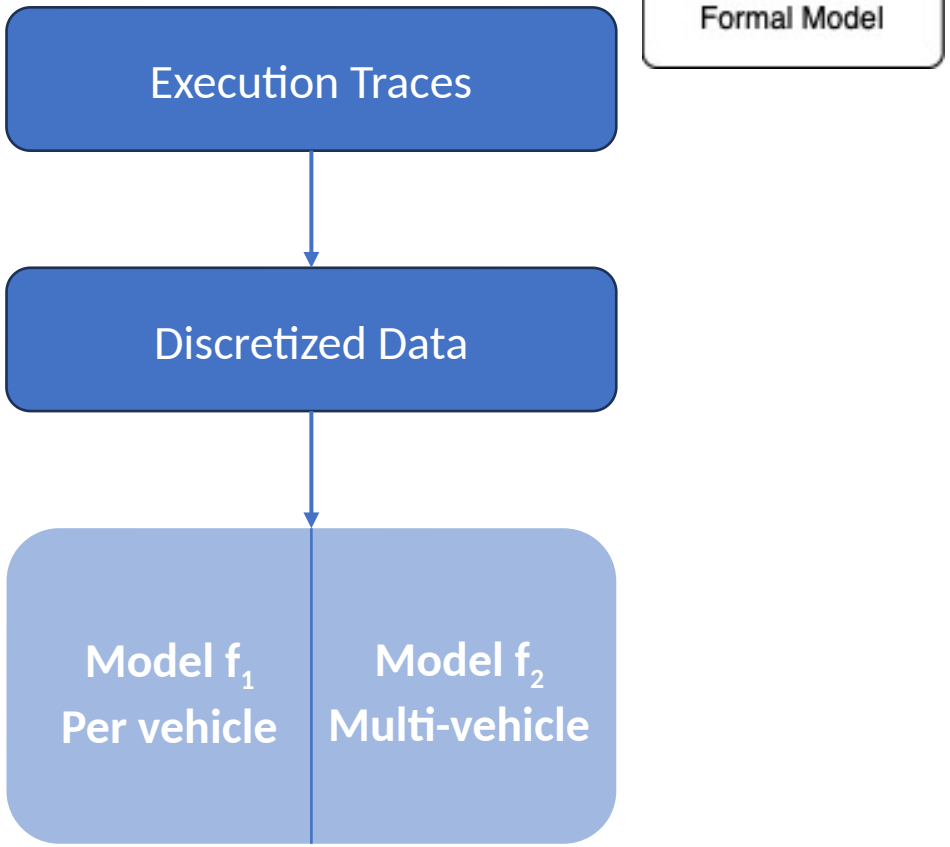
Our approach produces two types of formal models:

1. **Model f_1**: per-vehicle, time-based sequential behaviour
2. **Model f_2**: synchronized multi-vehicle behaviour in a platoon

The two models are complementary:

- f_1 captures **local evolution** of a single agent
- f_2 represents the **global coordination** and interactions

We use both models to analyze system behaviour at **micro** (individual) and **macro** (collective) levels.

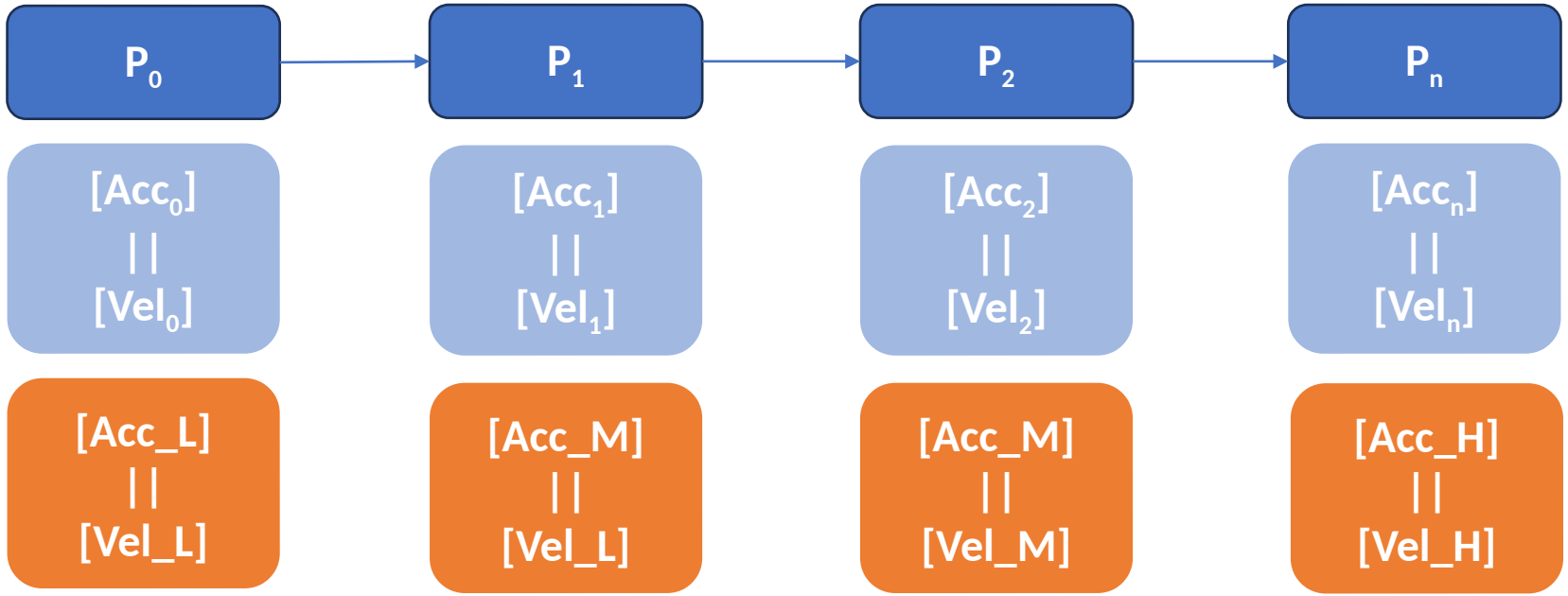


Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

Model f_1: Sequential Modelling of a Single Vehicle

Formal Model

- For each vehicle, we model the evolution over time
- Each time step is a **state**, labeled with discretized acceleration & velocity
- The model is a **sequence** of these states, using parallel composition of features + sequential composition in time



Sequential evolution of a vehicle's behaviour

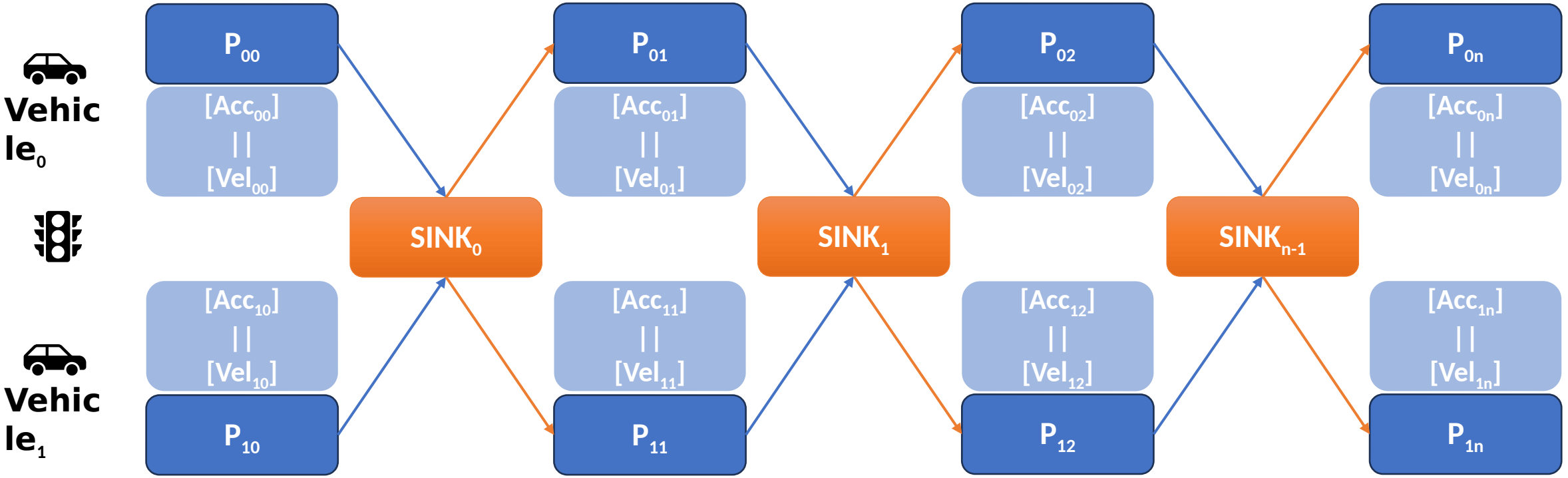
Process = (feature₁ ‖ feature₂ ‖ feature₃); next state

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

Model f_2: Synchronizing Multiple Vehicles

Formal Model

- We combine all f_1 models in **parallel**
- Introduce a sink process to synchronize all vehicles at each step
- Ensures all vehicles "tick" together → represents **platoon coordination**



The **sink** orchestrates all vehicles' transitions. f_2 captures **collective behaviour** of the platoon

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

Property Verification: Analyzing System Behaviour

The formal models $\mathbf{f_1}$ and $\mathbf{f_2}$ are used to verify **specific behavioural properties** of the platooning system.
Examples of properties checked:

- "If the leader vehicle accelerates, the following vehicles accelerate in sequence."
- "If the leader vehicle brakes, the rest of the platoon reacts accordingly."

These properties are verified using **model checking techniques**, enabling automated validation.

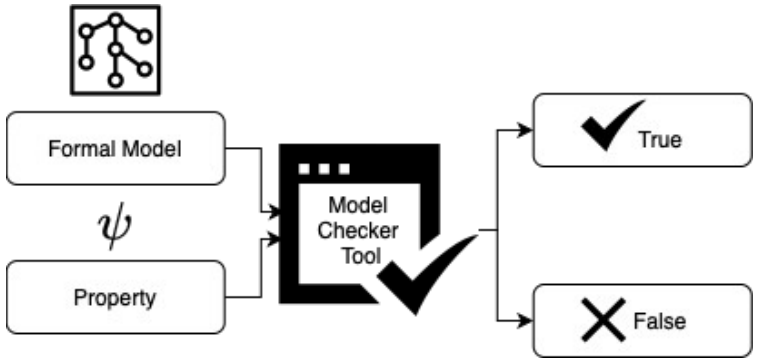
Property	Verification Result
Leader acceleration propagation	✅ Verified
Leader deceleration propagation	✅ Verified

Property verification helps us understand system dynamics and identify potential weaknesses or areas for improvement.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

Model Checking: Verifying System Properties

- **Model checking** is a formal verification technique used to automatically check whether a system model satisfies a set of desired properties.
- The process involves:
  - Defining a **formal model** of the system (e.g., using transition systems or process algebra)
  - Expressing the **properties** to be checked in a formal logic (e.g., LTL, CTL, or μ-calculus)
  - Automatically exploring all possible system behaviours to verify whether the properties hold
- If a property is **not satisfied**, the tool provides a **counterexample**, showing a specific execution trace where the property is violated.



Model checking enables **exhaustive and automated verification**, which is crucial in safety-critical and autonomous systems.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

Future Work: From Data Abstraction to Security Verification

1. **Enhancing Discretization**
   - Explore adaptive binning and dynamic thresholds
   - Introduce **context-aware discretization**, adjusting based on platoon state (e.g., traffic conditions, role in the platoon)
   - Experiment with **multivariate discretization** (acceleration + velocity + spacing)
2. **Improving Model Abstraction**
   - Extend the formal model with **more vehicle parameters** (e.g., inter-distance, braking signals)
   - Incorporate **time-based transitions** or delays to model timing behaviour
   - Move toward **compositional modelling**: e.g., vehicles as individual components with shared rules
3. **Towards Security Verification**
   - Define properties that detect **abnormal propagation** of behaviours (e.g., delayed or missing reactions)
   - Model and verify scenarios involving:
     - **Compromised nodes** (e.g., a vehicle not relaying commands)
     - **Communication delays or spoofed data**
   - Use model checking to confirm the system maintains safety under adversarial conditions

Our goal is to use formal methods not only to ensure correctness, but also to support **resilience and trust** in platooning systems under real-world uncertainties.