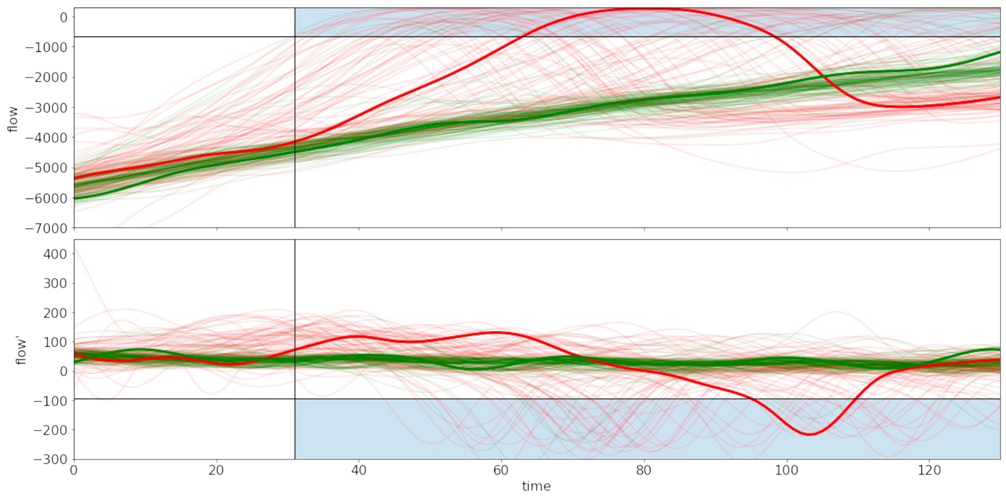
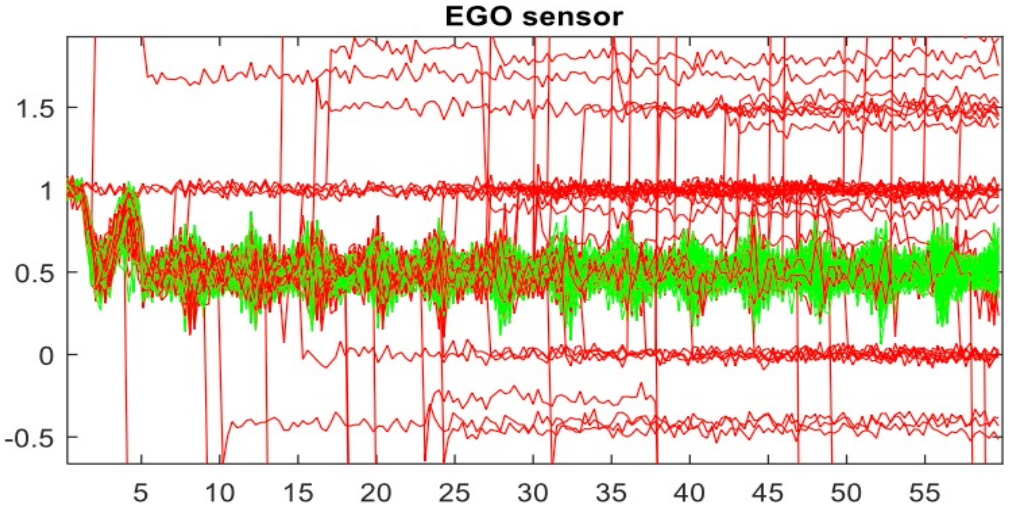
Learning Temporal Logic Formulas from Time-series Data

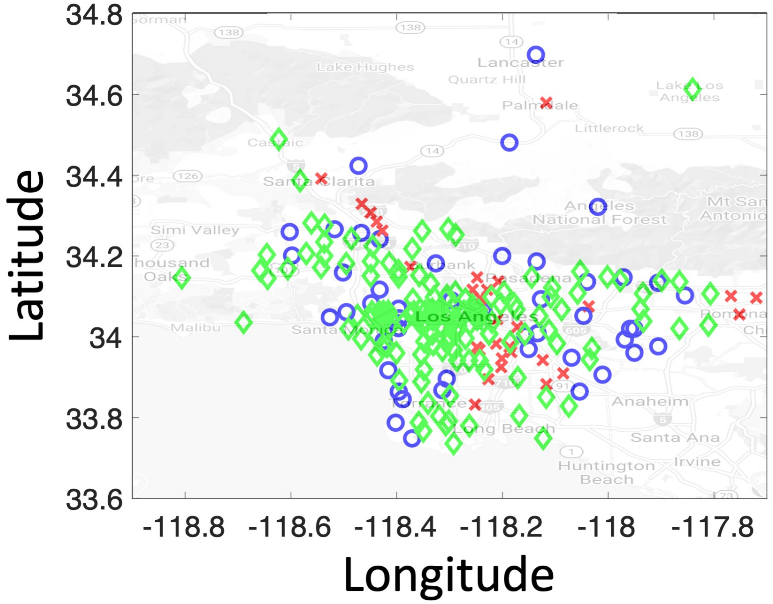
Laura Nenzi

March 30, Trieste 2022

Time-series Data

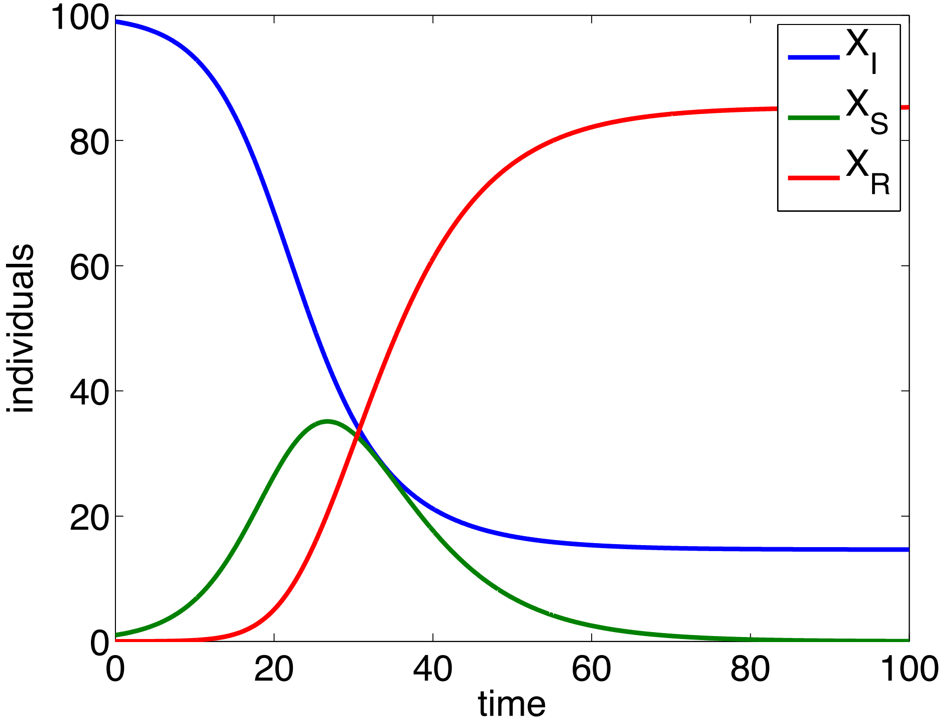






Formal Specification

SIMULATION



BEHAVIOUR

"Between 30 and 50 time units, the number of recovered individuals becomes more than 60"

PROPERTY

(Temporal Logic Formula)

$$F_{[30,50]}(X_R > 60)$$

Specification-based Monitoring

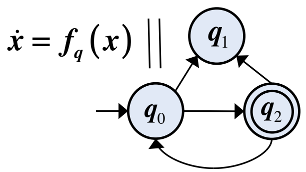
Informal Specification

Between 2s and 4.5s the output signal is between -2 and 2

Formal Specification Language (STL)

$\varphi \coloneqq G_{[2,4.5]}(-2 < x[t] < 2)$

CPS model



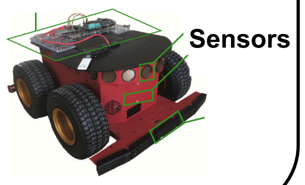
Hybrid System

Simulation

$q_0 \rightarrow q_1 \rightarrow \ldots$

$x(t)$

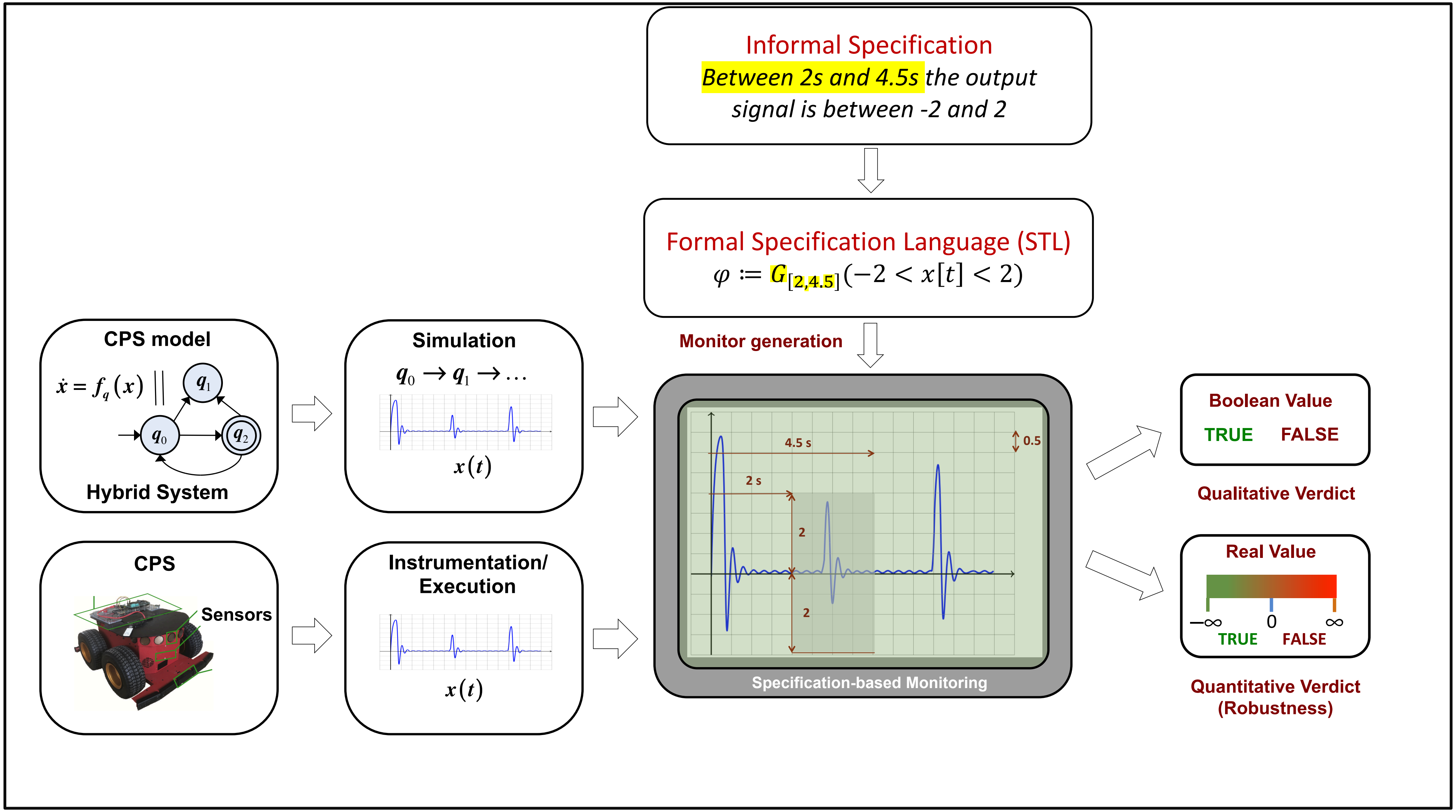
CPS



Instrumentation/ Execution

$x(t)$

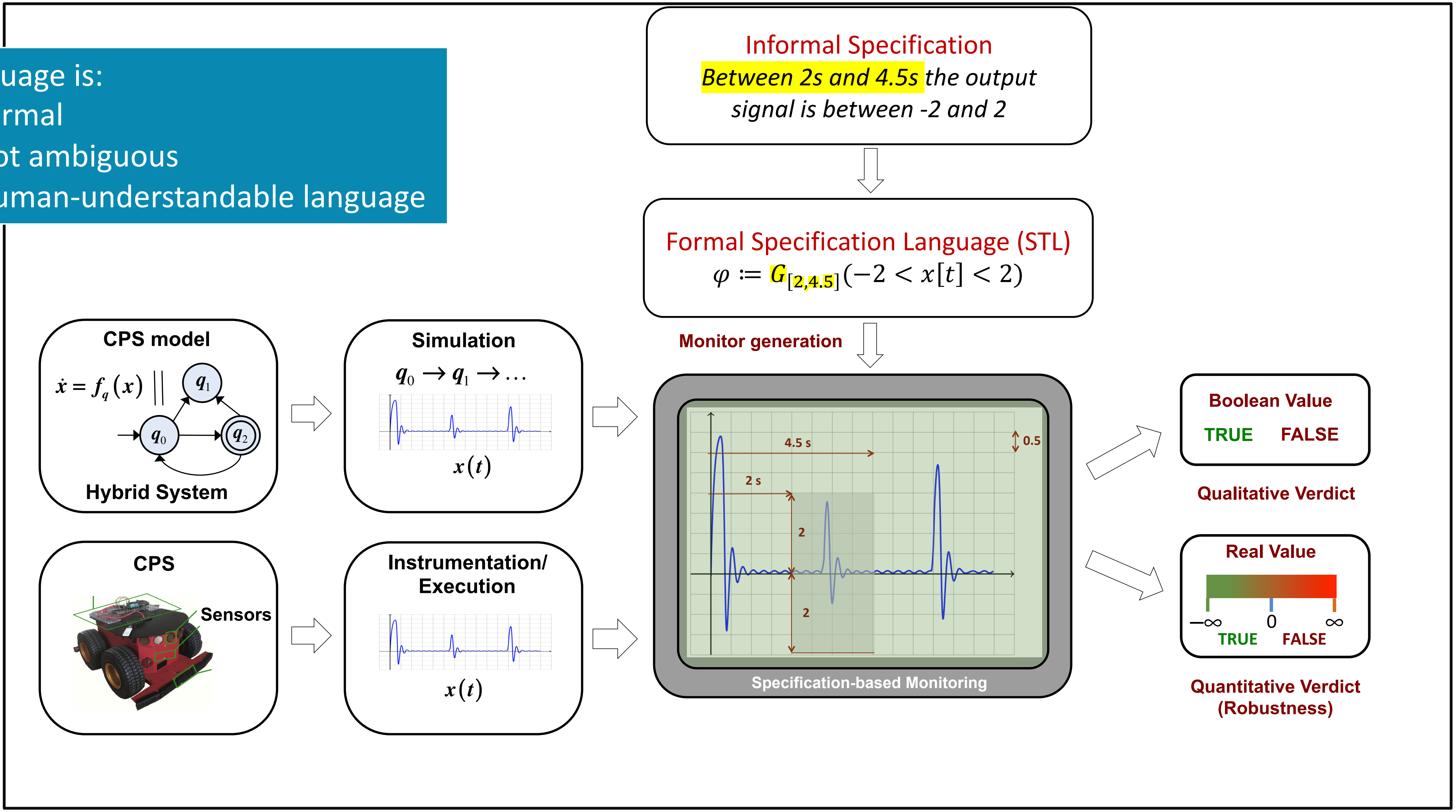
Specification-based Monitoring



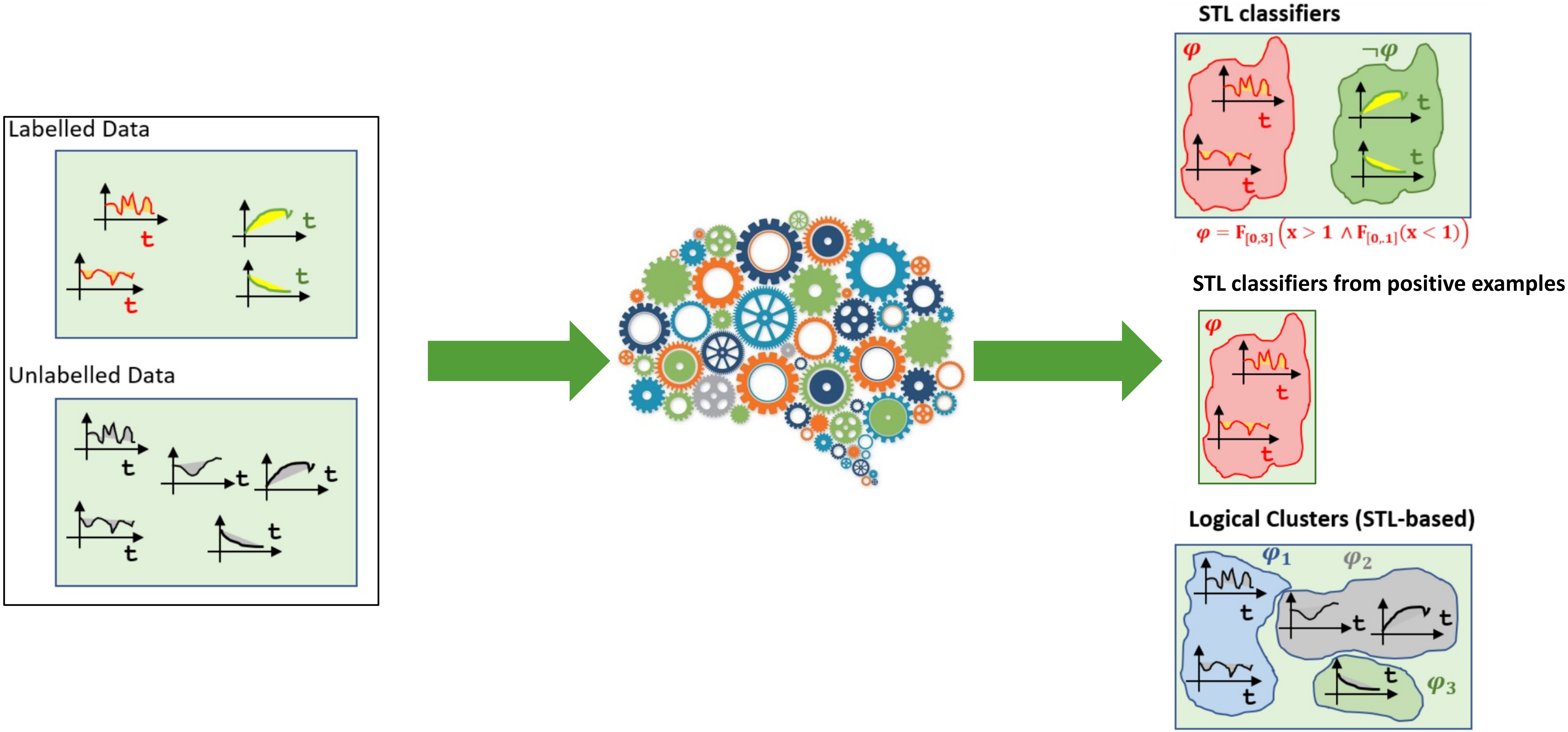
Specification-based Monitoring

Logic language is:

- formal
- not ambiguous
- human-understandable language

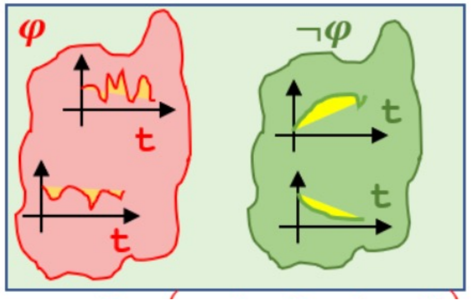


Learning from Time-series Data

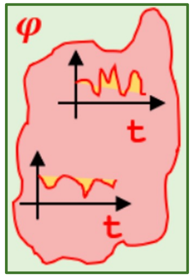


Learning from Time-series Data

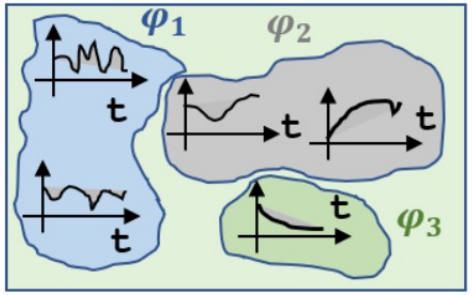
STL classifiers



STL classifiers from positive examples



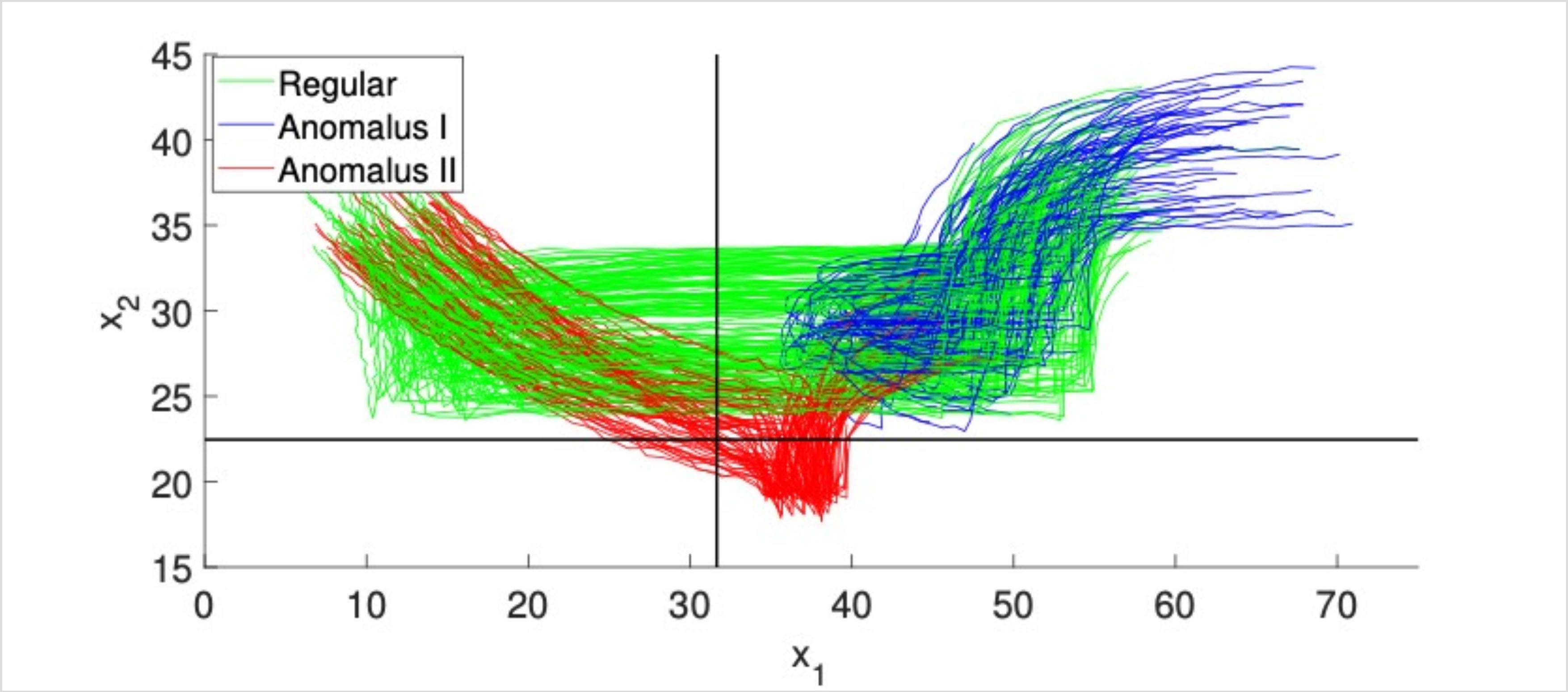
Logical Clusters (STL-based)



Advantages: explicability, easy to build monitors

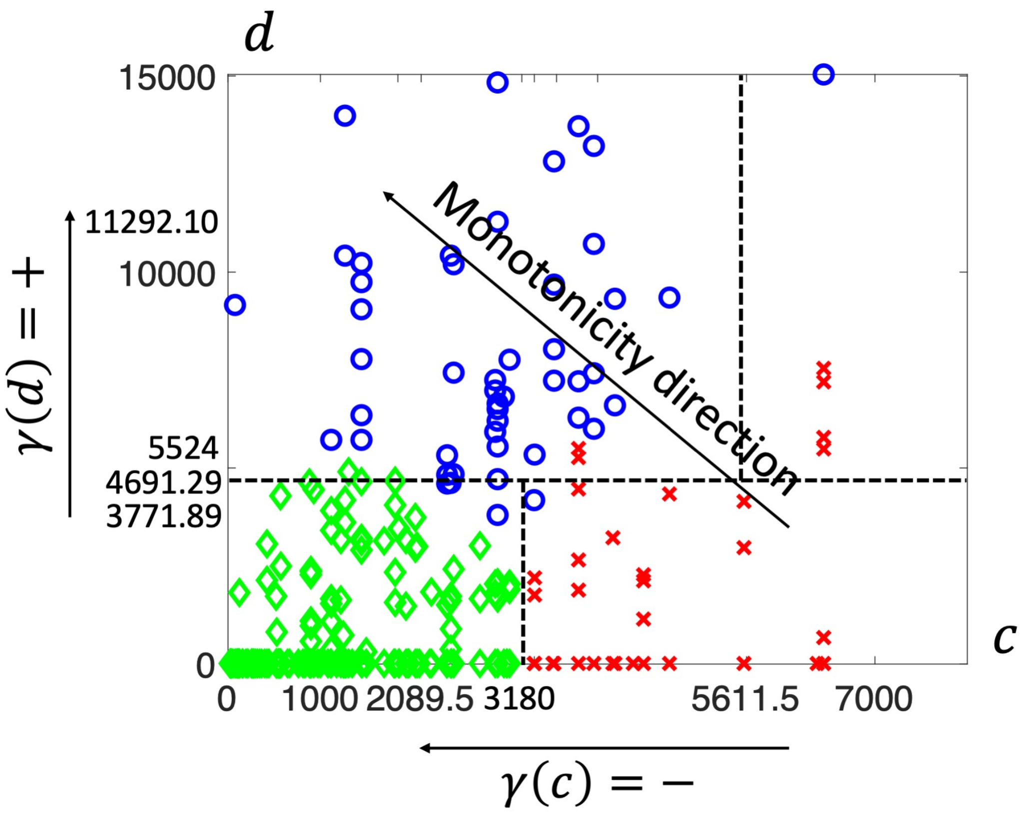
Applications: anomaly detection, specification synthesis

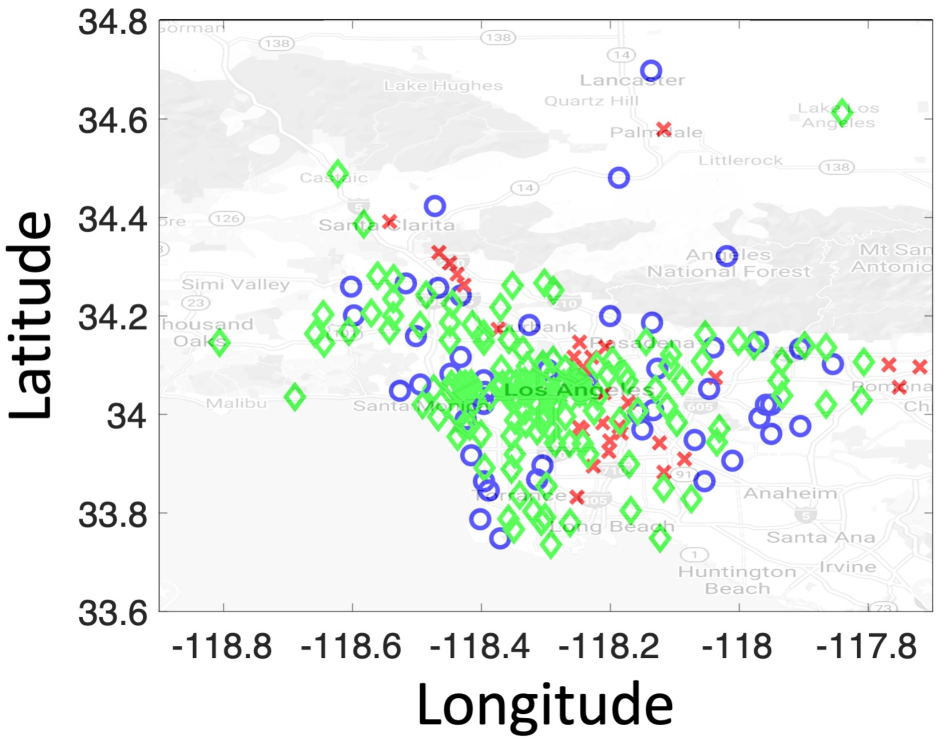
Learning STL Classifiers ((Semi-)Supervised Learning)



Goal: learning a specification/ classifier as a temporal logic formula to discriminate as much as possible between bad and good behaviours

Learning STL-based clustering (Unsupervised Learning)





Goal: clusterizing spatio-temporal data using formal logic

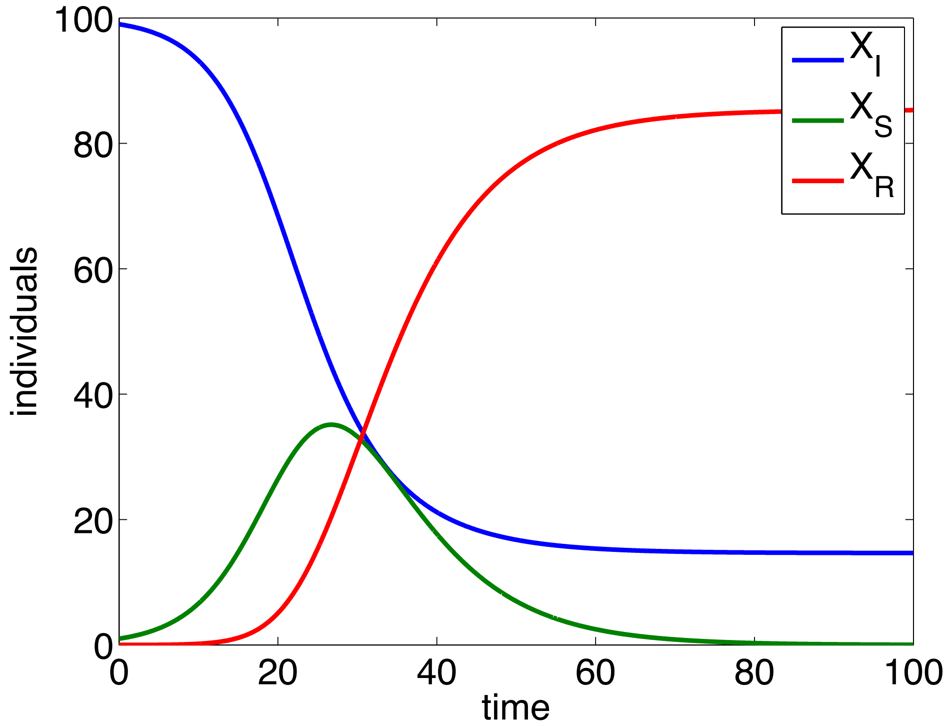
Agenda

- Signal Temporal Logic (STL)
- STL-based classification (supervised and semi-supervised)
- Spatio-Temporal Reach and Escape Logic (STREL)
- STL-based clustering of time-series data

Signal Temporal Logic (STL)

STL Syntax	$\varphi := true \mid \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathrm{U}_I \, \varphi_2$

In addition $F_I\varphi := \top \, U_I\varphi \qquad G_I\varphi := \neg F_I\neg\varphi$



"Between 30 and 50 time units, the number of recovered individuals becomes more than 60"

$$F_{[30,50]}(X_R > 60)$$

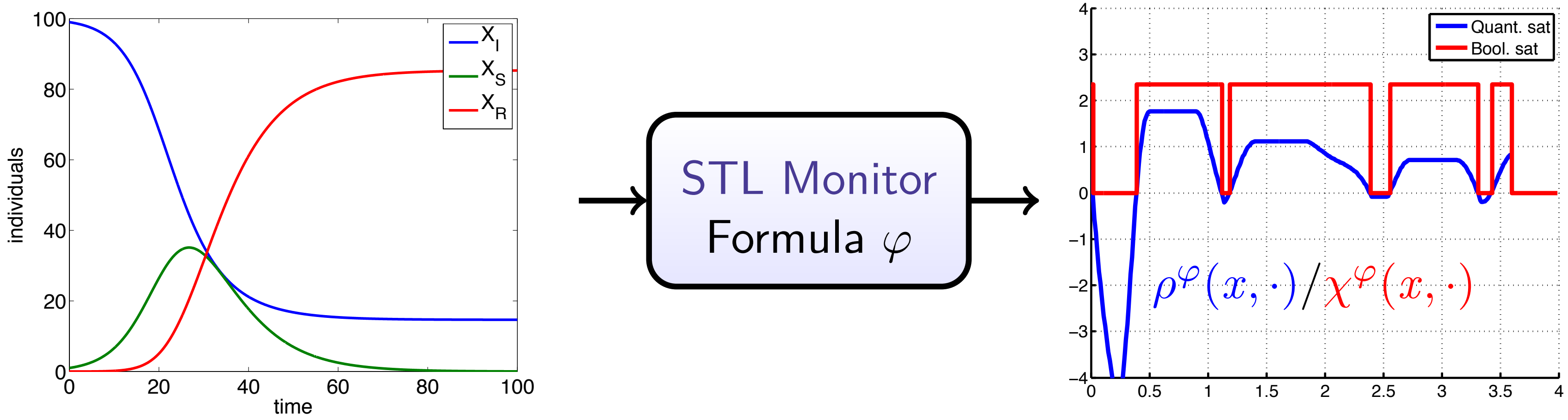
Signal Temporal Logic (STL)

Boolean Signal

$s_\varphi : [0, T] :\rightarrow \{0,1\}$ s.t. $s_\varphi(t) = 1 \Leftrightarrow (\vec{x}, t) \models \varphi$

Quantitative Signal

$\rho_\varphi : [0, T] :\rightarrow \mathbb{R} \cup \{\pm\infty\}$ s.t. $\rho_\varphi(t) = \rho(\varphi, \vec{x}, t)$

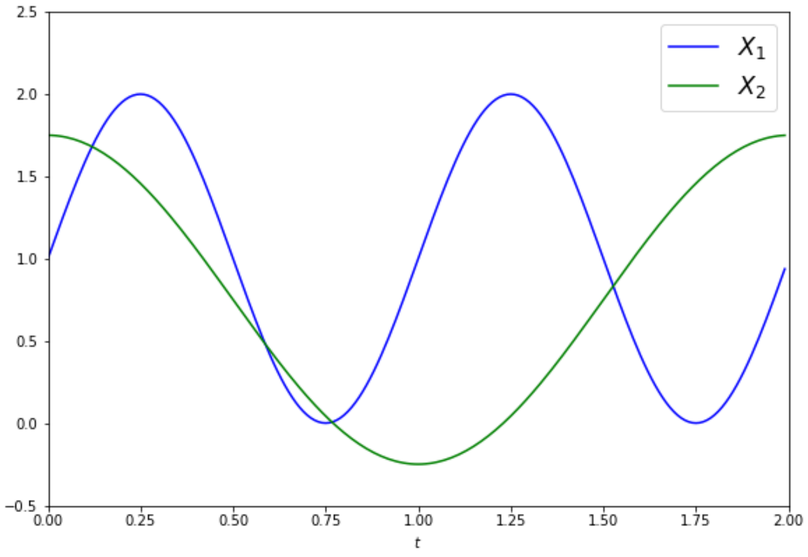


Signal Temporal Logic (STL)

Boolean Semantics $\chi(\vec{x}, t, \varphi) \in \{0, 1\}$

$$\boxed{t \xrightarrow{\vec{x}} \mathbb{R}^m} \xrightarrow{f} \mathbb{R} \xrightarrow{f \geq 0} \{0, 1\}$$

primary

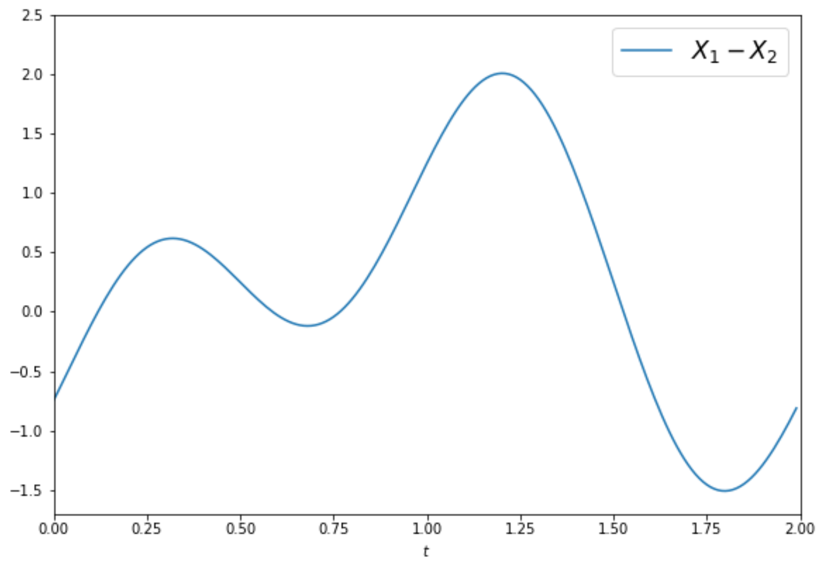


Signal Temporal Logic (STL)

Boolean Semantics

$$\chi(\vec{x}, t, \varphi) \in \{0,1\}$$

$$\underbrace{\underbrace{t \xrightarrow{\vec{x}} \mathbb{R}^m}_{\text{primary}} \xrightarrow{f}}_{\text{secondary}} \mathbb{R} \xrightarrow{f \geq 0} \{0,1\}$$



Signal Temporal Logic (STL)

Boolean Semantics

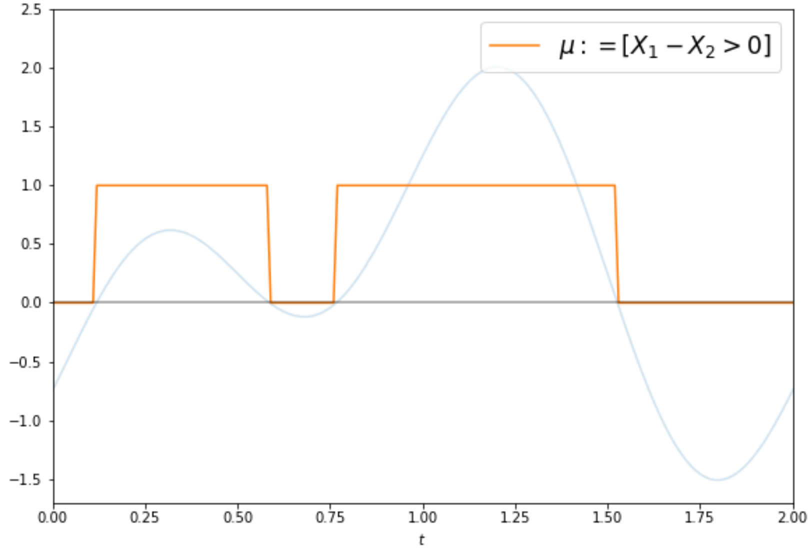
$$\chi(\vec{x}, t, \varphi) \in \{0, 1\}$$

$$t \xrightarrow{\vec{x}} \mathbb{R}^m \xrightarrow{f} \mathbb{R} \xrightarrow{f \geq 0} \{0, 1\}$$

primary

secondary

μ



Signal Temporal Logic (STL)

$$\cancel{t \xrightarrow{\vec{x}} \mathbb{R}^m \xrightarrow{f} \mathbb{R}} \xrightarrow{f \geq 0} \{0, 1\}$$

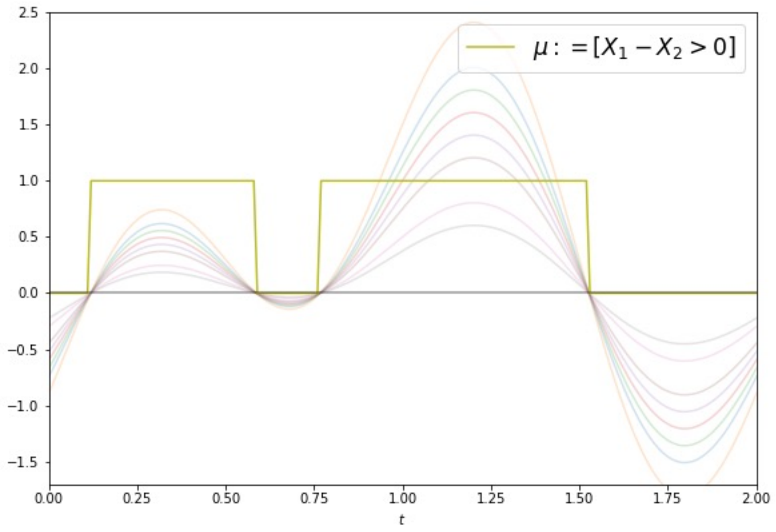
~~primary~~

~~secondary~~

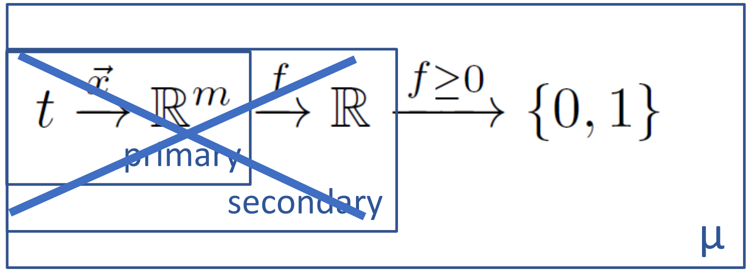
μ

Boolean Semantics

$$\chi(\vec{x}, t, \varphi) \in \{0, 1\}$$

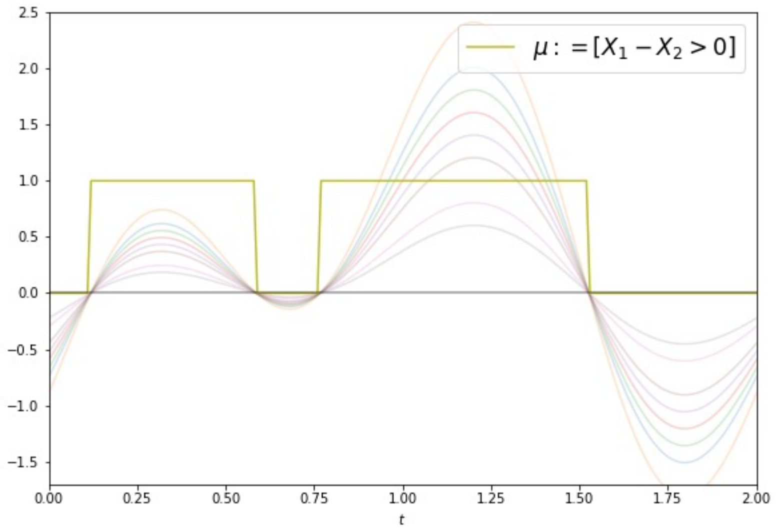


Signal Temporal Logic (STL)



Boolean Semantics

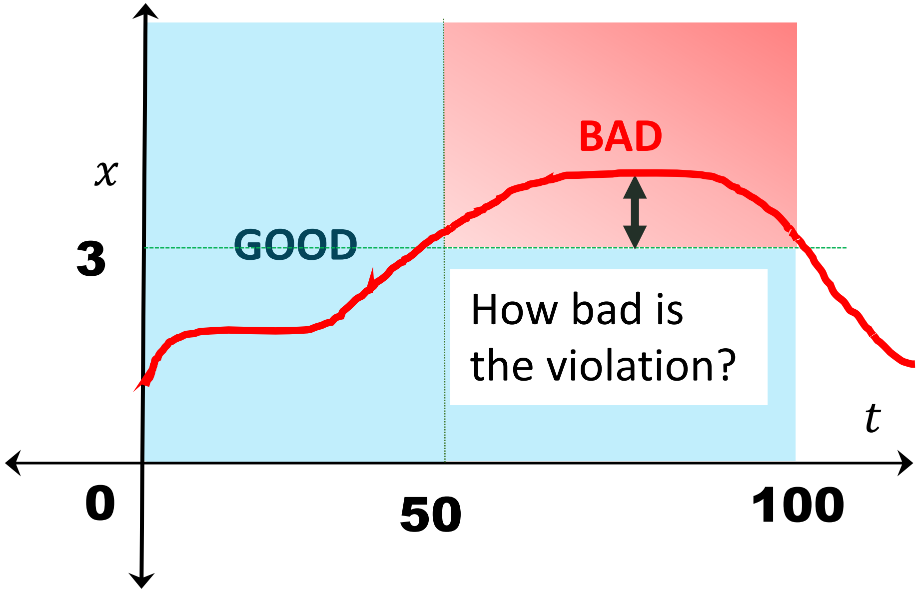
$$\chi(\vec{x}, t, \varphi) \in \{0, 1\}$$



Quantitative Semantics

$$\rho(\vec{x}, t, \varphi) \in \mathbb{R} \cup \{+\infty, -\infty\}$$

Distance to violation/satisfaction



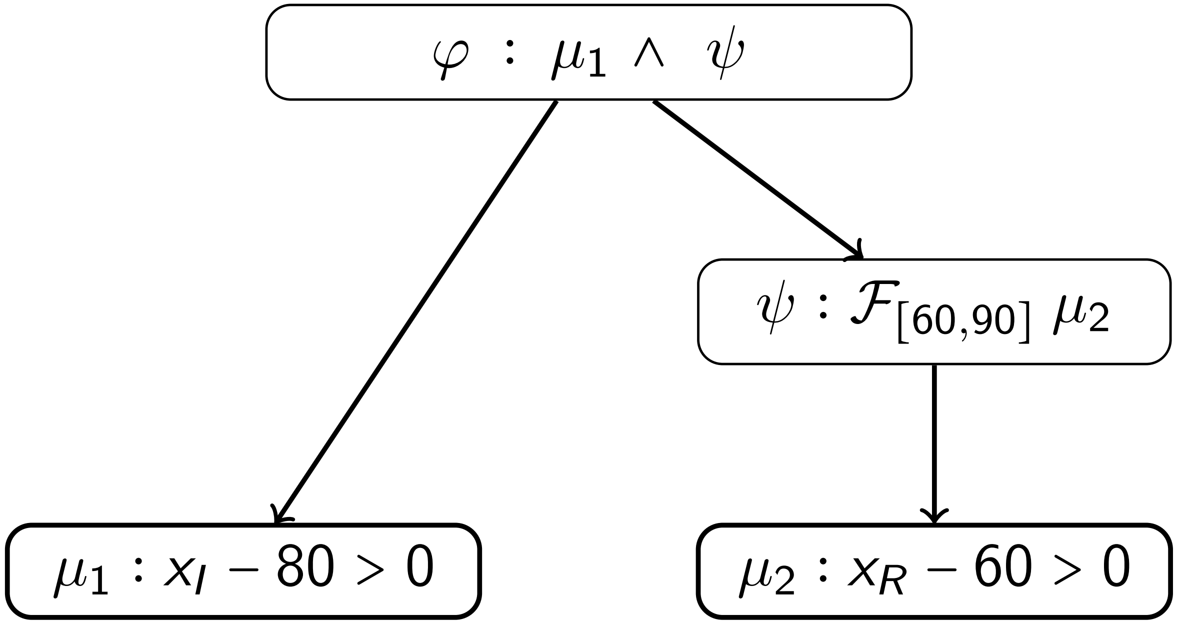
$$\mathbf{G}_{[50,100]}(x(t) < 3)$$

Recursive Quantitative Semantics

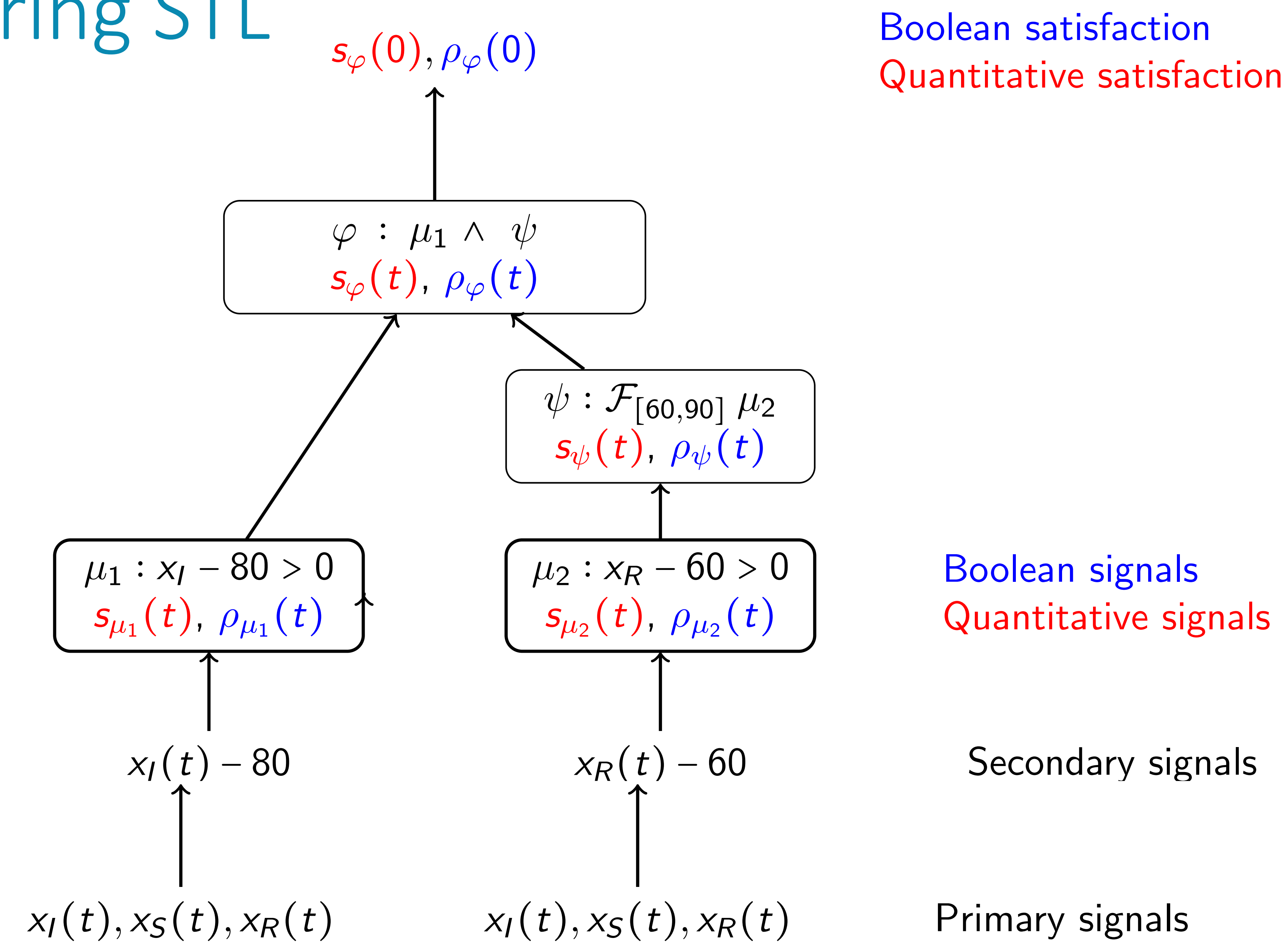
φ	$\rho(\varphi, \mathbf{x}, t)$
$f(\mathbf{x}) > 0$, $f(\mathbf{x}) \geq 0$	$f(\mathbf{x}(t))$
$\neg\varphi$	$-\rho(\varphi, \mathbf{x}, t)$
$\varphi_1 \wedge \varphi_2$	$\min\big(\rho(\varphi_1, \mathbf{x}, \mathrm{t}) \wedge \rho(\varphi_2, \mathbf{x}, \mathrm{t})\big)$
$\mathbf{F}_{[a,b]}\varphi$	$\sup_{\tau \in [t+a, t+b]} \rho(\varphi, \mathbf{x}, \tau)$
$\mathbf{G}_{[a,b]}\varphi$	$\inf_{\tau \in [t+a, t+b]} \rho(\varphi, \mathbf{x}, \tau)$
$\varphi\ \mathbf{U}_{[a,b]}\ \psi$	$\sup_{\tau \in [t+a, t+b]} \left(\min\left(\rho(\psi, \mathbf{x}, \tau), \inf_{\tau' \in [t, \tau)} \rho(\varphi, \mathbf{x}, t)\right)\right)$

Monitoring STL

$$\varphi : (x_I > 80) \wedge \mathcal{F}_{[60,90]} (x_R > 60)$$



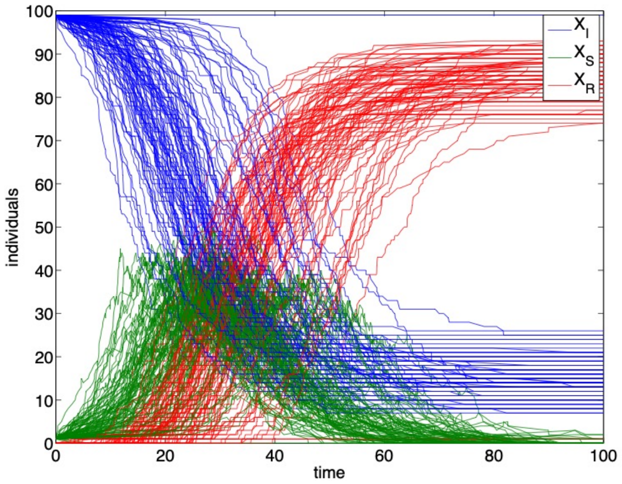
Monitoring STL

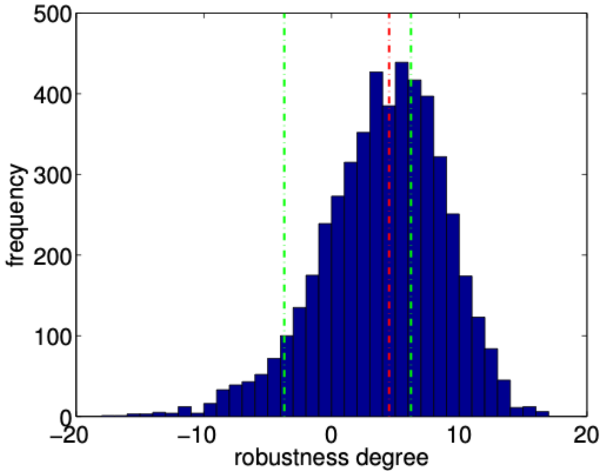


Average Robustness

Robustness Distribution

$$\mathbb{P}\left(R_\varphi(\mathbf{X}) \in [a,b]\right) = \mathbb{P}\left(\mathbf{X} \in \{\mathbf{x} \in \mathcal{D} \mid \rho(\varphi, \mathbf{x}, 0) \in [a,b]\}\right)$$





Indicators

- $\mathbb{E}(R_\varphi)$ (the average robustness degree)
- $\mathbb{E}(R_\varphi \mid R_\varphi > 0)$ and $\mathbb{E}(R_\varphi \mid R_\varphi < 0)$ (the conditional averages)

Parametric Signal Temporal Logic

Definition (PSTL syntax)

$$\phi := (x_i \bowtie \pi) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \,\mathcal{U}_{[\tau_1,\tau_2]}\, \varphi_2$$

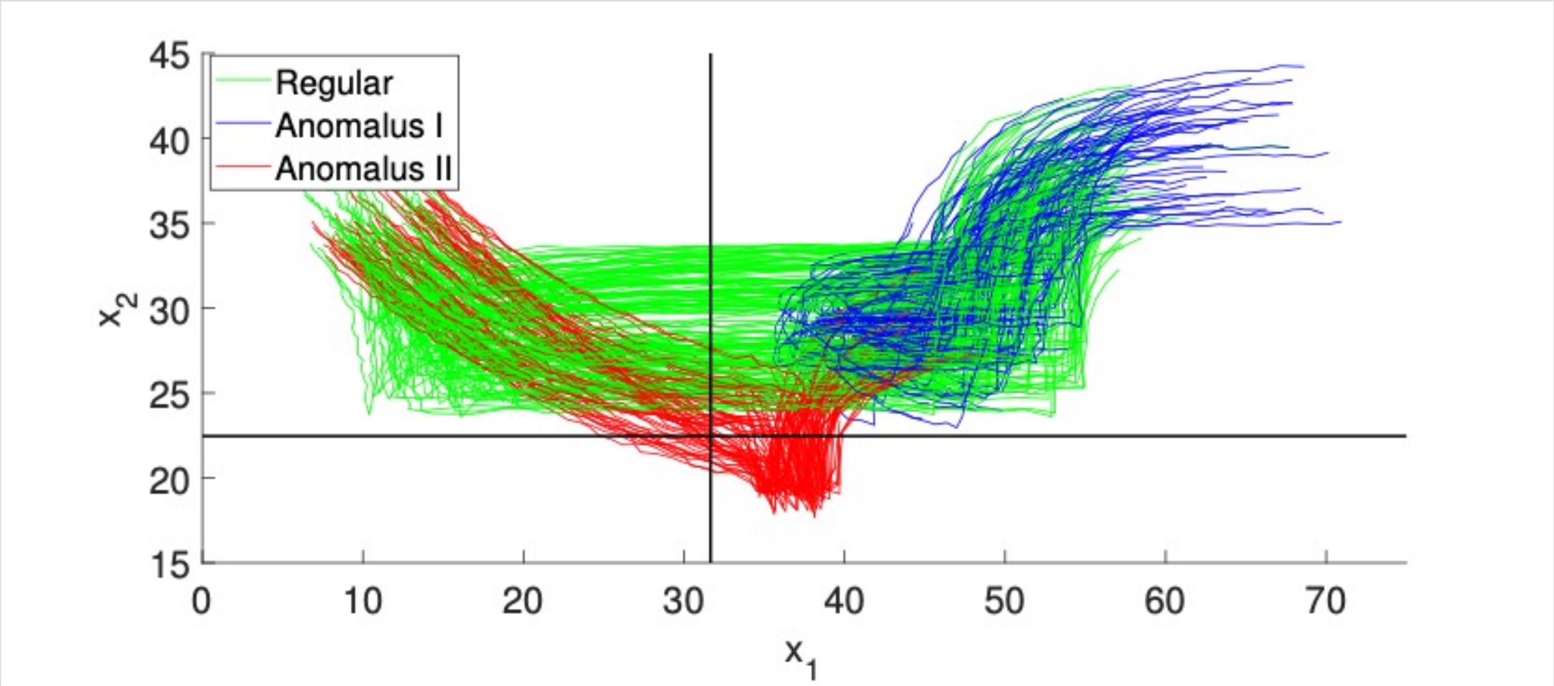
with $\bowtie \in \{>, \leq\}$

- π is **threshold** parameter
- τ_1, τ_2 are **temporal** parameters

- $\mathbb{K} = (\mathcal{T} \times \mathcal{C})$ be the **parameter space**
- $\boldsymbol{\theta} \in \mathbb{K}$ is a **parameter configuration**

e.g., $\phi = \mathcal{F}_{[a,b]}(x_i > k), \boldsymbol{\theta} = (0, 2, 3.5)$ then $\phi_{\boldsymbol{\theta}} = \mathcal{F}_{[0,2]}(x_i > 3.5)$.

Learning STL classifiers



Goal: learning a specification/ classifier as a temporal logic formula to discriminate as much as possible between bad and good behaviours

Advantages: explicability, easy to build monitors

Application: anomaly detection, specification synthesis

Methodology

- *Single-level* variant: learning formula structure and parameter using Context Free Grammar Genetic Programming (CFGGP)

- *Bi-level* variant:
  - learning formula structure CFGGP
  - learn parameters of the formula using by **Bayesian Optimisation**

A fitness function f measures the quality of candidate solutions and depends on the kind of problem at hand (two-classes, one-class)

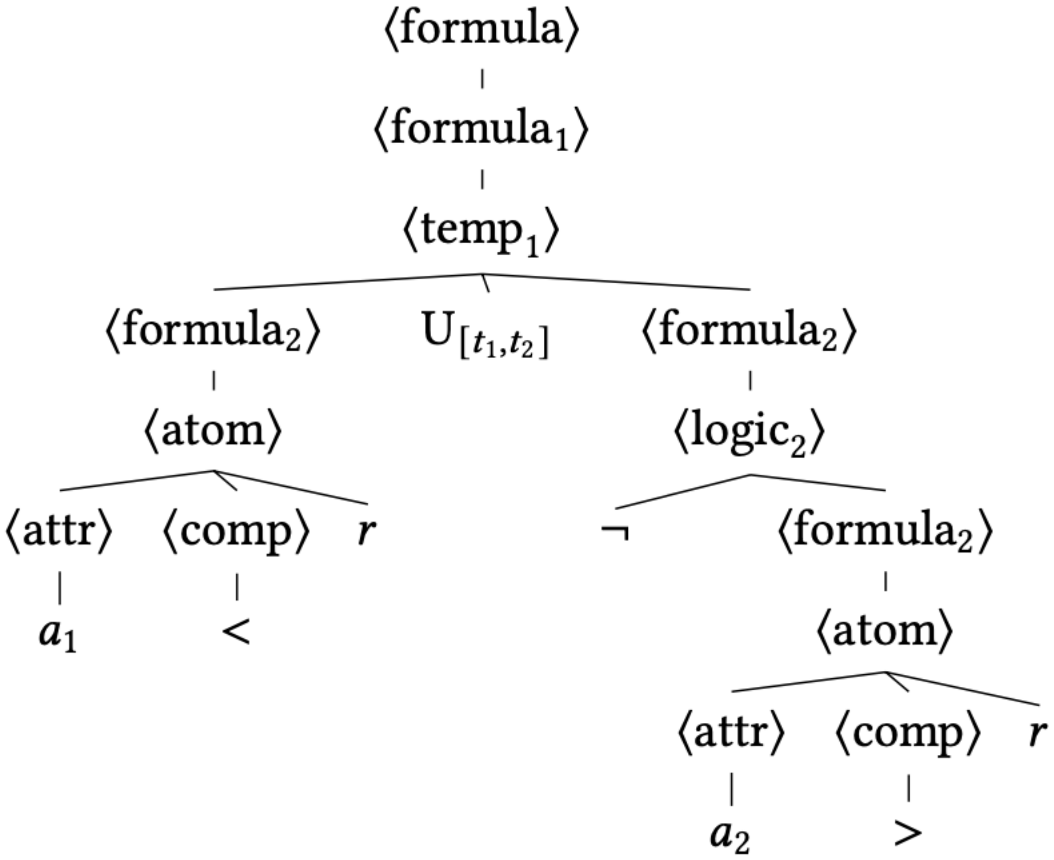
Evolutionary algorithm

- It builds the offspring population P'
- It merges the parent and offspring populations
- It shrinks the resulting new population P

```
function evolve():
    P ← initialize(G, n_pop)
    foreach i ∈ {1, ..., n_gen} do
        P' ← ∅
        while |P'| ≤ n_pop do
            i ← 0
            repeat
                if ~ U(0, 1) ≤ p_xover then
                    (φ_p,1, f_p,1) ← select(P)
                    (φ_p,2, f_p,2) ← select(P)
                    φ_c ← crossover(φ_p,1, φ_p,2; G)
                else
                    (φ_p, f_p) ← select(P)
                    φ_c ← mutate(φ_p; G)
                end
                i ← i + 1
            until (φ_c ∉ P ∪ P') ∧ (i ≤ n_atts)
            P' ← P' ∪ {(φ_c, f_opt(φ_c; L))}
        end
        P ← P ∪ P'
        while |P| ≥ n_pop do
            P ← P \ {worst(P)}
        end
    end
    return best(P)
end
```

Building the populations

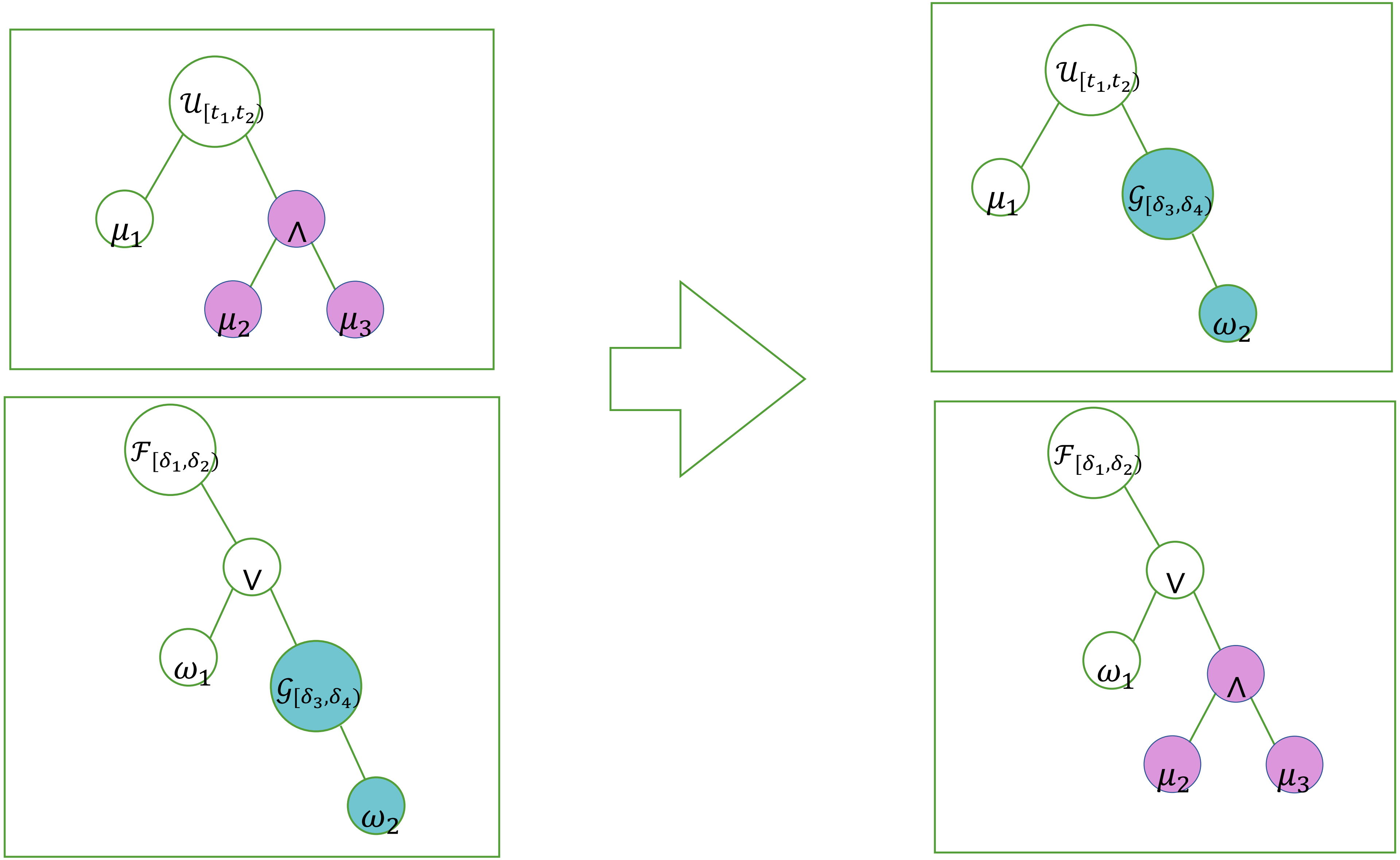
- Candidate formulas are represented as derivation trees of a grammar



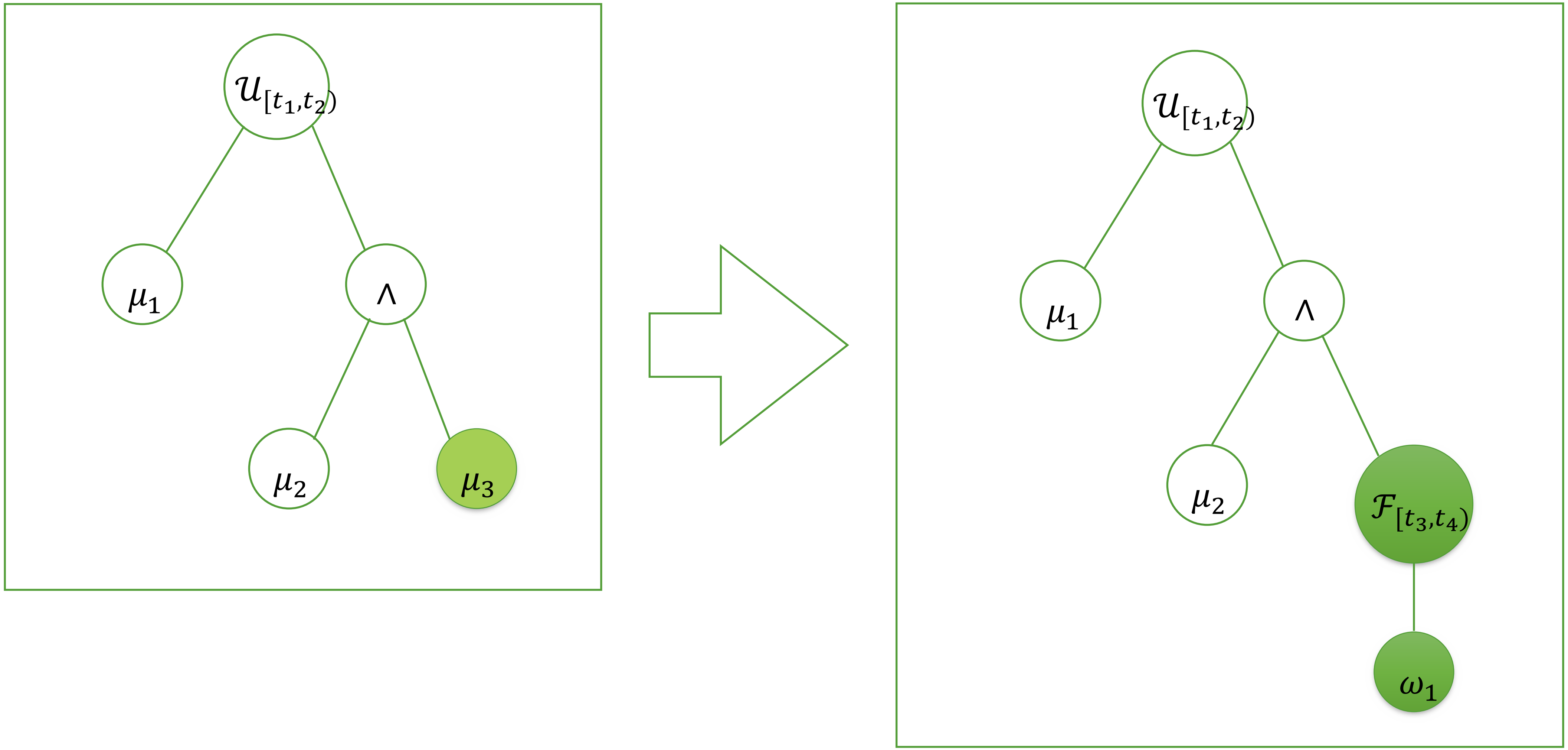
Context Free Grammar

$$
\begin{aligned}
\langle \text{formula} \rangle &::= \langle \text{formula}_1 \rangle \\
\langle \text{formula}_i \rangle &::= \begin{cases} \langle \text{atom} \rangle \mid \langle \text{logic}_i \rangle \mid \langle \text{temp}_1 \rangle & \text{if } i < i_{\max} \\ \langle \text{atom} \rangle \mid \langle \text{logic}_i \rangle & \text{otherwise} \end{cases} \\
\langle \text{logic}_i \rangle &::= \neg \langle \text{formula}_i \rangle \mid \langle \text{formula}_i \rangle \wedge \langle \text{formula}_i \rangle \\
\langle \text{temp}_i \rangle &::= \langle \text{formula}_{i+1} \rangle \mathrm{U}_{\langle \text{interval} \rangle} \langle \text{formula}_{i+1} \rangle \mid \\
&\quad\; \mathrm{G}_{\langle \text{interval} \rangle} \langle \text{formula}_{i+1} \rangle \mid \mathrm{F}_{\langle \text{interval} \rangle} \langle \text{formula}_{i+1} \rangle \\
\langle \text{interval} \rangle &::= [\langle \text{num} \rangle, \langle \text{num} \rangle] \\
\langle \text{atom} \rangle &::= \langle \text{attr} \rangle \langle \text{comp} \rangle \langle \text{num} \rangle \\
\langle \text{attr} \rangle &::= a_1 \mid a_2 \mid \ldots \mid a_{|A|} \\
\langle \text{comp} \rangle &::= \; < \mid > \\
\langle \text{num} \rangle &::= \langle \text{digit} \rangle \langle \text{digit} \rangle \\
\langle \text{digit} \rangle &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{aligned}
$$

Crossover operator



Mutation operator



Learning the Parameters

Problem

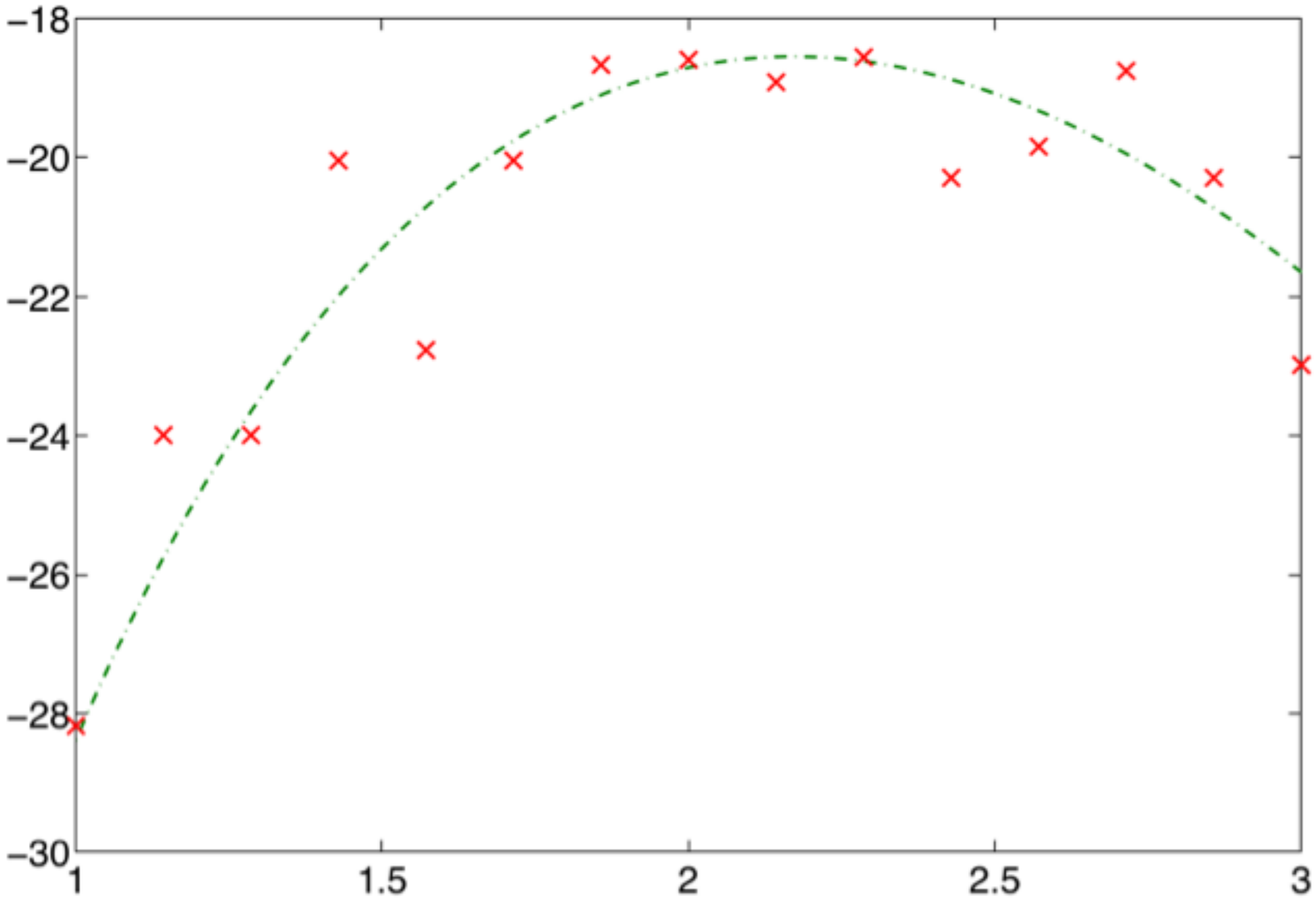
Given a PSTL formula ϕ, a parameter space K, find Θ^* that maximises the discrimination function $f_{opt}(\varphi_\Theta)$

Methodology

1. Sample $\{(\theta_{(i)}, y_{(i)}), i = 1,\ldots,n\}$
2. Emulate (**GP Regression**): $G[R_\phi] \sim GP(\mu,k)$
3. Optimize the emulation via **GP-UCB algorithm**, new $\theta_{(n+1)}$

(1) The $G(\phi_\theta)$ Computation

Collection of the training set $\{(\theta^{(i)}, y^{(i)}), i = 1,\ldots,m\}$ for parameters values θ.

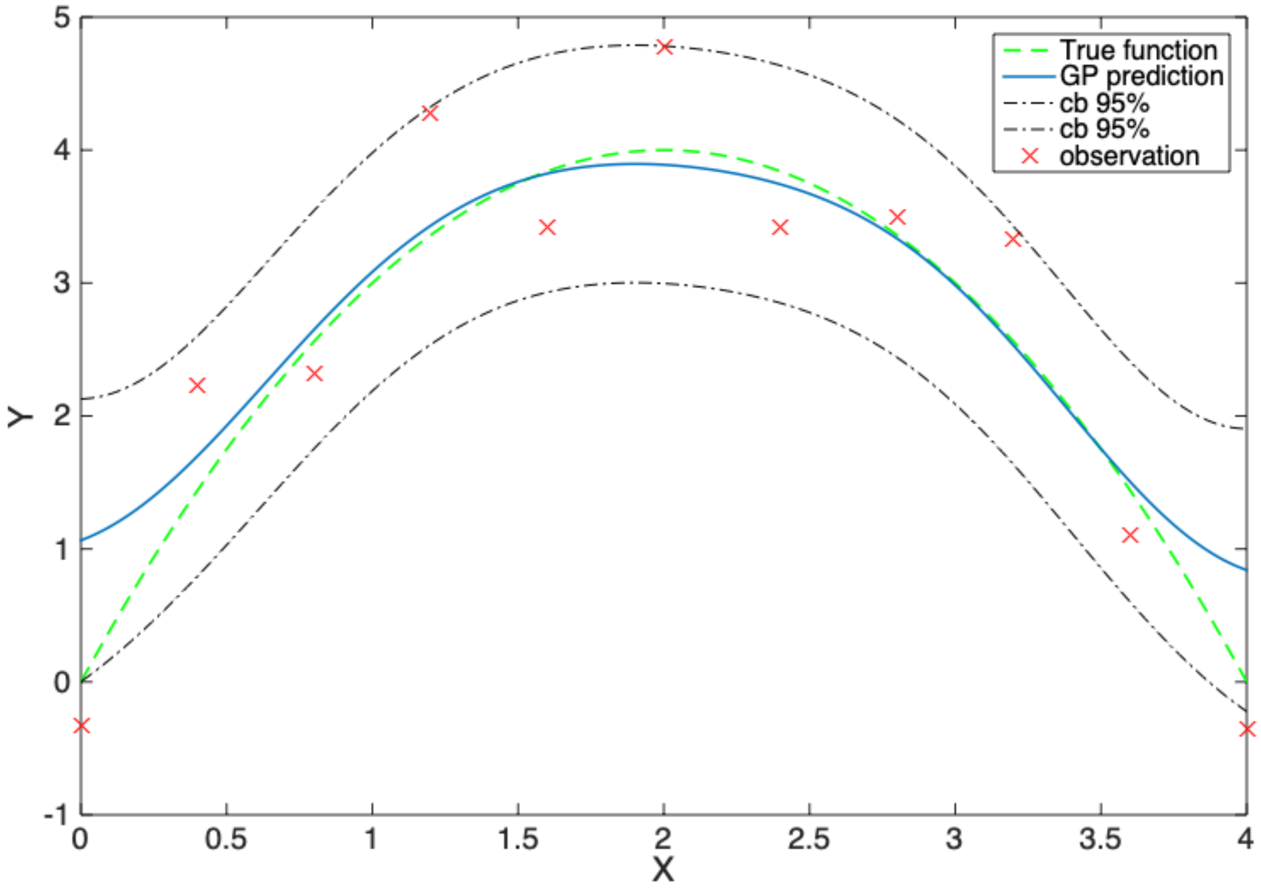


(2) The GP Regression

We have noisy observations y of the function value distributed around an unknown true value $f(\theta)$ with spherical Gaussian noise

(2) The GP Regression

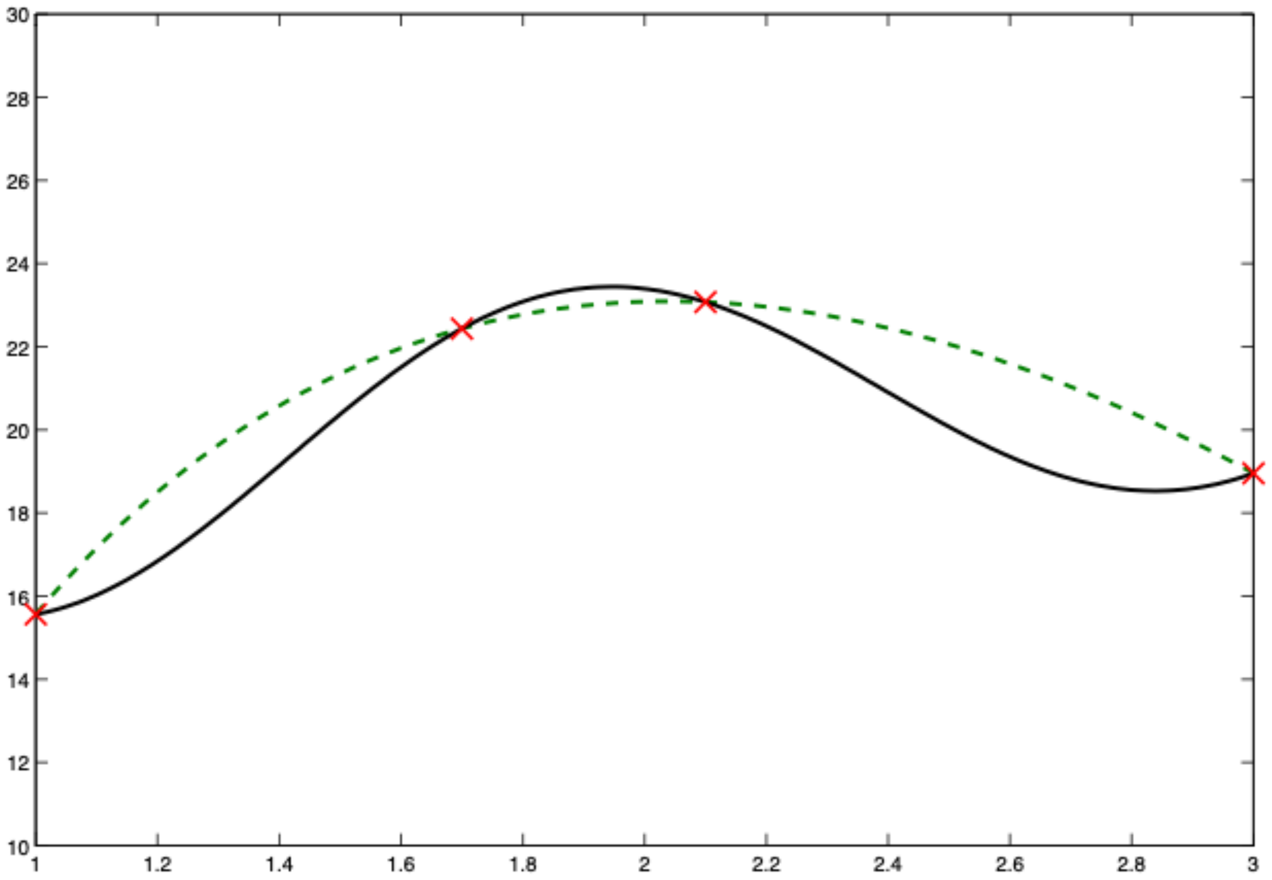
We have noisy observations y of the function value distributed around an unknown true value f (θ) with spherical Gaussian noise



(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the

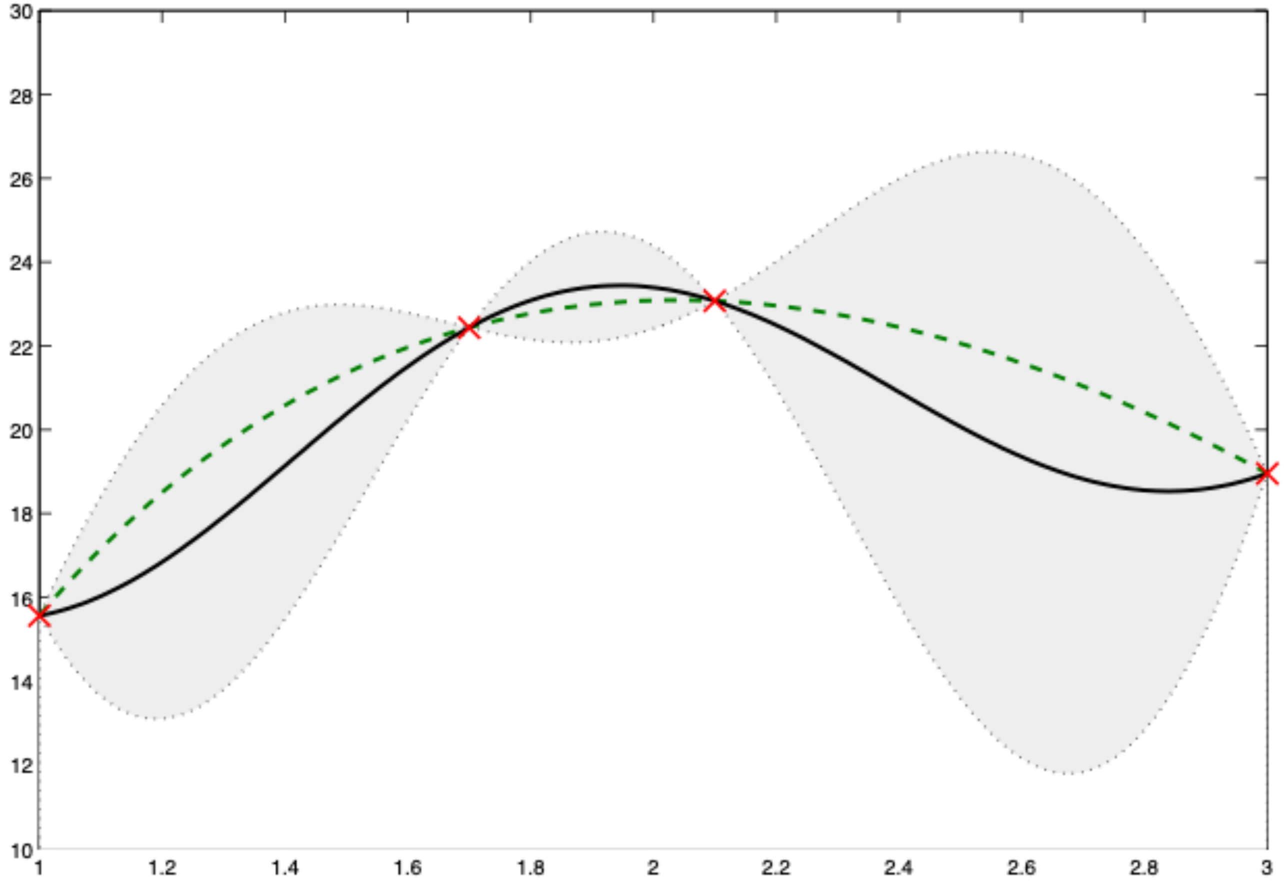
95% upper quantile of the distribution: $\theta_{t+1} = argmax_{\theta}[\mu^*(\theta) + \beta_t\sqrt{k^*(\theta,\theta)}]$



(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the

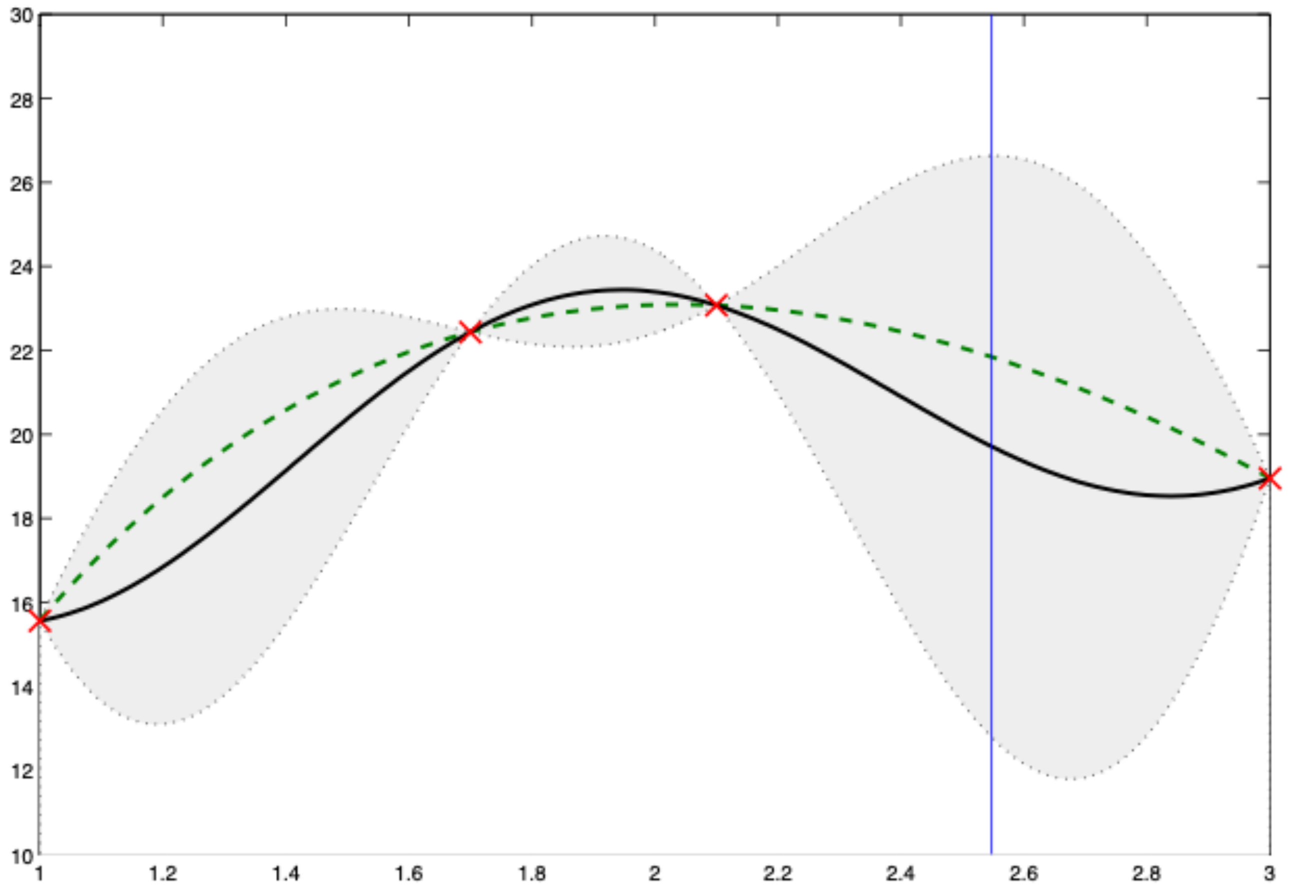
95% upper quantile of the distribution: $\theta_{t+1} = argmax_{\theta}[\mu^*(\theta) + \beta_t\sqrt{k^*(\theta,\theta)}]$



(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the

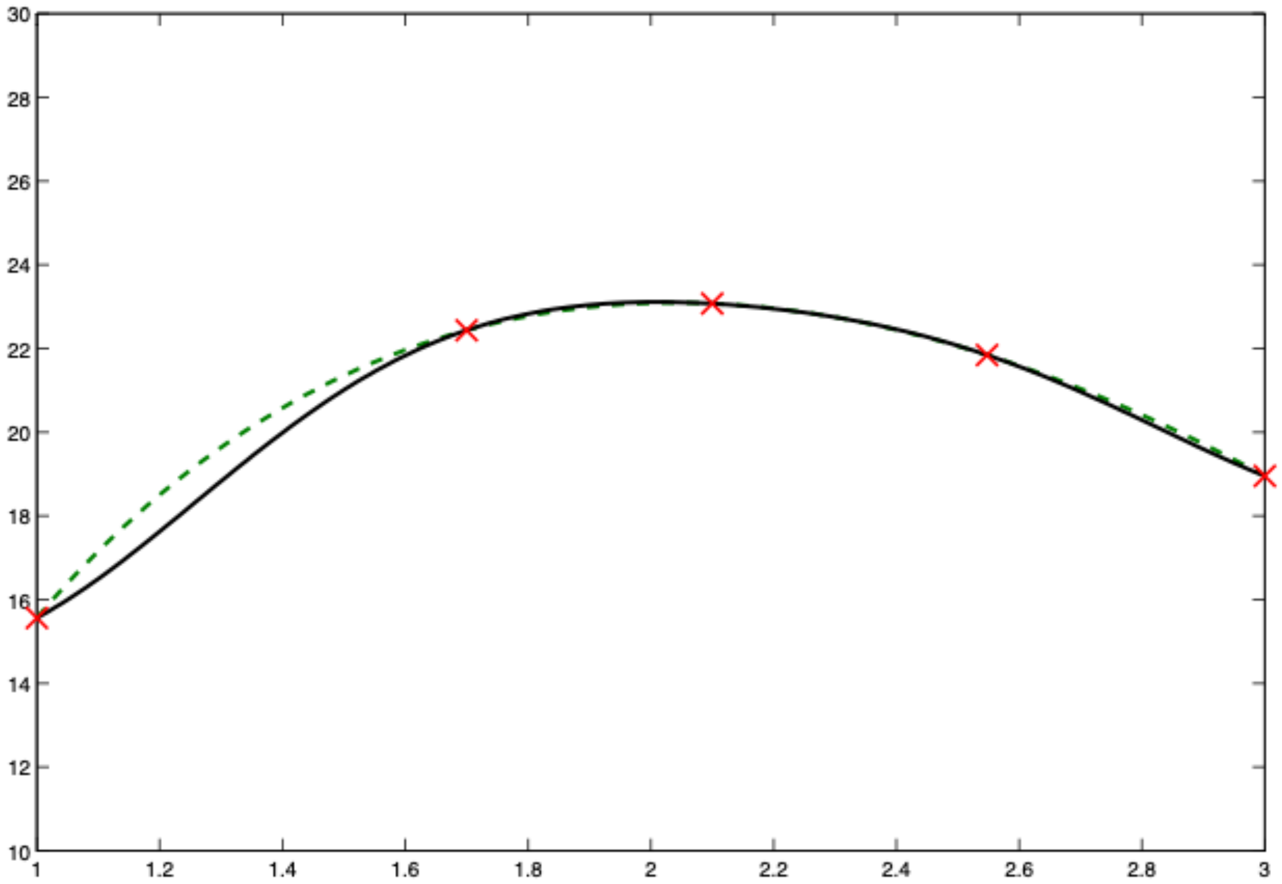
95% upper quantile of the distribution: $\theta_{t+1} = argmax_{\theta}[\mu^*(\theta) + \beta_t \sqrt{k^*(\theta,\theta)}]$



(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the

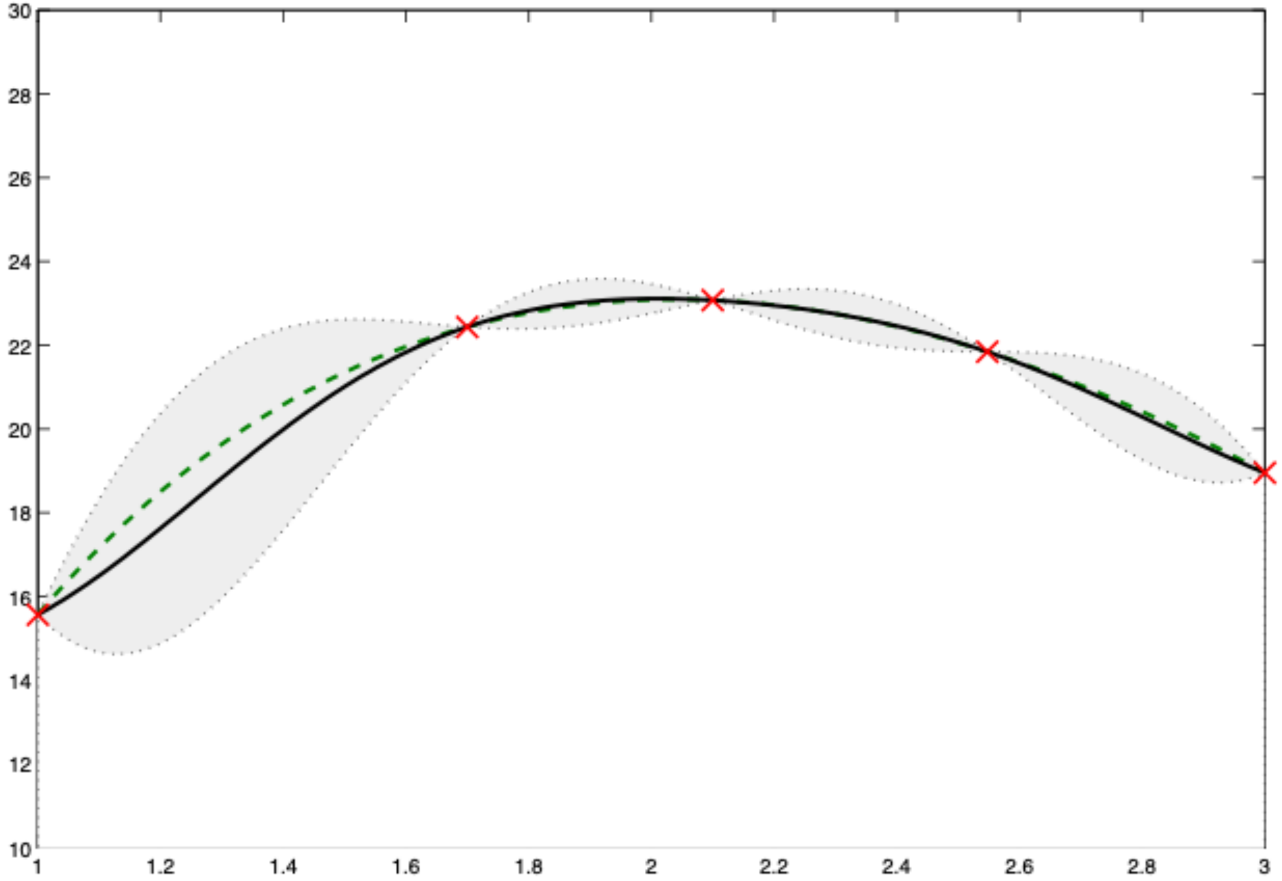
95% upper quantile of the distribution: $\theta_{t+1} = argmax_{\theta}[\mu^*(\theta) + \beta_t\sqrt{k^*(\theta,\theta)}]$



(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the

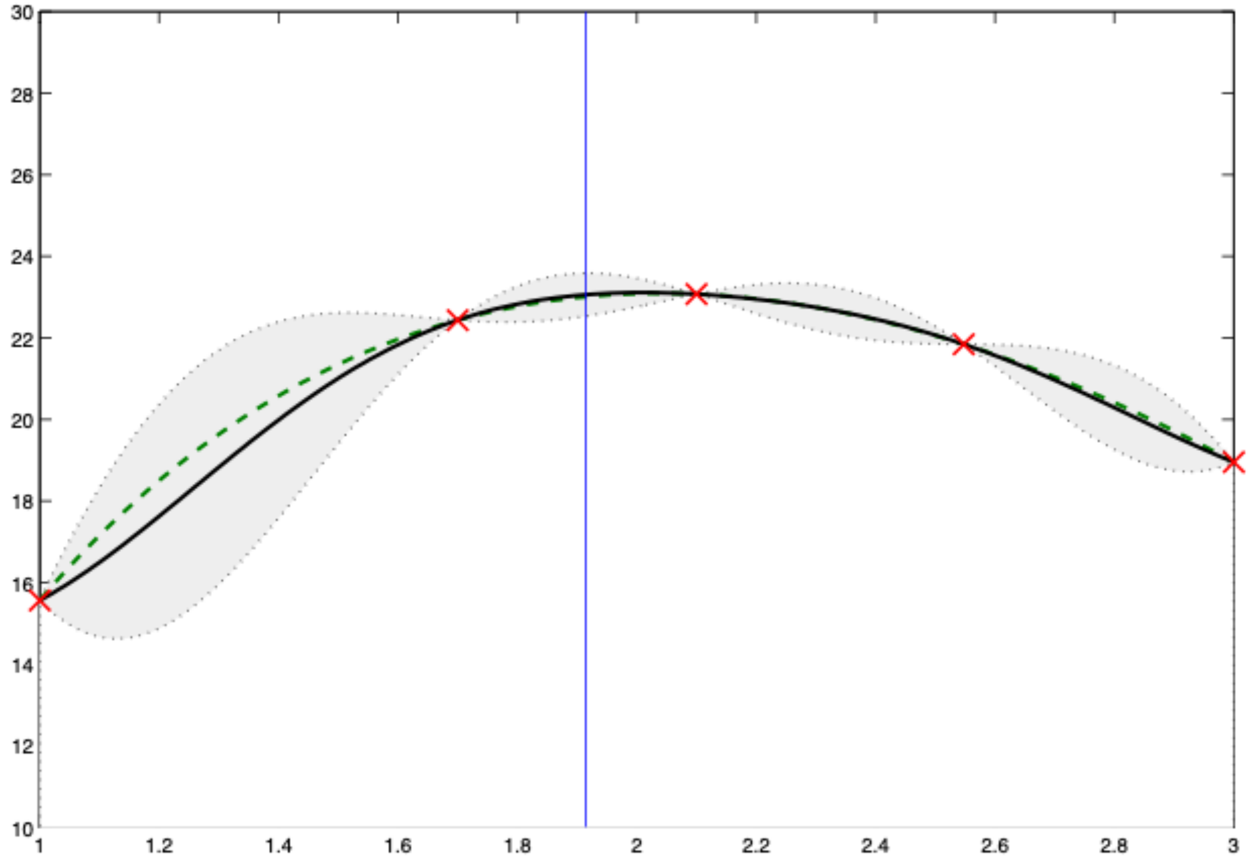
95% upper quantile of the distribution: $\theta_{t+1} = argmax_\theta[\mu^*(\theta) + \beta_t\sqrt{k^*(\theta,\theta)}]$



(3) The GP-UCB Algorithm

Balance Exploration and Exploitation: we maximise the

95% upper quantile of the distribution: $\theta_{t+1} = argmax_{\theta}\left[\mu^*(\theta) + \beta_t\sqrt{k^*(\theta,\theta)}\right]$



Fitness Function for the two-classes problem

$$f(\varphi; X_{\mathcal{L}}^{+}, X_{\mathcal{L}}^{-}) = -\frac{\mu_{\varphi,X_{\mathcal{L}}^{+}} - \mu_{\varphi,X_{\mathcal{L}}^{-}}}{\sigma_{\varphi,X_{\mathcal{L}}^{+}} + \sigma_{\varphi,X_{\mathcal{L}}^{-}}}$$

$$\mu_{\varphi,X} = \frac{1}{|X|} \sum_{\boldsymbol{x} \in X} \rho(\varphi, \boldsymbol{x})$$

$$\sigma_{\varphi,X} = \sqrt{\frac{1}{|X|} \sum_{\boldsymbol{x} \in X} \left(\rho(\varphi, \boldsymbol{x}) - \mu_{\varphi,X}\right)^{2}}$$

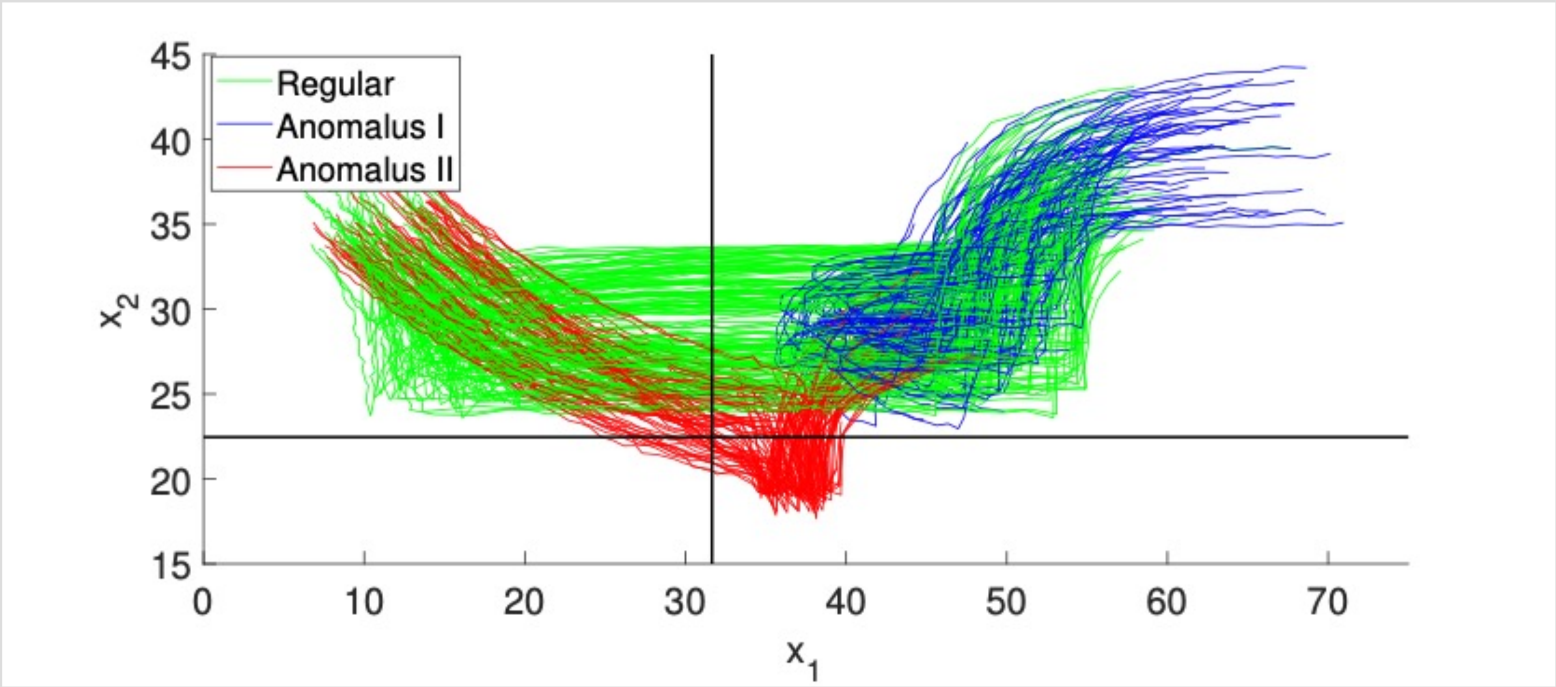
Fitness Function for the one-class problem

$$f(\varphi; X^+_{\mathcal{L}}) = \alpha \frac{1}{\left|X^+_{\mathcal{L}}\right|} \left|\{\boldsymbol{x} \in X^+_{\mathcal{L}} : \boldsymbol{x} \not\models \varphi\}\right| + \frac{1}{\sigma'_{\varphi, X^+_{\mathcal{L}}} \left|X^+_{\mathcal{L}}\right|} \sum_{\boldsymbol{x} \in X^+_{\mathcal{L}}} |\rho(\varphi, \boldsymbol{x})|$$

$$\sigma'_{\varphi, X} = \sqrt{\frac{1}{|X|} \sum_{\boldsymbol{x} \in X} \left(|\rho(\varphi, \boldsymbol{x})| - \frac{1}{|X|} \sum_{\boldsymbol{x} \in X} |\rho(\varphi, \boldsymbol{x})| \right)^2}$$

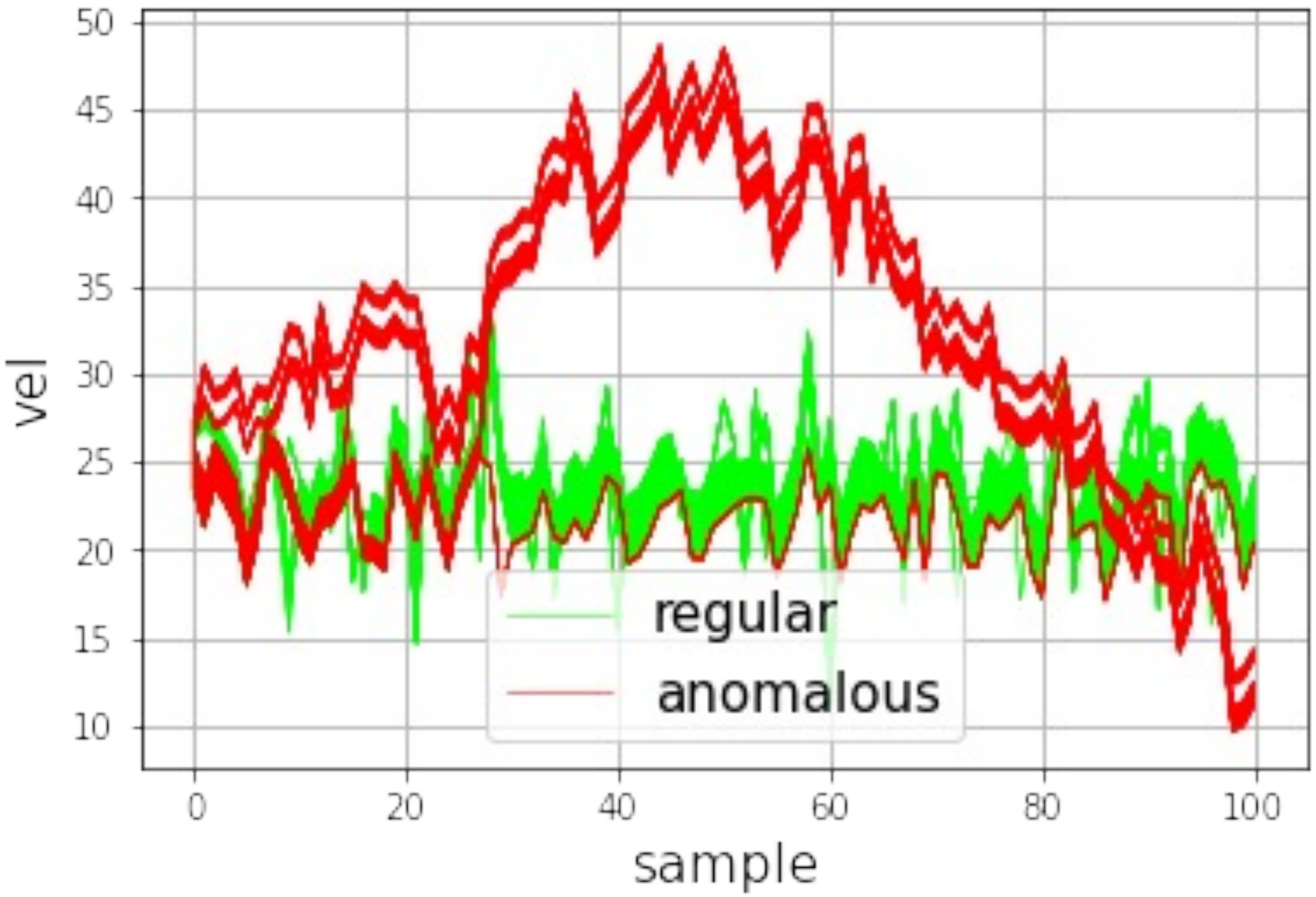
Maritime Surveillance

Synthetic dataset of naval surveillance of 2-dimensional coordinates traces of vessels behaviours.



$$\phi = ((x_2 > 22.46)\,\mathcal{U}_{[49,287]}\,(x_1 \leq 31.65))$$

Train Cruise

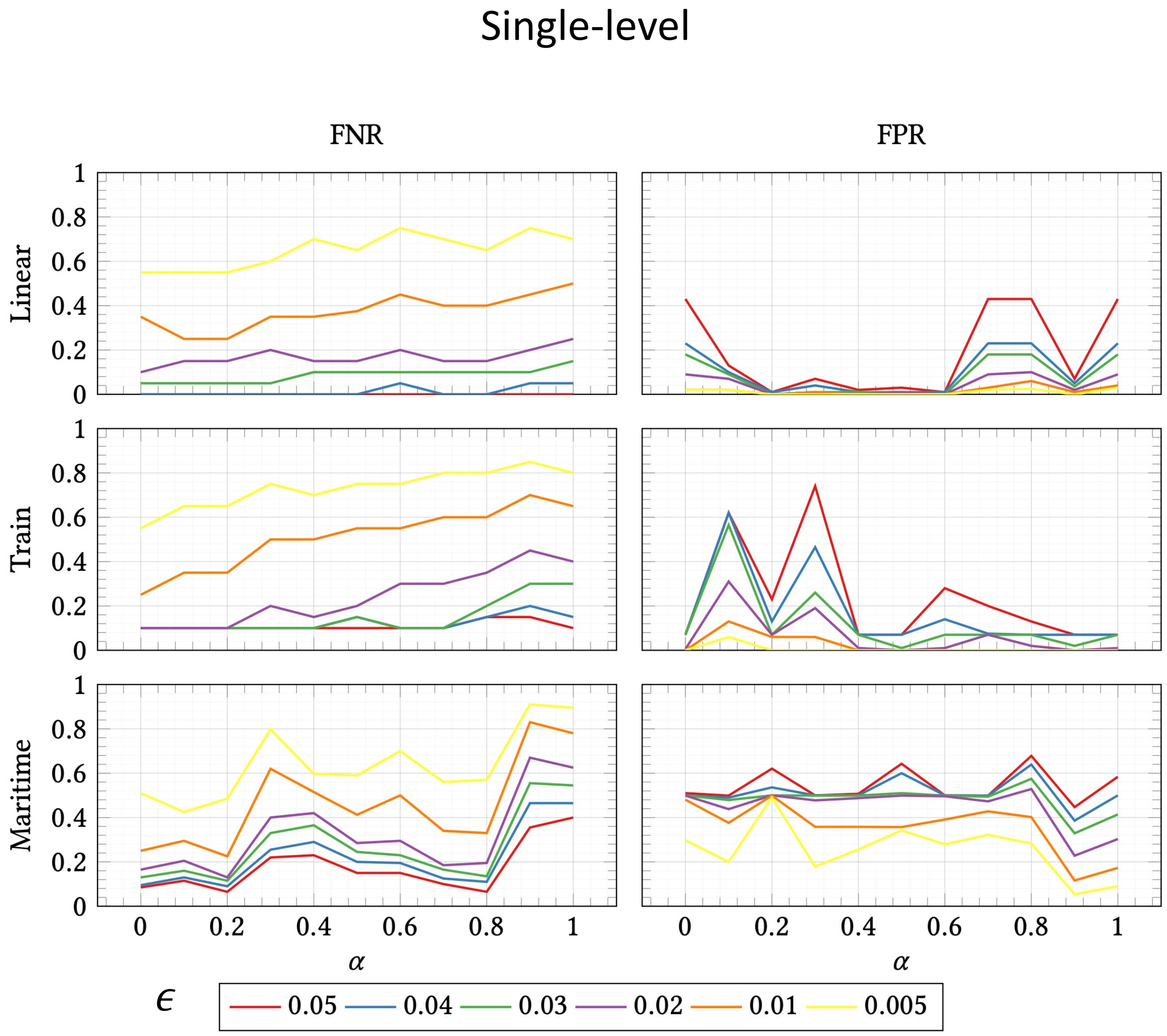


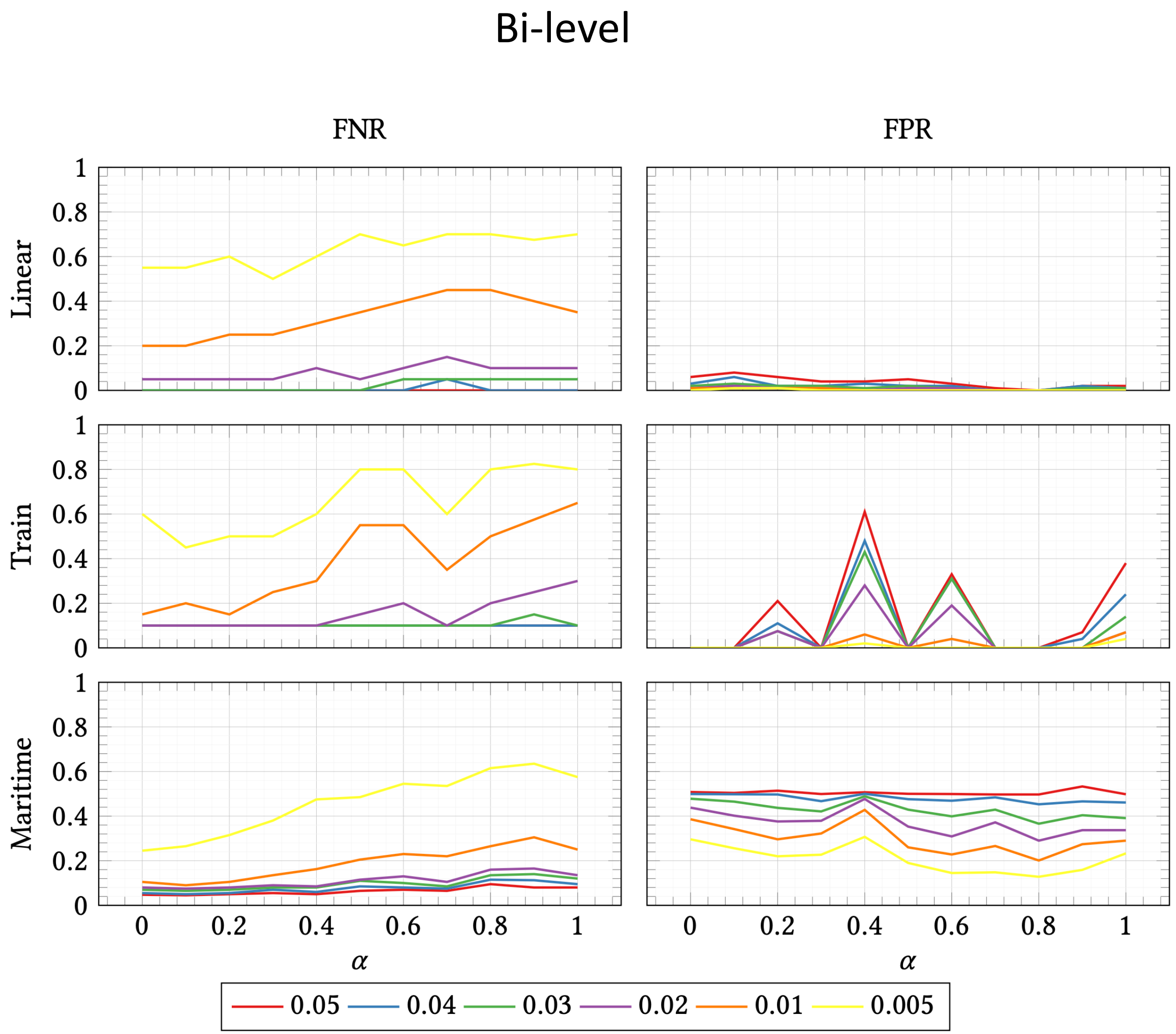
$$(F_{[22,40]}(\mathtt{vel} > 24.48)) \wedge (F_{[46,49]}(19.00 < \mathtt{vel} < 26.44))$$

Results (supervised learning)

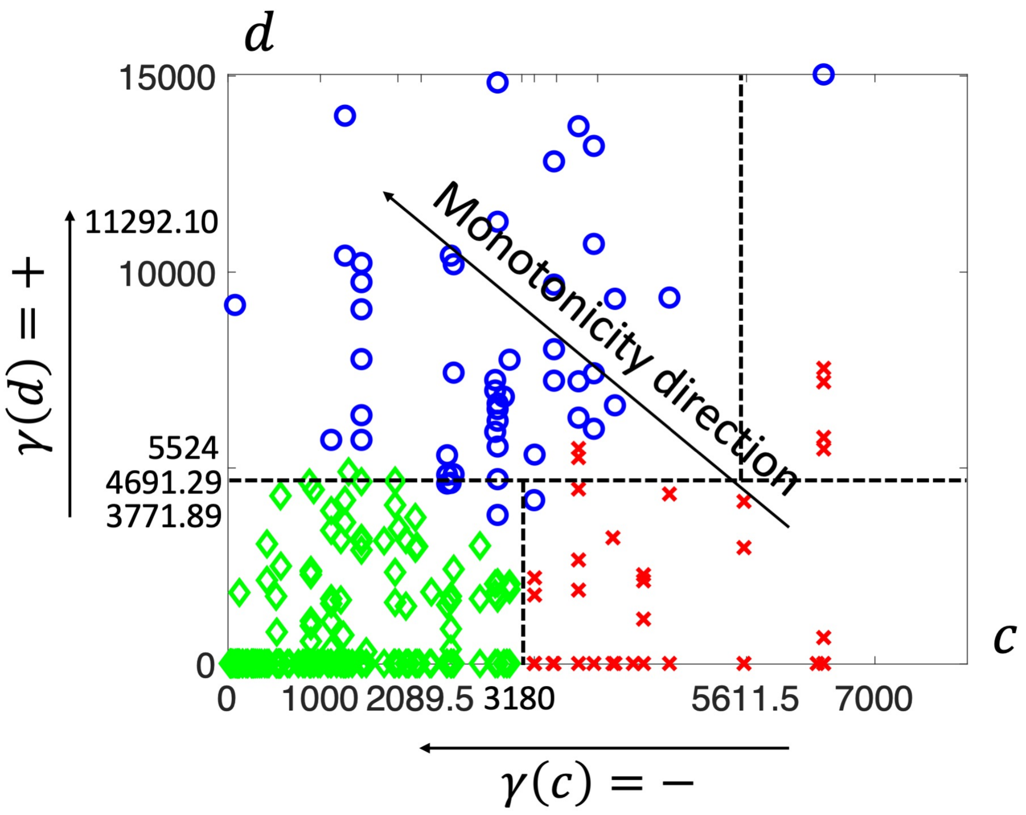
Dataset	Algorithm	FNR	FPR	MCR	Time
Maritime	BUSTLE (single-level)	0.00	0.00	0.00	109
	BUSTLE (bi-level)	0.00	0.00	0.00	1477
	[23]	0.00	0.00	0.00	N/A
	[22]	0.05	0.02	0.04	73
	[6]	N/A	N/A	0.02	140
Linear	BUSTLE (single-level)	0.00	0.00	0.00	15
	BUSTLE (bi-level)	0.00	0.00	0.00	112
	[23]	0.01	0.01	0.01	N/A
	[22]	N/A	N/A	0.02	39
Train	BUSTLE (single-level)	0.03	0.05	0.04	26
	BUSTLE (bi-level)	0.00	0.03	0.02	523
	[23]	0.07	0.32	0.19	N/A
	[22]	N/A	N/A	0.02	32

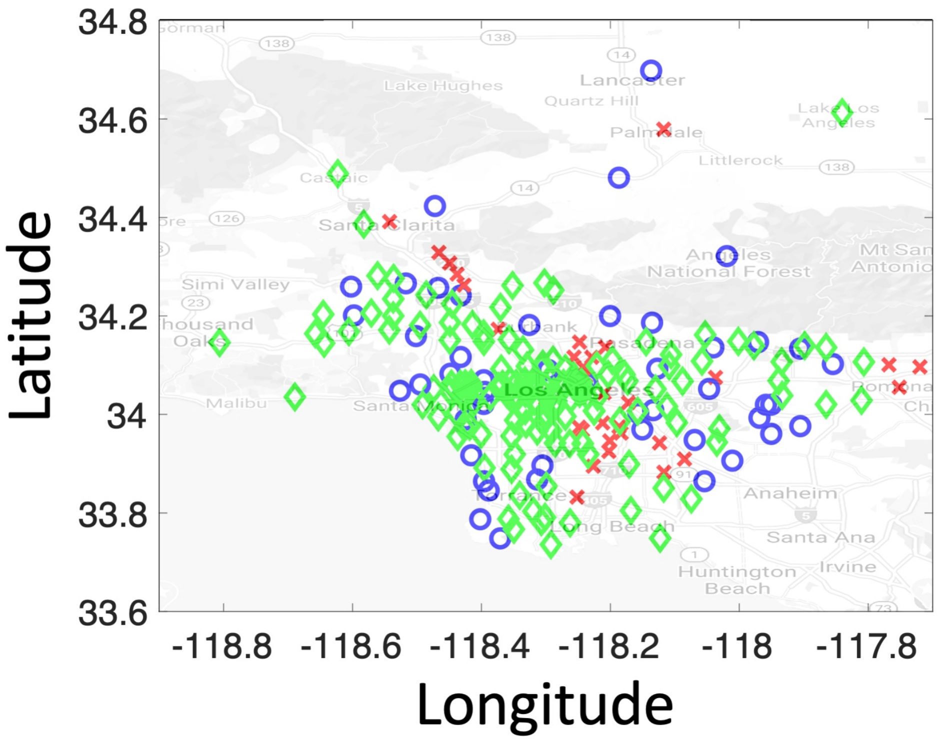
Results (semi-supervised learning)





Learning STL-based clustering (Unsupervised Learning)





Goal: clusterizing spatio-temporal data using formal logic

[2] Mining Interpretable Spatio-temporal Logic Properties for Spatially Distributed Systems, ATVA, [Mohammadinejad et al., 2021]

STL-based clustering of time-series data:

- Considerable interest in learning logical properties of temporal data using logics such as Signal Temporal Logic (STL)
- Signal Temporal Logic (STL):
  - A logic over Boolean and temporal combinations of signal predicates
- There is limited work on discovering such relations on spatio-temporal data

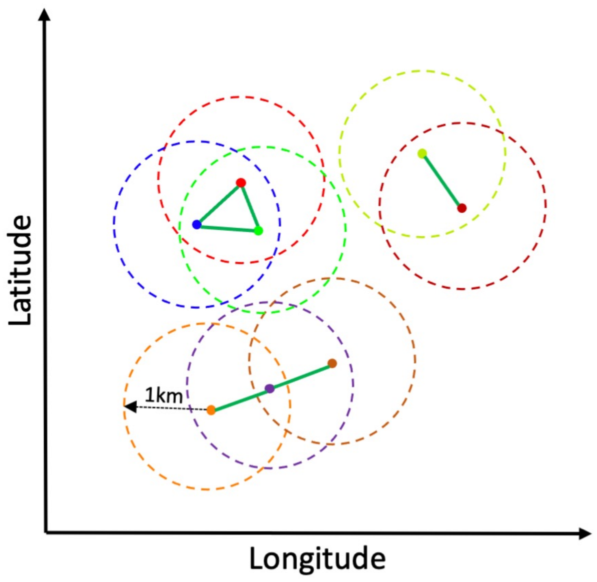
We propose the first set of algorithms for unsupervised learning of spatio-temporal data using formal logics

Spatial Model:

We model the spatial configuration as a weighted graph $S = \langle L, W \rangle$

L: set of locations

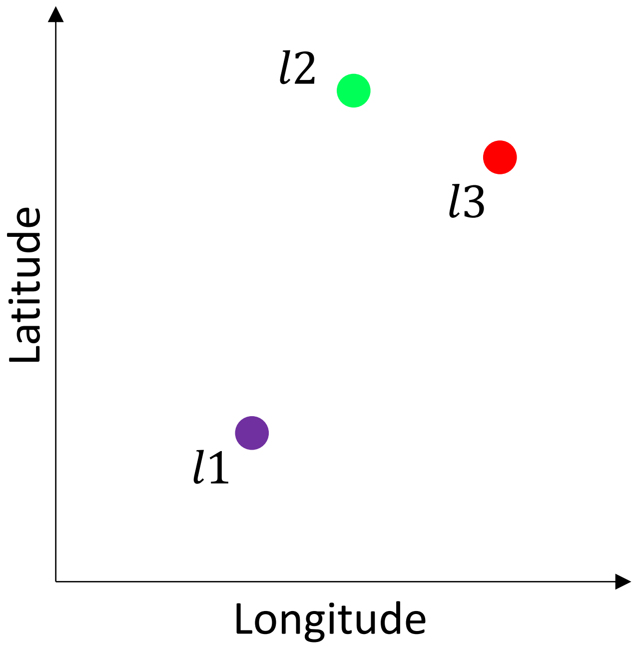
W: proximity relation between locations

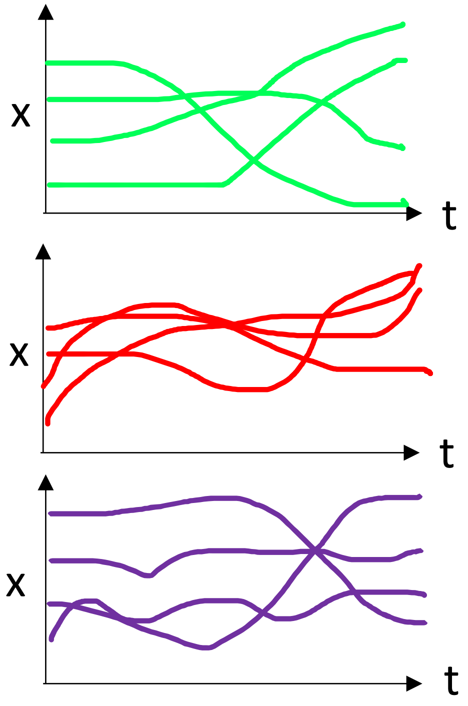


Connectivity graph
$W: spatial\ proximity$

Spatio-temporal trace:

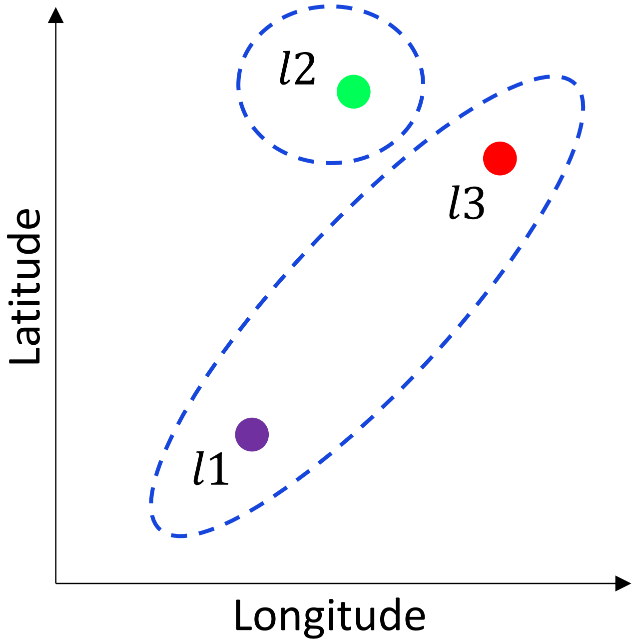
- Time-series data (trace/signal): a sequence of data values indexed by time stamps
- A spatio-temporal trace associates each location in a spatial model with a time-series trace

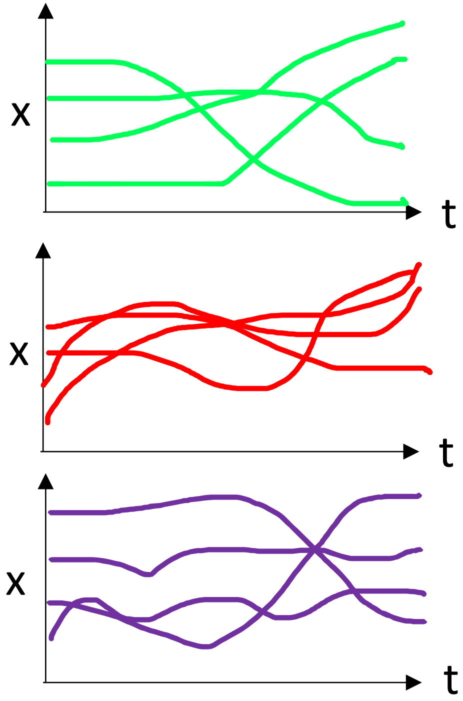




Spatio-temporal data clustering:

- It is a process of grouping data with similar spatial attributes, temporal attributes, or both [1]





Spatio-Temporal Reach and Escape Logic (STREL):

- An extension of STL with two spatial operators: Reach and Escape

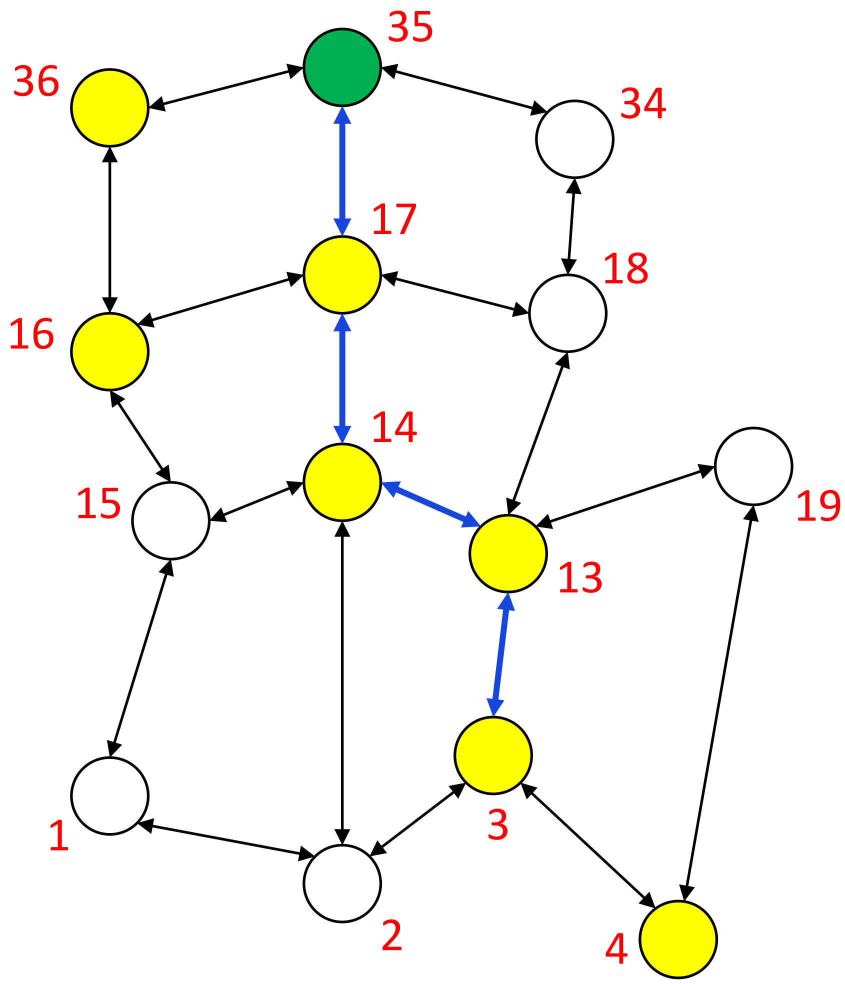
- Somewhere, Everywhere and Surround operators can be derived from Reach and Escape

- I will explain Reach and Everywhere operators

  Refer to [2] and [3] to learn more about other spatial operators

[2] Monitoring spatio-temporal properties (invited tutorial) [Nenzi et al., 2020]
[3] https://www.youtube.com/watch?v=EfB1r9htG6M&t=179s

Reach operator (R)

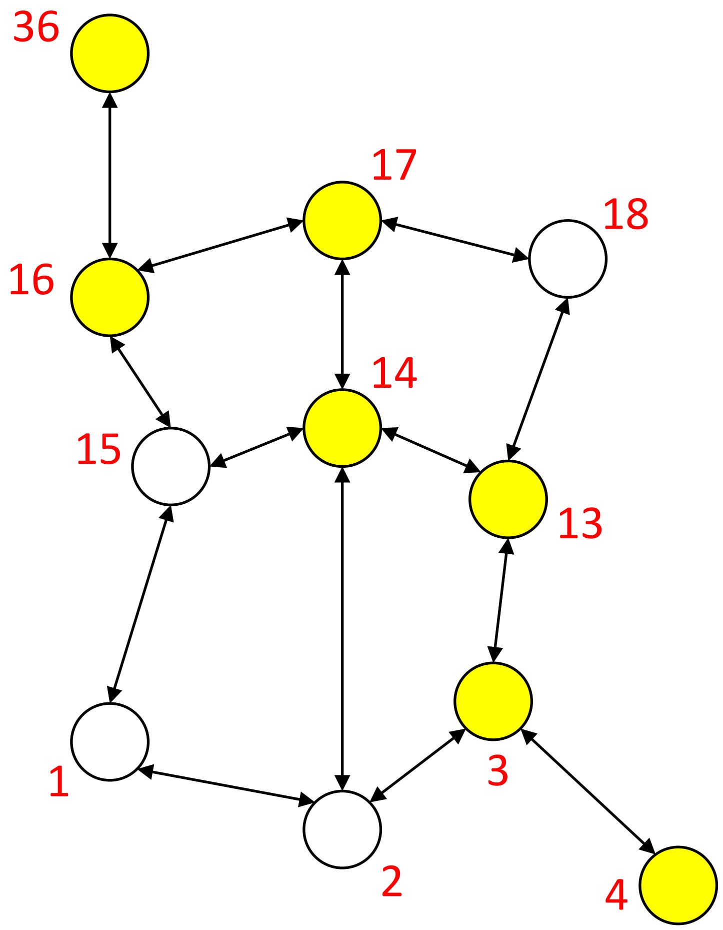


$$\varphi = yellow \; R_{[1,4]} green$$

l_3 *satisfies* φ
$path = l_3, l_{13} l_{14} l_{17} l_{35}$

l_4 *does not satisfy* φ

Everywhere operator (□)



$\varphi = \Box_{[2,3]} yellow$

l_1 *satisfies* φ

l_2 *does not satisfy* φ

[2] Monitoring spatio-temporal properties (invited tutorial) [Nenzi et al., 2020]
[3] https://www.youtube.com/watch?v=EfB1r9htG6M&t=179s

Parametric STREL (PSTREL):

- Replacing values in STREL by parameters

$$\boxed{\varphi_1 R_{[0,1000]} \varphi_2}$$

$$\Downarrow$$

$$\boxed{\varphi_1 R_{[d_1,d2]} \varphi_2}$$

Monotonic PSTREL $\varphi(p)$:

- The polarity of a parameter p is:
  - $+$ if it is easier to satisfy φ as we increase the value of p
  - $-$ if it is easier to satisfy φ as we decrease the value of p

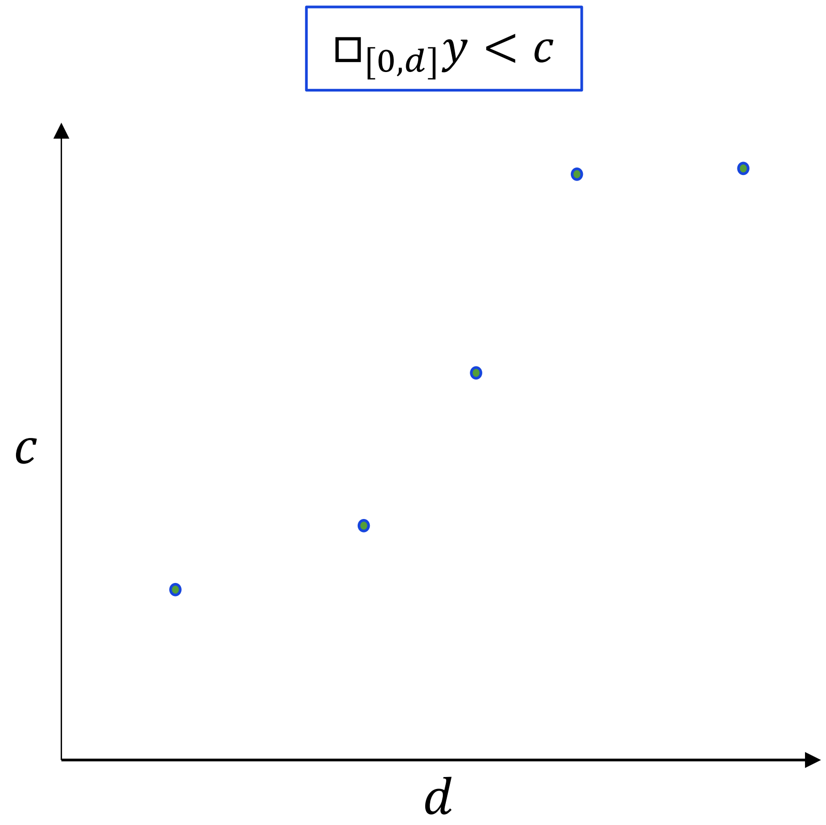
- Monotonic PSTREL:
  - All parameters have either $+$ or $-$ polarity

- Example: $\square_{[0,d]}\varphi$
  - Polarity of d is $-$

Validity Domain of PSTREL $\varphi(p)$

- Given a location l
- A set of spatio-temporal traces X associated with l
- The set of all valuations to p such that each trace in X satisfies the STREL formula
- Boundary of the validity domain:

  The robustness value with respect to at least one trace in X is ≈ 0
- Robustness means distance to satisfaction or violation



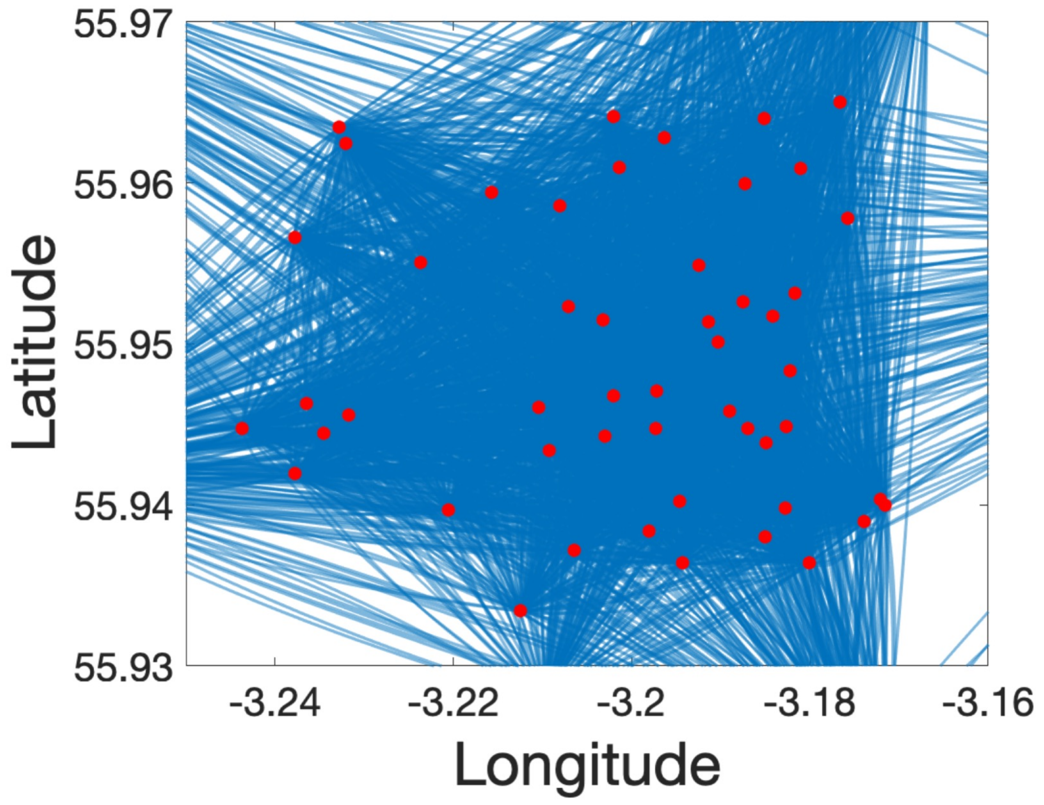
High-level steps:

- Constructing the spatial model
- Projecting each spatio-temporal trace to a tight valuation in the parameter space of a given PSTREL formula
- Clustering the trace projections
- Learning bounding boxes for each cluster using a Decision Tree based approach
- Learning a STREL formula for each cluster
- Improving the interpretability of the learned STREL formulas

Constructing Spatial Model:

Approach 1: fully connected graph

- Pros: gives the most accurate result
- Cons: computationally expensive

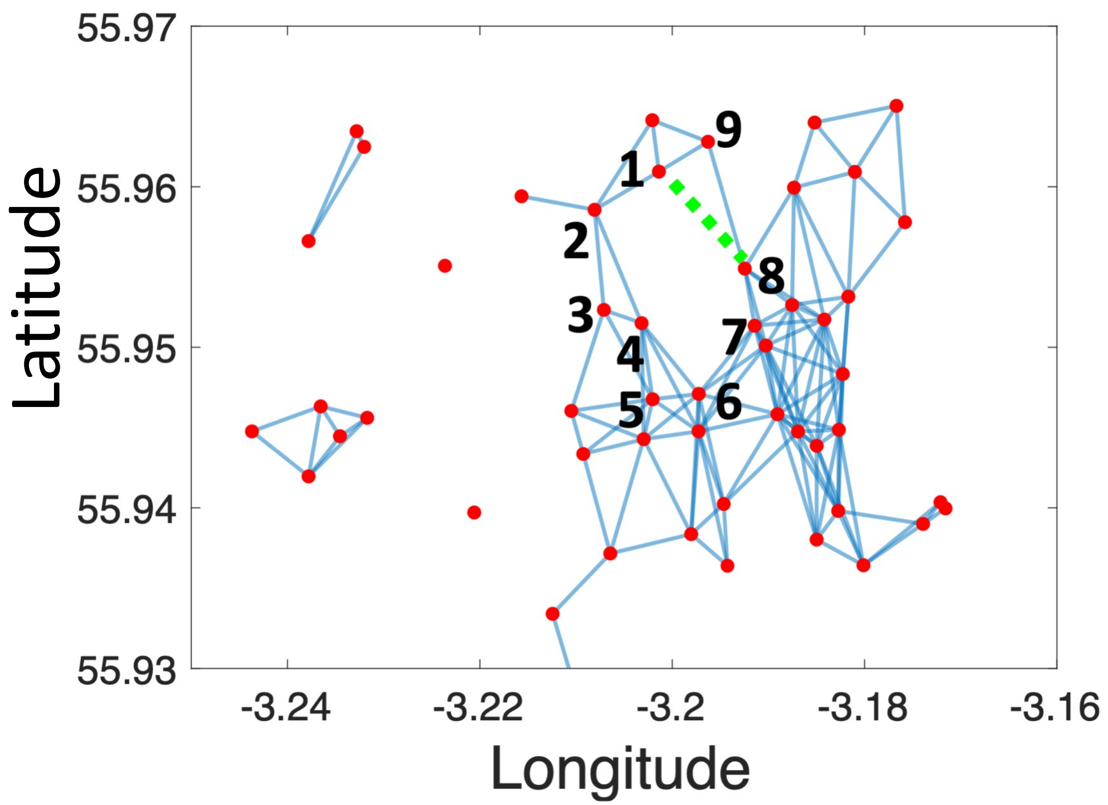


Constructing Spatial Model:

Approach 2: Connectivity graph that connects locations with distance less than a threshold

- Pros: lower cost
- Cons: disconnected spatial model

which affects the accuracy

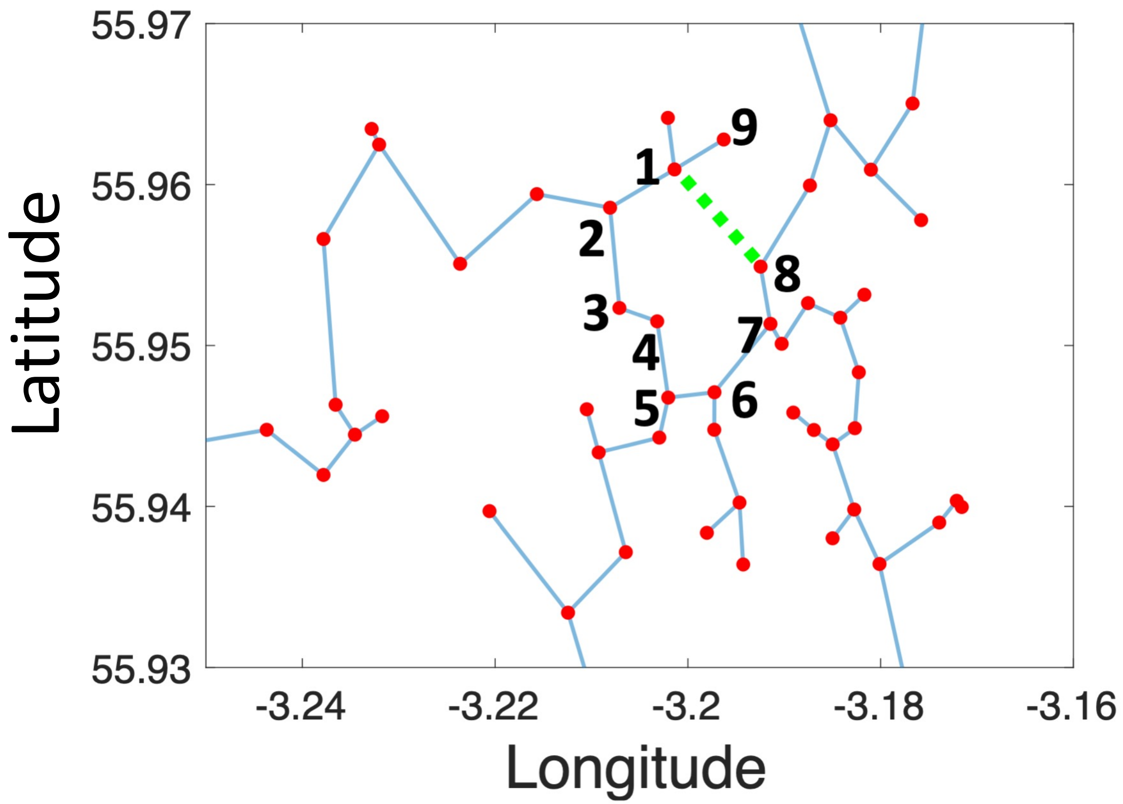


Constructing Spatial Model:

Approach 3: Minimum Spanning Tree (MST)

- Pros: low cost and connected graph
- Cons: overestimation of distance

between some nodes



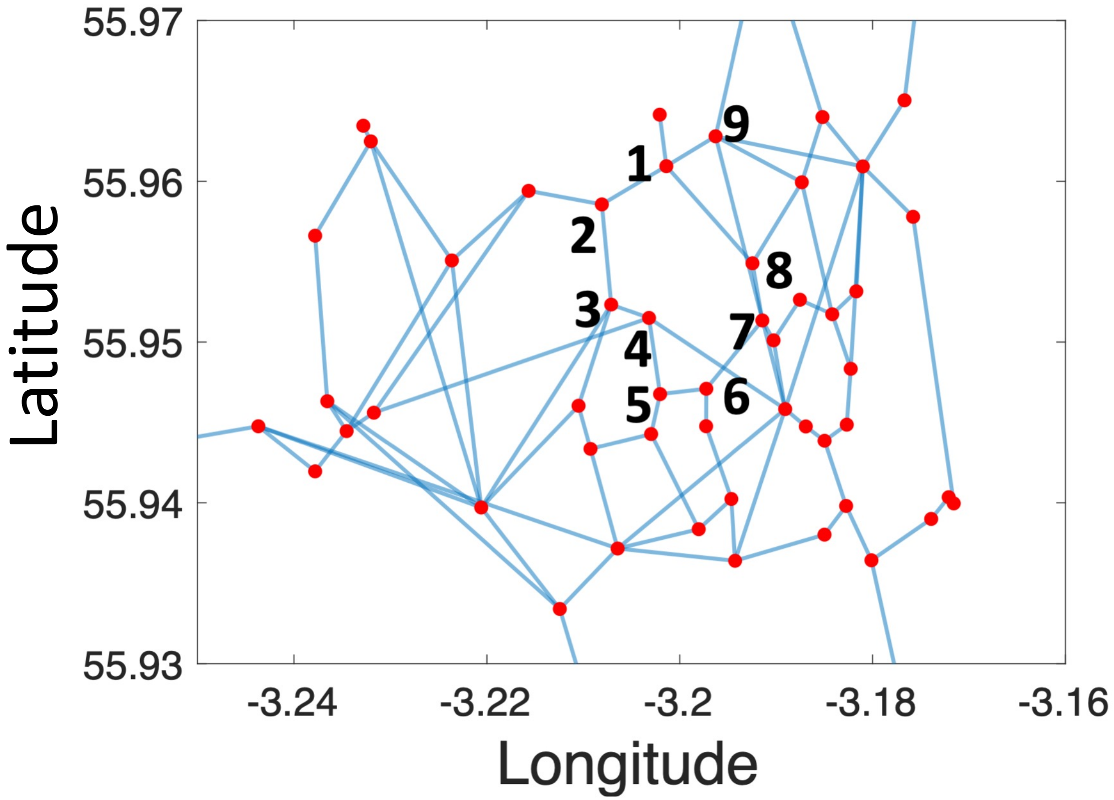
Constructing Spatial Model:

Approach 4: Enhanced Minimum Spanning Graph

Step1: create an MST

Step2: connect nodes that their shortest distance through MST is more than α times their actual distance (default $\alpha = 2$)

- Pros: low cost, connected graph and more accurate distance between nodes
- Cons: not as accurate as fully connected graph

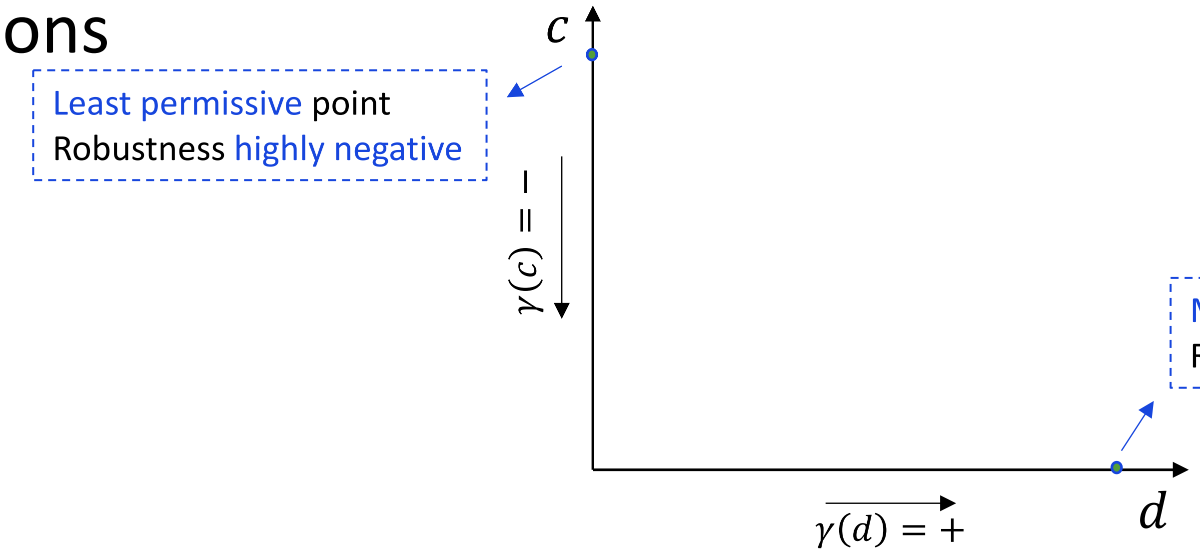


Spatio-temporal trace projection [4] :

- The user provides a PSTREL formula

$$\boxed{G_{[0,3hours]} \diamond_{[0,d]} (Bikes > c)}$$

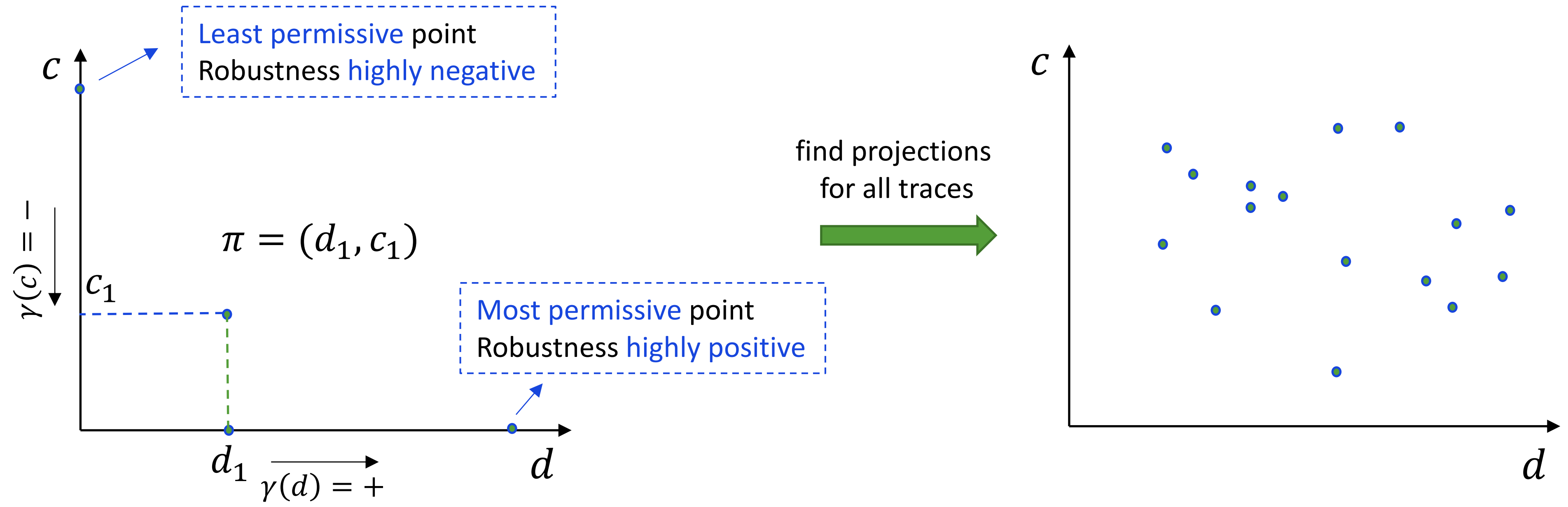
- The goal is to learn the tight parameter valuations for each spatio-temporal trace
- Tight parameter valuation is not unique, and each point on the boundary of validity domain corresponds to a tight parameter valuations



[4]: Logical clustering and learning for time-series data [V-Chanlatte et al., 2017]

Spatio-temporal trace projection [4] :

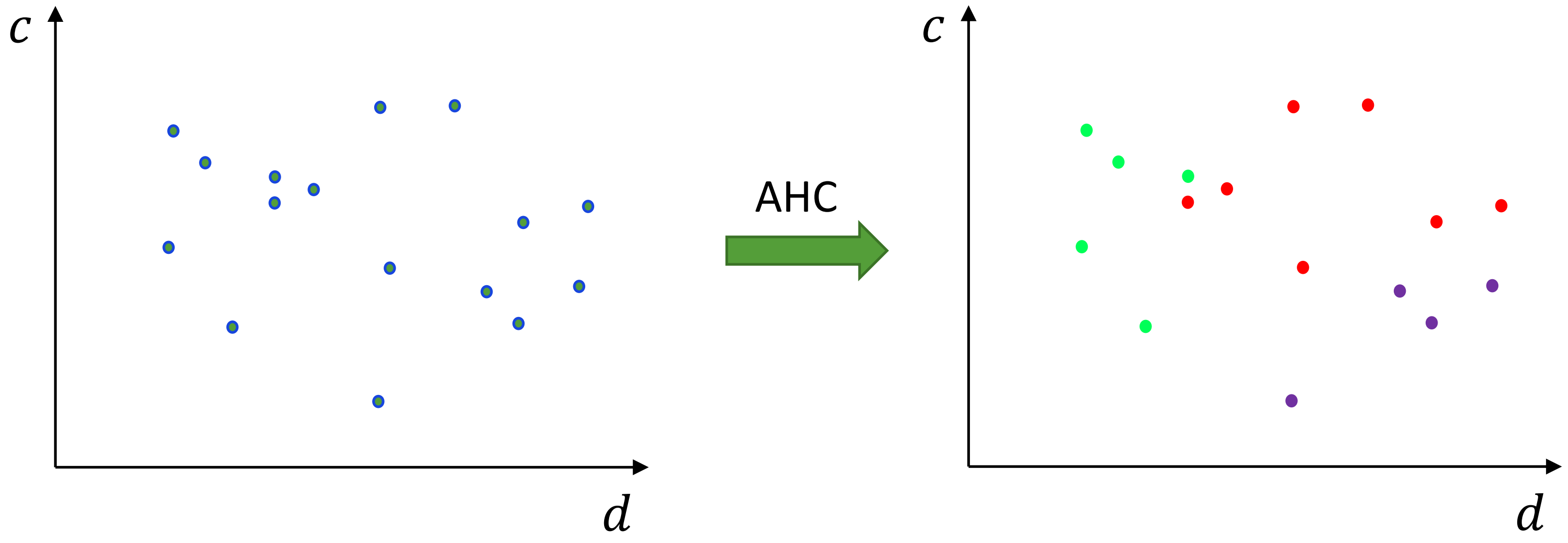
- We assume some ordering or priority on parameter space, e.g., $d >_p c$, provided by user
    1. Bisection search on d
    2. Bisection search on c



[4]: Logical clustering and learning for time-series data [V-Chanlatte et al., 2017]

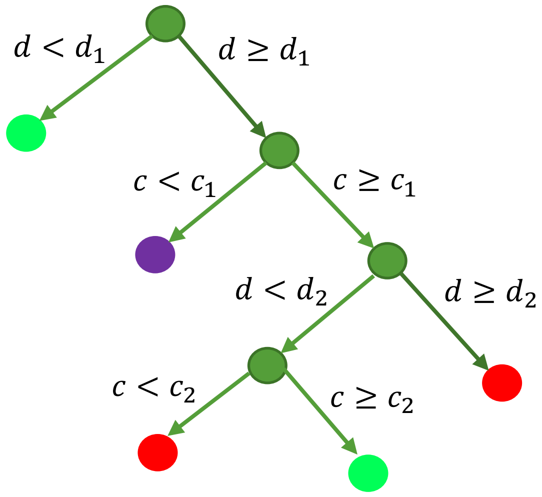
Clustering:

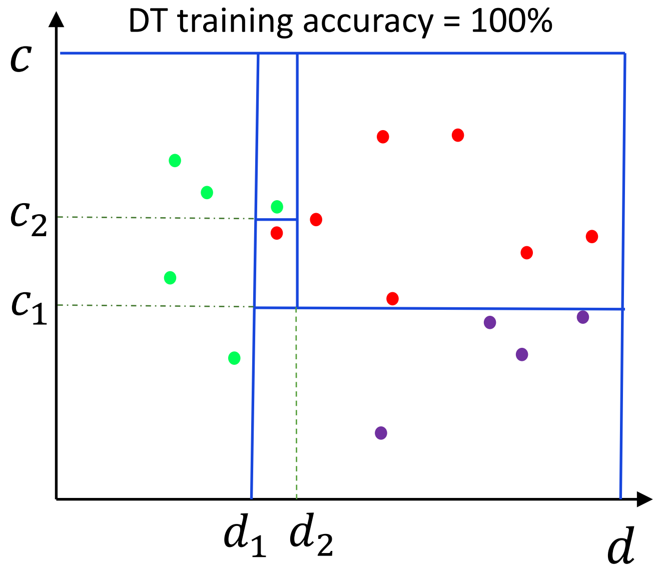
- The parameter valuation points serve as features for off-the-shelf clustering algorithms
- We use the Agglomerative Hierarchical Clustering (AHC) technique
- Number of clusters to choose:
  - Domain knowledge/Silhouette metric



Learning bounding boxes for each cluster:

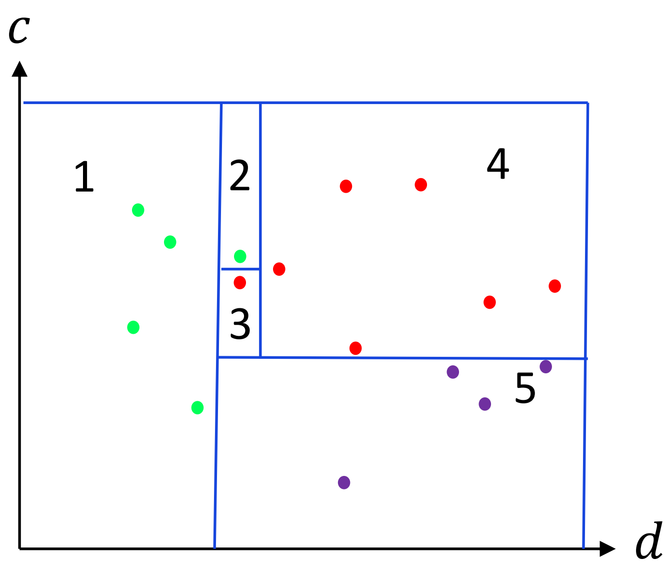
- We label each parameter valuation with its cluster
  - Labels = (green, red, purple)
- We use off-the-shelf Decision Tree (DT) algorithms to learn bounding boxes



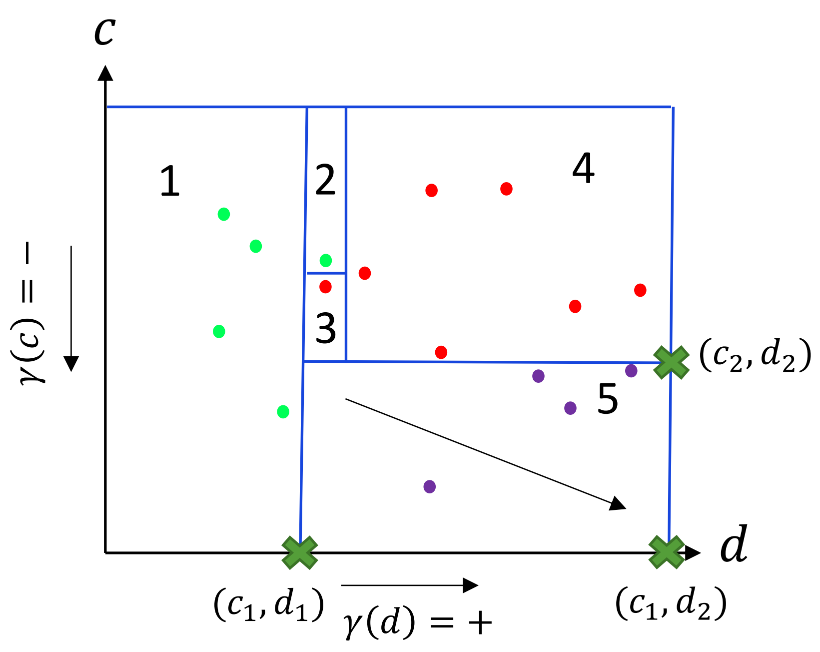


Learning a STREL Formula for each Cluster:

- $\varphi_{green} = \varphi_1 \vee \varphi_2$
- $\varphi_{red} = \varphi_3 \vee \varphi_4$
- $\varphi_{purple} = \varphi_5$



Learning a STREL Formula for each Cluster:



$$\varphi_5 = \varphi(\mathrm{c}_1, \mathrm{d}_2) \wedge \neg\varphi(\mathrm{c}_1, \mathrm{d}_1) \wedge \neg\varphi(c_2, d_2)$$

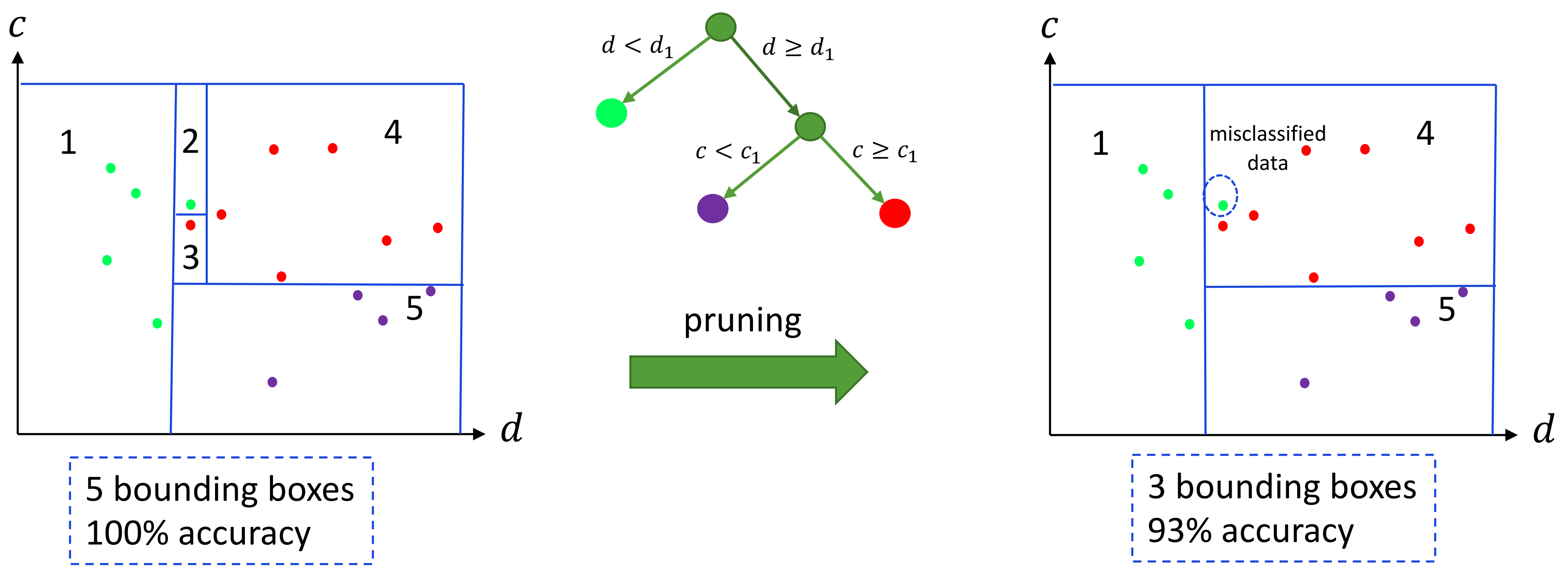
$$\varphi = G_{[0,3hours]} \diamond_{[0,d]} (Bikes > c)$$

$$\varphi_5 = G_{[0,3hours]} \diamond_{[0,d_2]} (Bikes > c_1)$$
$$\wedge \neg G_{[0,3hours]} \diamond_{[0,d_1]} (Bikes > c_1)$$
$$\wedge \neg G_{[0,3hours]} \diamond_{[0,d_2]} (Bikes > c_2)$$

[4]: Logical clustering and learning for time-series data [V-Chanlatte et al., 2017]

Pruning the Decision Tree:

- In some cases, achieving 100% accuracy can result in long and hence less interpretable formulas
- We prune the DT using a K-fold cross validation approach



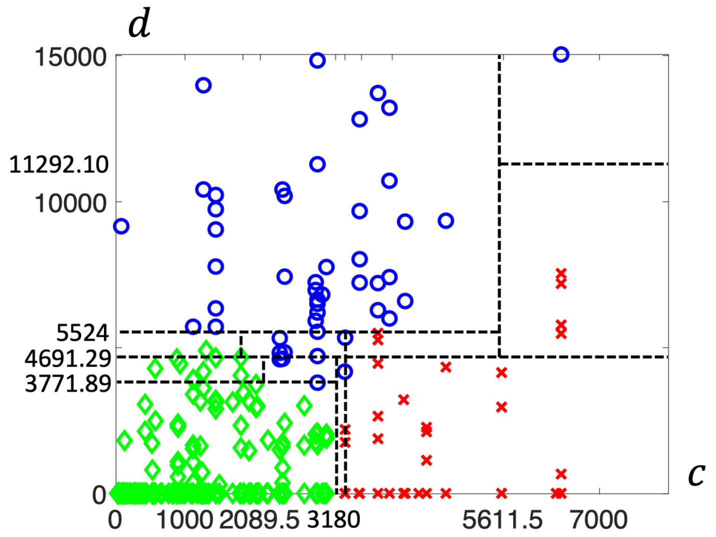
Benchmarks:

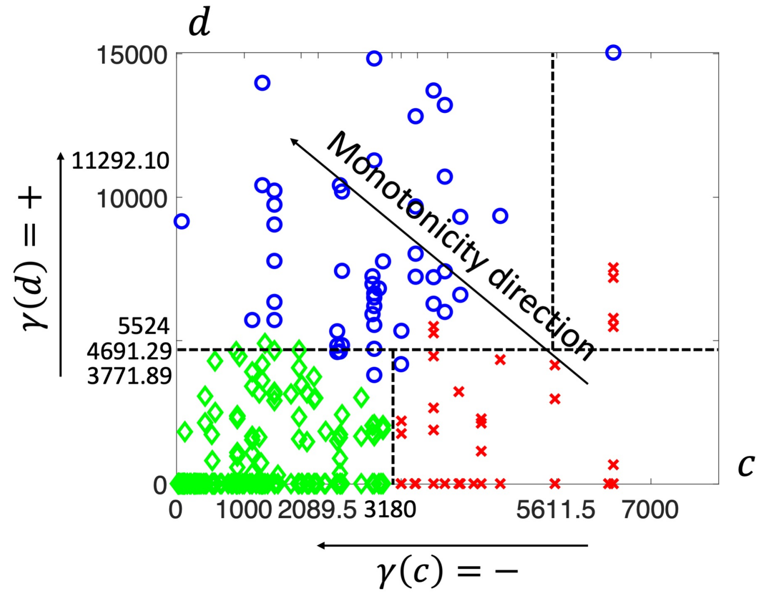
- COVID-19 data from LA County
  - COVID-19 pandemic has extremely affected our lives
  - Understanding the spread pattern of COVID-19 in different areas is vital to stop the spread of the disease.
  - We focus on number of new positive cases in each region of the LA county
- BSS data from the city of Edinburgh
  - The BSS consists of a number of bike stations, distributed over a geographic area
  - We focus on the number of bikes (B) and empty slots (S) in each bike station
  - We are interested in analyzing the behavior of each station
- Outdoor Air Quality data from California
- Synthetic data for a food court building

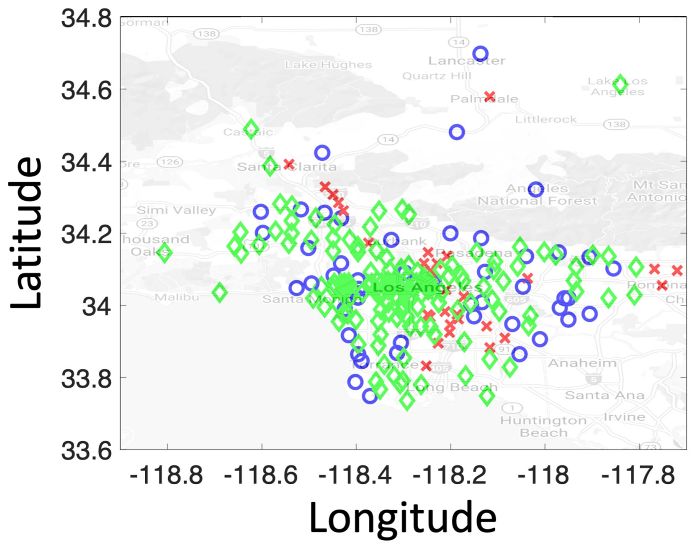
COVID-19 data from LA County

PSTREL formula: $\Diamond_{[0,d]}\{F_{[0,\tau]}(x > c)\}$

- We fix τ to 10 days
- Small d and large c are **hot spots**







$$\varphi_{red} = \Diamond_{[0,4691.29]}\{F_{[0,10]}(x \geq 3181)\} \vee \Diamond_{[0,15000]}\{F_{[0,10]}(x \geq 5612)\}$$

BSS data from the city of Edinburgh

PSTREL formula:

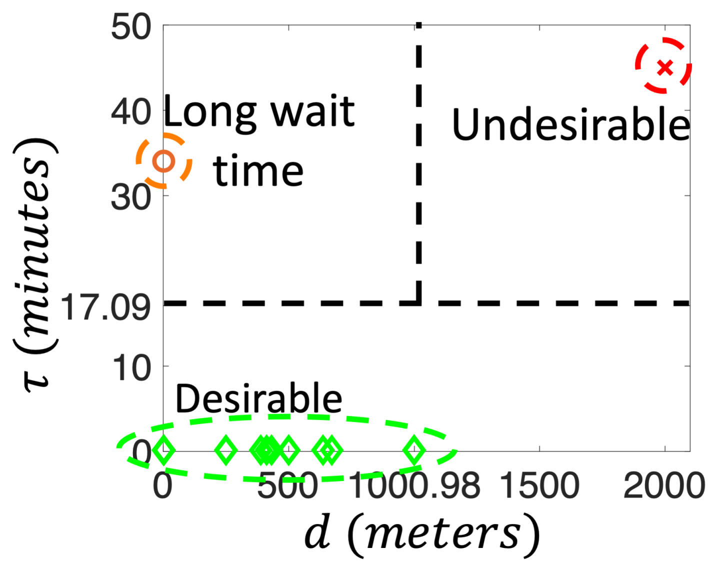
$$\varphi(\tau, d) = G_{[0,\tau]}\big(\varphi_{wait}(\tau) \vee \varphi_{walk}(d)\big)$$

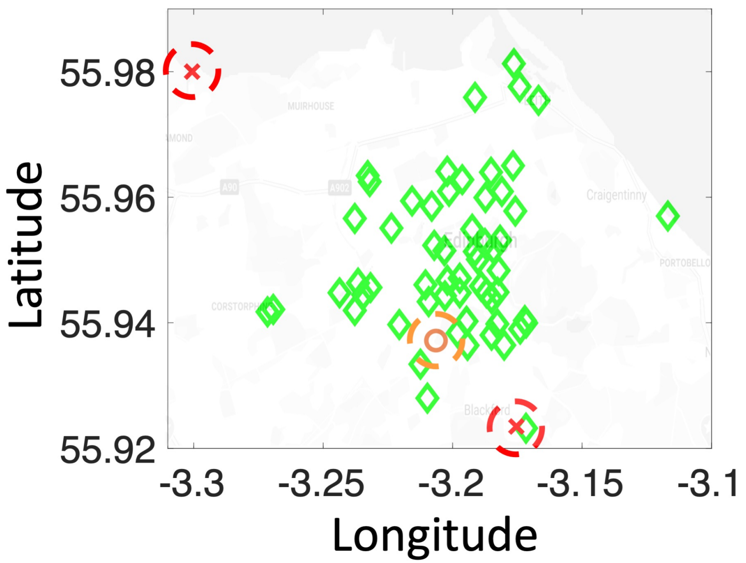
- Within the next 3 hours either $\varphi_{wait}(\tau)$ or $\varphi_{walk}(d)$ is True

$$\varphi_{wait}(\tau) = F_{[0,\tau]}(B \geq 1) \wedge F_{[0,\tau]}(S \geq 1),$$
$$\varphi_{walk}(d) = \diamond_{[0,d]}\,(B \geq 1) \wedge \diamond_{[0,d]}\,(S \geq 1)$$

- Locations with large τ: long wait times
- Locations with large d: far from stations with Bikes/Slots availability

BSS data from the city of Edinburgh





$$\varphi_{red} = \neg G_{[0,3]}\big(\varphi_{wait}(17.09) \vee \varphi_{walk}(2100)\big) \wedge \neg G_{[0,3]}(\varphi_{wait}(50) \vee \varphi_{walk}(1000.98))$$

Results summary:

Case	$\|L\|$	$\|W\|$	$runtime(secs)$	numC	$\|\varphi_{cluster}\|$
COVID-19	235	427	813.65	3	$3.\|\varphi\| + 4$
BSS	61	91	681.78	3	$2.\|\varphi\| + 4$
Air Quality	107	60	136.02	8	$5.\|\varphi\| + 7$
Food Court	20	35	78.24	8	$3.\|\varphi\| + 4$

In a nutshell:

- We proposed a technique to learn interpretable STREL formulas from spatio-temporal data
- We proposed a new method for creating a spatial model with a restrict number of edges that preserves connectivity of the spatial model.
- We leveraged robustness of STREL combined with bisection search to extract features for spatiotemporal time-series clustering.
- We applied AHC on the extracted features followed by a DT based approach to learn an interpretable STREL formula for each cluster
- The results show that our method performs slower than ML approaches, but it is more interpretable

Thank you for listening!