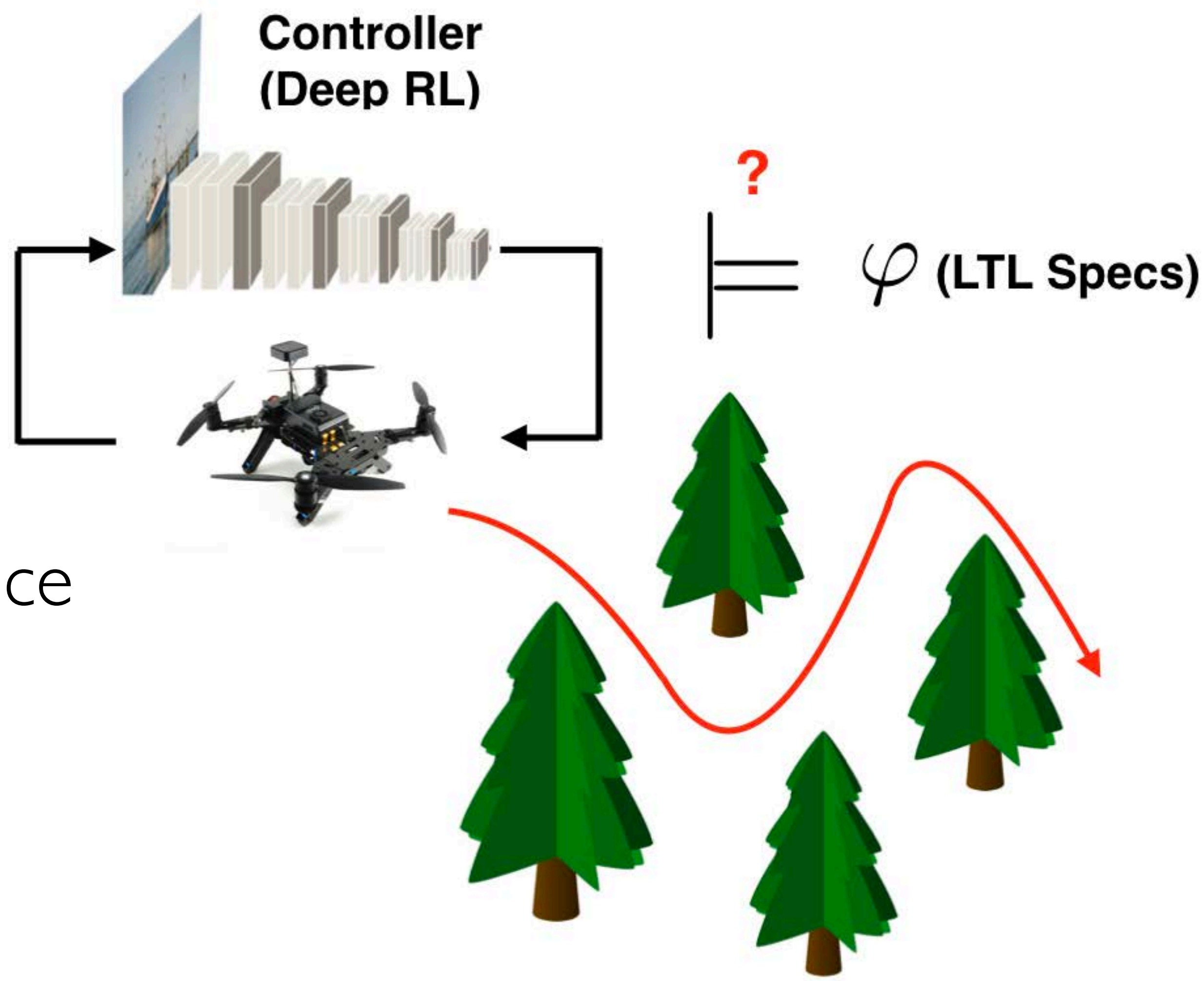
Provably Correct Training of Neural Network Controllers

Yasser Shoukry

Assistant Professor
Resilient Cyber-Physical Systems Lab
Electrical Engineering and Computer Science
University of California, Irvine



UCIrvine

Resilient CPS Lab

CPS Lab
Resilient

Video released of Uber self-driving crash that killed woman in Arizona

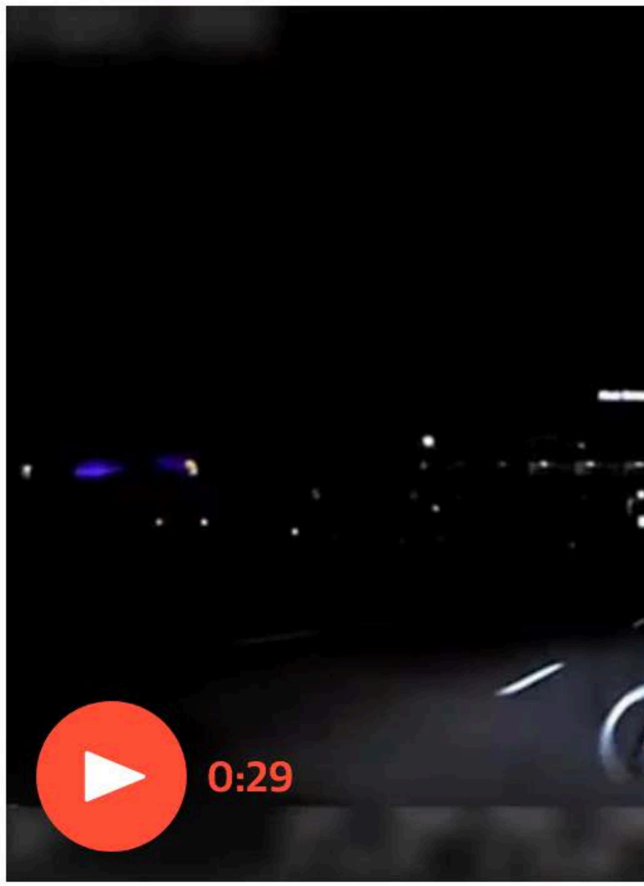
New footage of the crash that killed Elaine Herzberg raises fresh questions about why the self-driving car did not stop

▲ Uber dashcam footage shows lead up to fatal self-driving crash – video

The Guardian, Mar 22 2018

Video released of Uber self-driving crash that killed woman in Arizona

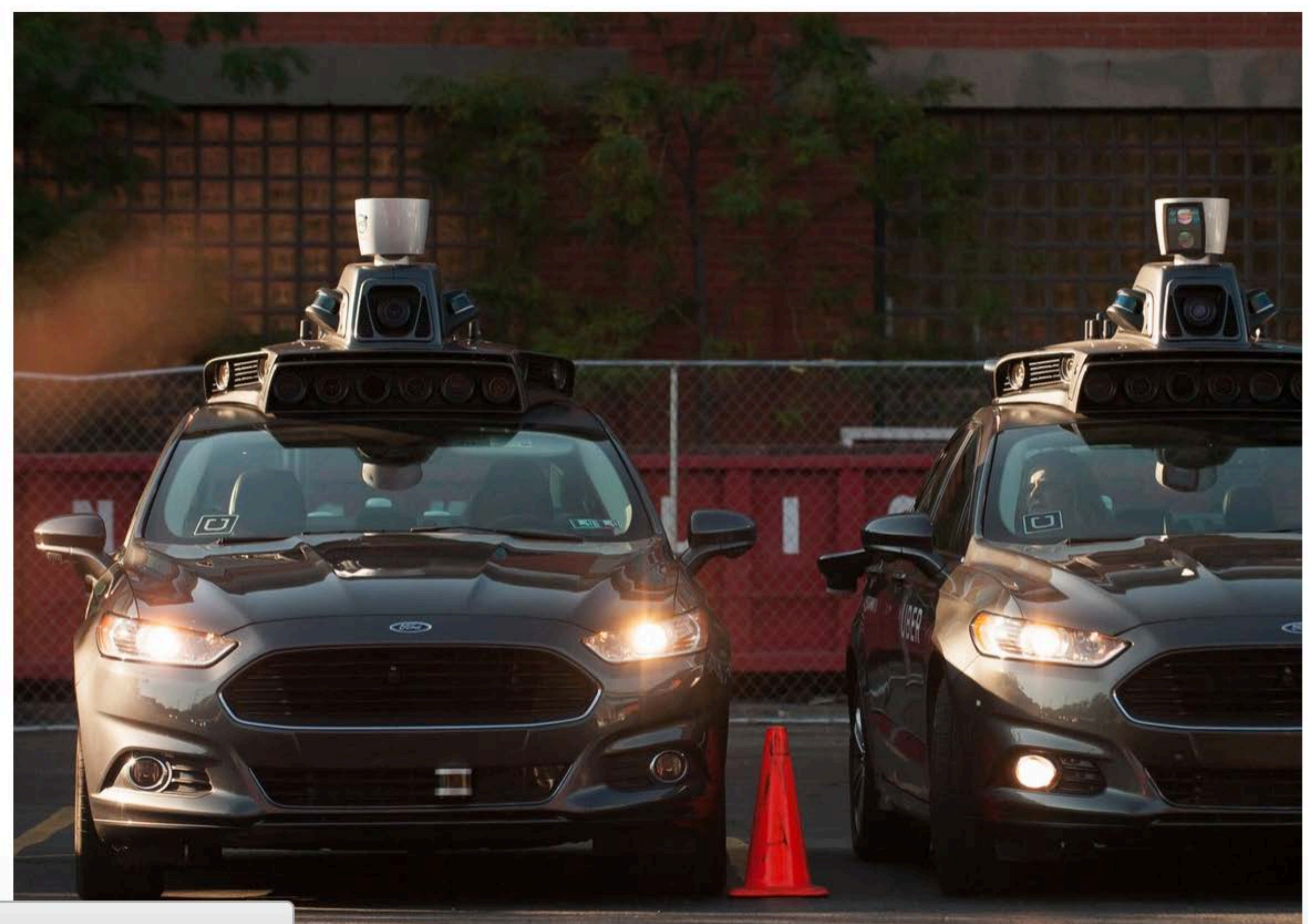
New footage of the crash that killed Elaine Herzberg raises fresh questions about why the self-driving car did not stop



▲ Uber dashcam footage shows lead up to fatal

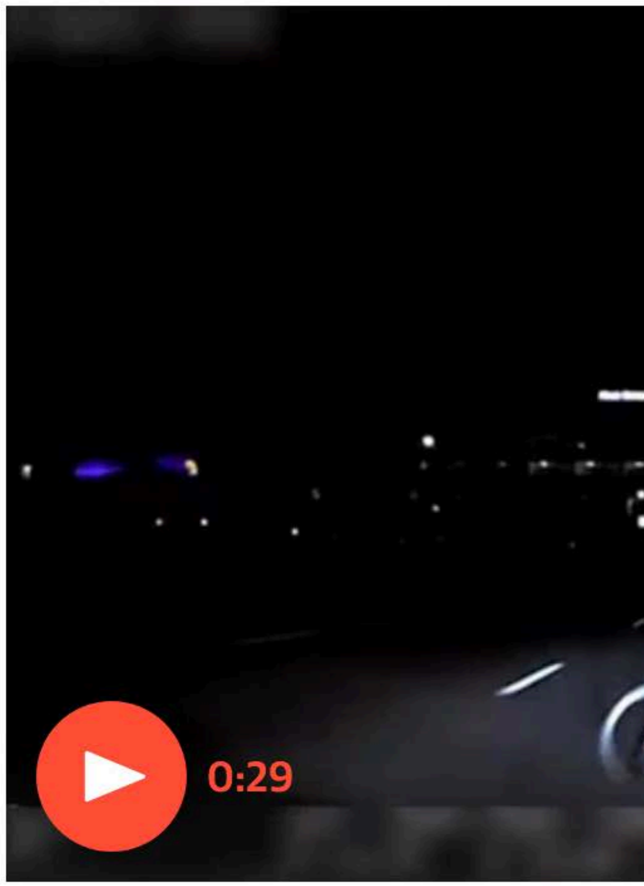
8

HOME FROM THE HONEYMOON, THE SELF-DRIVING CAR INDUSTRY FACES REALITY



Video released of Uber self-driving crash that killed woman in Arizona

New footage of the crash that killed Elaine Herzberg raises fresh questions about why the self-driving car did not stop



Uber dashcam footage shows lead up to fatal

HOME FROM THE HONEYMOON, THE SELF-DRIVING CAR INDUSTRY FACES REALITY

M

TECHNOLOGY | MEET THE TEAM | COMPANY NEWS | WE'RE HIRING

The End of Starsky Robotics

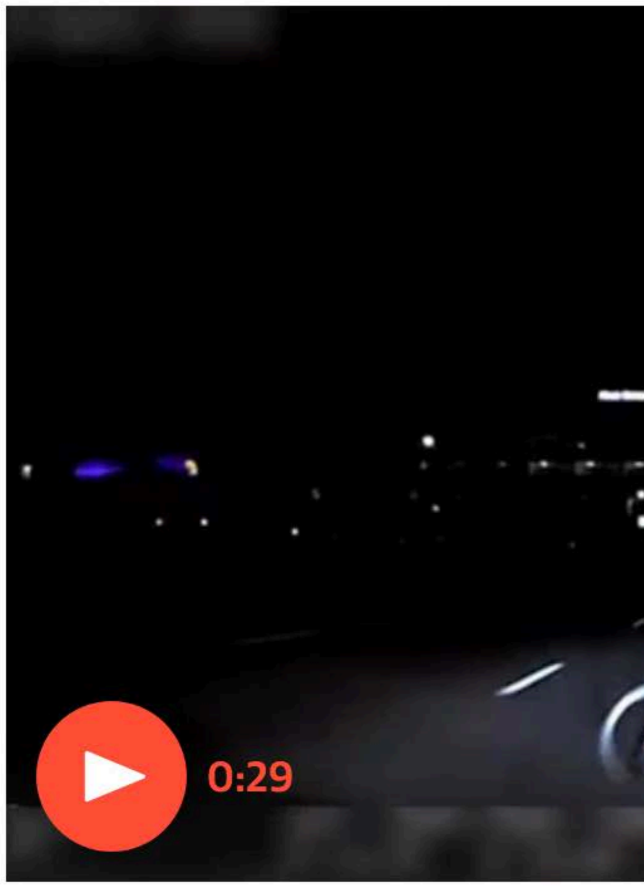
Stefan Seltz-Axmacher Follow
Mar 19 · 9 min read

In 2015, I got obsessed with the idea of driverless trucks and started Starsky Robotics. In 2016, we became the first street-legal vehicle to be paid to do real work without a person behind the wheel. In 2018, we became the first street-legal truck to do a fully unmanned run, albeit on a closed road. In 2019, our truck became the first fully-unmanned truck to drive on a live highway.

And in 2020, we're shutting down.

Video released of Uber self-driving crash that killed woman in Arizona

New footage of the crash that killed Elaine Herzberg raises fresh questions about why the self-driving car did not stop



▲ Uber dashcam footage shows lead up to fatal

HOME FROM THE HONEYMOON, THE SELF-DRIVING CAR INDUSTRY FACES REALITY

M

TECHNOLOGY | MEET THE TEAM | COMPANY NEWS | WE'RE HIRING

The End of Starsky Robotics

Stefan Seltz-Axmacher Follow
Mar 19 · 9 min read

Starsky Robotics
10–4 Labs
Starsky Robotics is a driverless truck startup which aims...

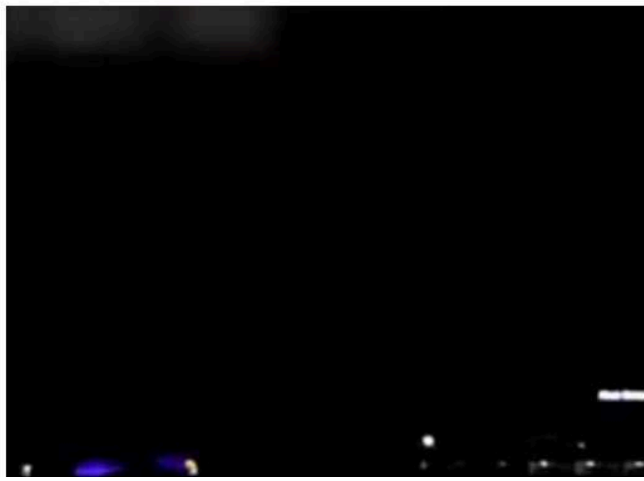
Follow

It took me way too long to realize that VCs would rather a $1b business with a 90% margin than a $5b business with a 50% margin, even if capital requirements and growth were the same.

And growth would be the same. The biggest limiter of autonomous deployments isn't sales, it's safety.

Video released of Uber self-driving crash that killed woman in Arizona

New footage of the crash that killed Elaine Herzberg raises fresh questions about why the self-driving car did not stop



HOME FROM THE HONEYMOON, THE SELF-DRIVING CAR INDUSTRY FACES REALITY

Starsky Robotics
10–4 Labs
Starsky Robotics is a driverless truck startup which aims...

Follow

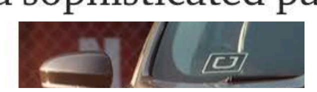
The AV Space

There are too many problems with the AV industry to detail here: the professorial pace at which most teams work, the lack of tangible deployment milestones, the open secret that there isn't a robotaxi business model, etc. The biggest, however, is that supervised machine learning doesn't live up to the hype. It isn't actual artificial intelligence akin to C-3PO, it's a sophisticated pattern-matching tool.

'RE HIRING

ırsky Robotics

ow



Starsky Robotics
10–4 Labs
Starsky Robotics is a driverless truck startup which aims...

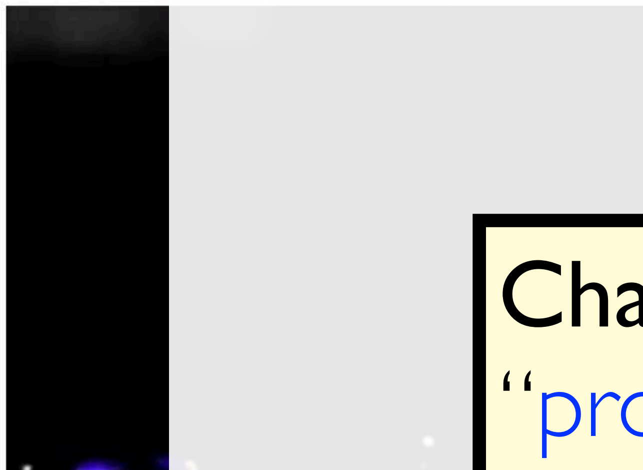
Follow

It took me way too long to realize that VCs would rather a $1b business with a 90% margin than a $5b business with a 50% margin, even if capital requirements and growth were the same.

And growth would be the same. The biggest limiter of autonomous deployments isn't sales, it's safety.

Video released of Uber self-driving crash that killed woman in Arizona

New footage of the crash that killed Elaine Herzberg raises fresh questions about why the self-driving car did not stop



HOME FROM THE HONEYMOON,

Challenge: Can we systematically design "provably correct" deep neural networks?

- Theory
- Algorithms
- Implementation

Starsky Rob
10–4 Labs
Starsky Robot
driverless tru
startup which
aims...

Follow

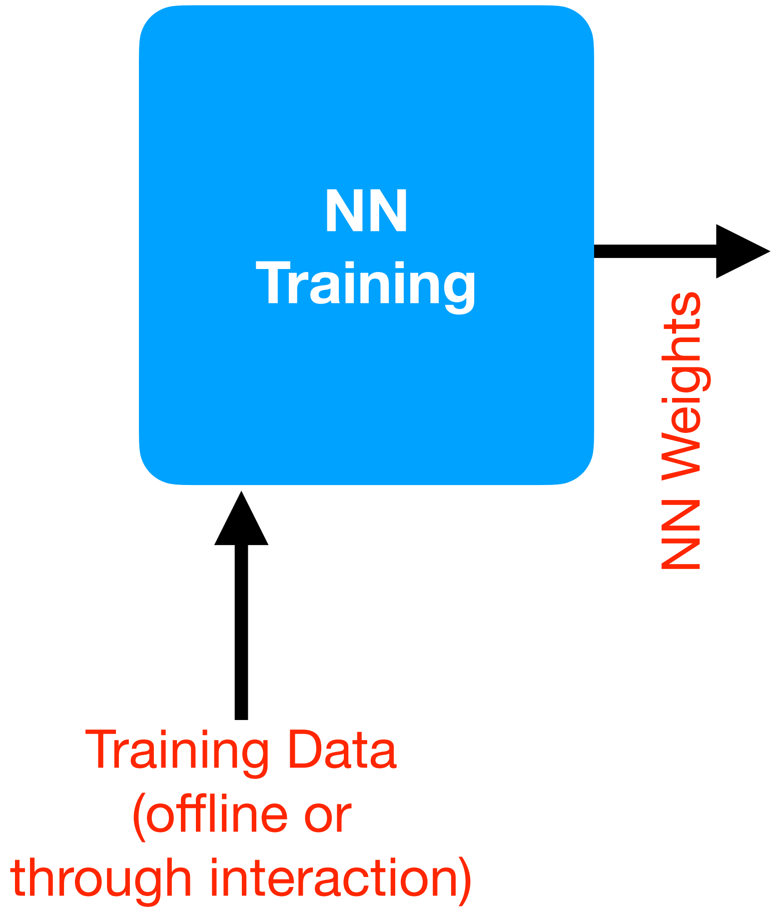
3PO, it's a sophisticated pattern-matching tool.

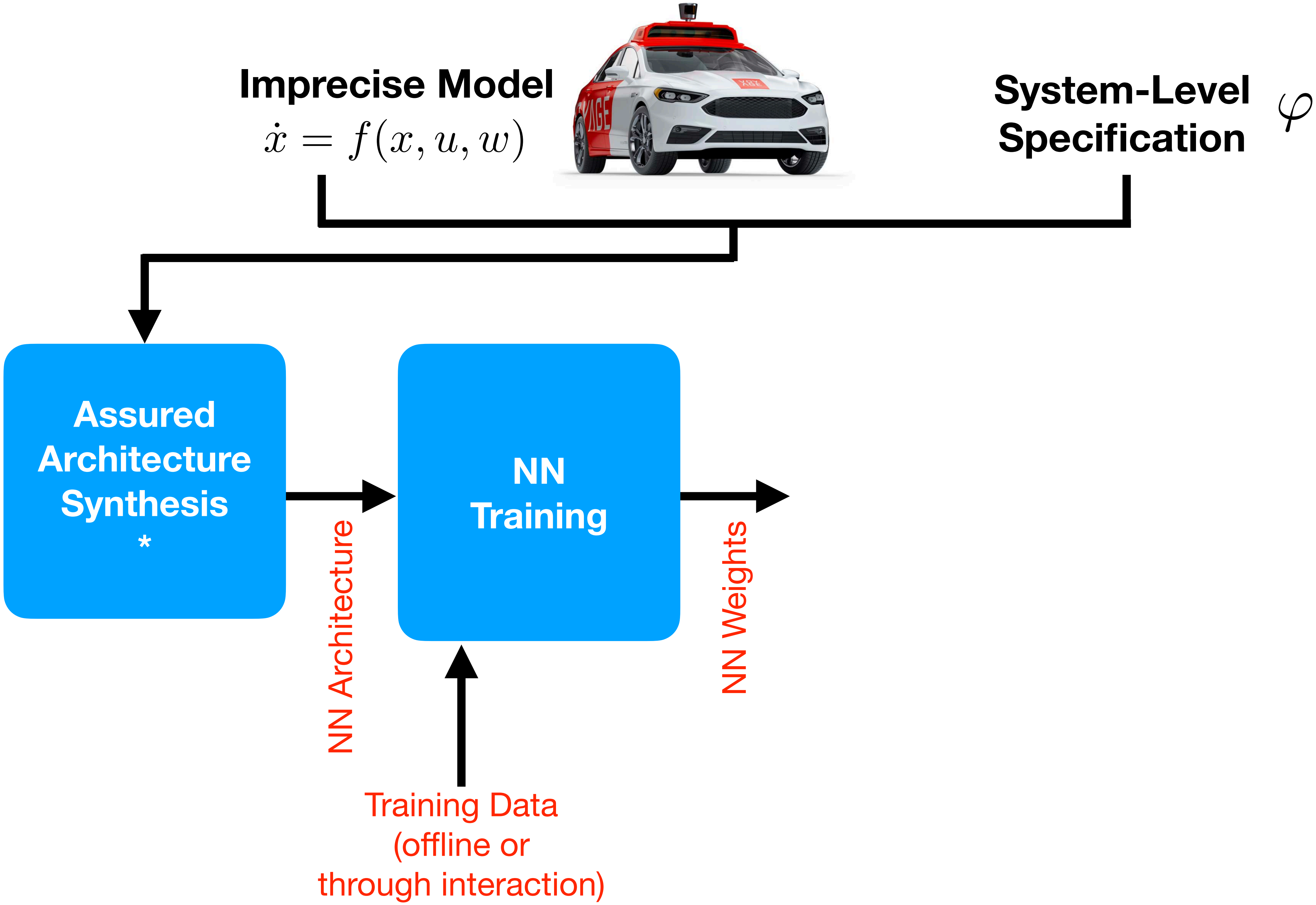
Starsky Robotics
10–4 Labs
Starsky Robotics is a driverless truck startup which aims...

Follow

It took me way too long to realize that VCs would rather a \$1b business with a 90% margin than a \$5b business with a 50% margin, even if capital requirements and growth were the same.

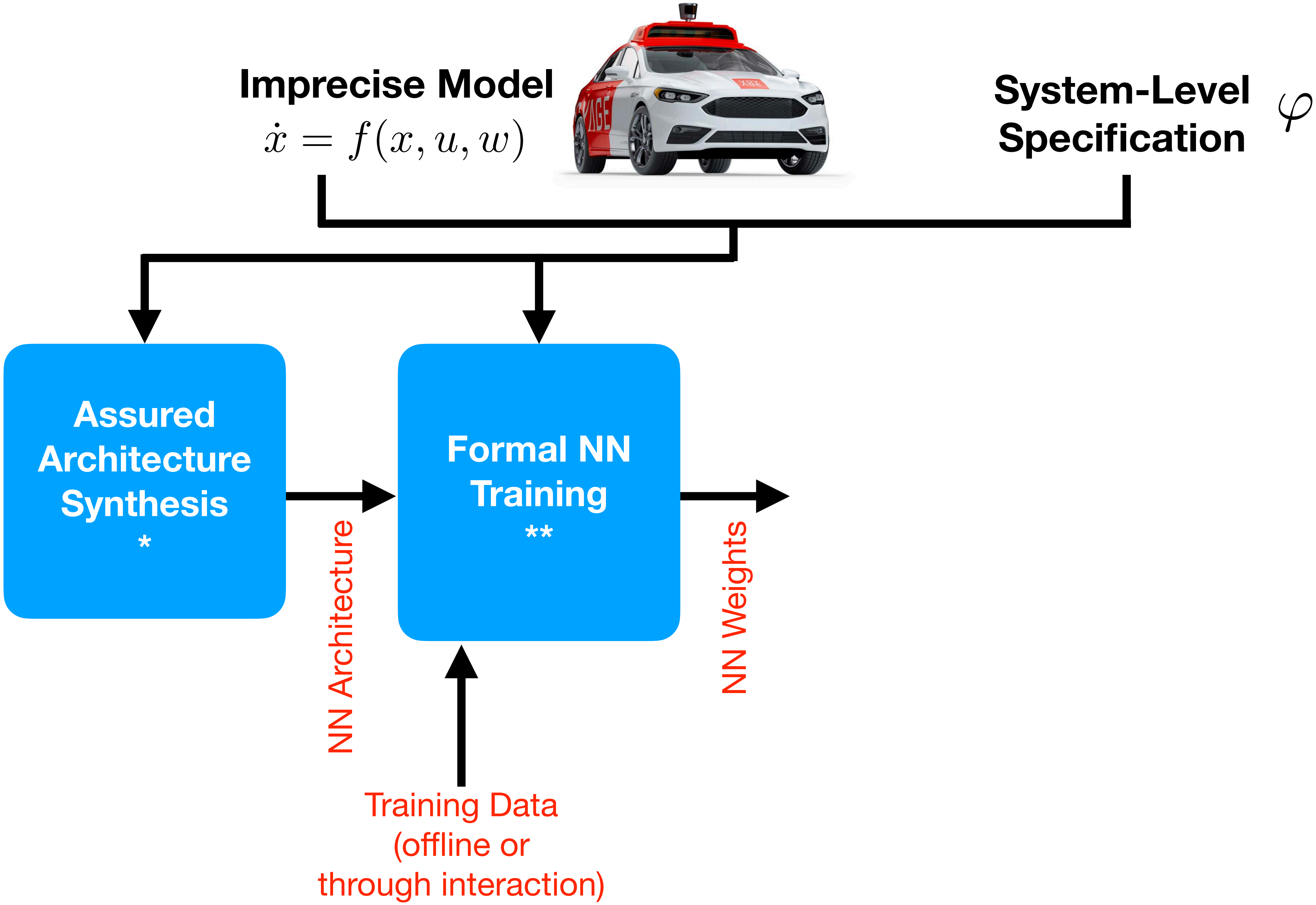
And growth would be the same. The biggest limiter of autonomous deployments isn't sales, it's safety.


NN Training
NN Weights
Training Data (offline or through interaction)



* J. Ferlez, X. Sun, and Y. Shoukry, "Two-Level Lattice Neural Network Architectures for Control of Nonlinear Systems," CDC 2020.

* J. Ferlez and Y. Shoukry, "AReN: Assured ReLU NN Architecture for Model Predictive Control of LTI Systems," HSCC, 2020.

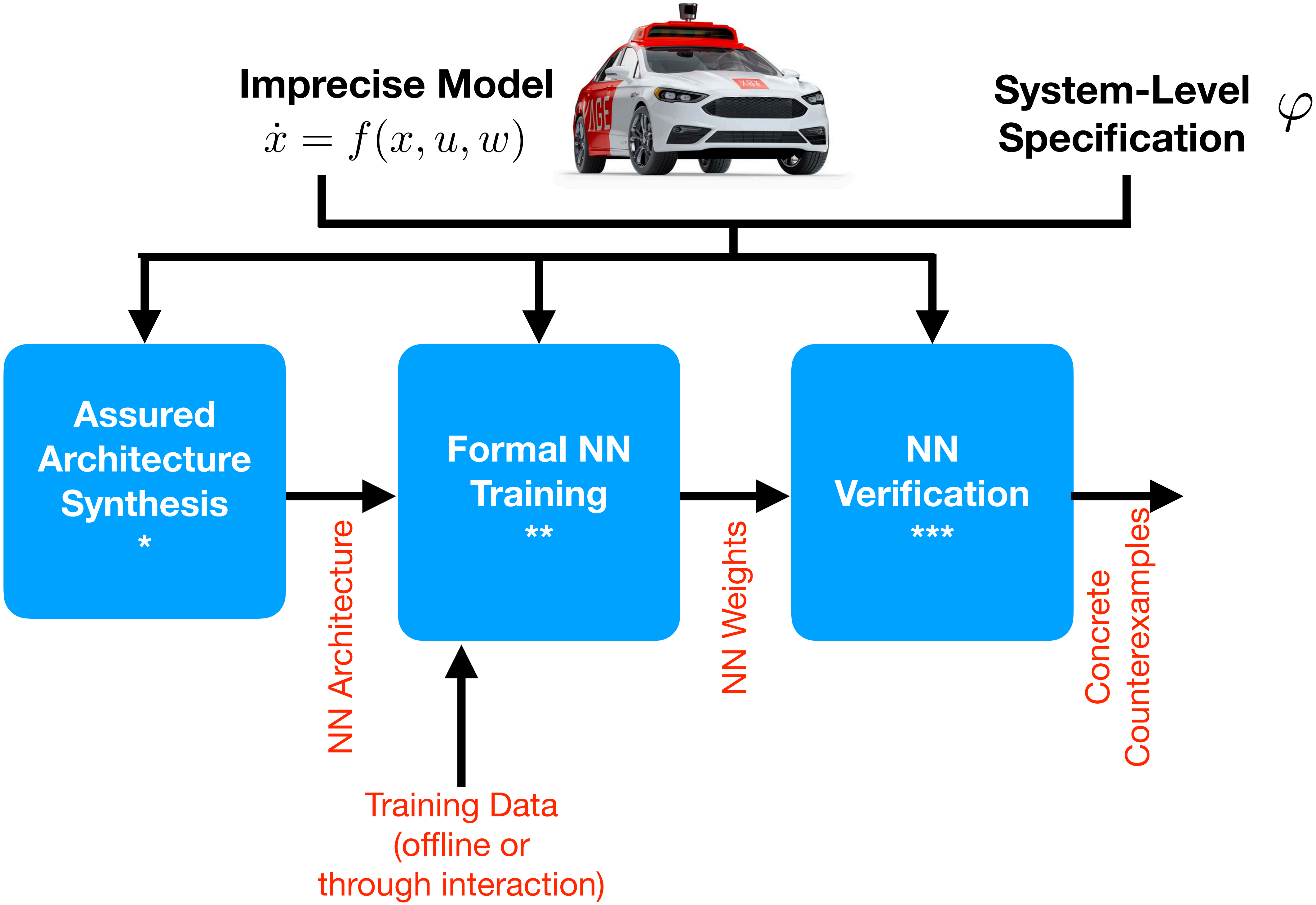


* J. Ferlez, X. Sun, and Y. Shoukry, "Two-Level Lattice Neural Network Architectures for Control of Nonlinear Systems," CDC 2020.

* J. Ferlez and Y. Shoukry, "AReN: Assured ReLU NN Architecture for Model Predictive Control of LTI Systems," HSCC, 2020.

** X. Sun and Y. Shoukry, "Provably Correct Training of Neural Network Controllers Using Reachability Analysis," arXiv 2021.

** X. Sun, W. Fatnassi, U. Santa Cruz, and Y. Shoukry, "Provably Safe Model-Based Meta Reinforcement Learning: An Abstraction-Based Approach," arXiv 2021.



* J. Ferlez, X. Sun, and Y. Shoukry, "Two-Level Lattice Neural Network Architectures for Control of Nonlinear Systems," CDC 2020.

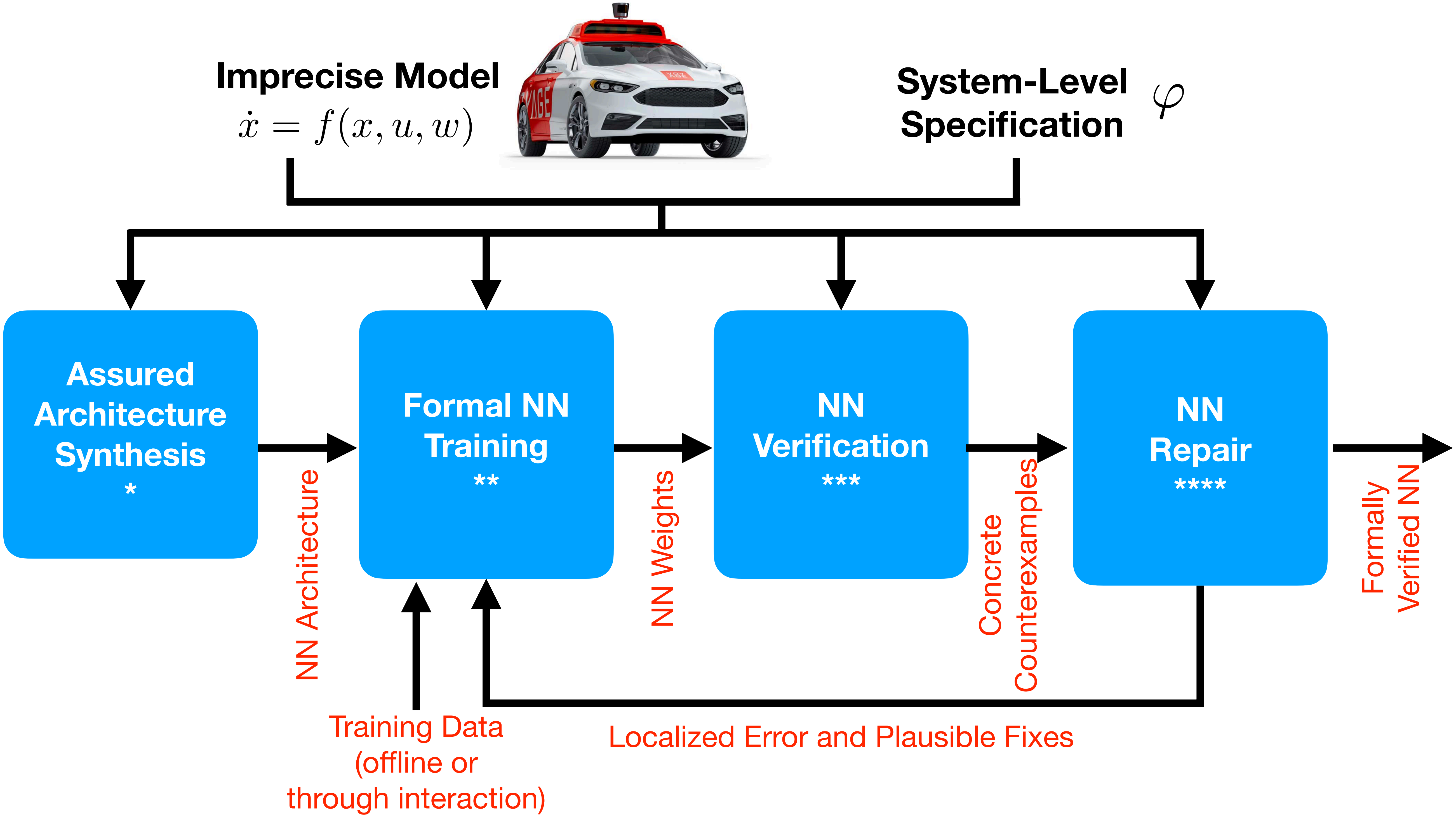
* J. Ferlez and Y. Shoukry, "AReN: Assured ReLU NN Architecture for Model Predictive Control of LTI Systems," HSCC, 2020.

** X. Sun and Y. Shoukry, "Provably Correct Training of Neural Network Controllers Using Reachability Analysis," arXiv 2021.

** X. Sun, W. Fatnassi, U. Santa Cruz, and Y. Shoukry, "Provably Safe Model-Based Meta Reinforcement Learning: An Abstraction-Based Approach," arXiv 2021.

*** H. Khedr, J. Ferlez, and Y. Shoukry, "PEREGRiNN: Penalized-Relaxation Greedy Neural Network Verifier,"CAV, 2021.

*** J. Ferlez and Y. Shoukry, "Bounding the Complexity of Formally Verifying Neural Networks: A Geometric Approach," arXiv 2020.



* J. Ferlez, X. Sun, and Y. Shoukry, "Two-Level Lattice Neural Network Architectures for Control of Nonlinear Systems," CDC 2020.

* J. Ferlez and Y. Shoukry, "AReN: Assured ReLU NN Architecture for Model Predictive Control of LTI Systems," HSCC, 2020.

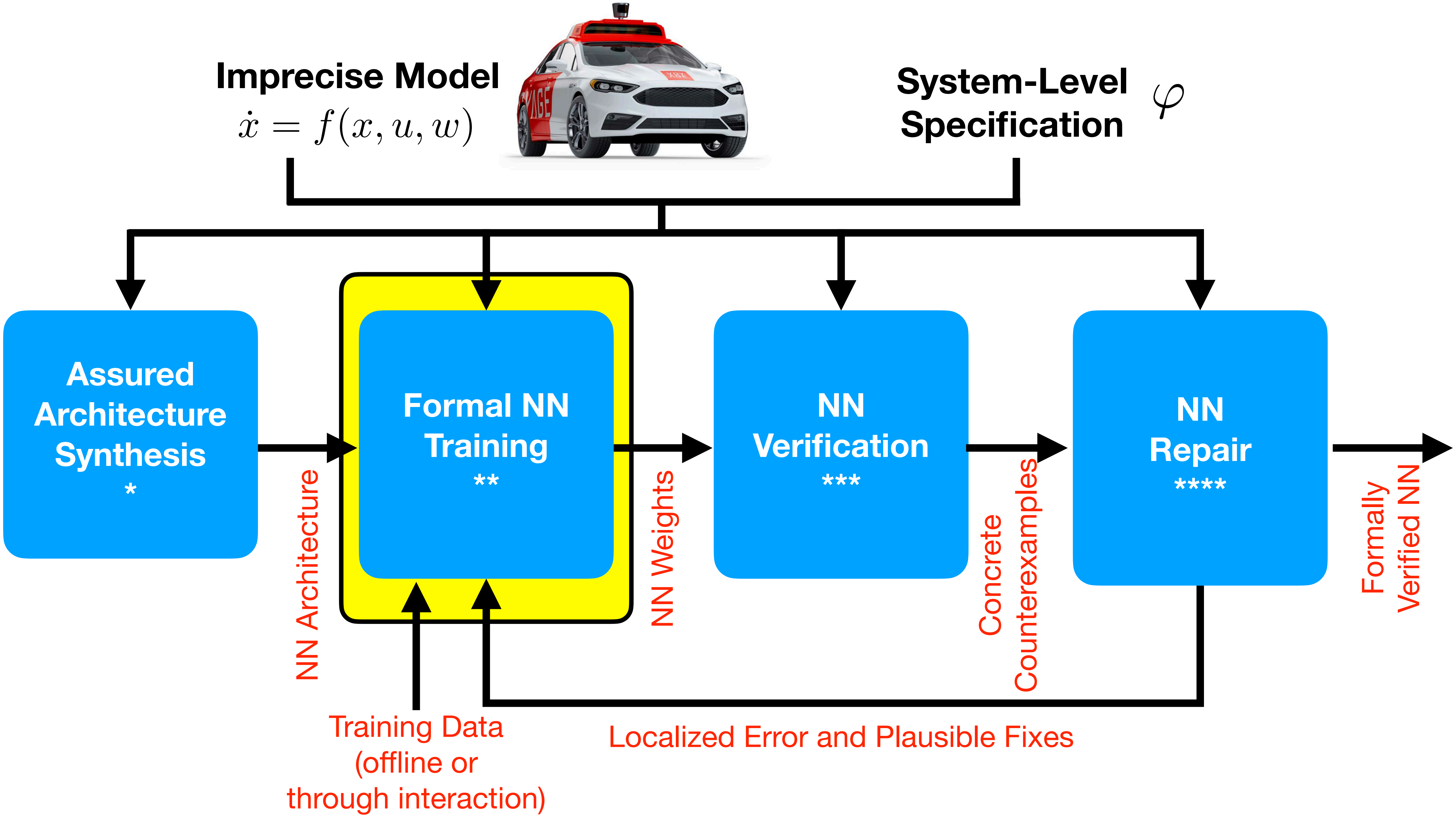
** X. Sun and Y. Shoukry, "Provably Correct Training of Neural Network Controllers Using Reachability Analysis," arXiv 2021.

** X. Sun, W. Fatnassi, U. Santa Cruz, and Y. Shoukry, "Provably Safe Model-Based Meta Reinforcement Learning: An Abstraction-Based Approach," arXiv 2021.

*** H. Khedr, J. Ferlez, and Y. Shoukry, "PEREGRiNN: Penalized-Relaxation Greedy Neural Network Verifier,"CAV, 2021.

*** J. Ferlez and Y. Shoukry, "Bounding the Complexity of Formally Verifying Neural Networks: A Geometric Approach," arXiv 2020.

**** U. Santa Cruz, J. Ferlez, and Y. Shoukry, "Safe-by-Repair: A Convex Optimization Approach for Repairing Unsafe Two-Level Lattice Neural Network Controllers," arXiv 2021.



* J. Ferlez, X. Sun, and Y. Shoukry, "Two-Level Lattice Neural Network Architectures for Control of Nonlinear Systems," CDC 2020.

* J. Ferlez and Y. Shoukry, "AReN: Assured ReLU NN Architecture for Model Predictive Control of LTI Systems," HSCC, 2020.

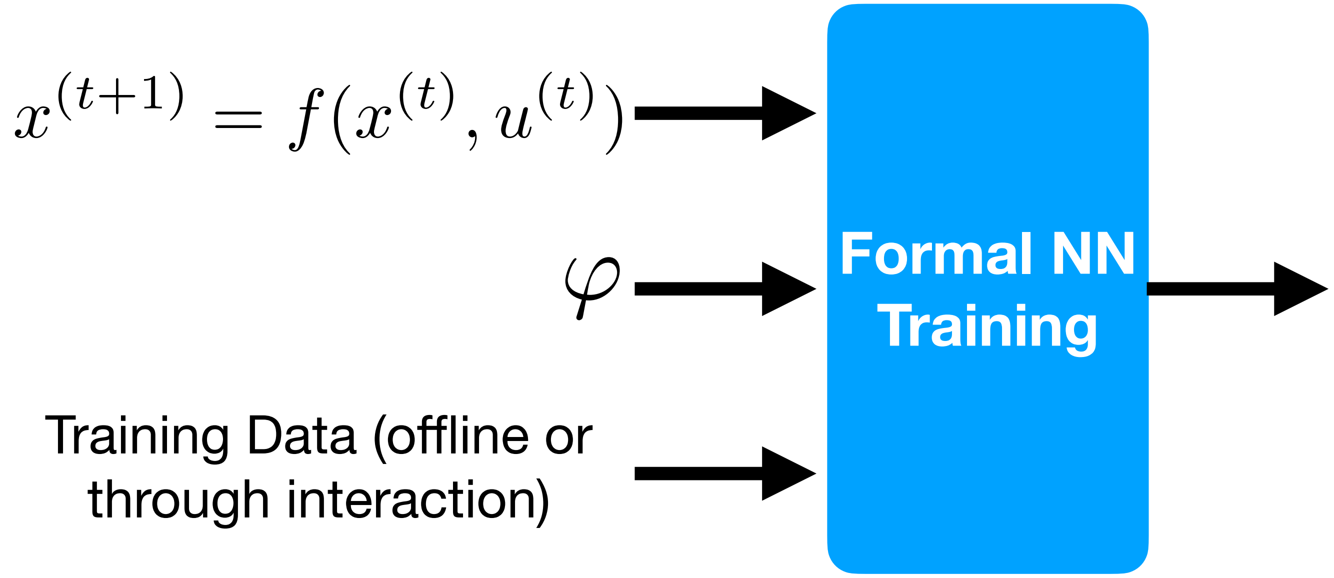
** X. Sun and Y. Shoukry, "Provably Correct Training of Neural Network Controllers Using Reachability Analysis," arXiv 2021.

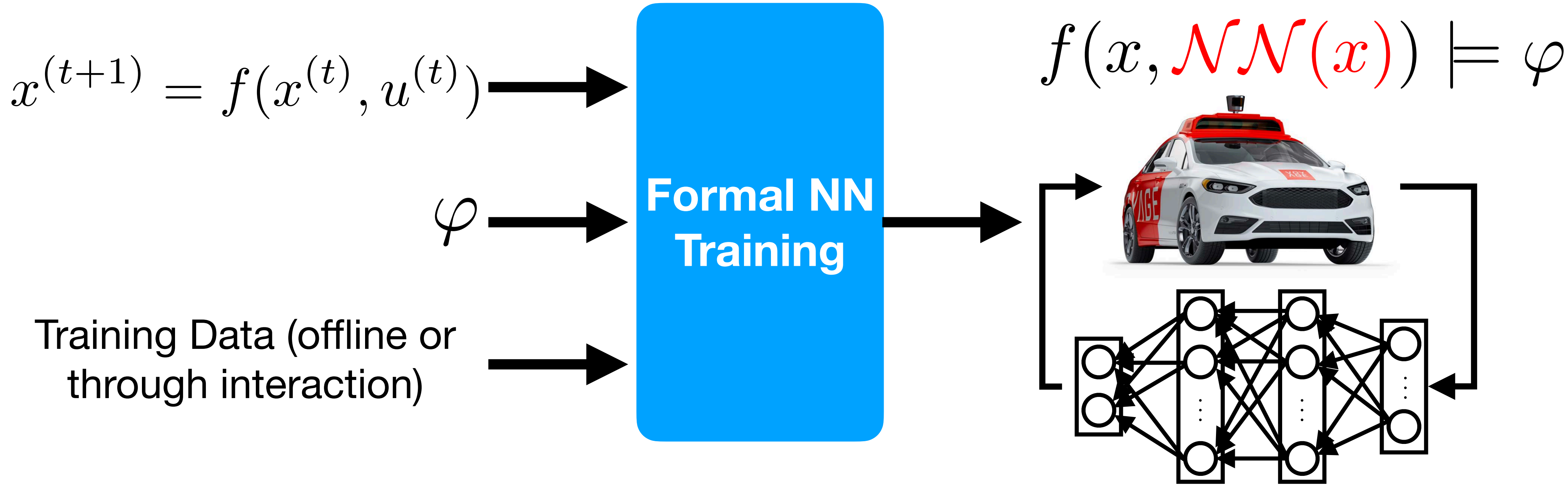
** X. Sun, W. Fatnassi, U. Santa Cruz, and Y. Shoukry, "Provably Safe Model-Based Meta Reinforcement Learning: An Abstraction-Based Approach," arXiv 2021.

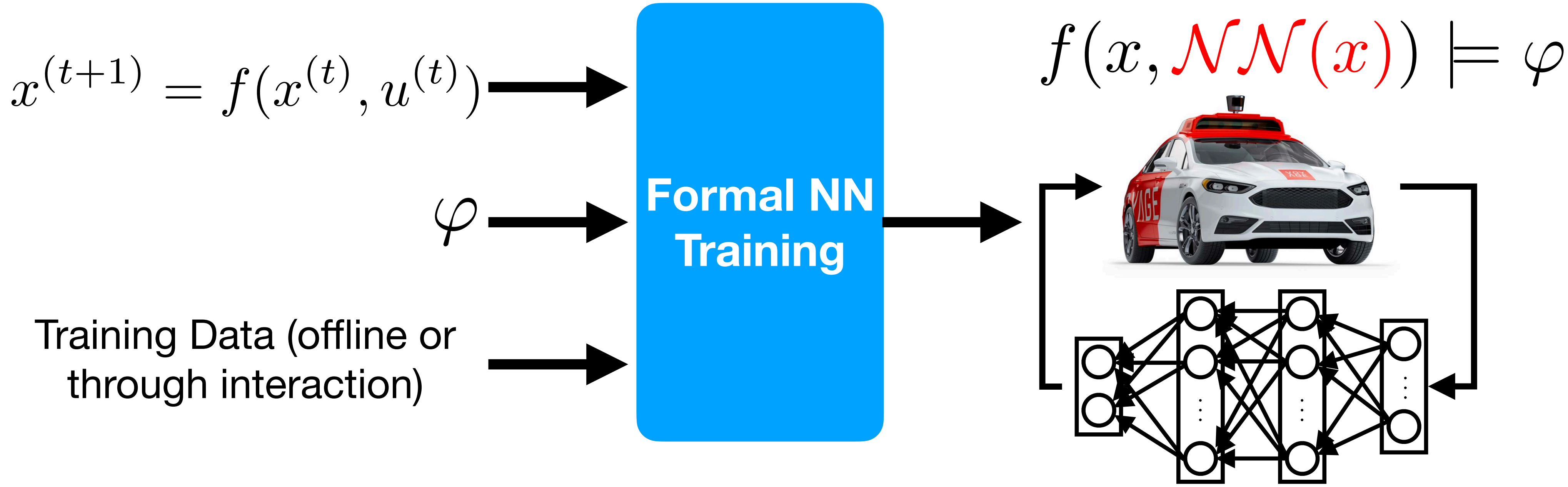
*** H. Khedr, J. Ferlez, and Y. Shoukry, "PEREGRiNN: Penalized-Relaxation Greedy Neural Network Verifier,"CAV, 2021.

*** J. Ferlez and Y. Shoukry, "Bounding the Complexity of Formally Verifying Neural Networks: A Geometric Approach," arXiv 2020.

**** U. Santa Cruz, J. Ferlez, and Y. Shoukry, "Safe-by-Repair: A Convex Optimization Approach for Repairing Unsafe Two-Level Lattice Neural Network Controllers," arXiv 2021.

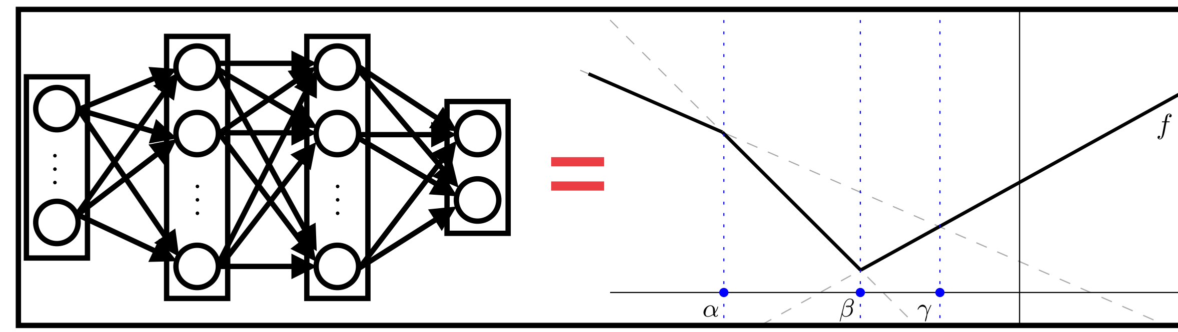

$x^{(t+1)} = f(x^{(t)}, u^{(t)})$
φ
Training Data (offline or through interaction)
Formal NN Training

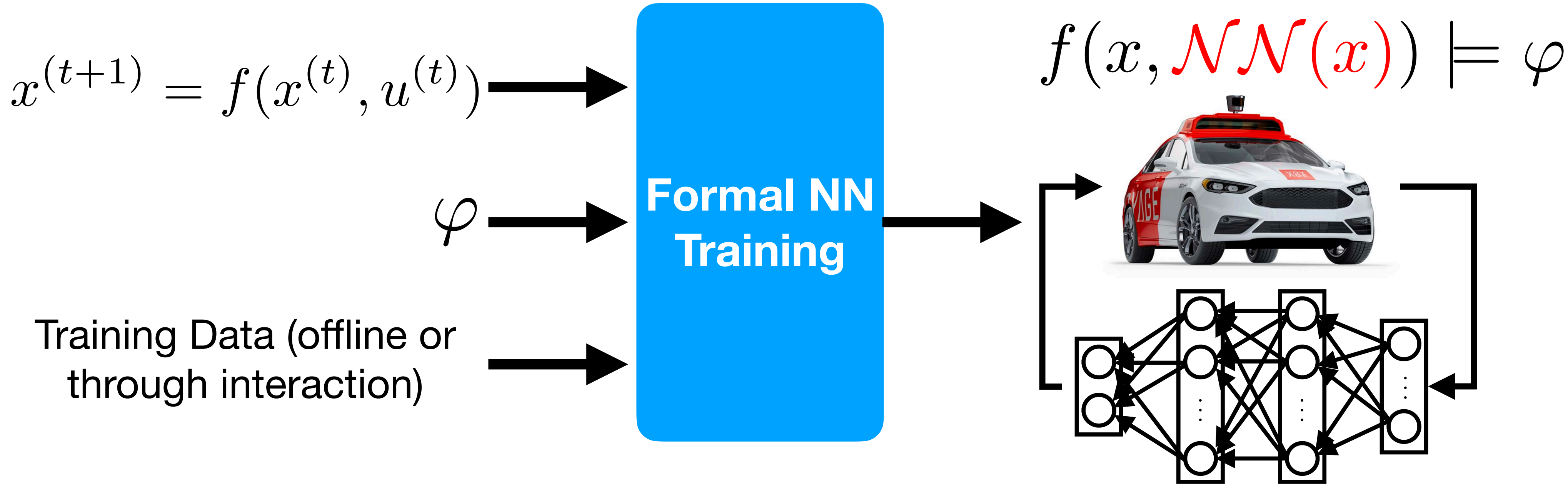

$x^{(t+1)} = f(x^{(t)}, u^{(t)})$
φ
Training Data (offline or through interaction)
Formal NN Training
$f(x, \mathcal{NN}(x)) \models \varphi$



Core idea:

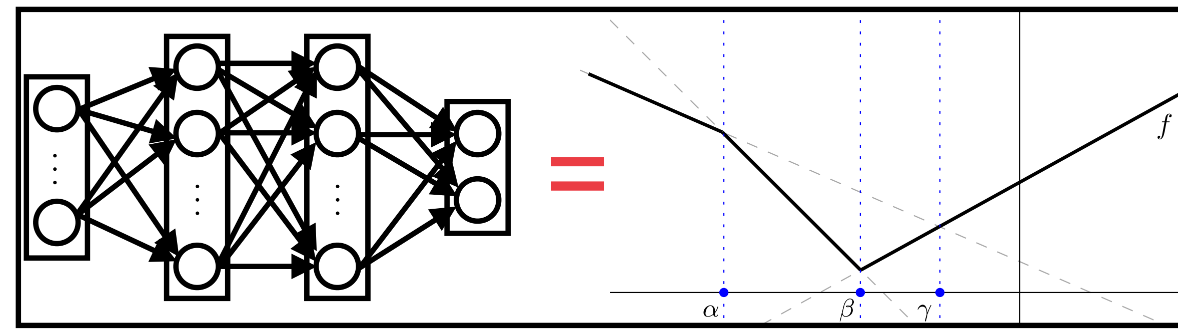
- Regression ReLU NN are Continuous Piece-Wise Affine (CPWA) functions





Core idea:

- Regression ReLU NN are Continuous Piece-Wise Affine (CPWA) functions



- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$f(x, K_{\mathrm{CPWA}}(x)) \models \varphi \quad \forall\, K_{\mathrm{CPWA}} \in \mathrm{CPWA}_{\varphi}$$

Resilient CPS Lab

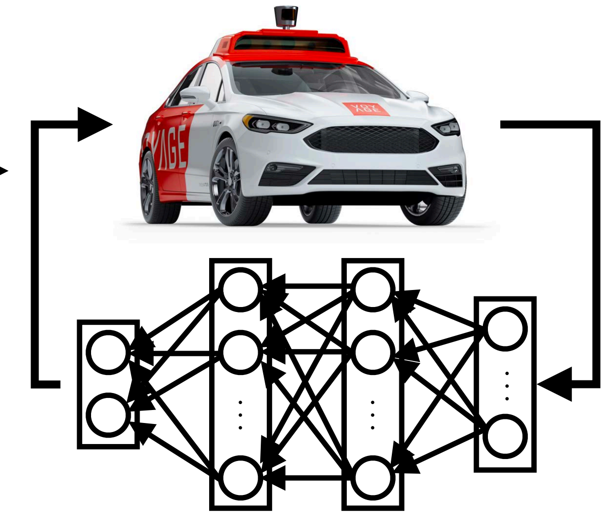
$x^{(t+1)} = f(x^{(t)}, u^{(t)})$ →

φ →

Training Data (offline or through interaction) →

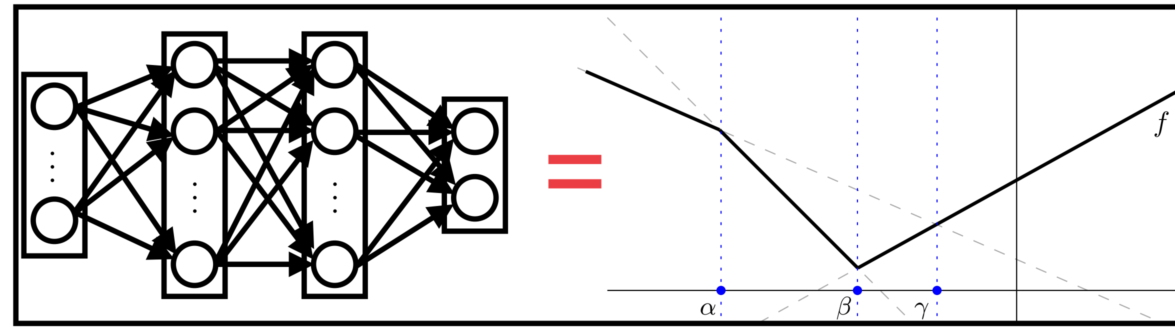
Formal NN Training →

$f(x, \color{red}{NN(x)}) \models \varphi$



Core idea:

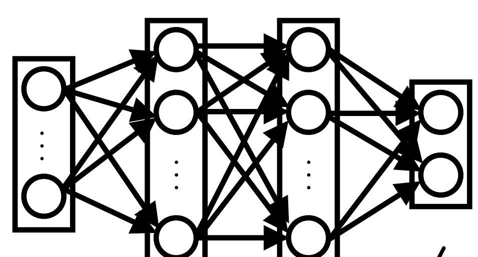
- Regression ReLU NN are Continuous Piece-Wise Affine (CPWA) functions



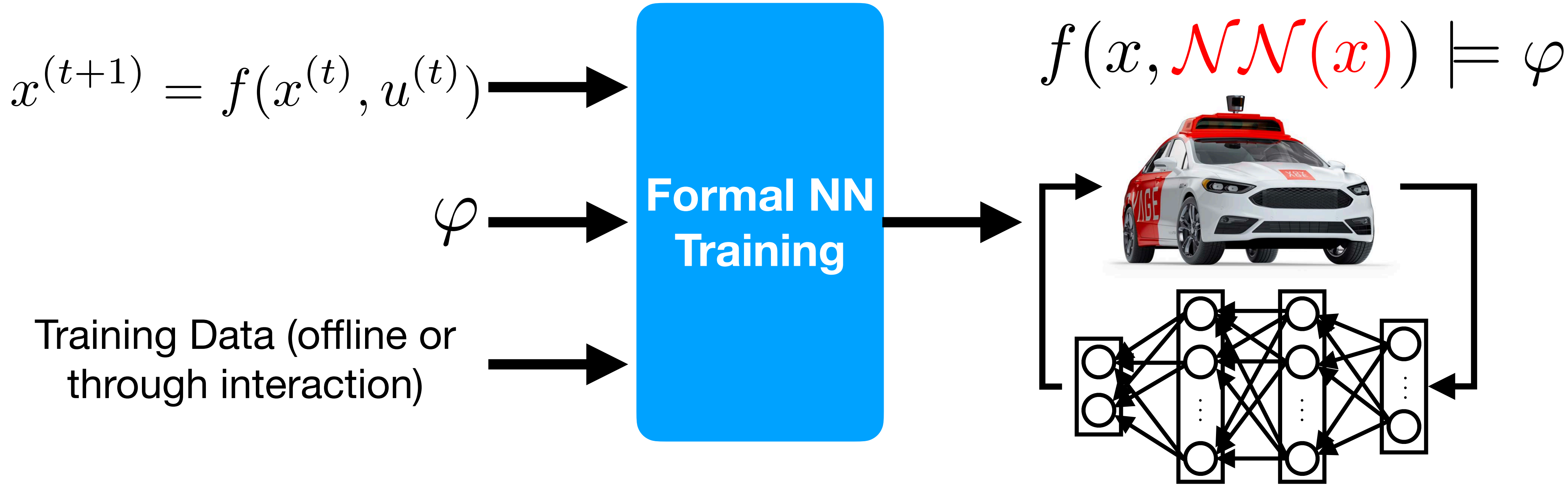
- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$f(x, K_{\text{CPWA}}(x)) \models \varphi \quad \forall\, K_{\text{CPWA}} \in \text{CPWA}_\varphi$$

Training Data (offline or through interaction) $\xrightarrow{\text{Training}}$

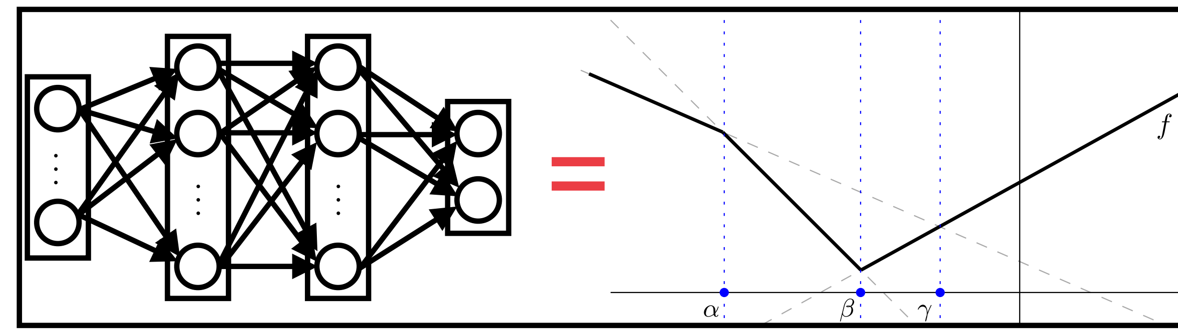


$\notin \text{CPWA}_\varphi$



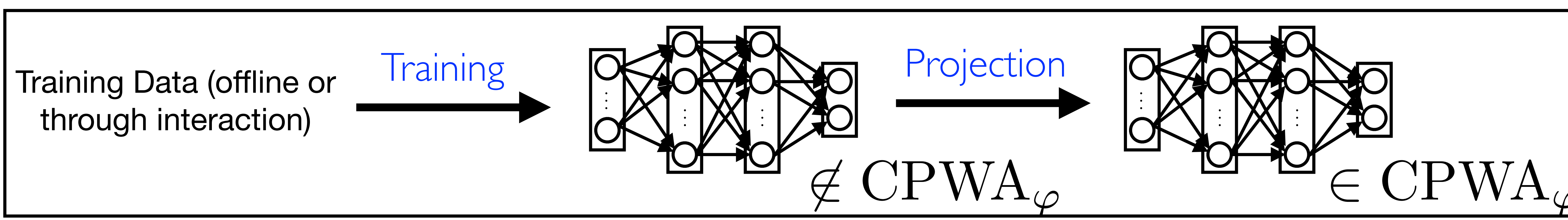
Core idea:

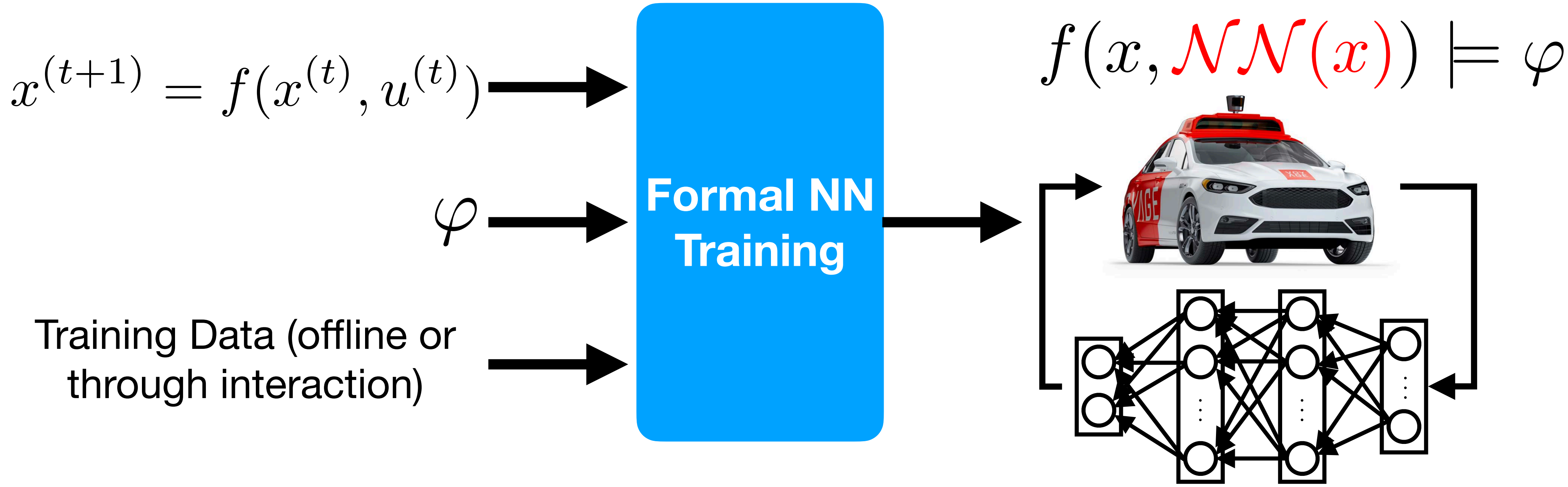
- Regression ReLU NN are Continuous Piece-Wise Affine (CPWA) functions



- Use reachability analysis to identify families of CPWA functions that satisfy the specs

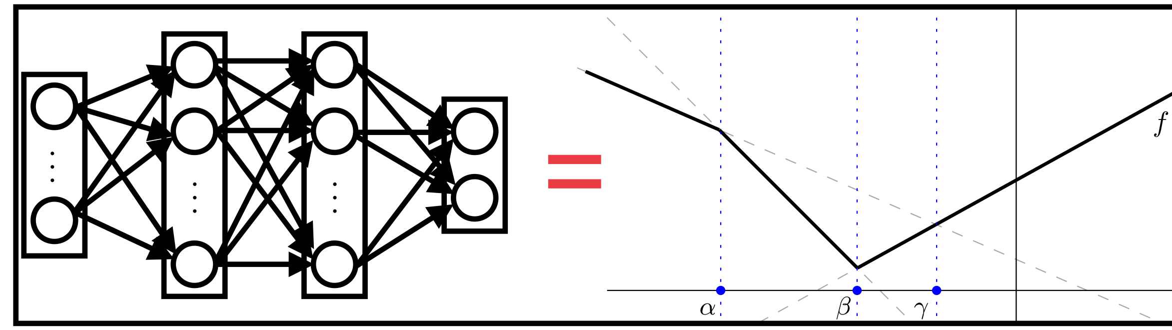
$$f(x, K_{\text{CPWA}}(x)) \models \varphi \quad \forall\, K_{\text{CPWA}} \in \text{CPWA}_{\varphi}$$





Core idea:

- Regression ReLU NN are Continuous Piece-Wise Affine (CPWA) functions

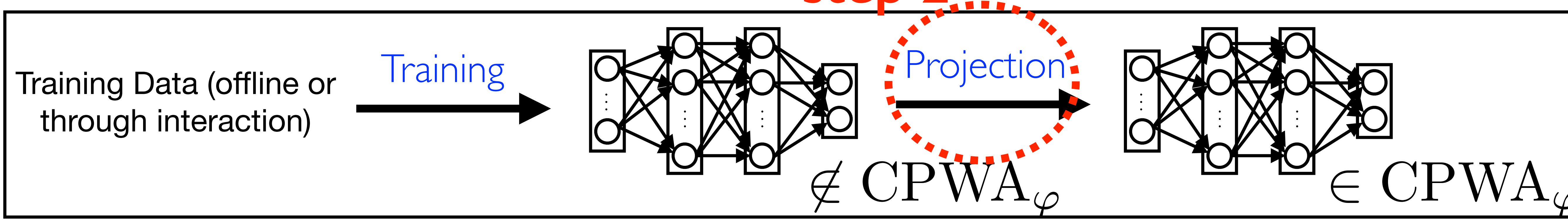


step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$f(x, K_{\mathrm{CPWA}}(x)) \models \varphi$$
$$\forall\, K_{\mathrm{CPWA}} \in \mathrm{CPWA}_{\varphi}$$

step 2



step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

step 1

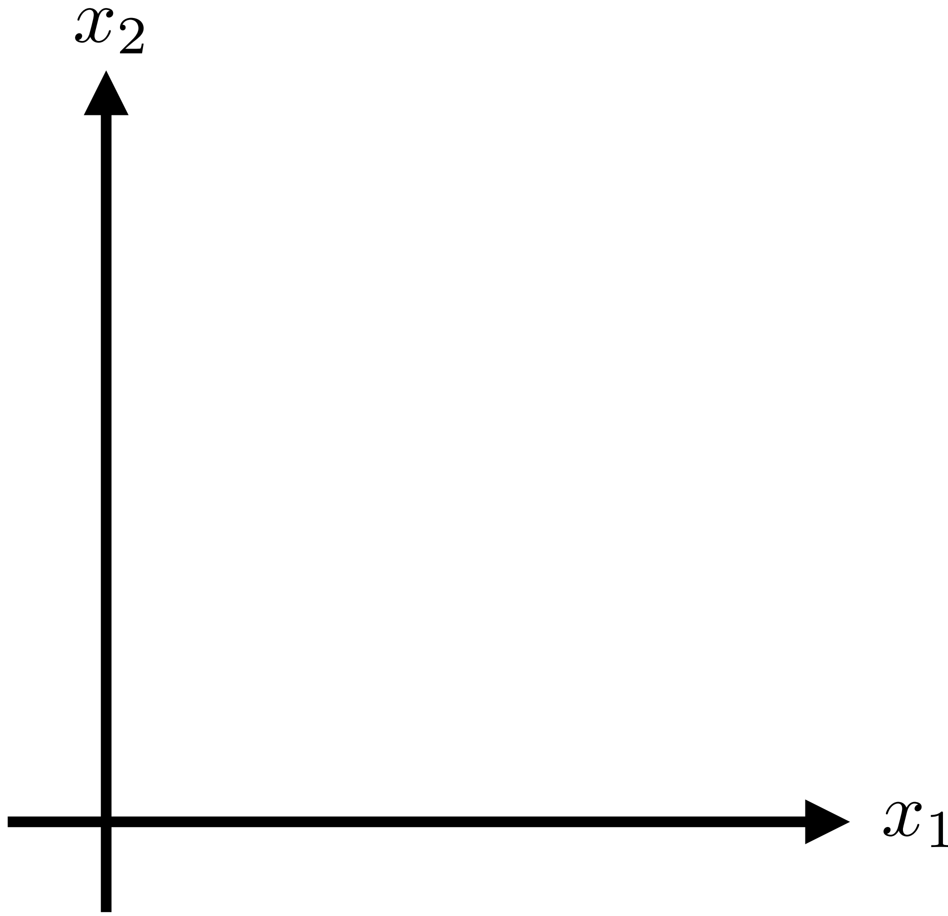
- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

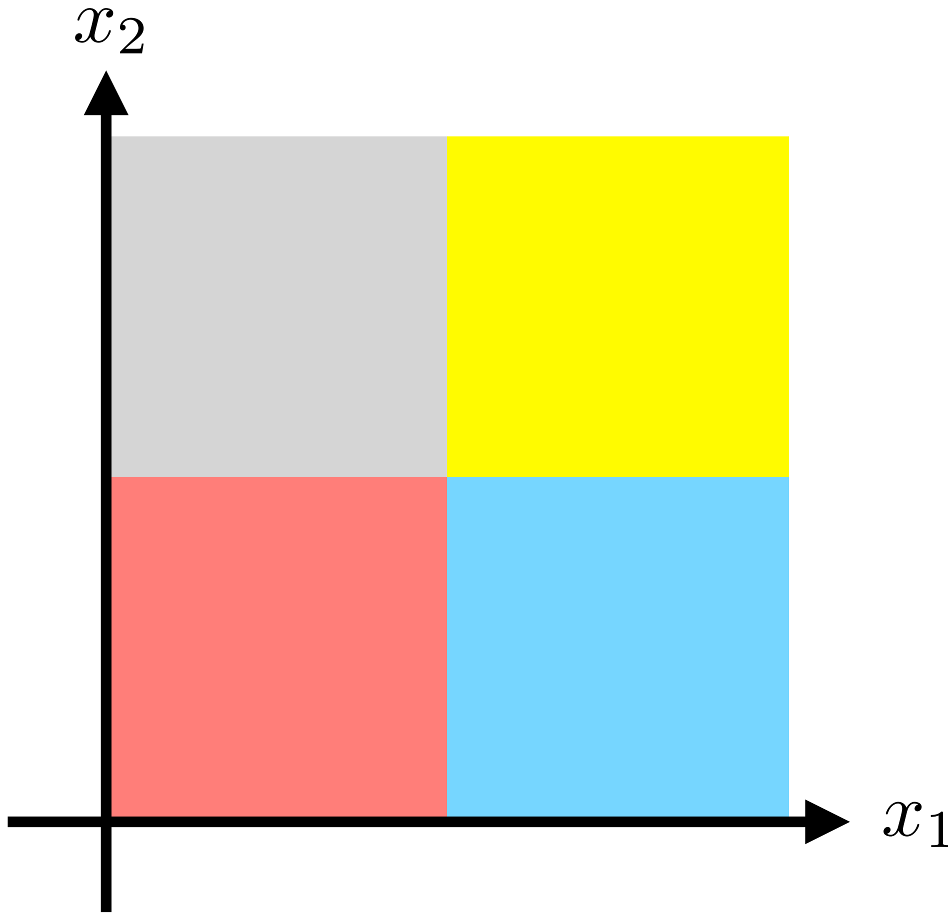
$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$



step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

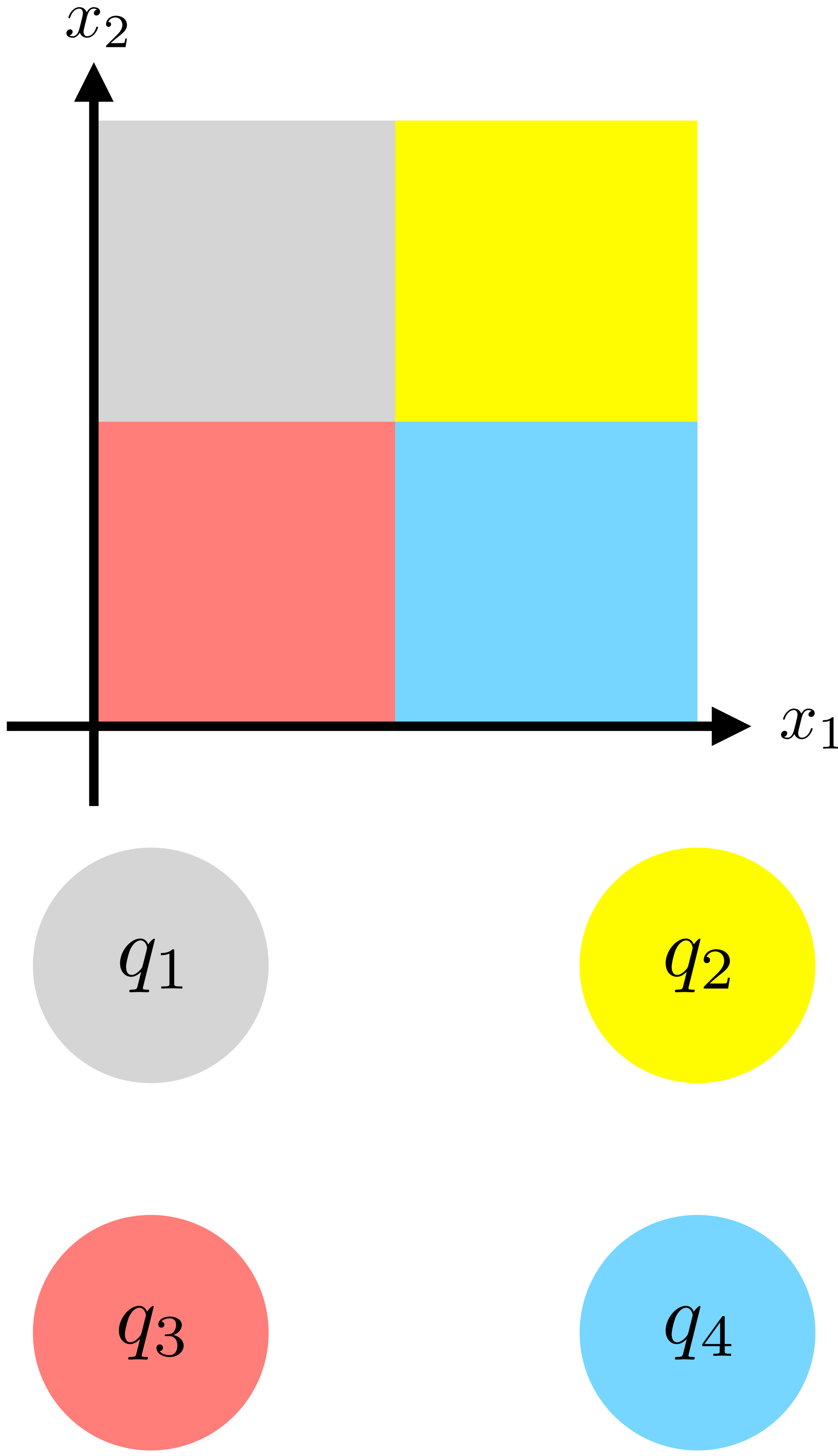
$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$



step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$



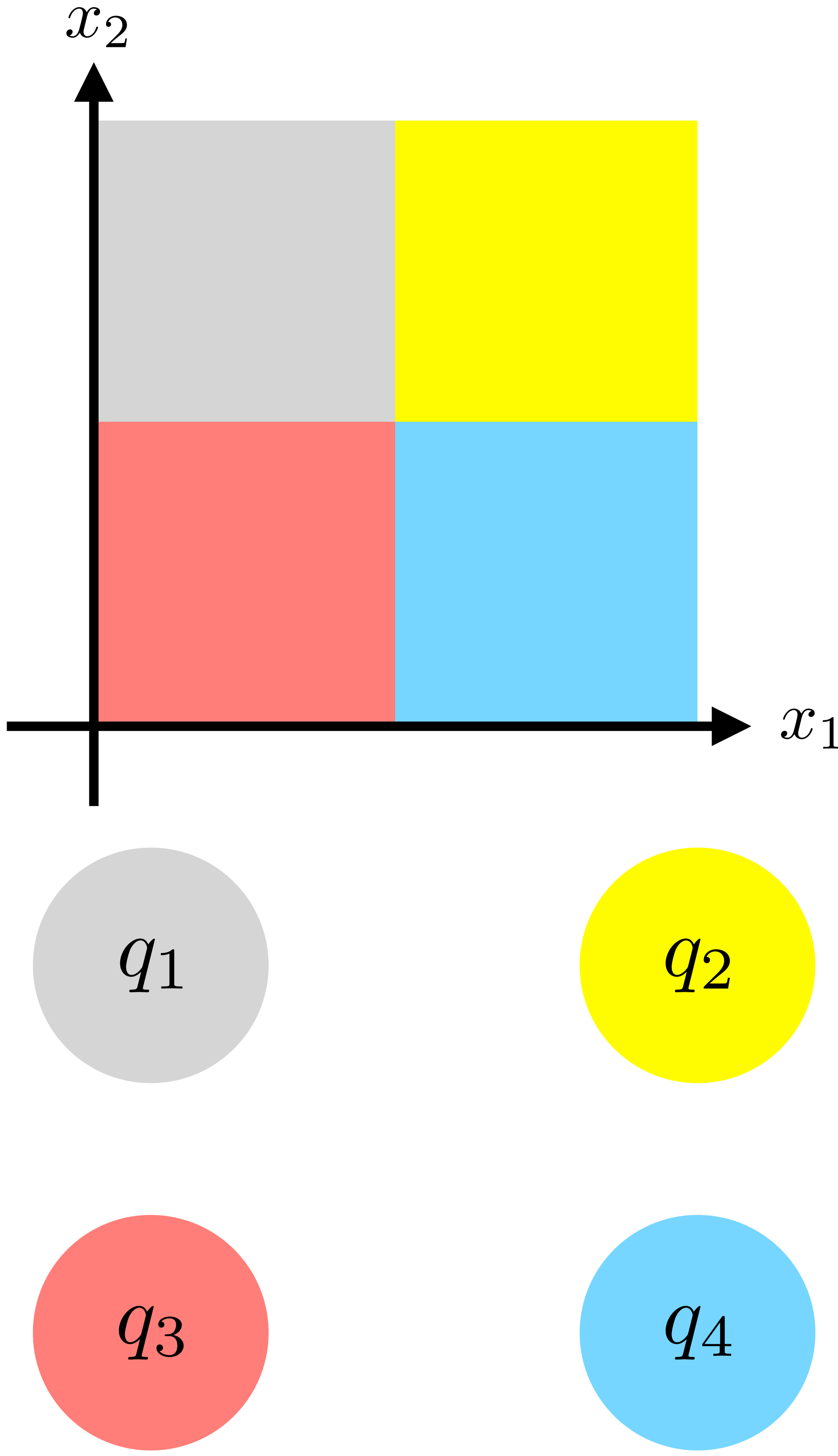
Resilient CPS Lab

step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

Abstract states:	$\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$



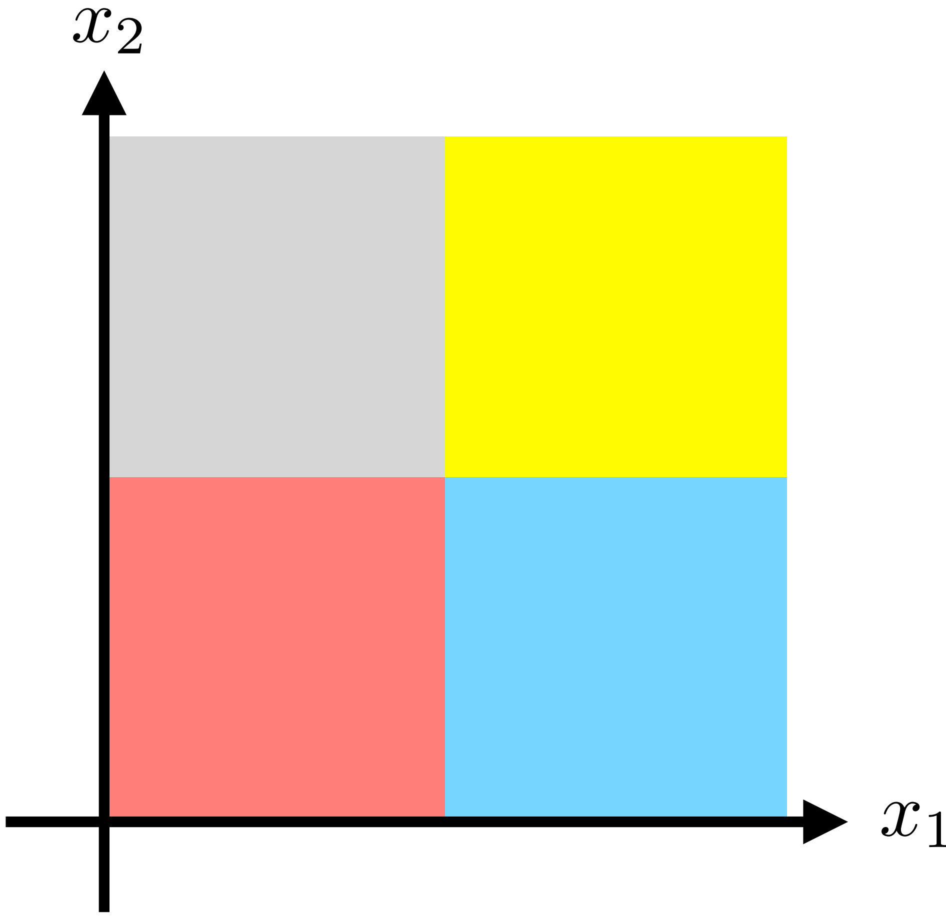
Resilient CPS Lab

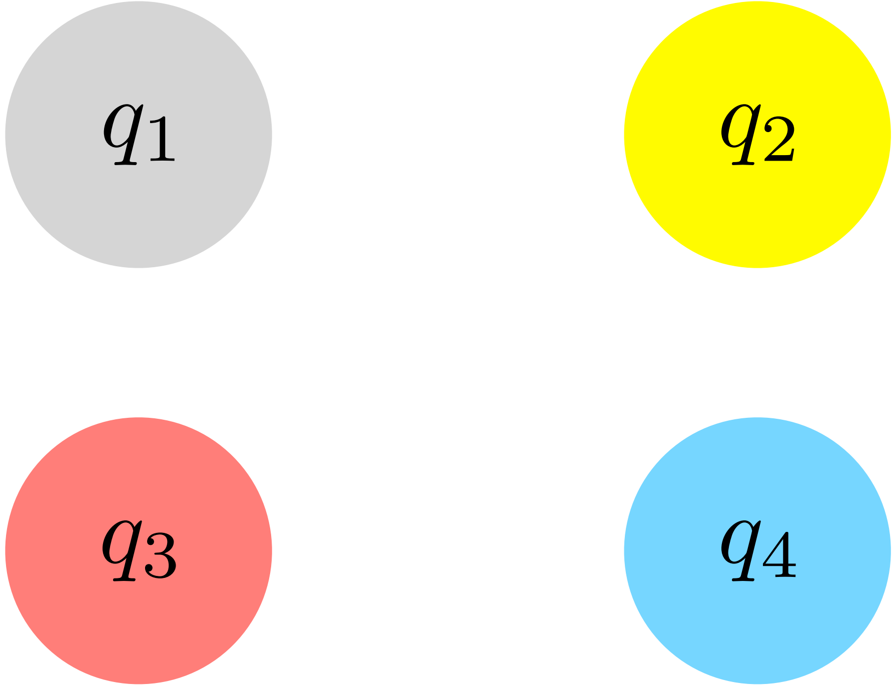
step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

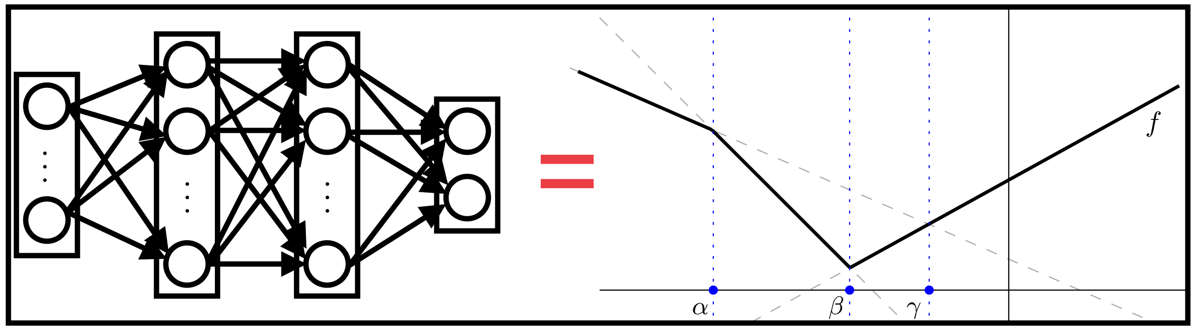
$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

$$\boxed{\text{Abstract states:} \qquad \mathbb{X} = \{q_1, q_2, \ldots, q_n\}}$$





Recall:



NN = Continuous Piece-Wise Affine (CPWA) functions

CPS Lab Resilient

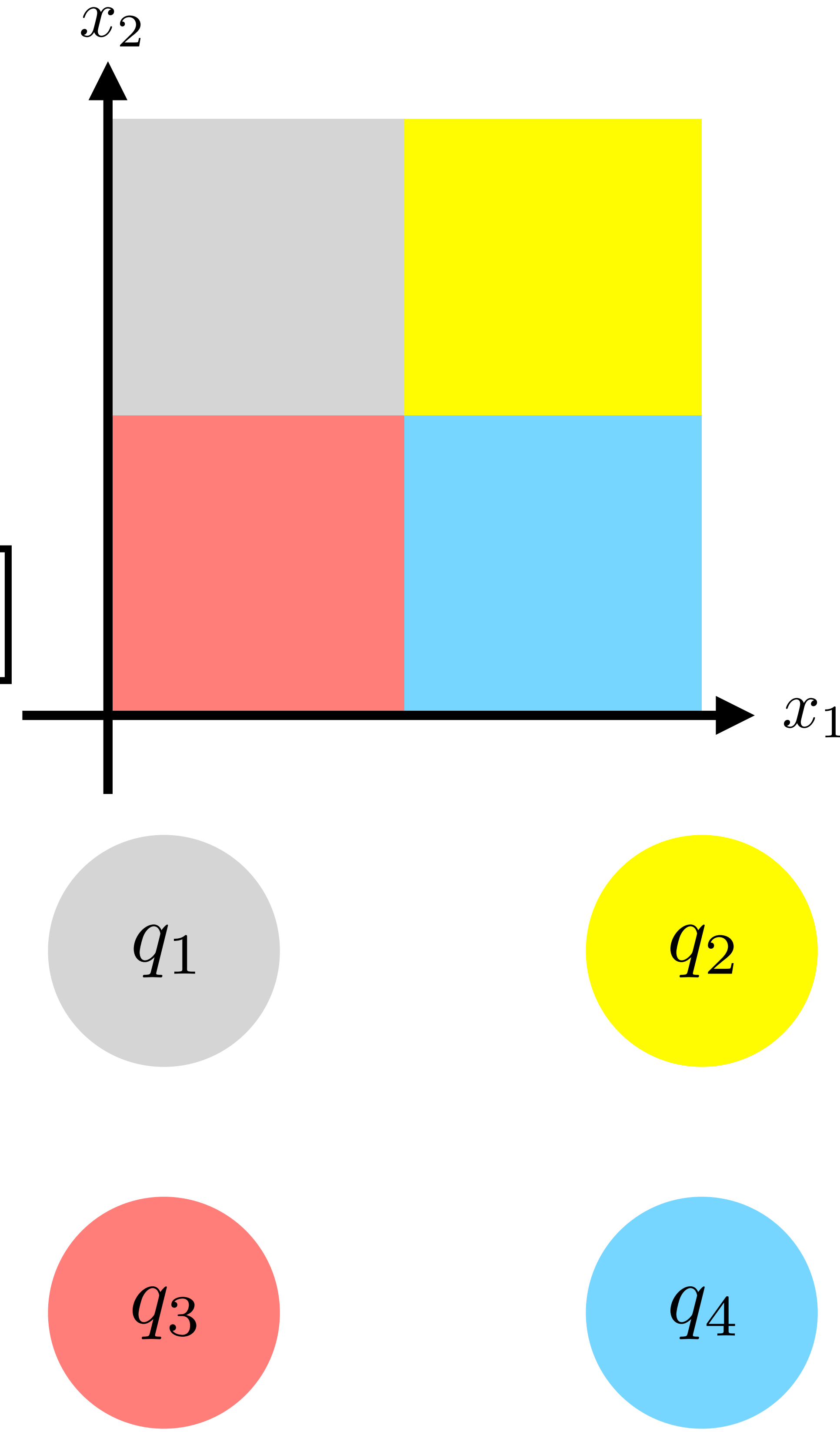
step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

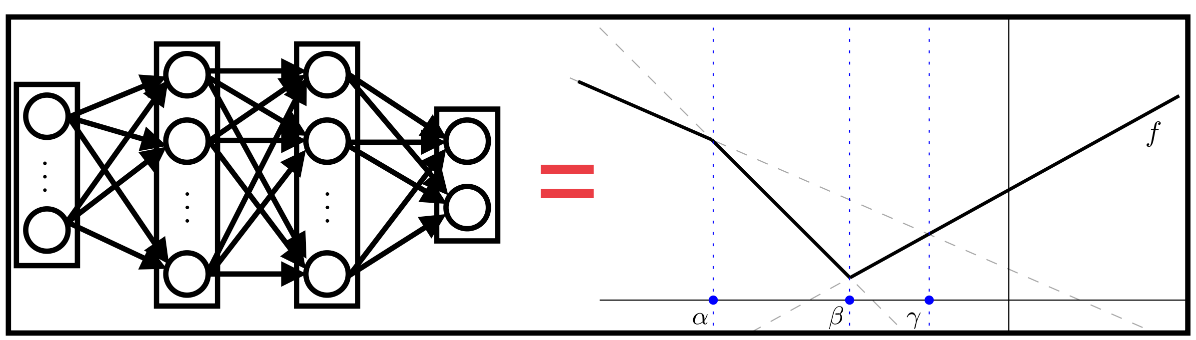
$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

Abstract states: $\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

$$u^{(t)} = K_i x^{(t)} + b_i$$



Recall:



NN = Continuous Piece-Wise Affine (CPWA) functions

step 1

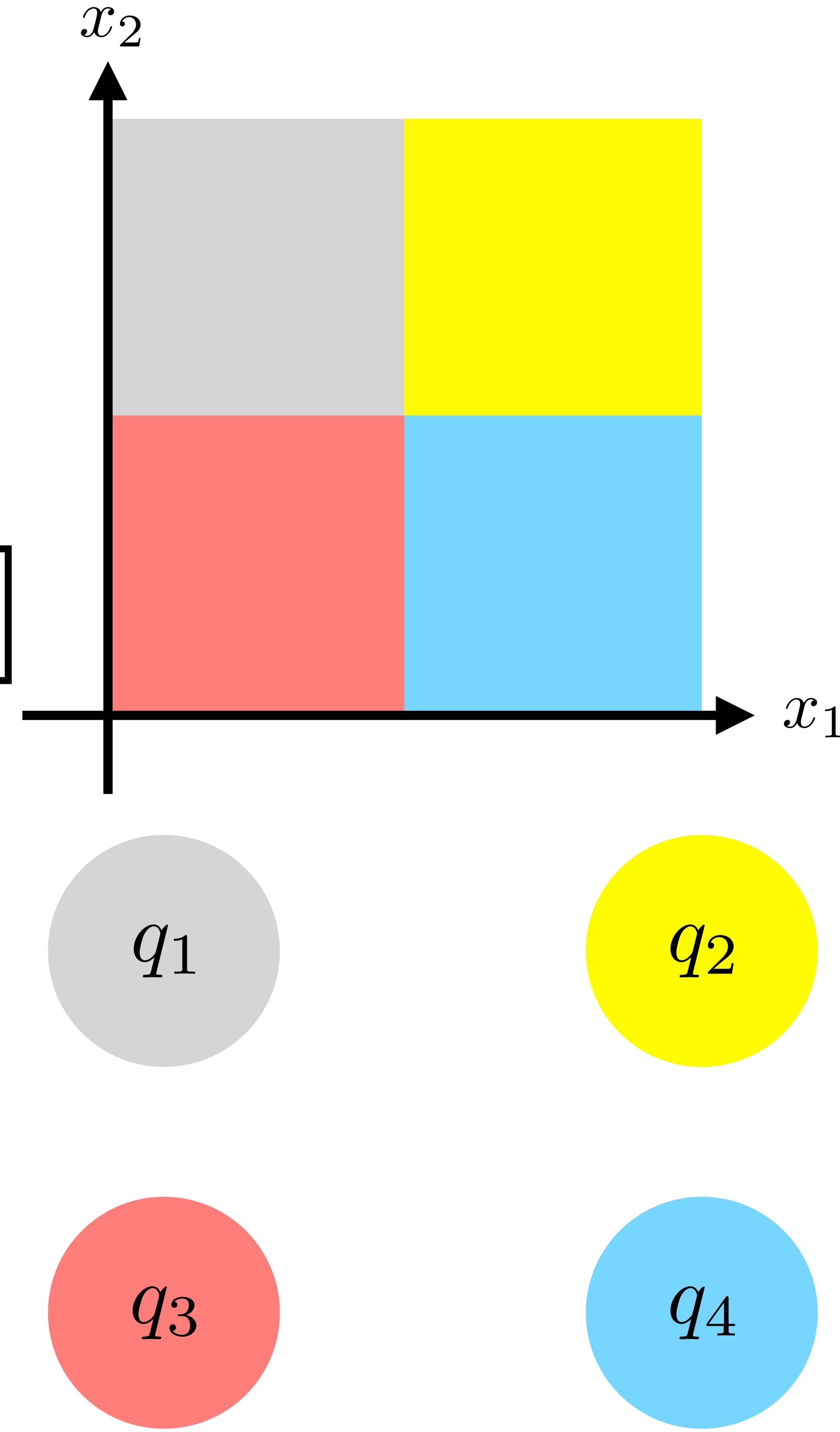
- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

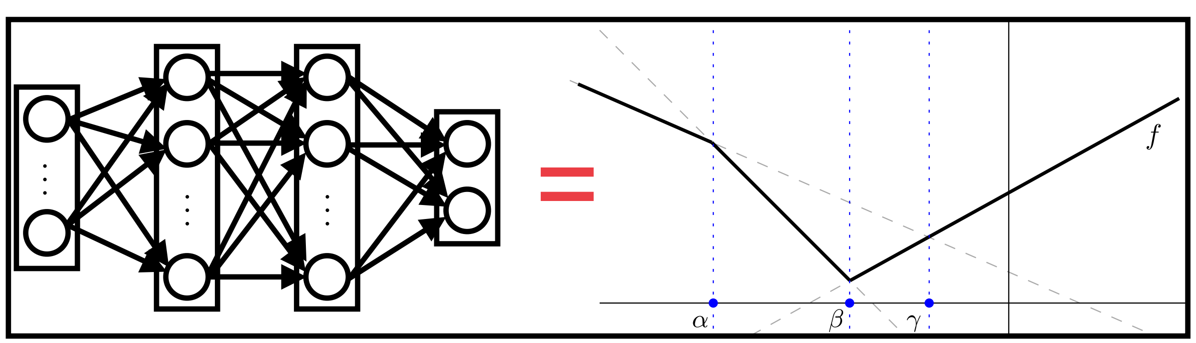
$$\boxed{\text{Abstract states:} \qquad \mathbb{X} = \{q_1, q_2, \ldots, q_n\}}$$

$$u^{(t)} = K_i x^{(t)} + b_i$$

$$\mathcal{P} = \{(K, b) \mid K \in \underbrace{\mathcal{K}}_{\text{polytopic}}, b \in \underbrace{\mathcal{B}}_{\text{polytopic}}\}$$



Recall:



NN = Continuous Piece-Wise Affine (CPWA) functions

step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

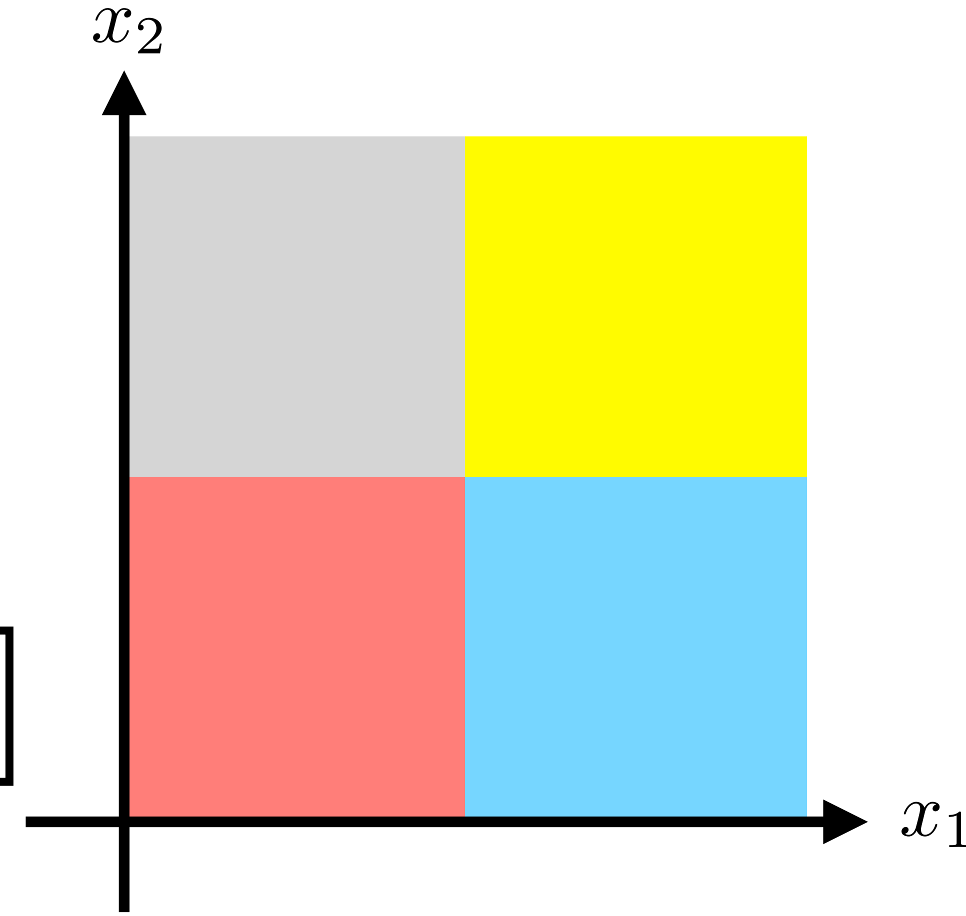
$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

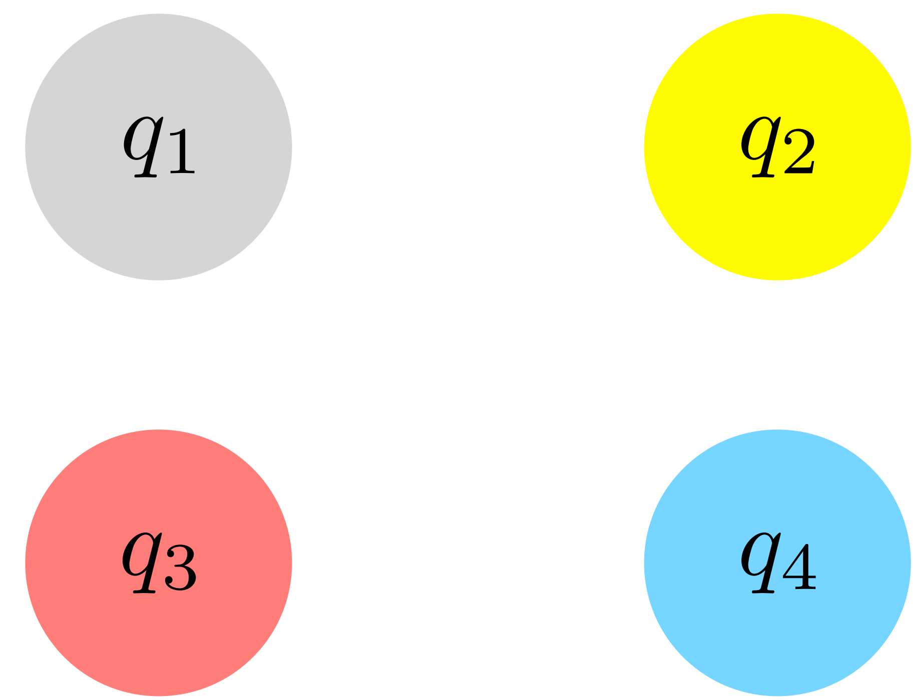
Abstract states:	$\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

$$u^{(t)} = K_i x^{(t)} + b_i$$

$$\mathcal{P} = \{(K, b) \mid K \in \underbrace{\mathcal{K}}_{\text{polytopic}}, b \in \underbrace{\mathcal{B}}_{\text{polytopic}}\}$$

Controller Partitions:	$\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$





step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

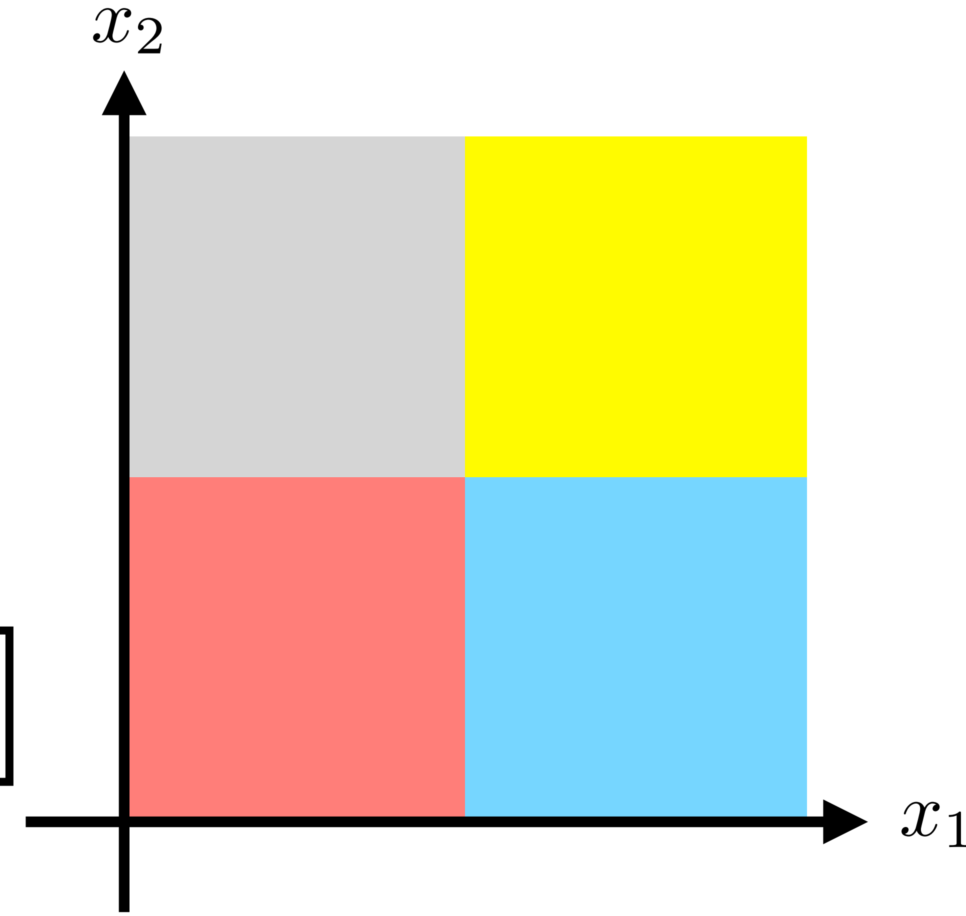
$\boxed{\text{Abstract states:} \quad \mathbb{X} = \{q_1, q_2, \ldots, q_n\}}$

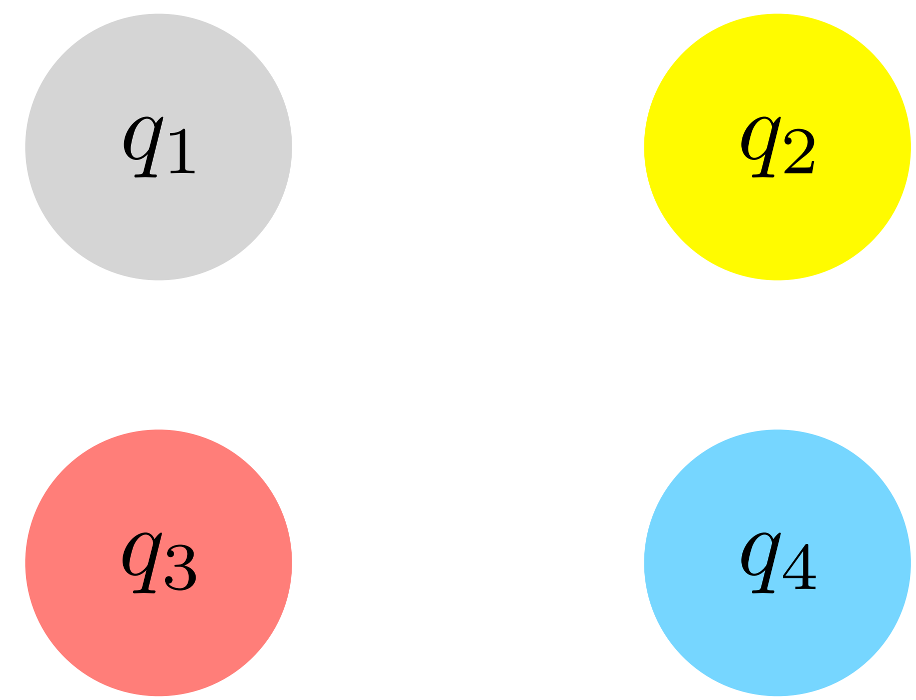
$$u^{(t)} = K_i x^{(t)} + b_i$$

$$\mathcal{P} = \{(K, b) \mid K \in \underbrace{\mathcal{K}}_{\text{polytopic}}, b \in \underbrace{\mathcal{B}}_{\text{polytopic}}\}$$

$\boxed{\text{Controller Partitions:} \quad \mathbb{P} = \{P_1, P_2, \ldots, P_m\}}$

$$\mathrm{Post}(q_1, P_1) \; ?$$





step I

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

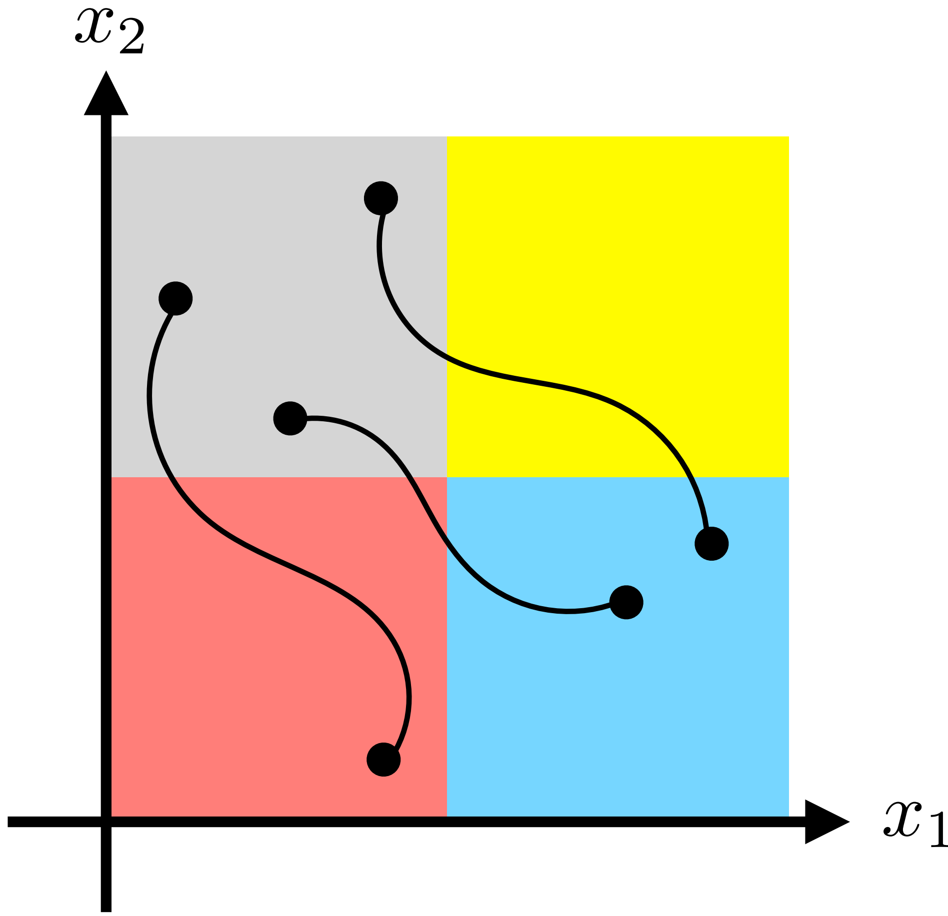
$\boxed{\text{Abstract states:} \quad \mathbb{X} = \{q_1, q_2, \ldots, q_n\}}$

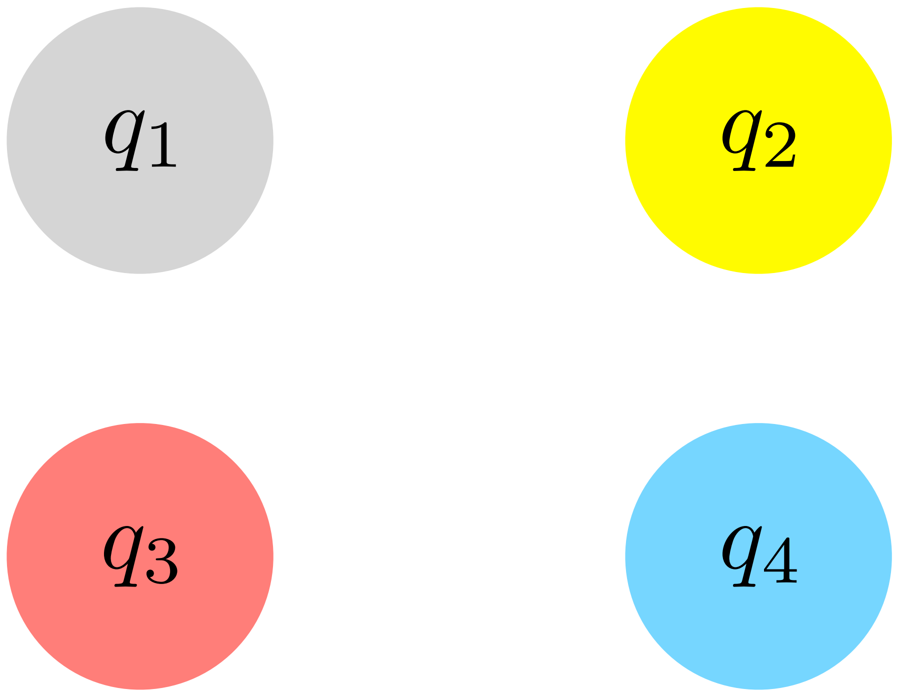
$$u^{(t)} = K_i x^{(t)} + b_i$$

$$\mathcal{P} = \{(K, b) \mid K \in \underbrace{\mathcal{K}}_{\text{polytopic}}, b \in \underbrace{\mathcal{B}}_{\text{polytopic}}\}$$

$\boxed{\text{Controller Partitions:} \quad \mathbb{P} = \{P_1, P_2, \ldots, P_m\}}$

$$\mathrm{Post}(q_1, P_1)\ ?$$





step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

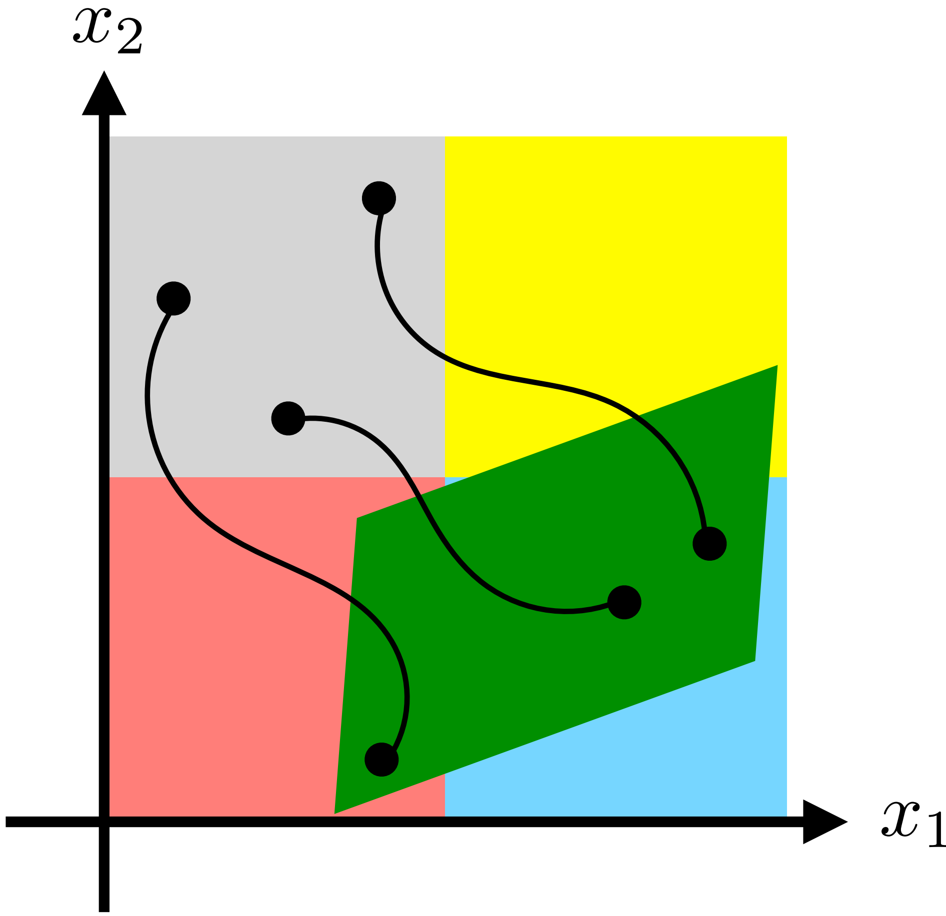
Abstract states: $\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

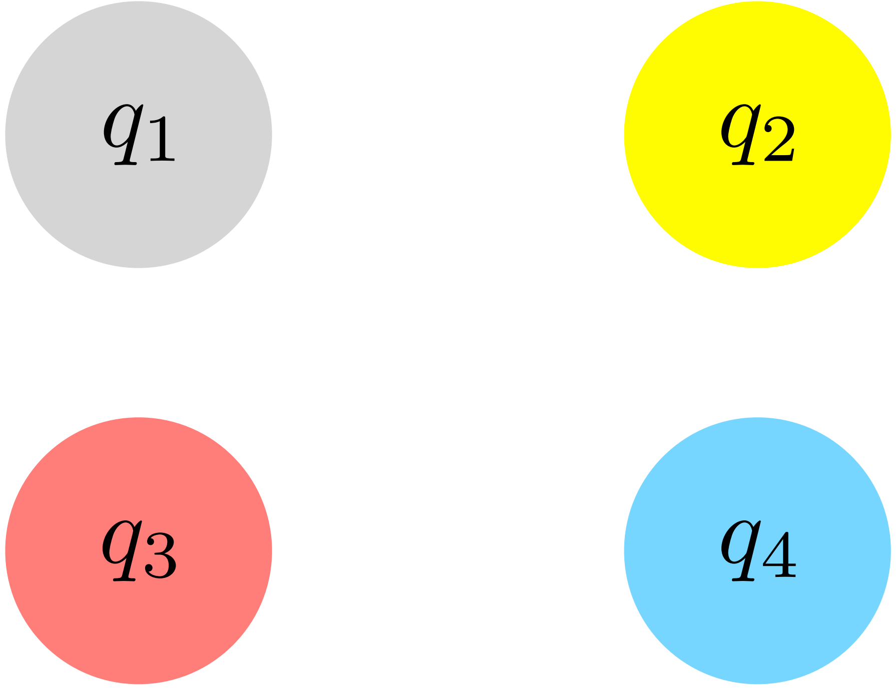
$$u^{(t)} = K_i x^{(t)} + b_i$$

$$\mathcal{P} = \{(K, b) \mid K \in \underbrace{\mathcal{K}}_{\text{polytopic}}, b \in \underbrace{\mathcal{B}}_{\text{polytopic}}\}$$

Controller Partitions: $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$

$$\mathrm{Post}(q_1, P_1) \text{ ?}$$

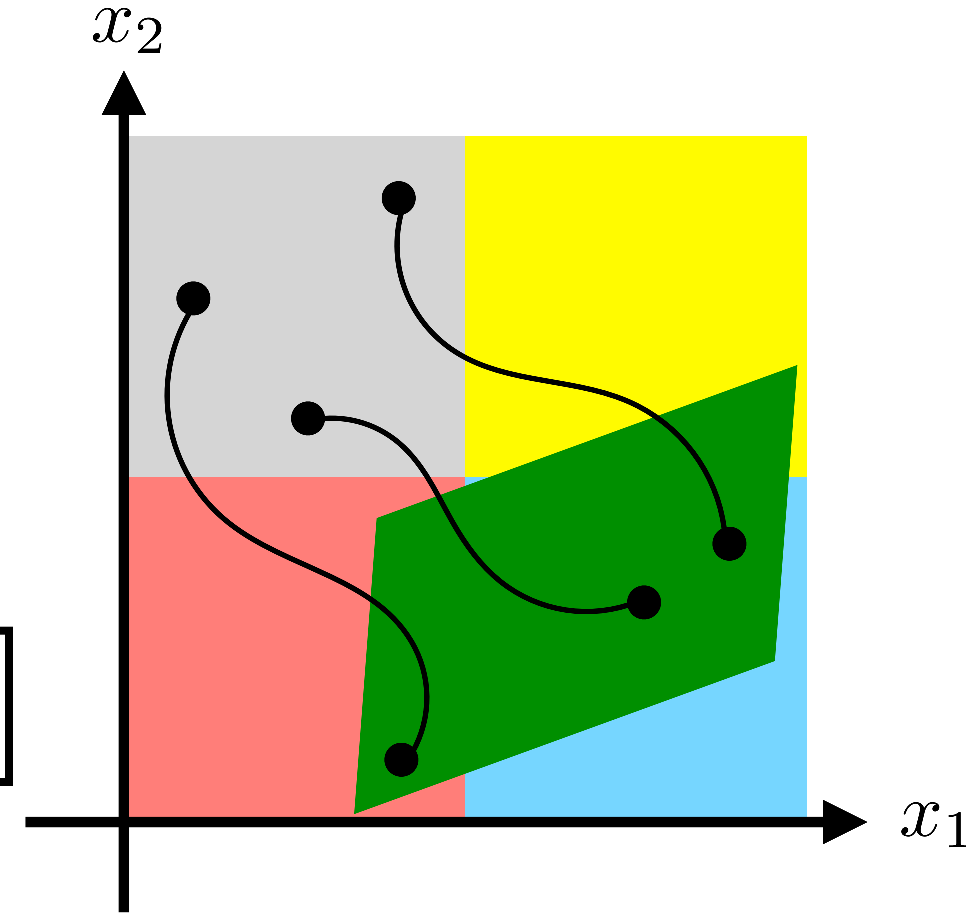




step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

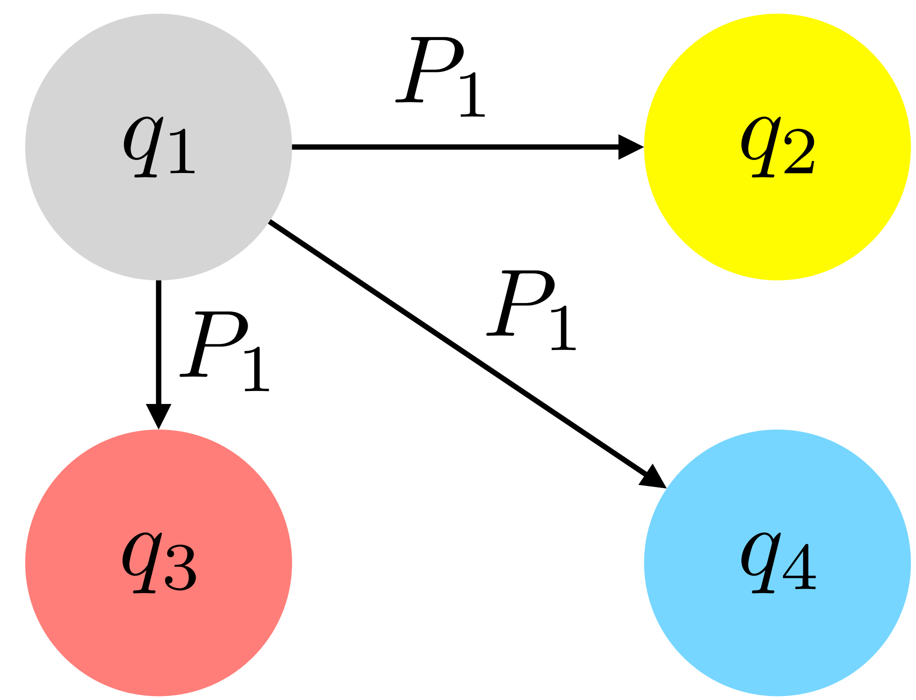
$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$



$$\boxed{\text{Abstract states:} \quad \mathbb{X} = \{q_1, q_2, \ldots, q_n\}}$$

$$u^{(t)} = K_i x^{(t)} + b_i$$

$$\mathcal{P} = \{(K, b) \mid K \in \underbrace{\mathcal{K}}_{\text{polytopic}}, b \in \underbrace{\mathcal{B}}_{\text{polytopic}}\}$$



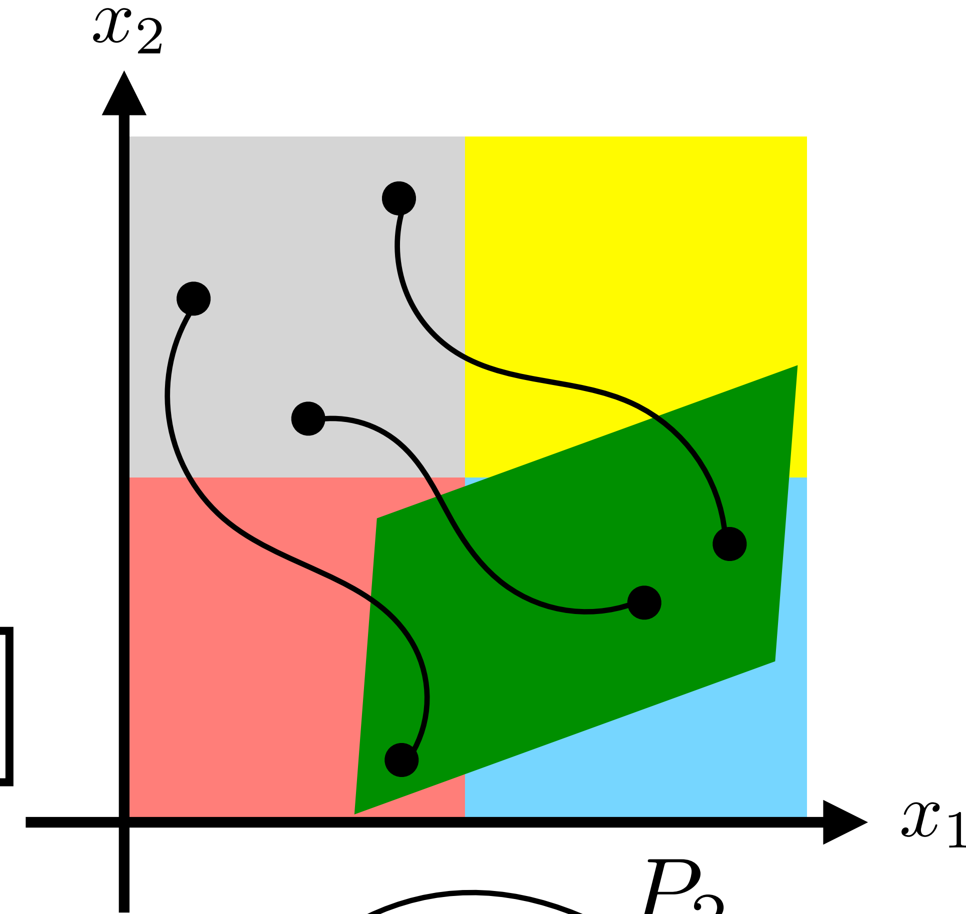
$$\boxed{\text{Controller Partitions:} \quad \mathbb{P} = \{P_1, P_2, \ldots, P_m\}}$$

$$\text{Post}(q_1, P_1) = \{q_2, q_3, q_4\}$$

Resilient CPS Lab

step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

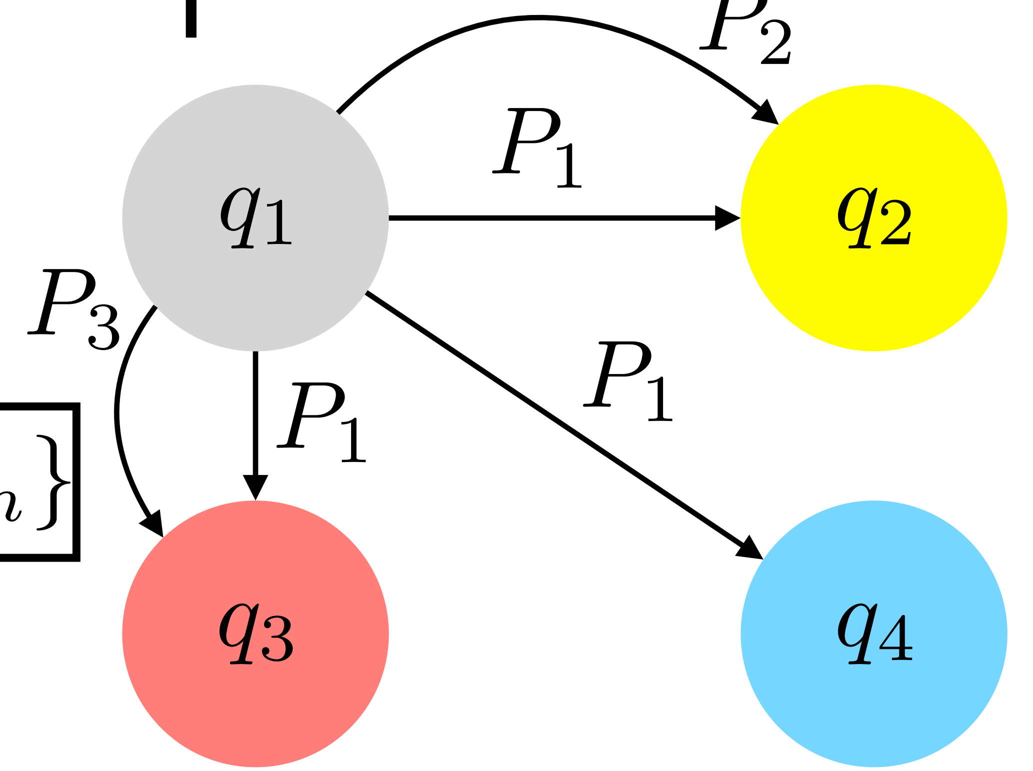


$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

Abstract states: $\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

$$u^{(t)} = K_i x^{(t)} + b_i$$

$$\mathcal{P} = \{(K, b) \mid K \in \underbrace{\mathcal{K}}_{\text{polytopic}}, b \in \underbrace{\mathcal{B}}_{\text{polytopic}}\}$$



Controller Partitions: $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$

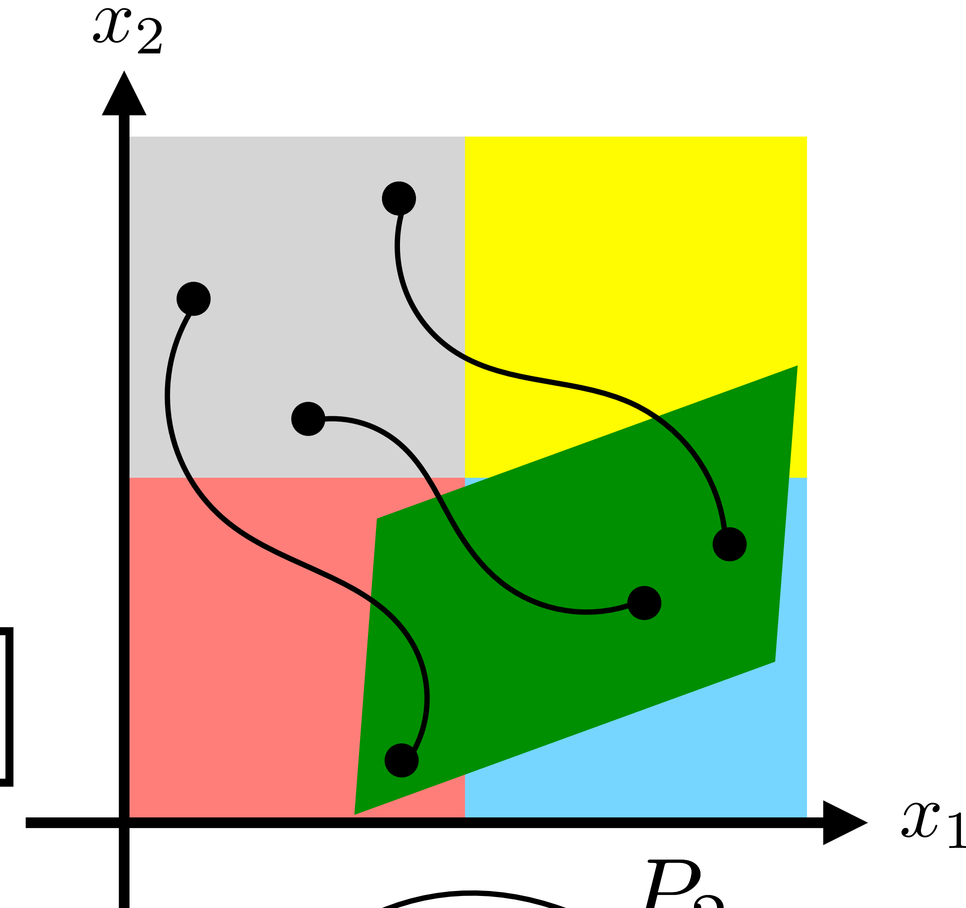
$$\mathrm{Post}(q_1, P_1) = \{q_2, q_3, q_4\}$$

Transitions: $\mathrm{Post}(q_i, P_j) = \{f(x, Kx + b) \mid x \in q_i, \ (K, b) \in P_j\}$

Resilient CPS Lab

step I

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

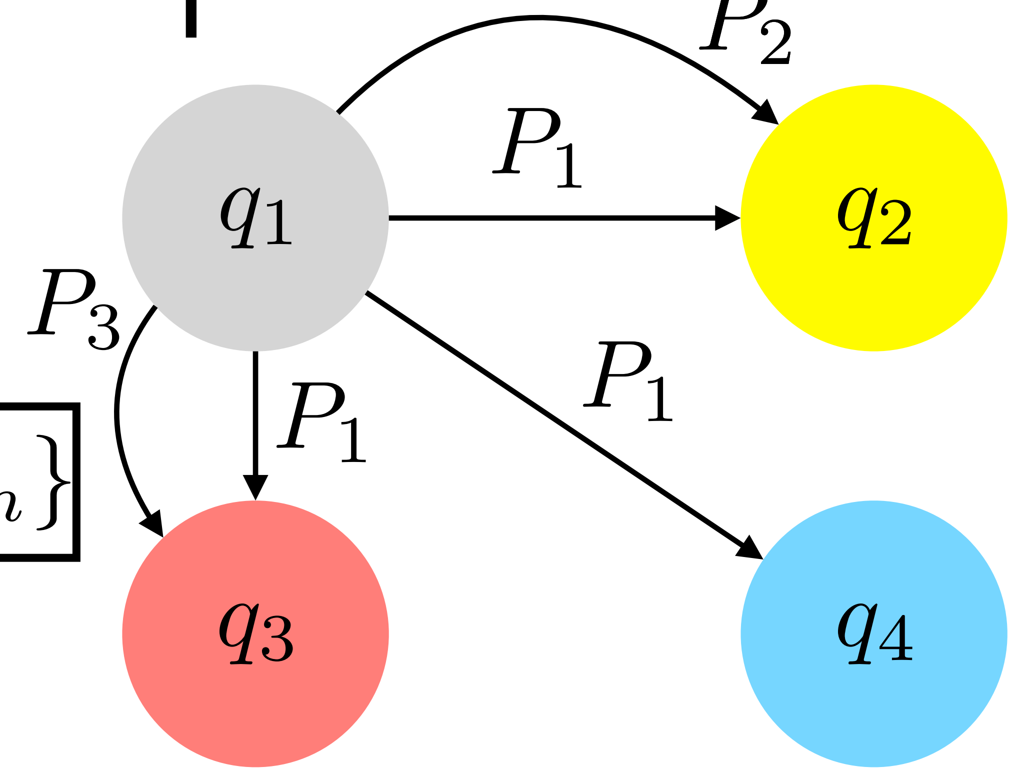


$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

Abstract states: $\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

$$u^{(t)} = K_i x^{(t)} + b_i$$

$$\mathcal{P} = \{(K, b) \mid K \in \underbrace{\mathcal{K}}_{\text{polytopic}}, b \in \underbrace{\mathcal{B}}_{\text{polytopic}}\}$$



Controller Partitions: $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$

$$\text{Post}(q_1, P_1) = \{q_2, q_3, q_4\}$$

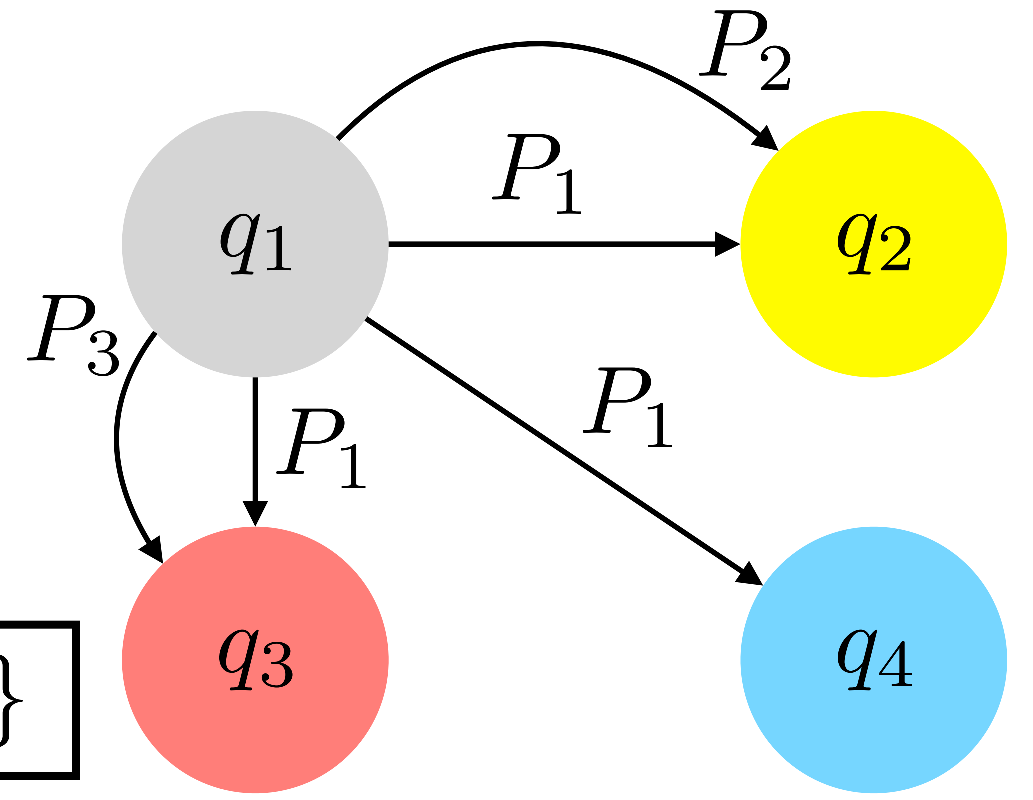
Transitions: $\text{Post}(q_i, P_j) = \{f(x, Kx + b) \mid x \in q_i, \ (K, b) \in P_j\}$

Note: Computing the **Post** operator can be done using existing techniques for reachability analysis of nonlinear systems (with the caveat that existing tools focus on partitioning the "**input**" space instead of the "**controller**" space).

step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$



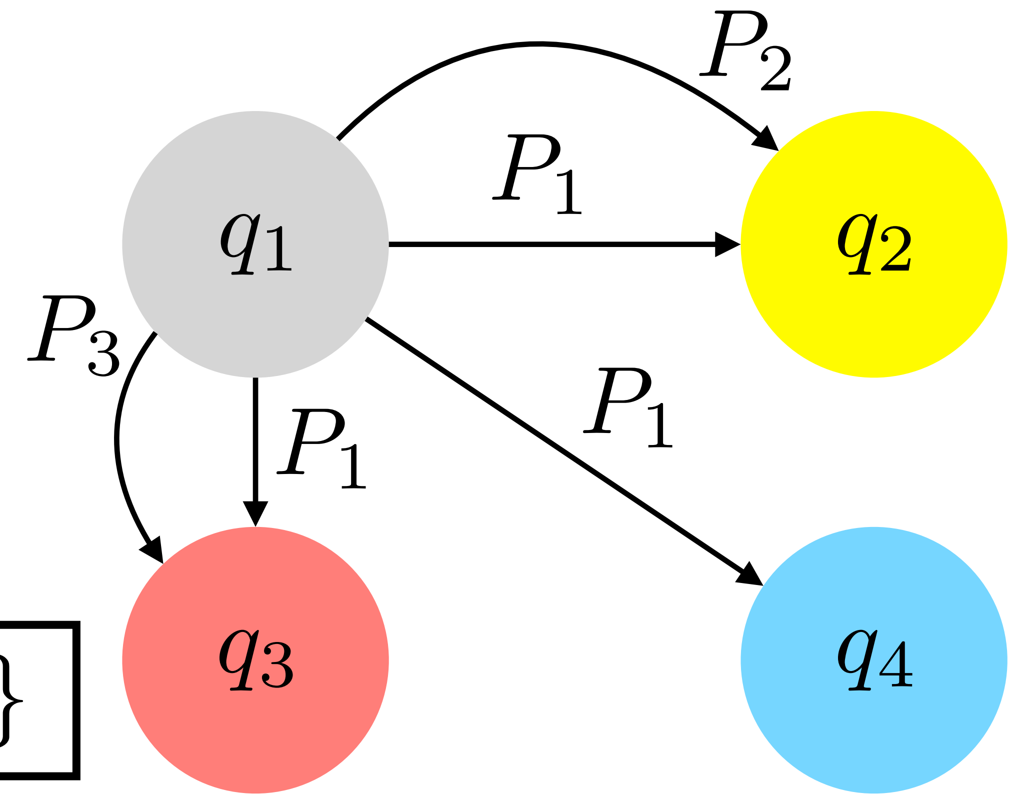
Abstract states: $\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

Controller Partitions: $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$

Transitions: $\mathrm{Post}(q_i, P_j) = \{f(x, Kx + b) \mid x \in q_i,\ (K, b) \in P_j\}$

step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs



$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

Abstract states: $\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

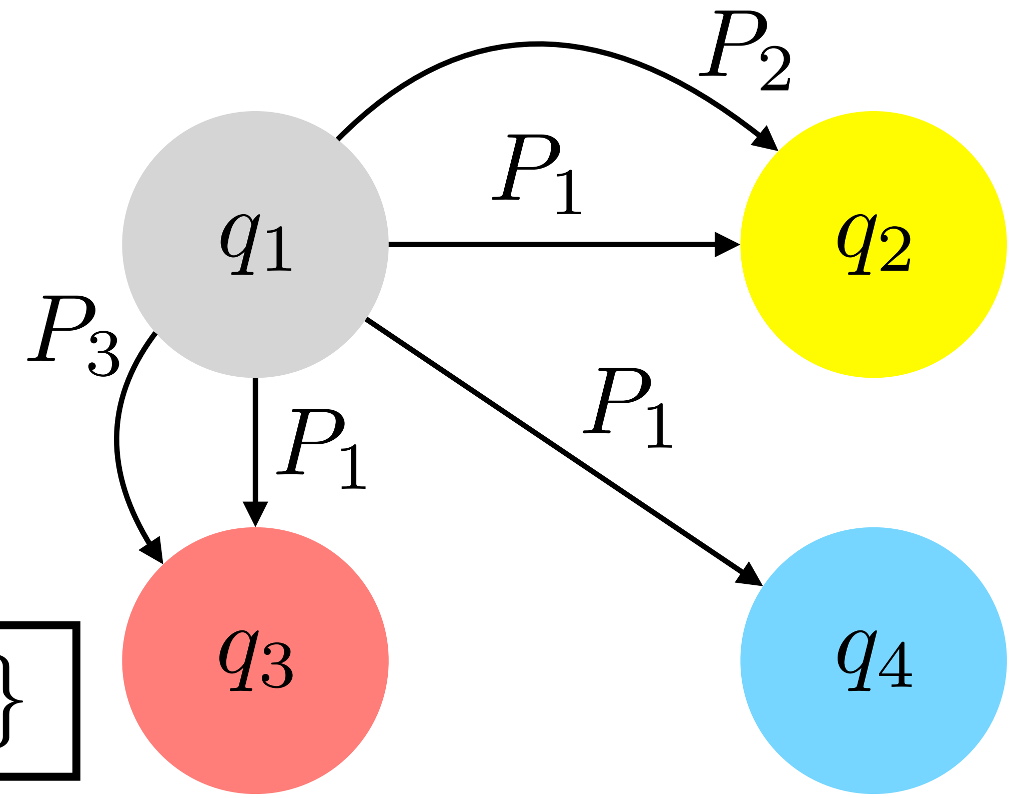
Controller Partitions: $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$

Transitions: $\text{Post}(q_i, P_j) = \{f(x, Kx + b) \mid x \in q_i, \ (K, b) \in P_j\}$

Specs (safety): $\varphi = \Box \neg q_4$

step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs



$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

Abstract states: $\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

Controller Partitions: $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$

Transitions: $\text{Post}(q_i, P_j) = \{f(x, Kx + b) \mid x \in q_i, \; (K, b) \in P_j\}$

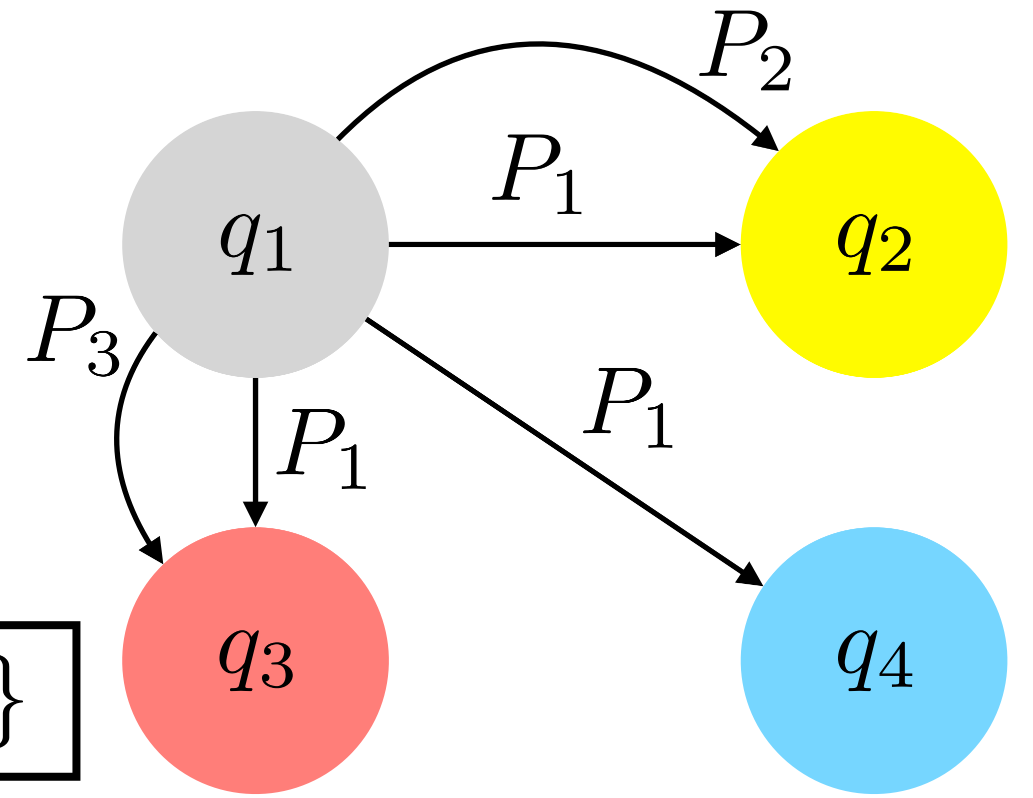
Specs (safety): $\varphi = \Box \neg q_4$

$$\text{CPWA}_\varphi(q_1) = P_2 \cup P_3$$

step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$



Abstract states: $\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

Controller Partitions: $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$

Transitions: $\mathrm{Post}(q_i, P_j) = \{f(x, Kx + b) \mid x \in q_i,\ (K, b) \in P_j\}$

Specs (safety): $\varphi = \Box\neg q_4$

$$\mathrm{CPWA}_\varphi(q_1) = P_2 \cup P_3$$

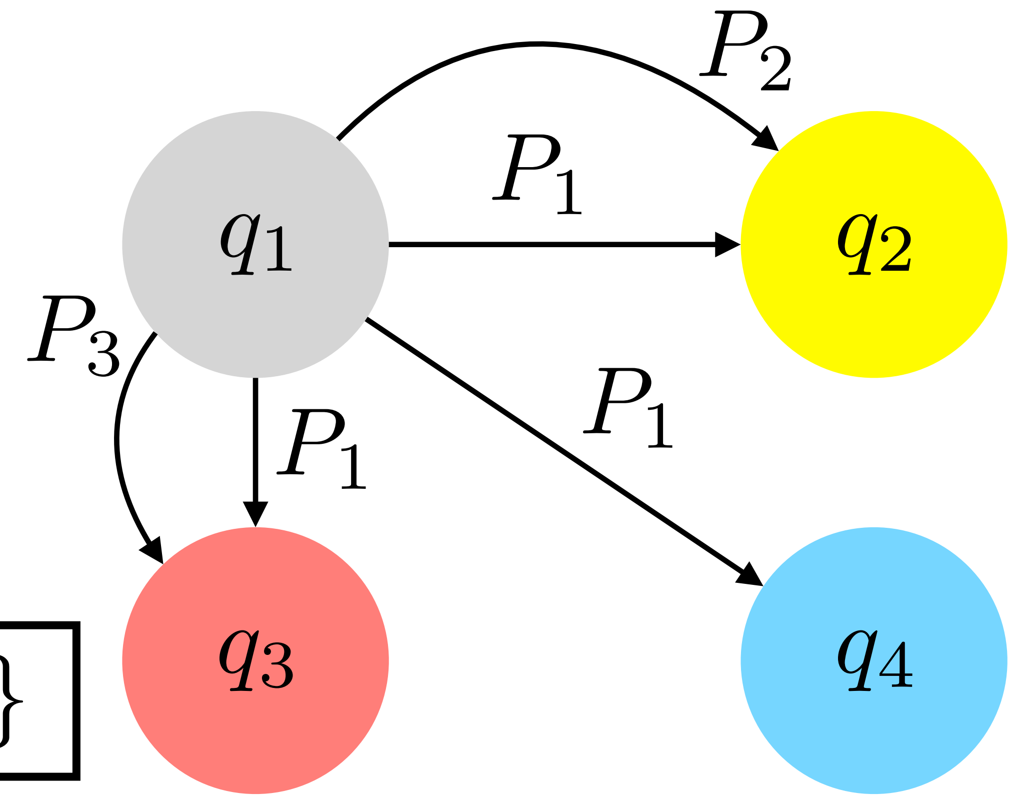
$$\mathrm{CPWA}_\varphi(q_2) = \ldots$$

$$\mathrm{CPWA}_\varphi(q_3) = \ldots$$

CPS Lab
Resilient

step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs



$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

Abstract states: $\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

Controller Partitions: $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$

Transitions: $\mathrm{Post}(q_i, P_j) = \{f(x, Kx + b) \mid x \in q_i, \ (K, b) \in P_j\}$

Specs (safety): $\varphi = \Box \neg q_4$

Note: Same can be extended to liveness properties using an abstract model built using the **Pre** operator instead of the **Post** operator

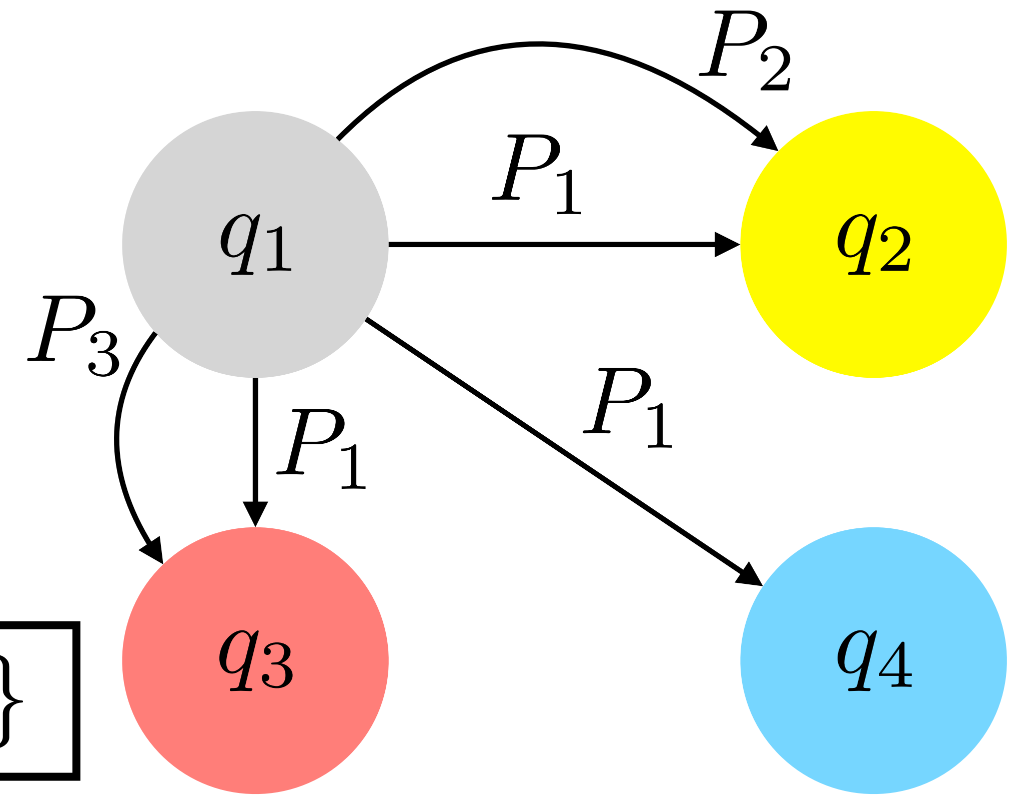
$$\mathrm{CPWA}_\varphi(q_1) = P_2 \cup P_3$$

$$\mathrm{CPWA}_\varphi(q_2) = \ldots$$

$$\mathrm{CPWA}_\varphi(q_3) = \ldots$$

step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs



$$x^{(t+1)} = f(x^{(t)}, u^{(t)})$$

Abstract states: $\mathbb{X} = \{q_1, q_2, \ldots, q_n\}$

Controller Partitions: $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$

Transitions: $\mathrm{Post}(q_i, P_j) = \{f(x, Kx + b) \mid x \in q_i,\ (K, b) \in P_j\}$

Specs (safety): $\varphi = \Box \neg q_4$

Note: Same can be extended to liveness properties using an abstract model built using the **Pre** operator instead of the **Post** operator

$$\mathrm{CPWA}_\varphi(q_1) = P_2 \cup P_3$$

$$\mathrm{CPWA}_\varphi(q_2) = \ldots$$

$$\mathrm{CPWA}_\varphi(q_3) = \ldots$$

$$f(x, K_{\mathrm{CPWA}}(x)) \models \varphi \quad \forall\, K_{\mathrm{CPWA}} \in \mathrm{CPWA}_\varphi$$

CPS Lab Resilient

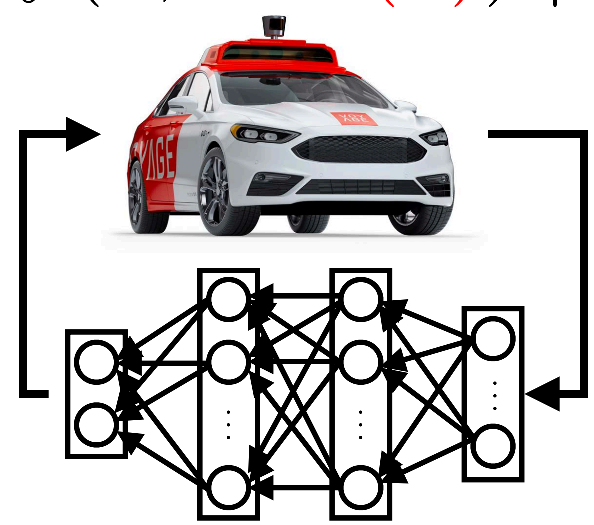
$$x^{(t+1)} = f(x^{(t)}, u^{(t)}) \rightarrow$$

$$\varphi \rightarrow$$

Training Data (offline or through interaction) →

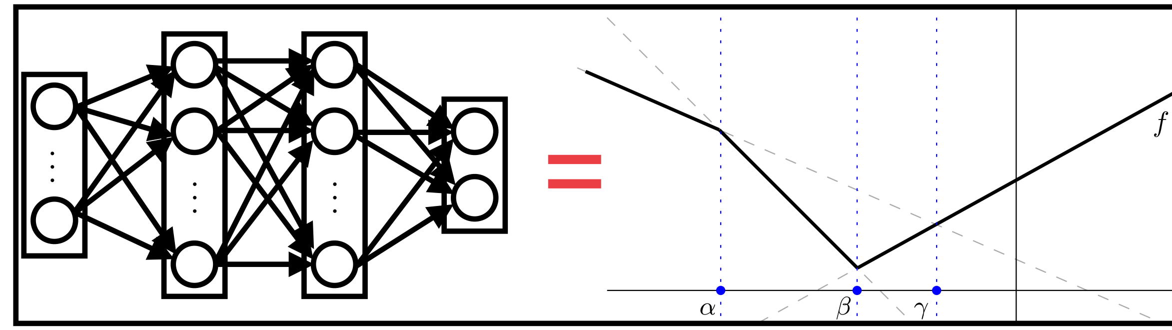
Formal NN Training →

$$f(x, \color{red}{\mathcal{NN}(x)}) \models \varphi$$



Core idea:

- Regression ReLU NN are Continuous Piece-Wise Affine (CPWA) functions

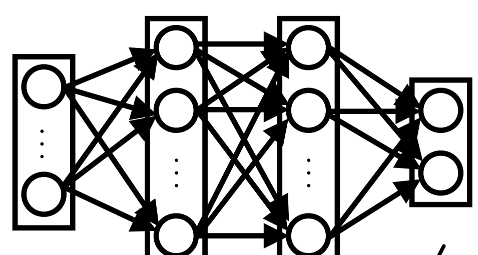


step 1

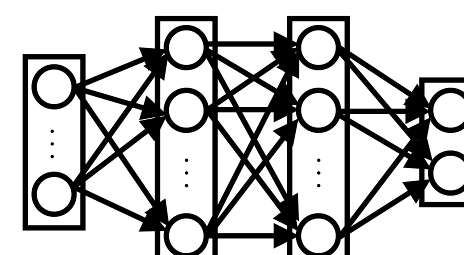
- Use reachability analysis to identify families of CPWA functions that satisfy the specs

$$f(x, \color{blue}{K_{\text{CPWA}}(x)}) \models \varphi \quad \forall\, K_{\text{CPWA}} \in \text{CPWA}_{\varphi}$$

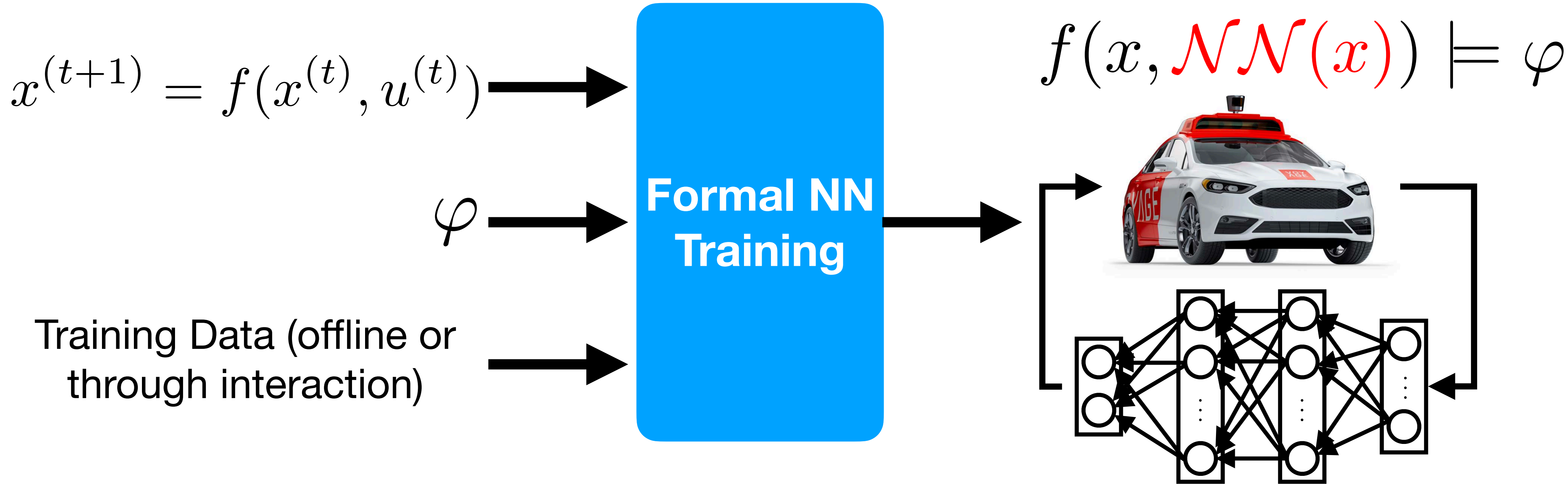
Training Data (offline or through interaction) $\xrightarrow{\text{Training}}$



$\notin \text{CPWA}_{\varphi}$ $\xrightarrow{\text{Projection}}$

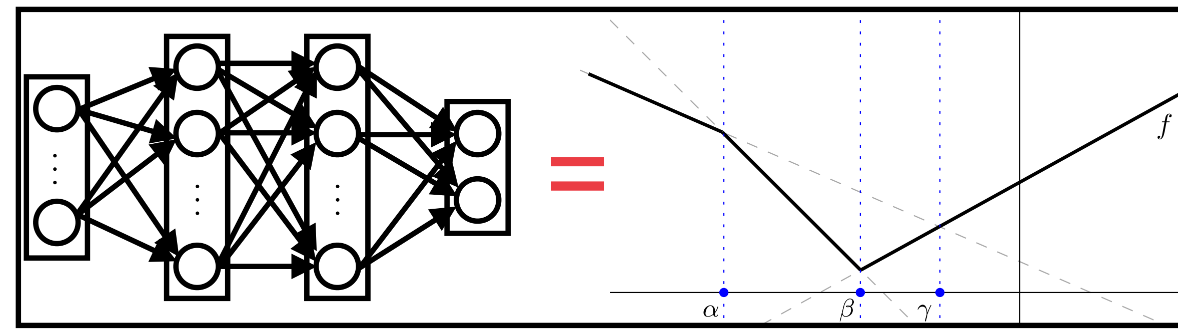


$\in \text{CPWA}_{\varphi}$



Core idea:

- Regression ReLU NN are Continuous Piece-Wise Affine (CPWA) functions

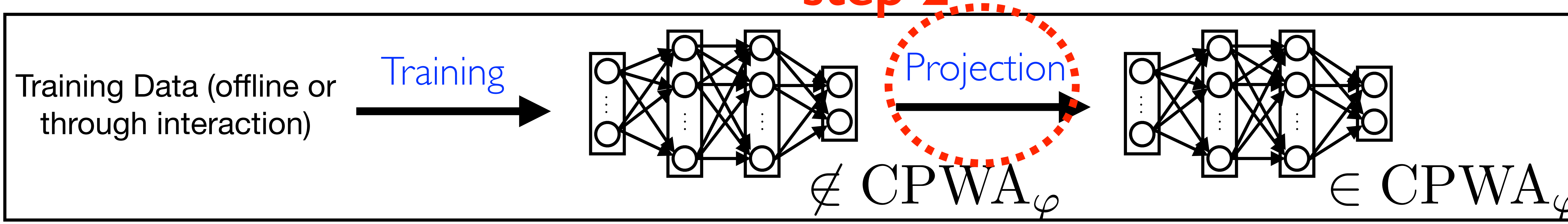


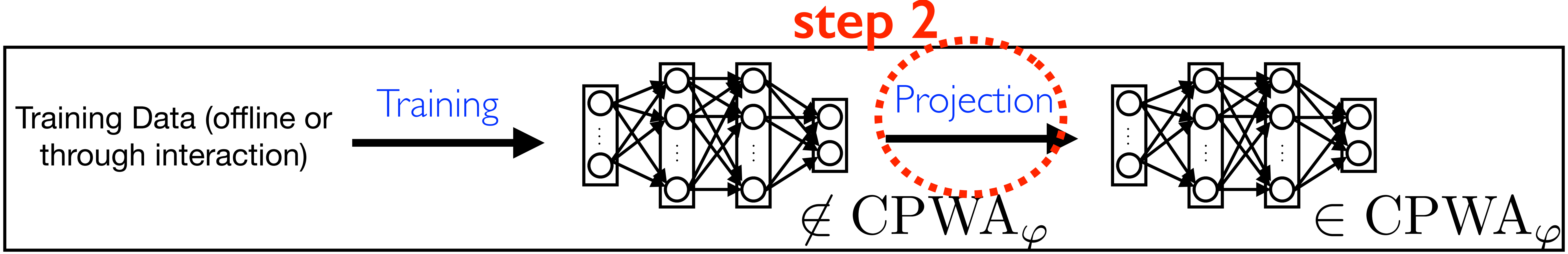
step 1

- Use reachability analysis to identify families of CPWA functions that satisfy the specs

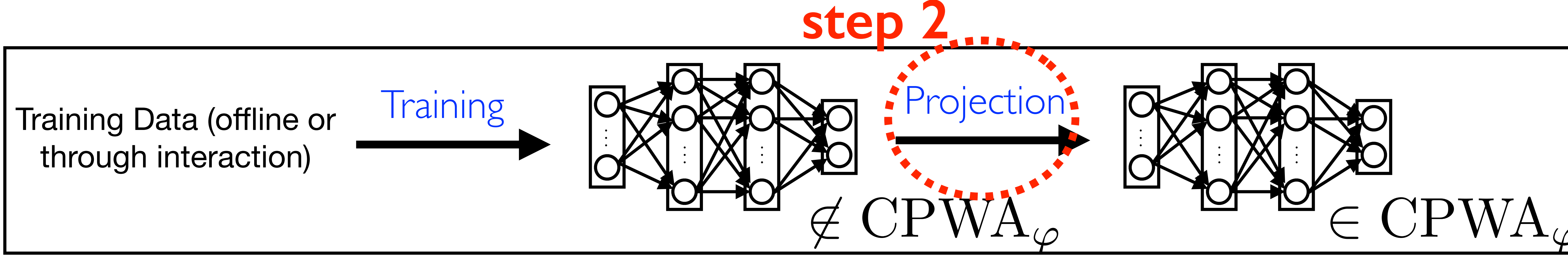
$$f(x, K_{\text{CPWA}}(x)) \models \varphi \quad \forall\, K_{\text{CPWA}} \in \text{CPWA}_{\varphi}$$

step 2

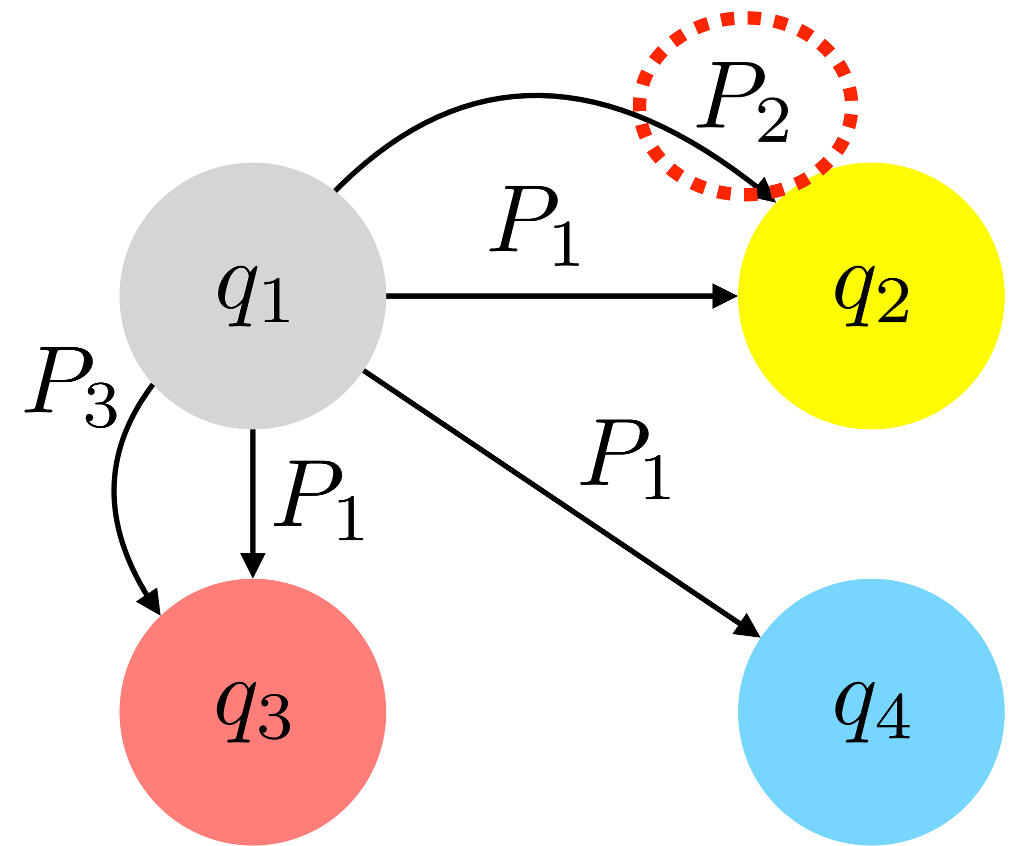




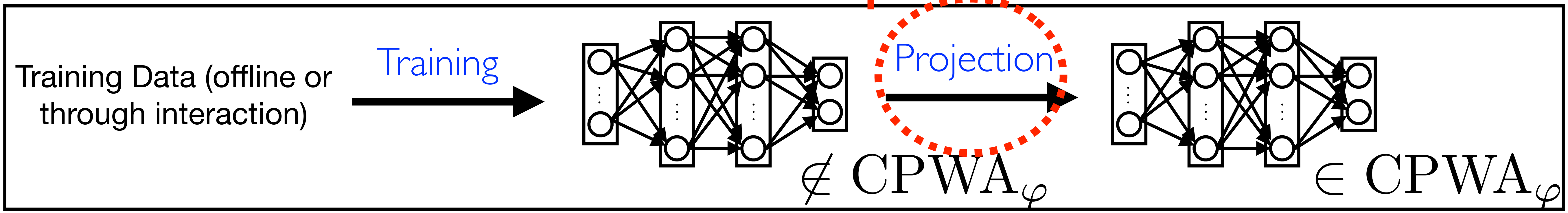
CPS Lab
Resilient

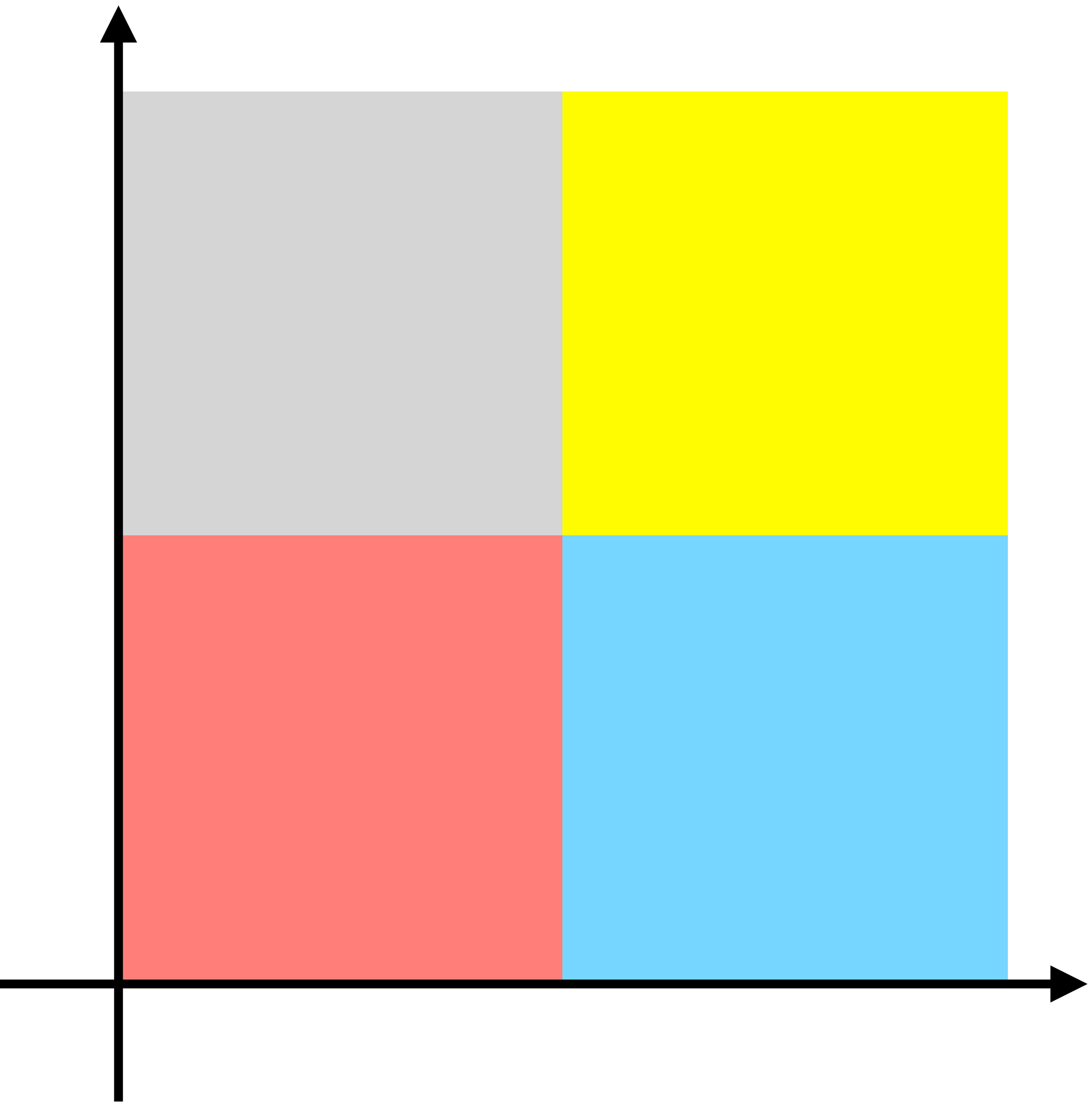


- For each abstract state, select one controller partition $P^\star$ from CPWA_φ

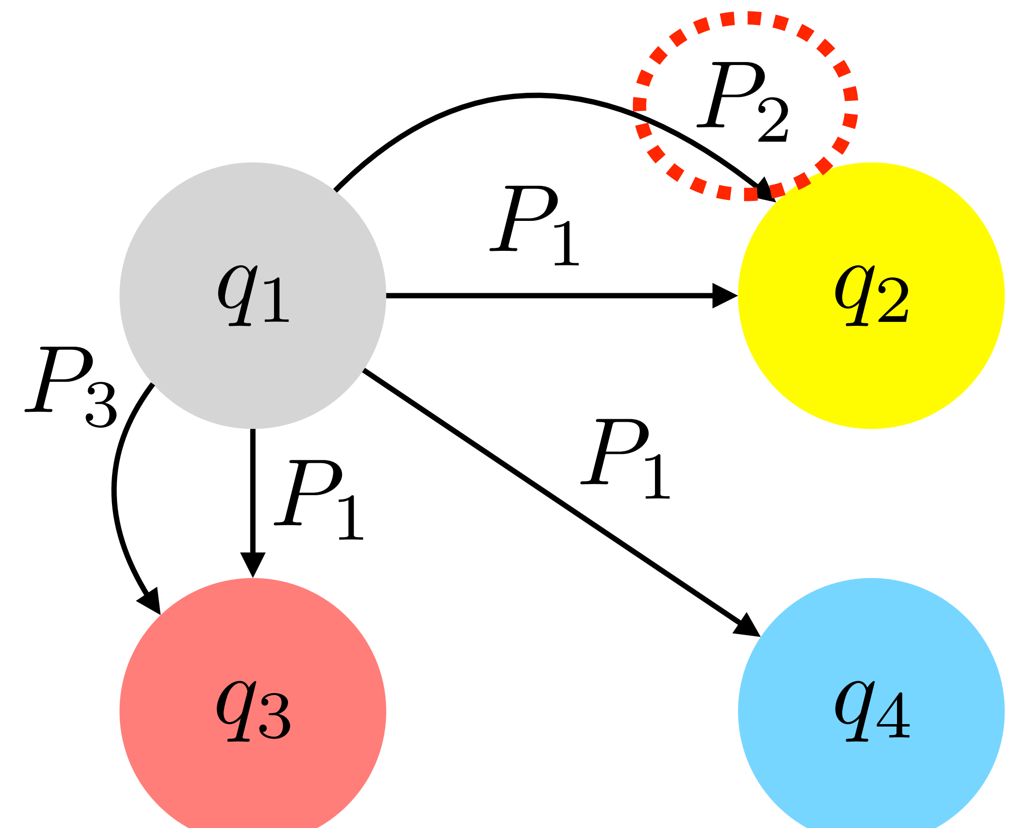


step 2

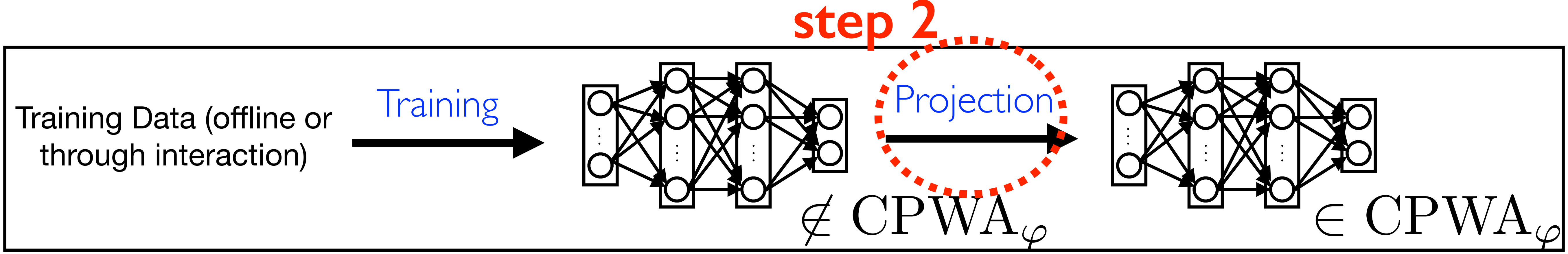


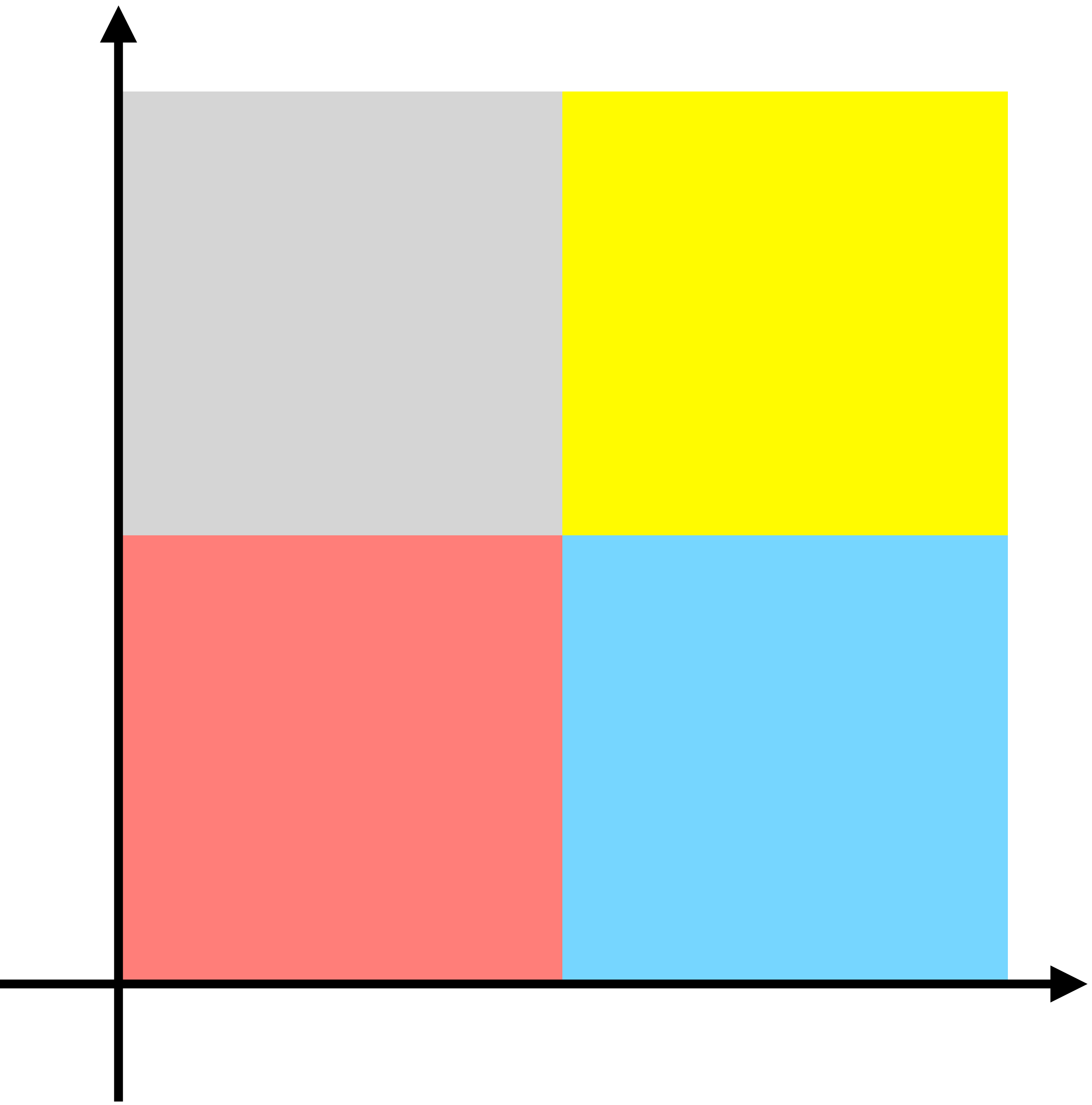


- For each abstract state, select one controller partition $P^\star$ from CPWA_φ



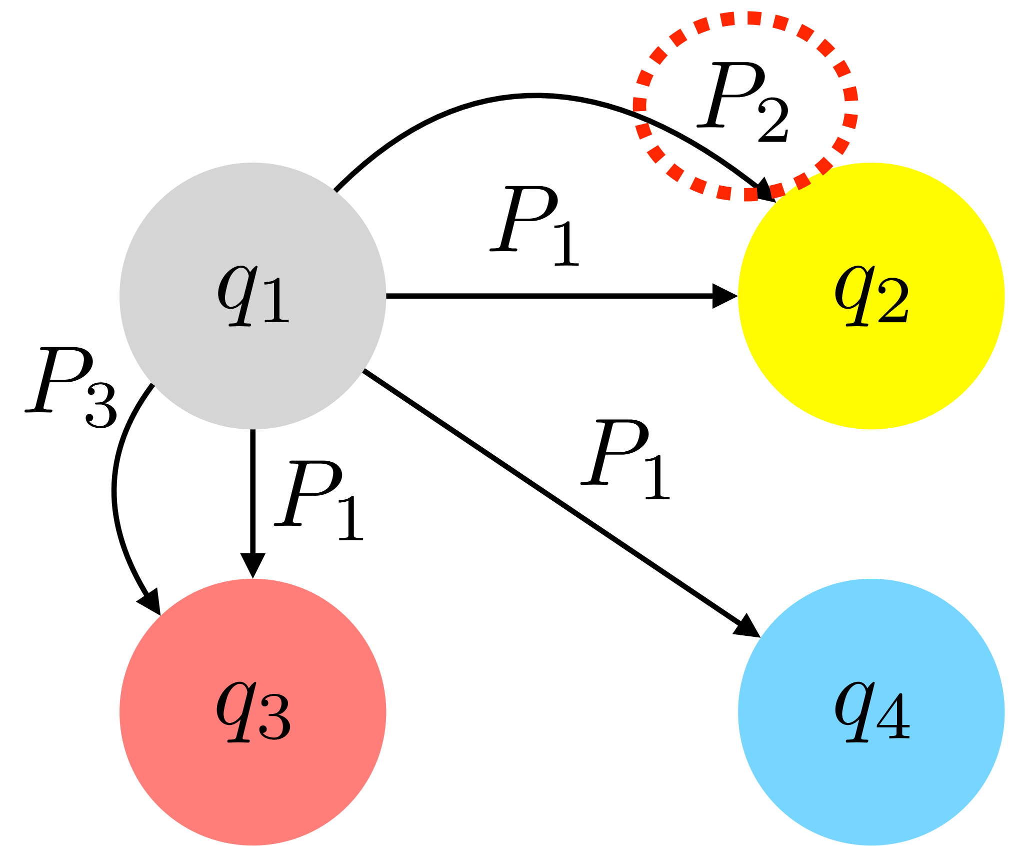
Resilient CPS Lab

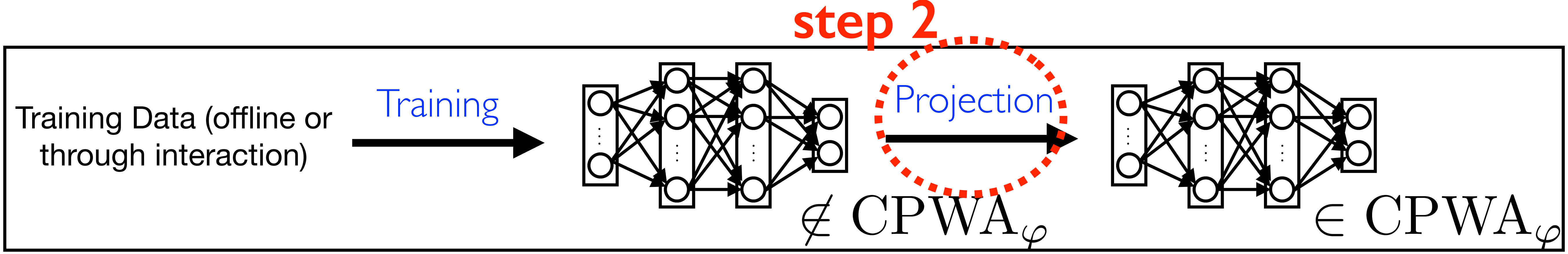


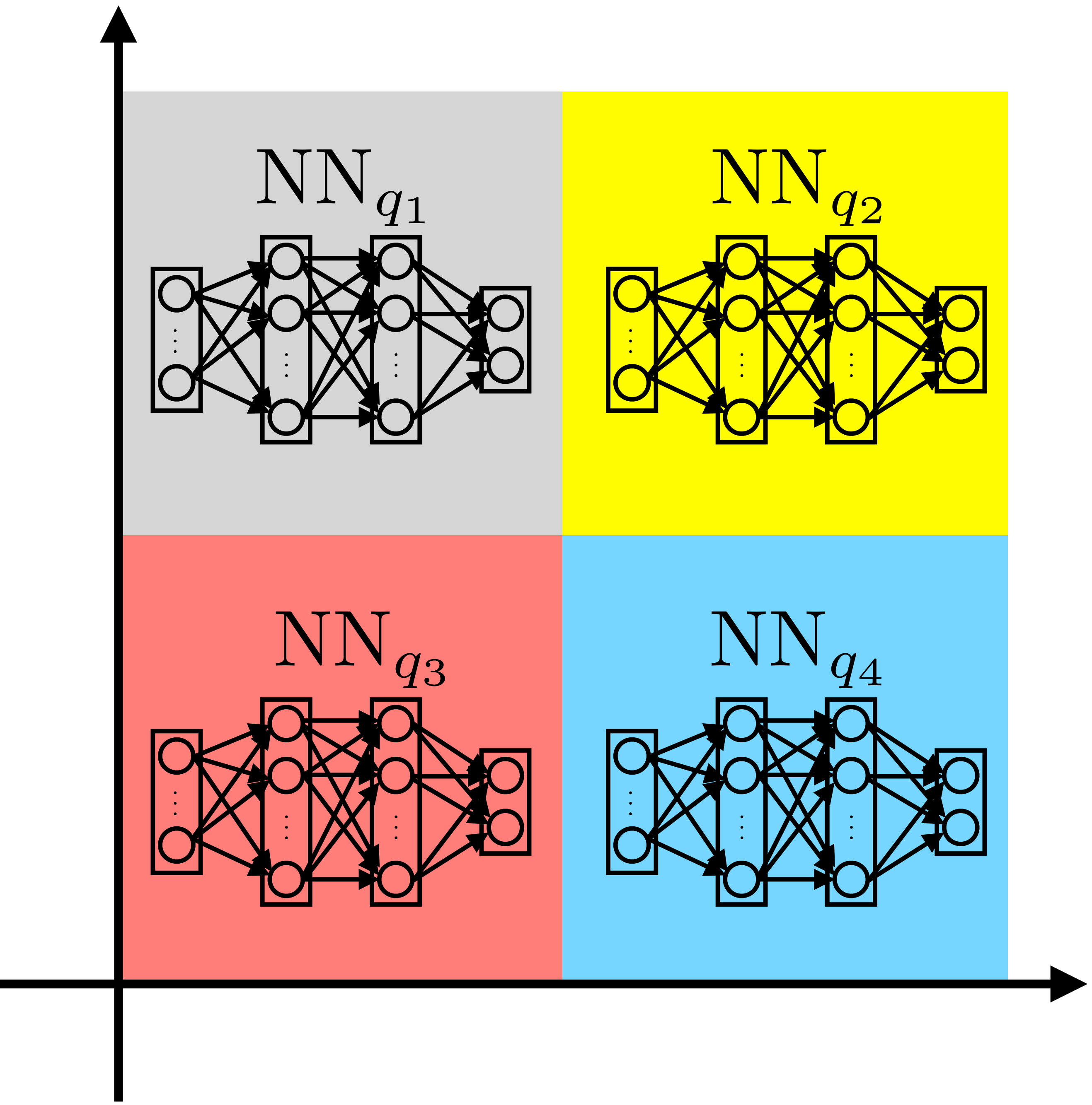


- For each abstract state, select one controller partition $P^\star$ from CPWA_φ

- Train one "local" neural network NN_q for each abstract state.

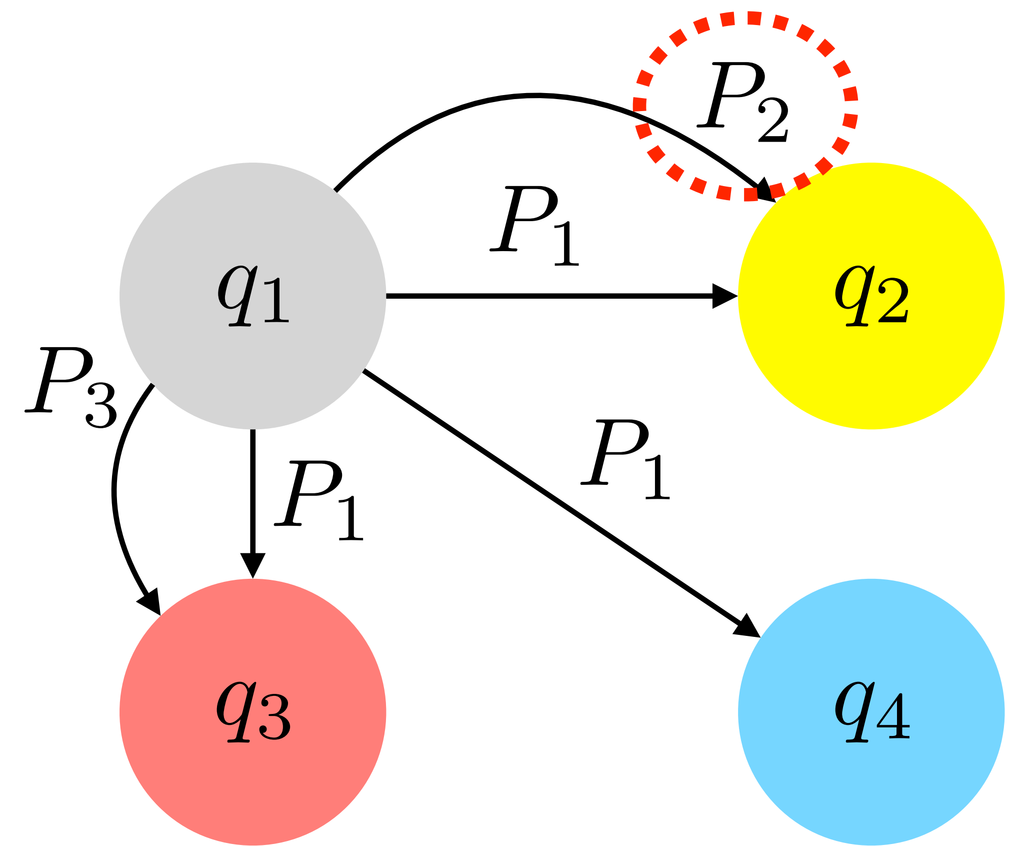


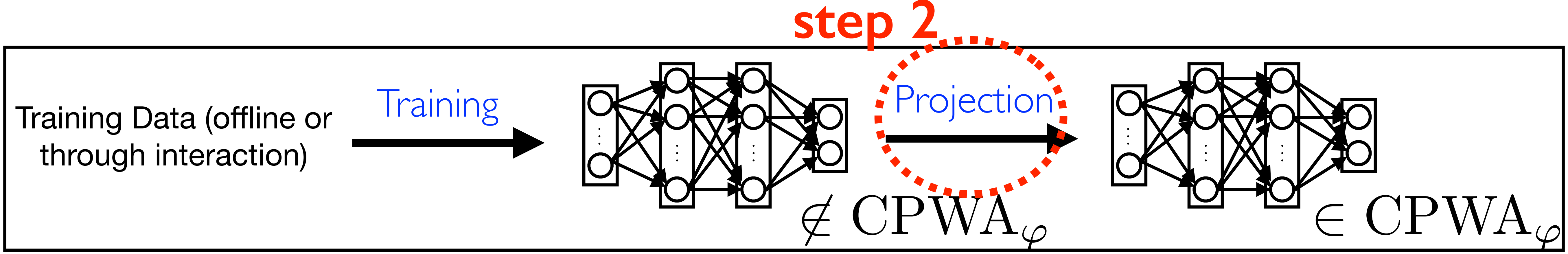


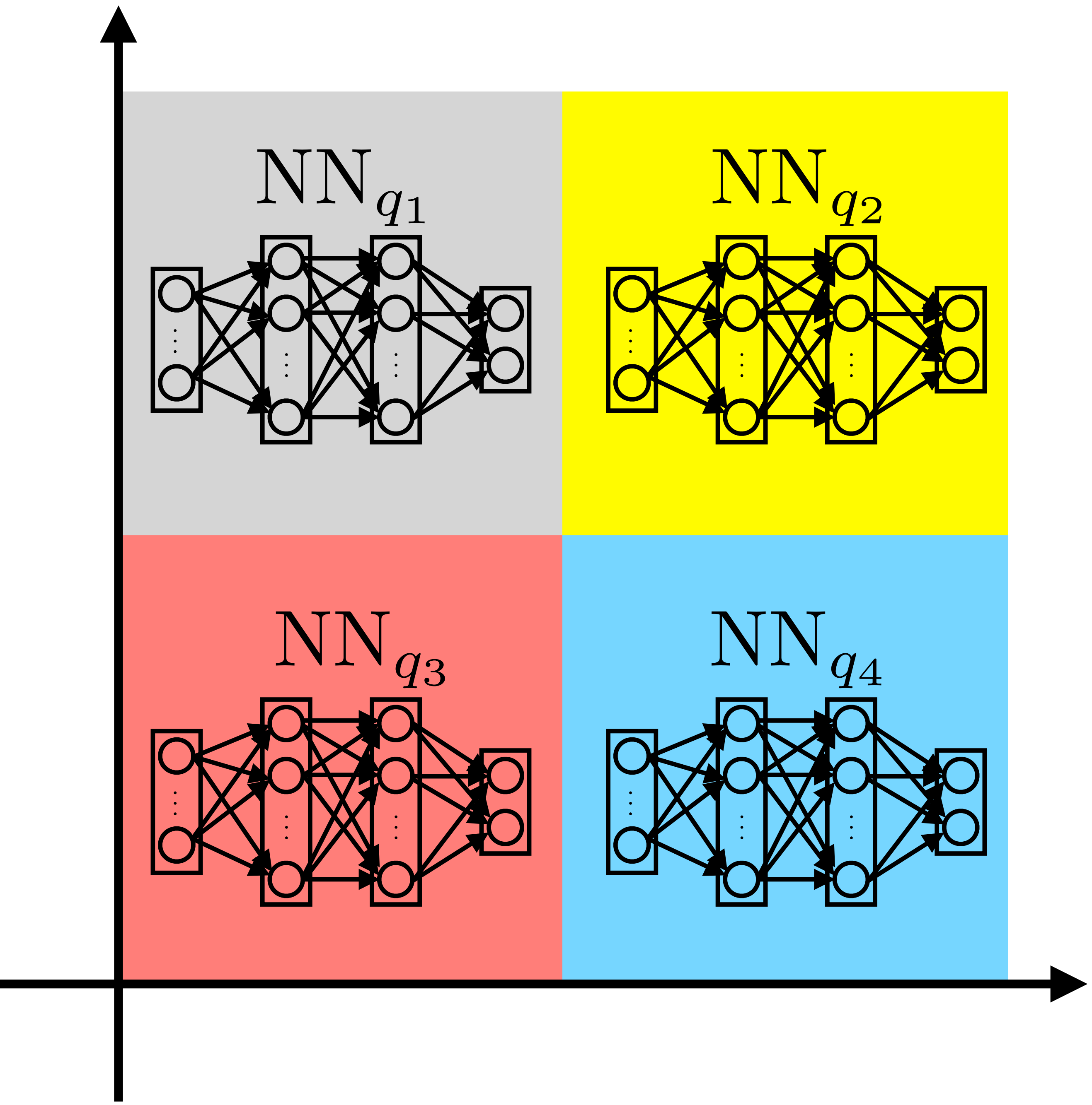


- For each abstract state, select one controller partition $P^\star$ from CPWA_φ

- Train one "local" neural network NN_q for each abstract state.

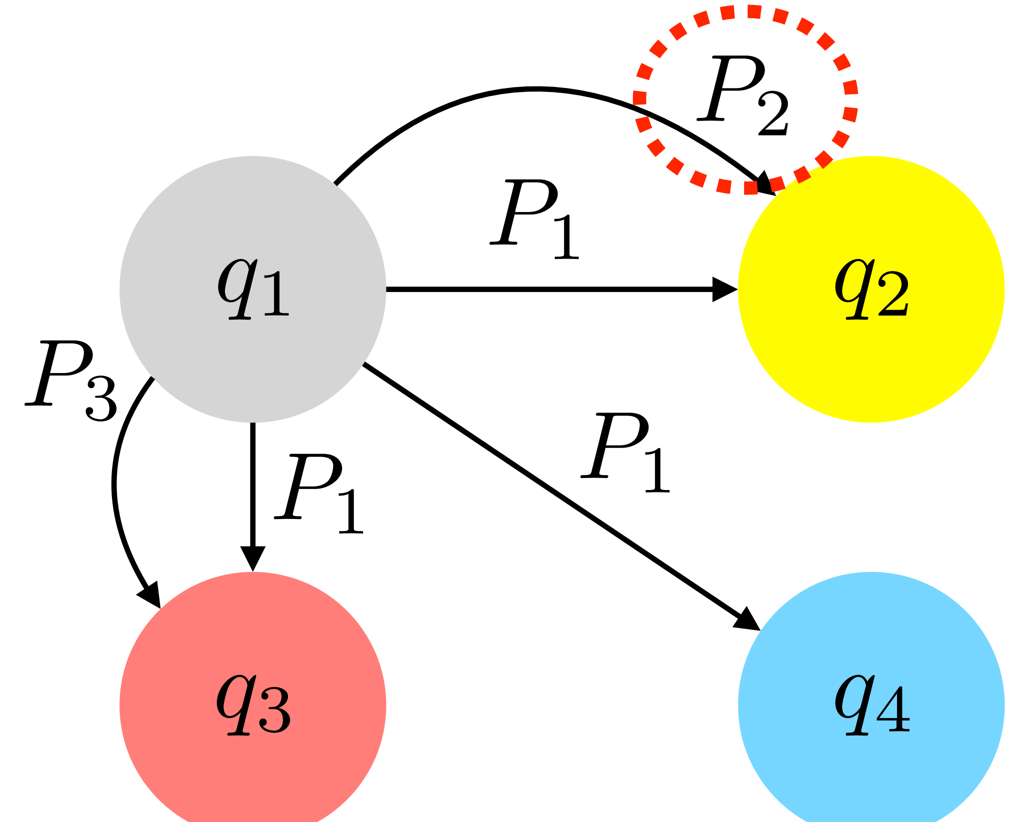




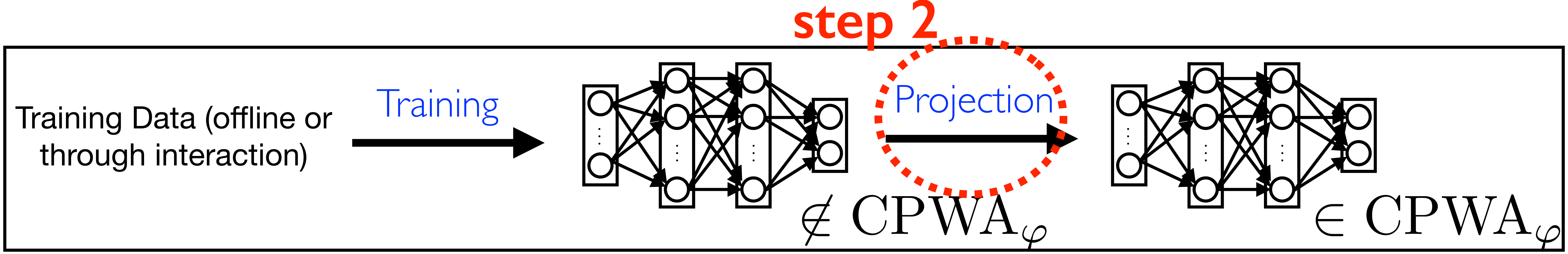


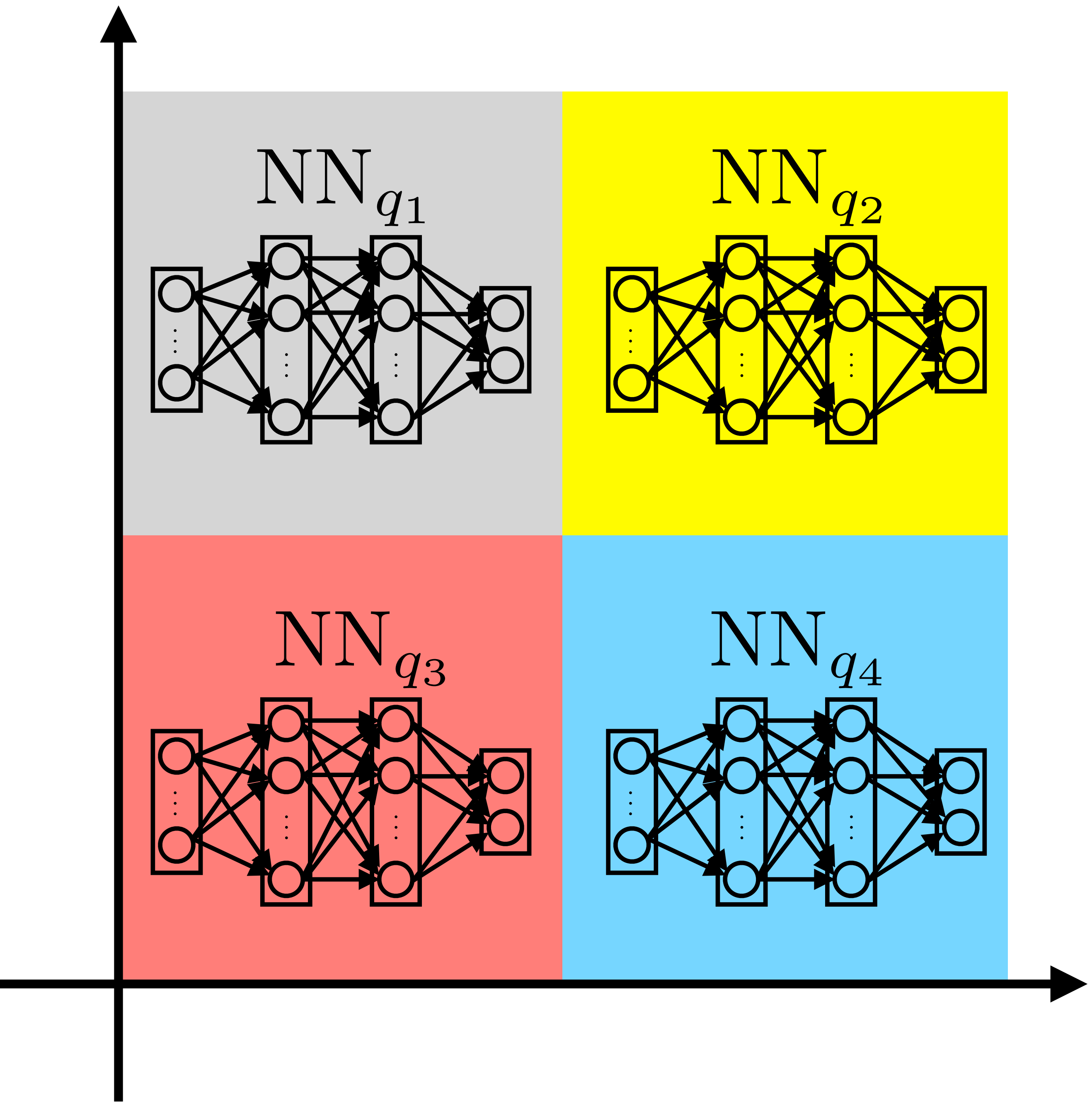
- For each abstract state, select one controller partition $P^\star$ from CPWA_φ

- Train one "local" neural network NN_q for each abstract state. Either using offline data (imitation learning) or interaction with the environment (Reinforcement learning)



Resilient CPS Lab

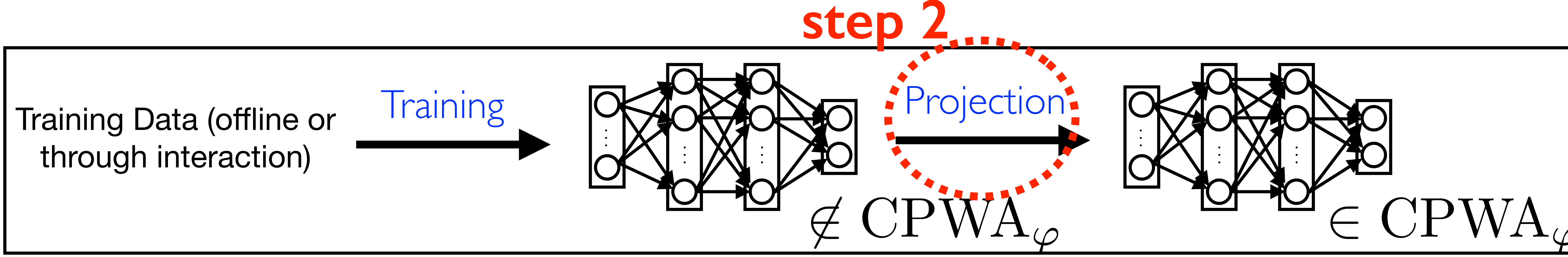


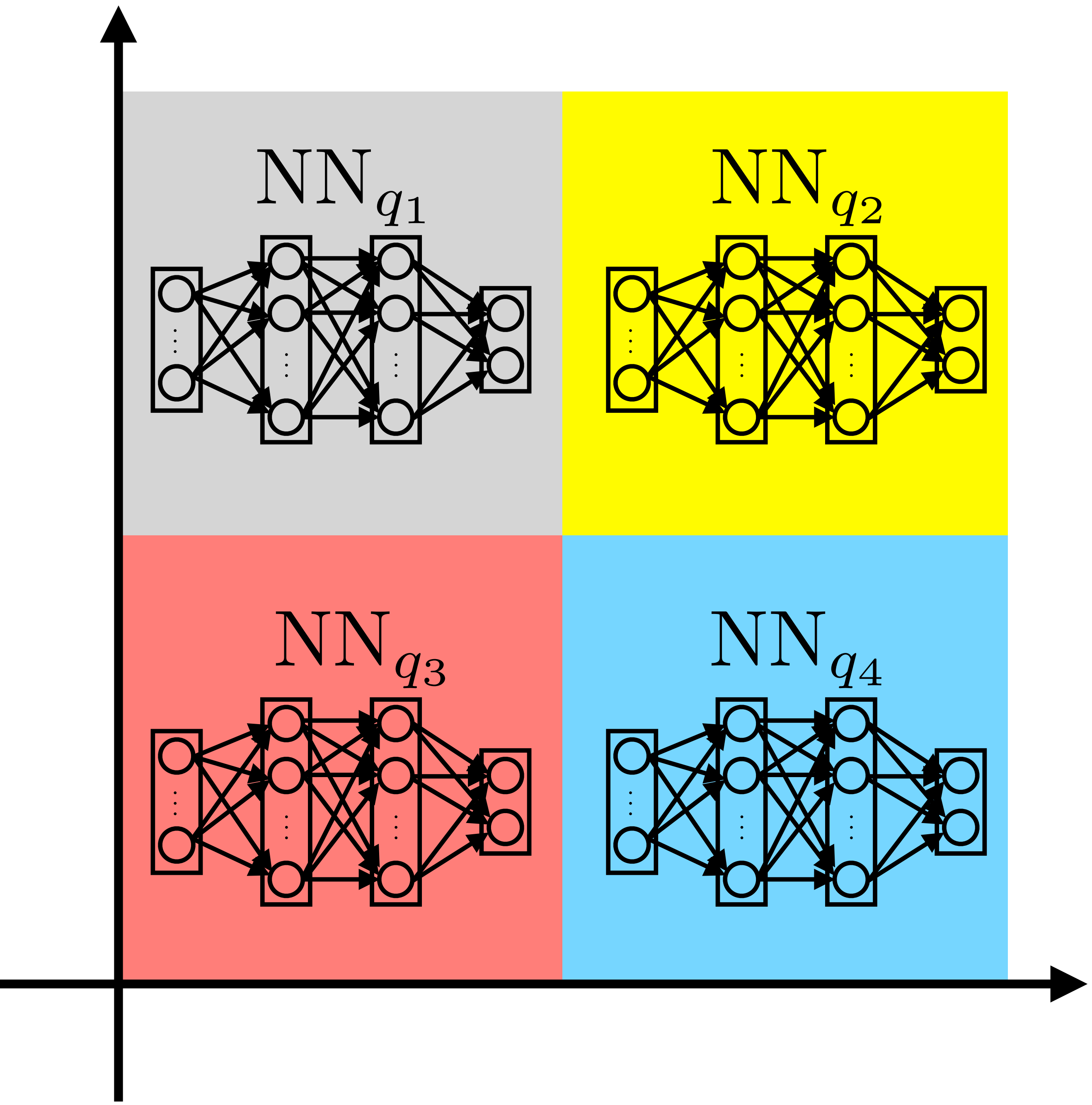


- For each abstract state, select one controller partition $P^\star$ from CPWA_φ

- Train one "local" neural network NN_q for each abstract state. Either using offline data (imitation learning) or interaction with the environment (Reinforcement learning)

- Enumerate all "affine" functions (K_i, b_i) in each local NN. Can be done efficiently since local NN are typically small.





- For each abstract state, select one controller partition $P^\star$ from CPWA_φ

- Train one "local" neural network NN_q for each abstract state. Either using offline data (imitation learning) or interaction with the environment (Reinforcement learning)

- Enumerate all "affine" functions (K_i, b_i) in each local NN. Can be done efficiently since local NN are typically small.

- Projection:

$$\min_{\widehat{W}} \|W - \widehat{W}\|$$

$$\text{s.t.} \quad (K_i, b_i) \in P^\star \quad \forall\ (K_i, b_i) \in \mathrm{NN}_q$$

(convex optimization problem if done layer-by-layer)

CPS Lab Resilient

Theorem (informal):

Consider the nonlinear system $x^+ = f(x, u)$ and a safety specification φ. Define a "global" neural network controller as the composition of "local" neural network controllers:

$$\mathrm{NN} = \mathrm{NN}_{q_1} || \mathrm{NN}_{q_2} || \dots \mathrm{NN}_{q_n}$$

Then:

$$f(x, \mathrm{NN}(x)) \models \varphi$$

X. Sun and Y. Shoukry, "Provably Correct Training of Neural Network Controllers Using Reachability Analysis," arXiv 2021.

Theorem (informal):

Consider the nonlinear system $x^+ = f(x,u)$ and a safety specification φ. Define a “global” neural network controller as the composition of “local” neural network controllers:

$$\mathrm{NN} = \mathrm{NN}_{q_1} || \mathrm{NN}_{q_2} || \ldots \mathrm{NN}_{q_n}$$

Then:

$$f(x, \mathrm{NN}(x)) \models \varphi$$

X. Sun and Y. Shoukry, "Provably Correct Training of Neural Network Controllers Using Reachability Analysis," arXiv 2021.

The result holds for liveness specifications under an additional assumption.

$$\zeta_x^{(t+\Delta t)} = \zeta_x^{(t)} + \Delta t \ v \ \cos(\theta^{(t)})$$
$$\zeta_y^{(t+\Delta t)} = \zeta_y^{(t)} + \Delta t \ v \ \sin(\theta^{(t)})$$
$$\theta^{(t+\Delta t)} = \theta^{(t)} + \Delta t \ u^{(t)}$$

$$\zeta_x^{(t+\Delta t)} = \zeta_x^{(t)} + \Delta t \ v \ \cos(\theta^{(t)})$$
$$\zeta_y^{(t+\Delta t)} = \zeta_y^{(t)} + \Delta t \ v \ \sin(\theta^{(t)})$$
$$\theta^{(t+\Delta t)} = \theta^{(t)} + \Delta t \ u^{(t)}$$

- Safe data collected and used for training
- Same data used in both experiments

$$\zeta_x^{(t+\Delta t)} = \zeta_x^{(t)} + \Delta t \; v \; \cos(\theta^{(t)})$$
$$\zeta_y^{(t+\Delta t)} = \zeta_y^{(t)} + \Delta t \; v \; \sin(\theta^{(t)})$$
$$\theta^{(t+\Delta t)} = \theta^{(t)} + \Delta t \; u^{(t)}$$

- Safe data collected and used for training
- Same data used in both experiments

Formal NN Training

NN Training

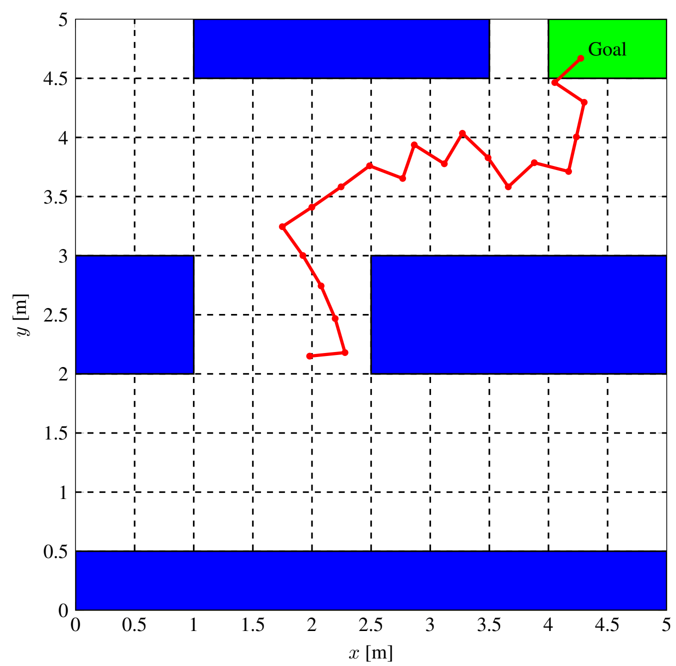
$$\zeta_x^{(t+\Delta t)} = \zeta_x^{(t)} + \Delta t \ v \ \cos(\theta^{(t)})$$

$$\zeta_y^{(t+\Delta t)} = \zeta_y^{(t)} + \Delta t \ v \ \sin(\theta^{(t)})$$

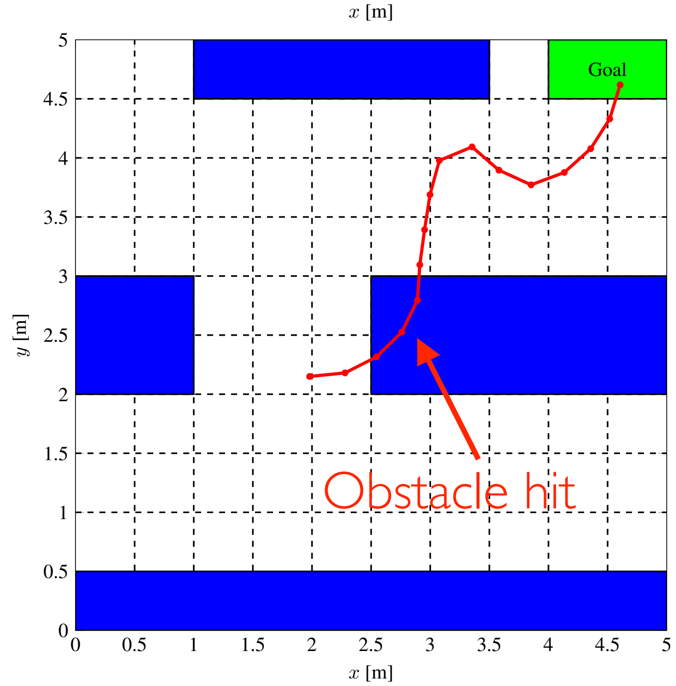
$$\theta^{(t+\Delta t)} = \theta^{(t)} + \Delta t \ u^{(t)}$$

- Safe data collected and used for training
- Same data used in both experiments

Formal NN Training



NN Training



CPS Lab
Resilient

$$\zeta_x^{(t+\Delta t)} = \zeta_x^{(t)} + \Delta t \; v \; \cos(\theta^{(t)})$$

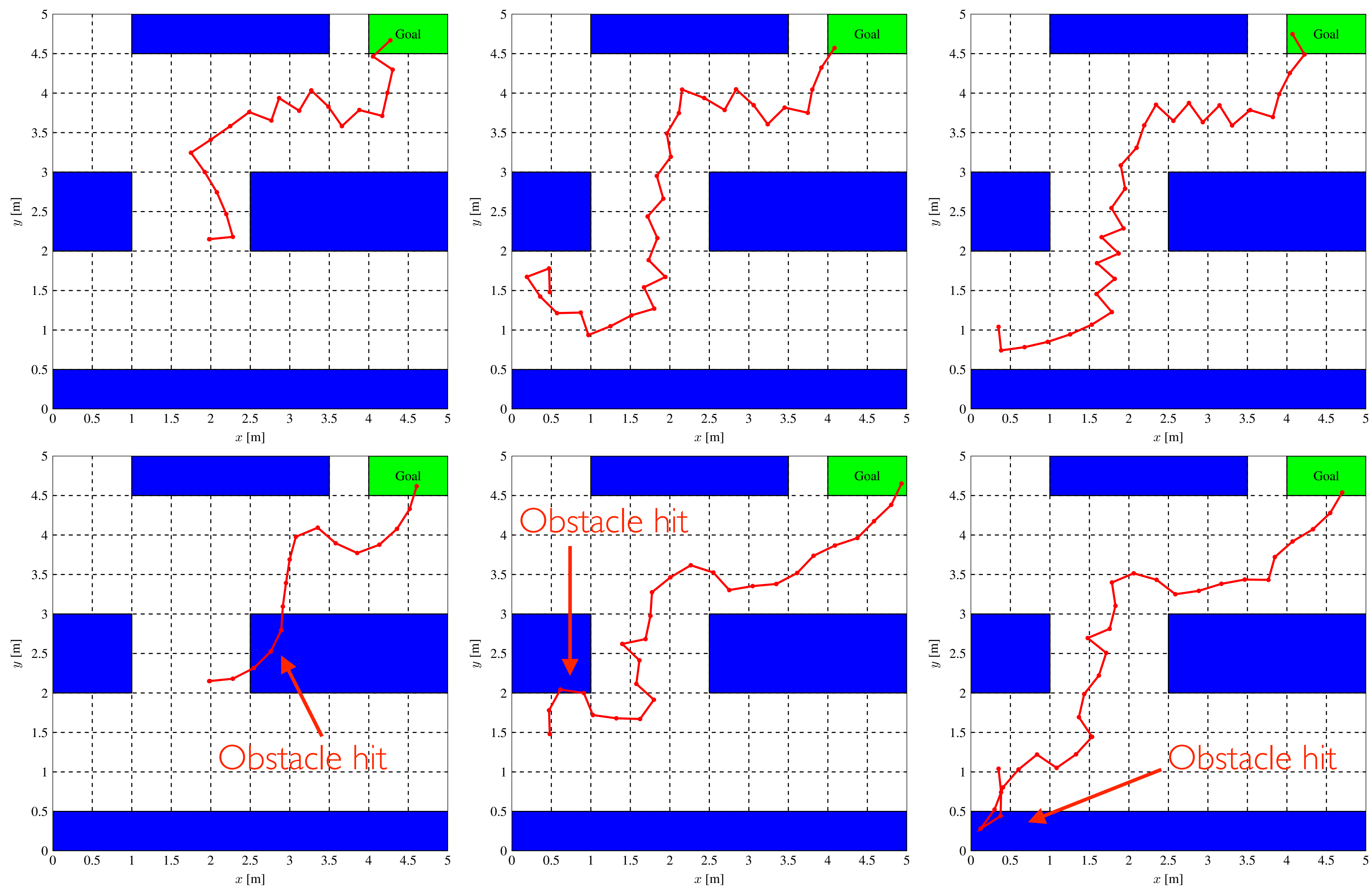
$$\zeta_y^{(t+\Delta t)} = \zeta_y^{(t)} + \Delta t \; v \; \sin(\theta^{(t)})$$

$$\theta^{(t+\Delta t)} = \theta^{(t)} + \Delta t \; u^{(t)}$$

- Safe data collected and used for training
- Same data used in both experiments

Formal NN Training

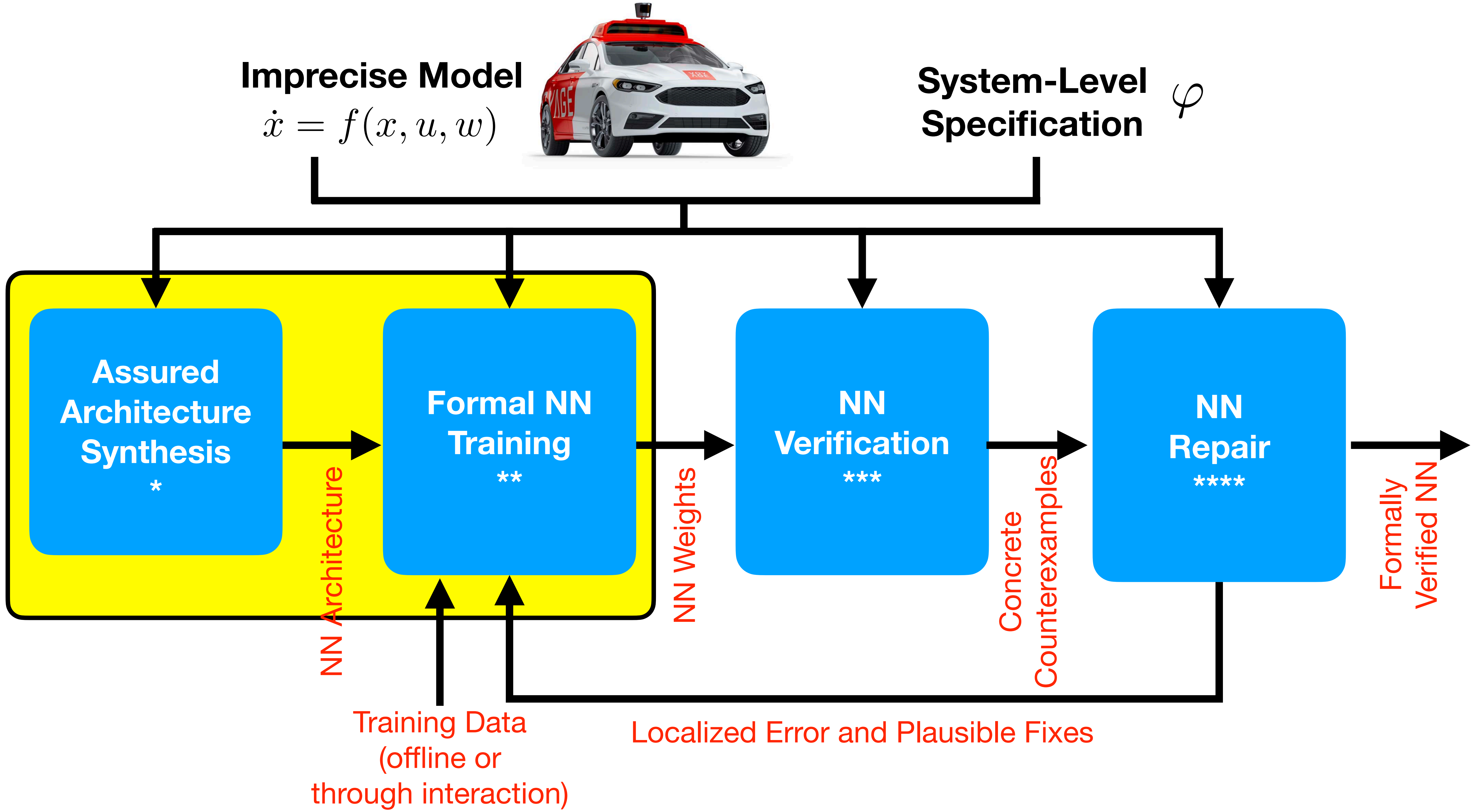
NN Training



Resilient CPS Lab

Workspace Index	Number of Abstract States	Number of Controller Partitions	Number of Safe & Reachable Abstract States	Compute Reachable Sets [s]	Construct Posterior Graph [s]	Compute Function P_{safe} [s]	Assign Controller Partitions [s]
1	552	160	400	52.6	82.3	0.06	0.7
1	552	320	400	107.5	160.3	0.1	0.9
1	552	640	400	223.1	329.6	0.2	1.7
1	1104	160	800	108.2	333.0	0.2	2.3
1	1104	320	800	219.6	684.2	0.4	2.7
1	1104	640	800	451.5	1297.4	0.6	4.2
2	904	160	632	88.1	159.1	0.1	1.0
2	904	320	632	203.6	313.2	0.2	1.1
2	904	640	632	393.2	660.8	0.3	1.7
2	1808	160	1264	202.1	634.6	0.3	3.4
2	1808	320	1264	388.6	1298.1	0.6	4.0
2	1808	640	1264	778.2	2564.4	0.9	5.9

System Dimension n	Number of Abstract States	Compute Reachable Sets [s]	Construct Posterior Graph [s]
2	69	0.6	0.7
4	276	2.7	2.6
6	1104	11.7	34.2
8	4416	57.1	521.0
10	17664	258.1	9840.4



* J. Ferlez, X. Sun, and Y. Shoukry, "Two-Level Lattice Neural Network Architectures for Control of Nonlinear Systems," CDC 2020.

* J. Ferlez and Y. Shoukry, "AReN: Assured ReLU NN Architecture for Model Predictive Control of LTI Systems," HSCC, 2020.

** X. Sun and Y. Shoukry, "Provably Correct Training of Neural Network Controllers Using Reachability Analysis," arXiv 2021.

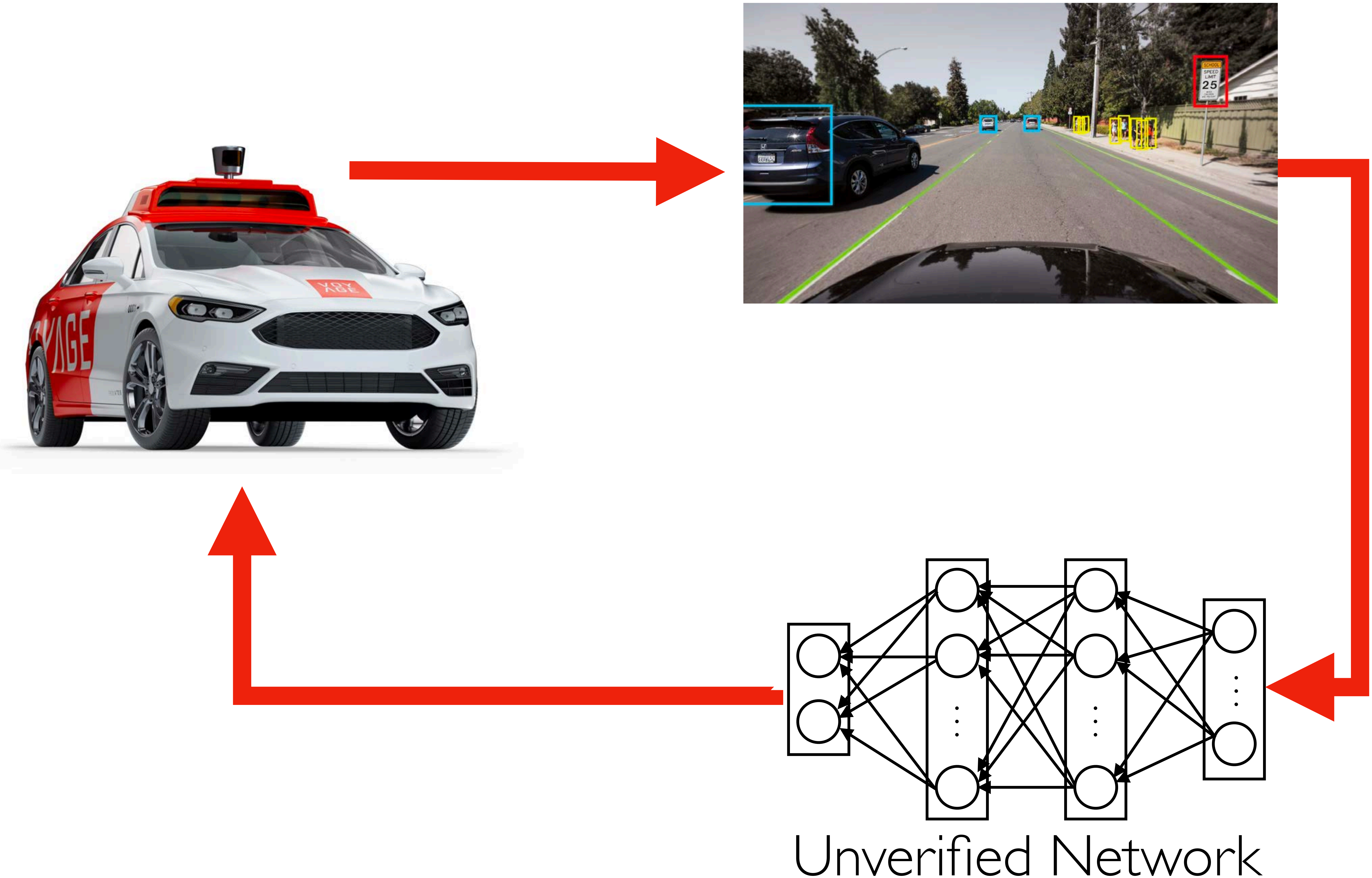
** X. Sun, W. Fatnassi, U. Santa Cruz, and Y. Shoukry, "Provably Safe Model-Based Meta Reinforcement Learning: An Abstraction-Based Approach," arXiv 2021.

*** H. Khedr, J. Ferlez, and Y. Shoukry, "PEREGRiNN: Penalized-Relaxation Greedy Neural Network Verifier,"CAV, 2021.

*** J. Ferlez and Y. Shoukry, "Bounding the Complexity of Formally Verifying Neural Networks: A Geometric Approach," arXiv 2020.

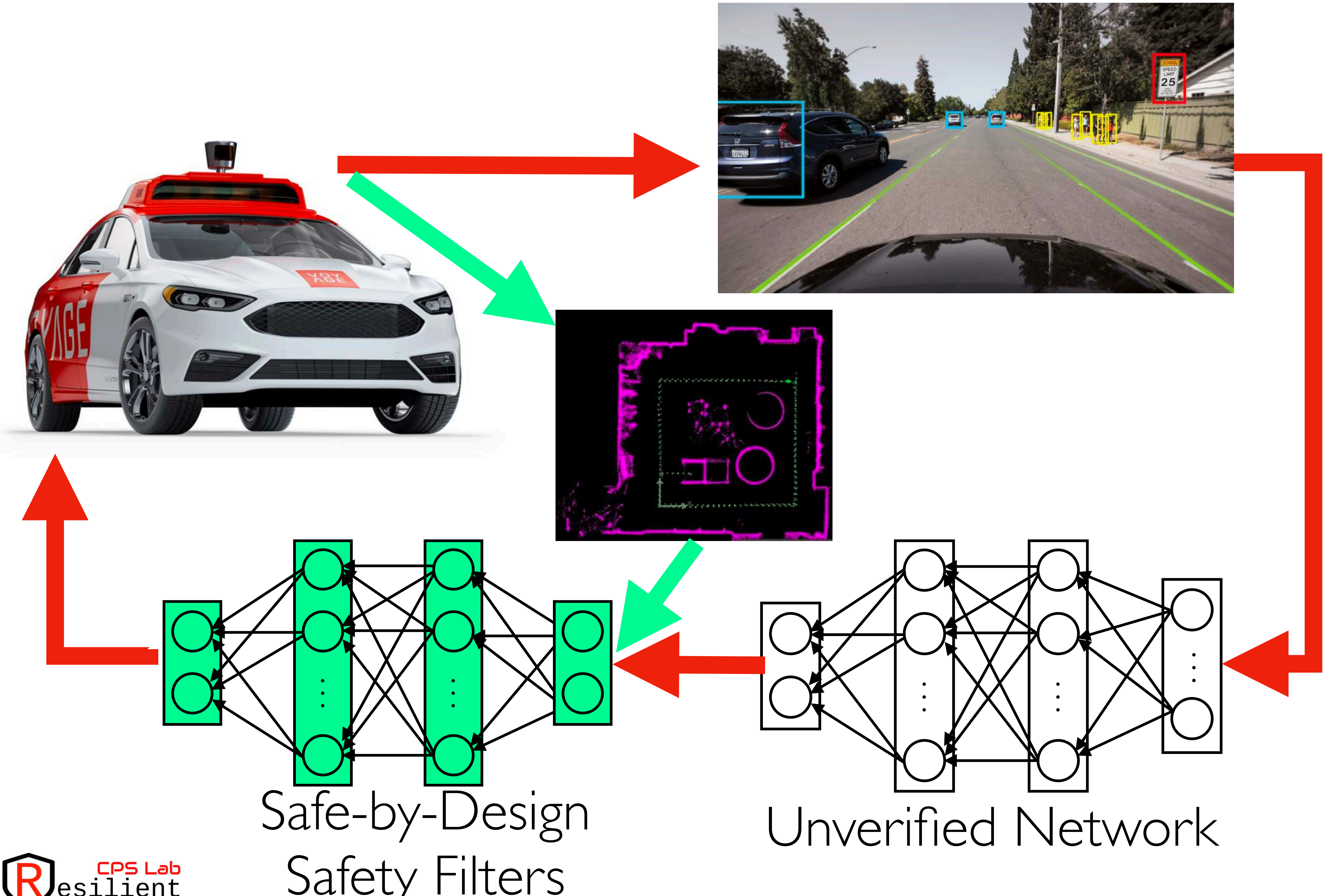
**** U. Santa Cruz, J. Ferlez, and Y. Shoukry, "Safe-by-Repair: A Convex Optimization Approach for Repairing Unsafe Two-Level Lattice Neural Network Controllers," arXiv 2021.

Synthesis of NN-based Safety Filters



Resilient CPS Lab

Synthesis of NN-based Safety Filters



Resilient CPS Lab

Synthesis of NN-based Safety Filters

Synthesis of NN-based Safety Filters

ROBUSTNESS DEMO #1

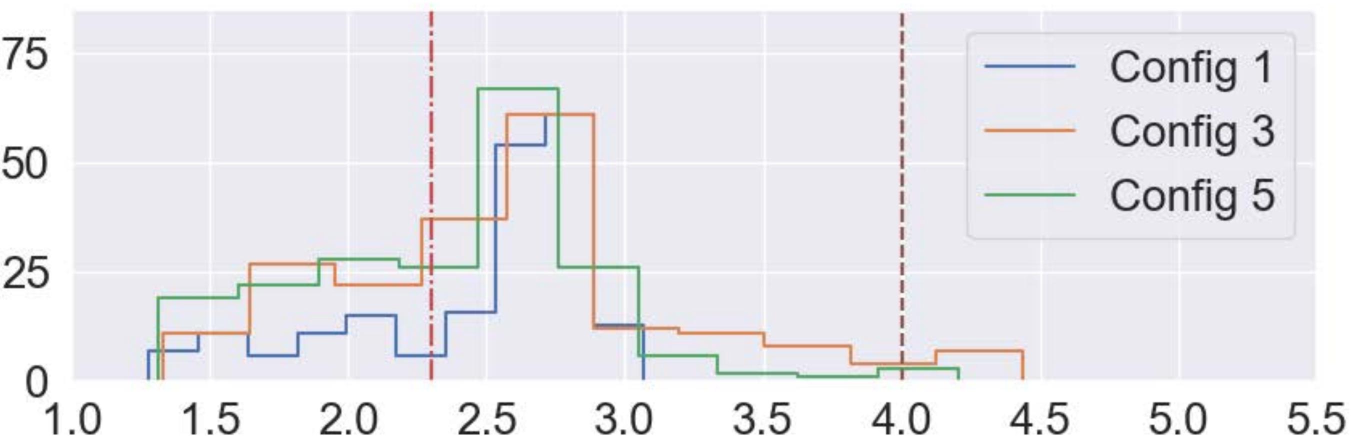
Agent #2

ShieldNN ON

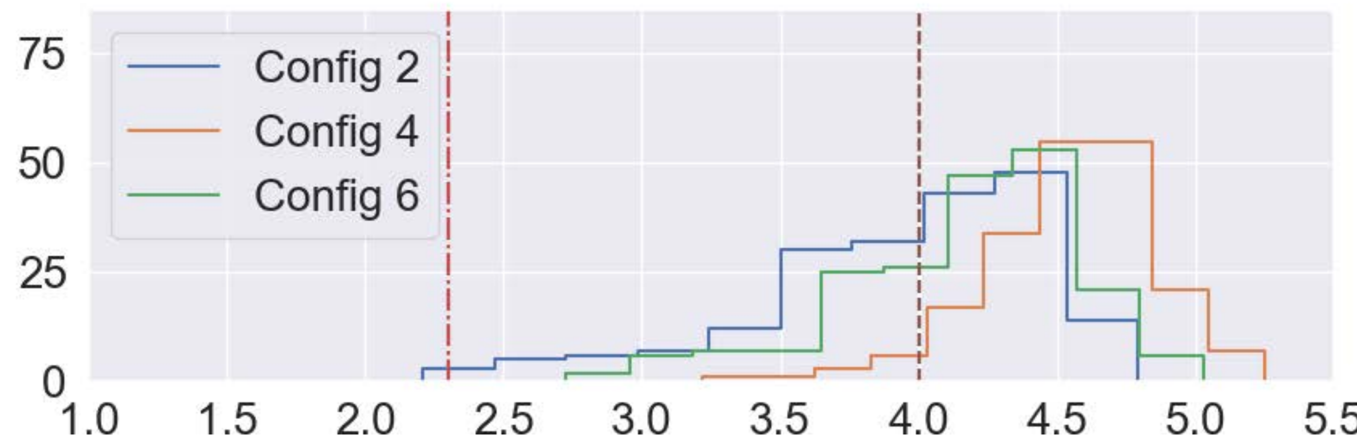
Agent #3

CPS Lab Resilient

Synthesis of NN-based Safety Filters



Without Root-of-Trust Network



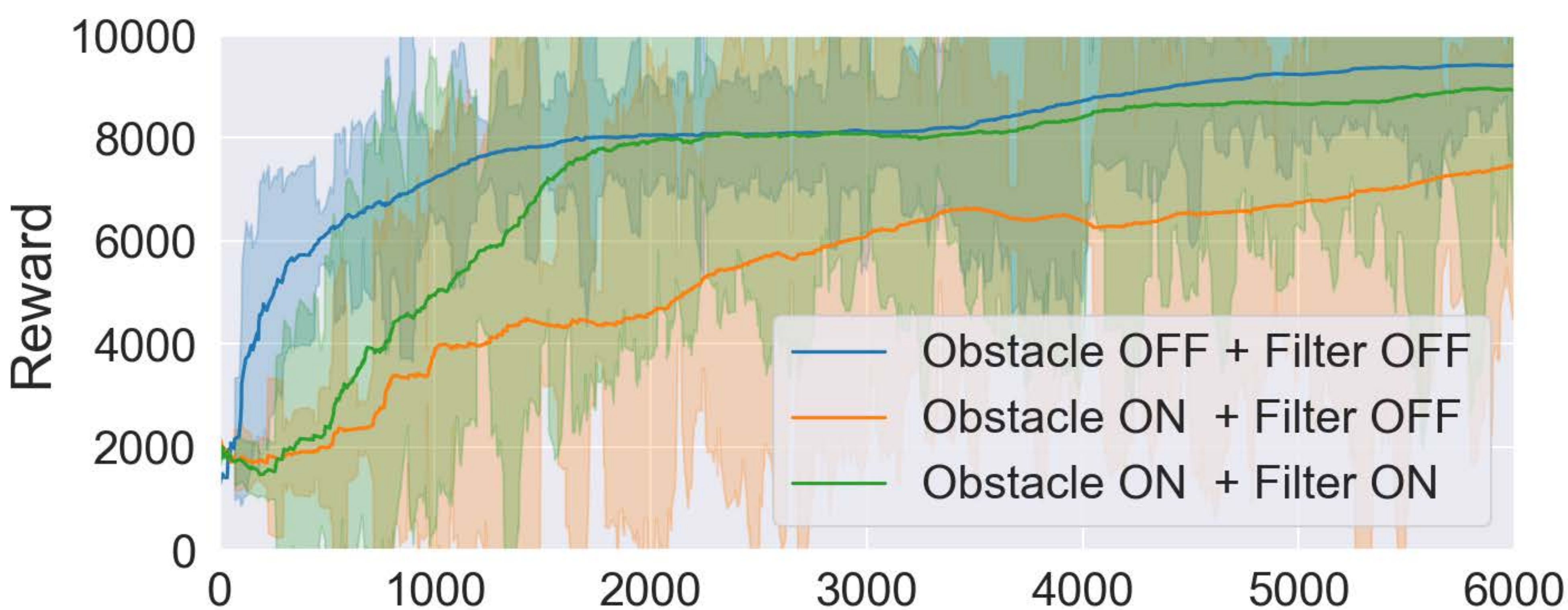
With Root-of-Trust Network

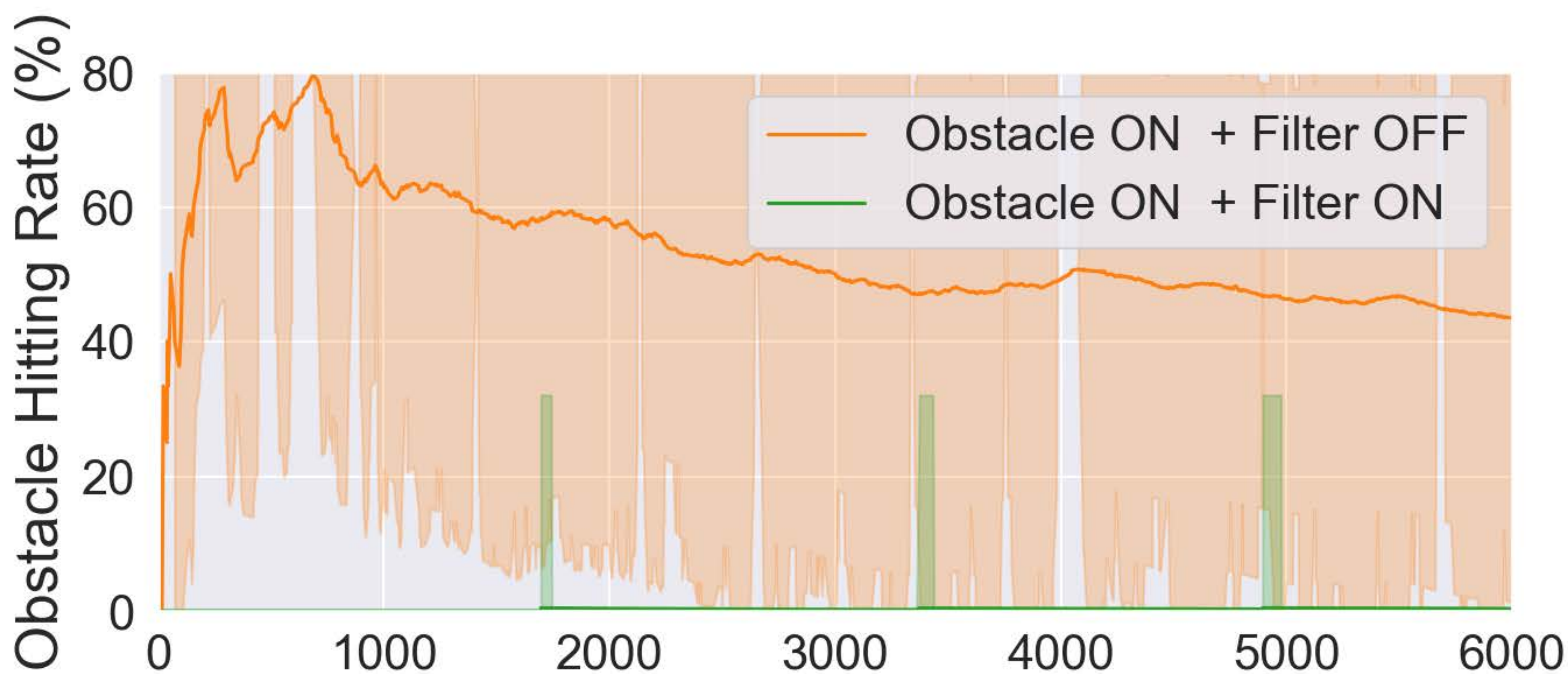
	Training		Testing	Experiment 1		Experiment 2	
Config	**Obstacle**	**Filter**	**Filter**	**TC%**[1]	**OHR%**[2]	**TC%**[1]	**OHR%**[2]
1	OFF	OFF	OFF	7.59	99.5	27.53	79.5
2	OFF	OFF	ON	98.82	0.5	98.73	0.5
3	ON	OFF	OFF	94.82	8.5	71.88	34
4	ON	OFF	ON	100	0	100	0
5	ON	ON	OFF	62.43	44	50.03	60
6	ON	ON	ON	100	0	100	0

[1] TC% := Track Completion % [2] OHR% := Obstacle Hit Rate %

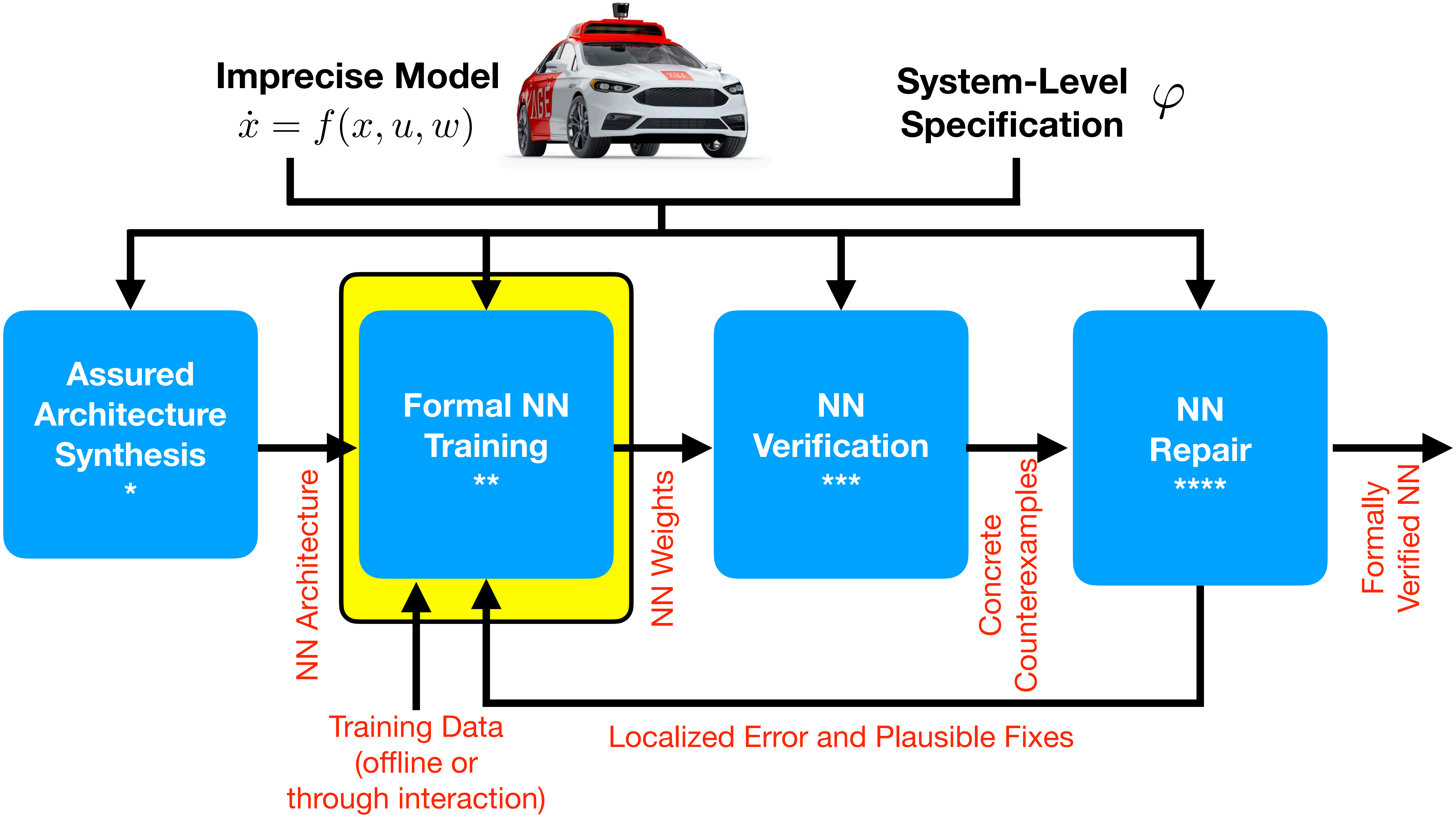
- J. Ferlez, M. Raffat, C. Fleming, and Y. Shoukry, "ShieldNN: A Provably Safe NN Filter for Unsafe NNControllers", ArXiv 2020.

Synthesis of NN-based Safety Filters





CPS Lab
Resilient



* J. Ferlez, X. Sun, and Y. Shoukry, "Two-Level Lattice Neural Network Architectures for Control of Nonlinear Systems," CDC 2020.

* J. Ferlez and Y. Shoukry, "AReN: Assured ReLU NN Architecture for Model Predictive Control of LTI Systems," HSCC, 2020.

** X. Sun and Y. Shoukry, "Provably Correct Training of Neural Network Controllers Using Reachability Analysis," arXiv 2021.

** X. Sun, W. Fatnassi, U. Santa Cruz, and Y. Shoukry, "Provably Safe Model-Based Meta Reinforcement Learning: An Abstraction-Based Approach," arXiv 2021.

*** H. Khedr, J. Ferlez, and Y. Shoukry, "PEREGRiNN: Penalized-Relaxation Greedy Neural Network Verifier,"CAV, 2021.

*** J. Ferlez and Y. Shoukry, "Bounding the Complexity of Formally Verifying Neural Networks: A Geometric Approach," arXiv 2020.

**** U. Santa Cruz, J. Ferlez, and Y. Shoukry, "Safe-by-Repair: A Convex Optimization Approach for Repairing Unsafe Two-Level Lattice Neural Network Controllers," arXiv 2021.

Thanks !

Xiaowu Sun

Haitham Khedr

Wael Fatnassi

Ulices Santa Cruz Leal

Momina Sajid

Dr. James Ferlez

NSF

NORTHROP GRUMMAN

CPS Lab Resilient