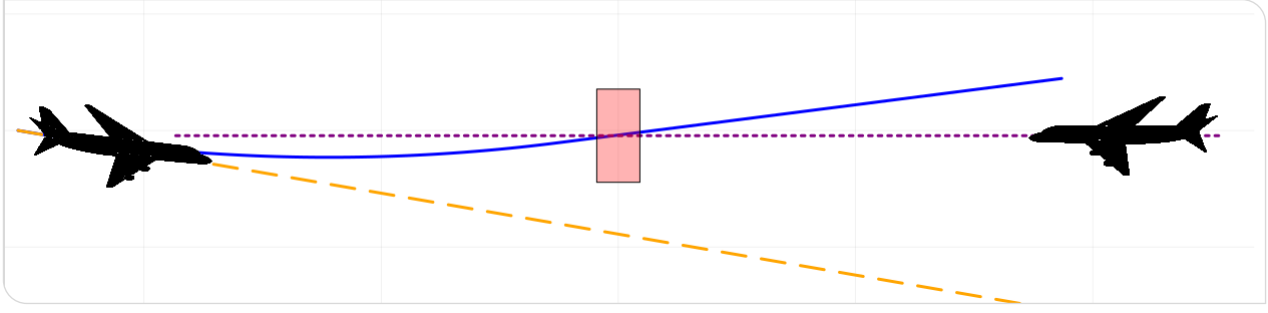
Karlsruhe Institute of Technology

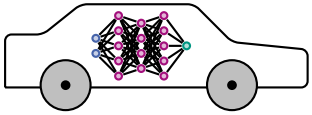
Provably Safe Neural Network Controllers via Differential Dynamic Logic

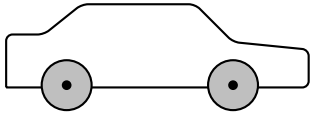
Symposium on AI Verification 2024

Samuel Teuber, Stefan Mitsch, André Platzer | 2024

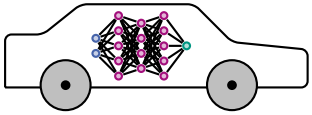


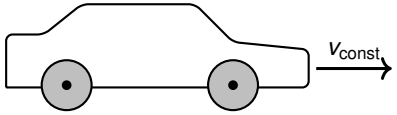
Motivation: Neural Network Control Systems



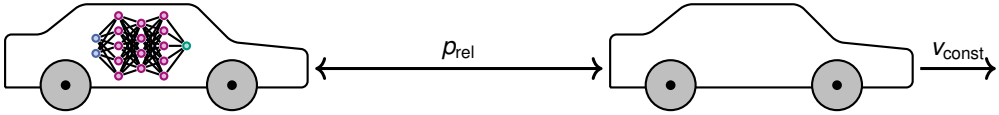


Motivation: Neural Network Control Systems

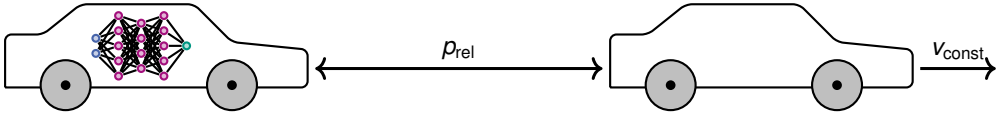




Motivation: Neural Network Control Systems



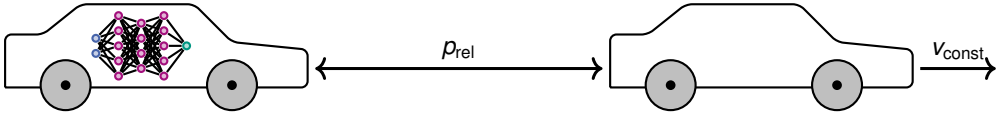
Motivation: Neural Network Control Systems



$$p'_{rel} = v_{\text{rel}}$$
$$v'_{rel} = -a_{rel} = -g\,(p_{rel}, v_{rel})$$

How can we prove the safety of a strategy g?

Motivation: Neural Network Control Systems



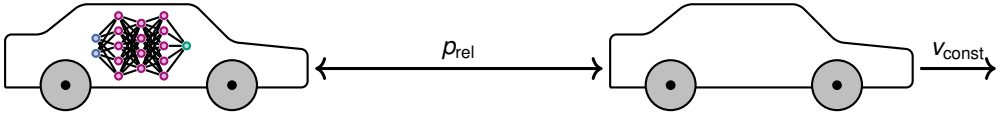
$$p'_{rel} = v_{\text{rel}}$$
$$v'_{rel} = -a_{rel} = -g\,(p_{rel}, v_{rel})$$

How can we prove the safety of a strategy g?

$g\,(p_{rel}, v_{rel}) = -B < 0$

Safe if cars start far enough apart (depends on B).

Motivation: Neural Network Control Systems



$$p'_{rel} = v_{\text{rel}}$$
$$v'_{rel} = -a_{rel} = -g\,(p_{rel}, v_{rel})$$

How can we prove the safety of a strategy g?

$g\,(p_{rel}, v_{rel}) = -B < 0$

Safe if cars start far enough apart (depends on B).

Secondary objectives

- Follow front car
- Passenger comfort/Energy efficiency

Objective

Given:

- A **safe** differential dynamic logic model of the system with a **controller** α_{ctrl}
- A **neural network controller** g

Objective

Given:

- A **safe** differential dynamic logic model of the system with a **controller** α_{ctrl}
- A **neural network controller** g

Question:

If we **replace** the control envelope α_{ctrl} by the NN g,
does the resulting system retain the same safety guarantees?

Differential Dynamic Logic by Example

We will use $d\mathcal{L}$ as the tool to prove the safety of cyber-physical systems.

[Platzer 2008]

Differential Dynamic Logic by Example

We will use $d\mathcal{L}$ as the tool to prove the safety of cyber-physical systems.

hybrid program

[Platzer 2008]

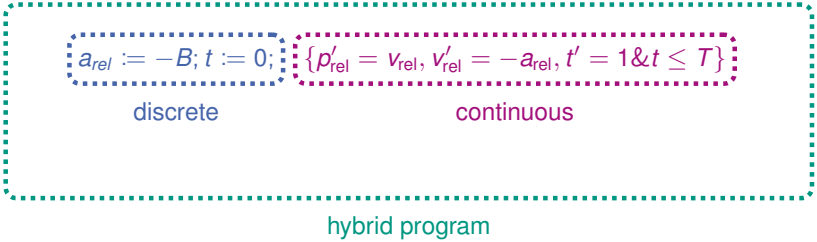
Differential Dynamic Logic by Example

We will use $d\mathcal{L}$ as the tool to prove the safety of cyber-physical systems.

[Platzer 2008]

Differential Dynamic Logic by Example

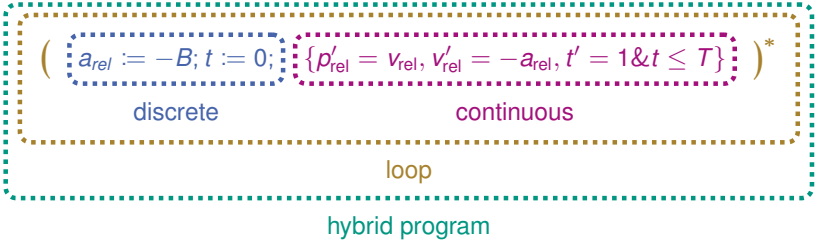
We will use $d\mathcal{L}$ as the tool to prove the safety of cyber-physical systems.



[Platzer 2008]

Differential Dynamic Logic by Example

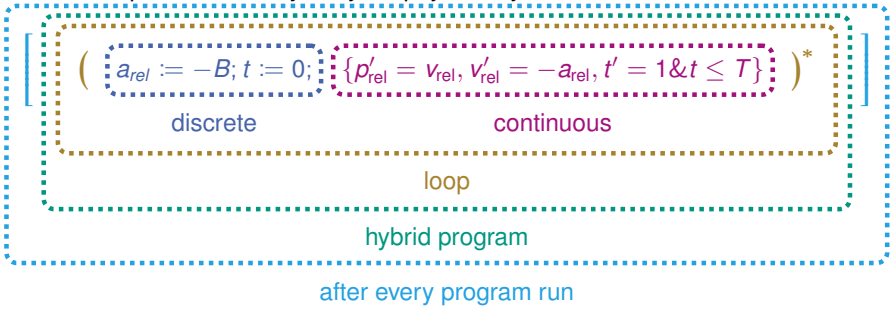
We will use $d\mathcal{L}$ as the tool to prove the safety of cyber-physical systems.



[Platzer 2008]

Differential Dynamic Logic by Example

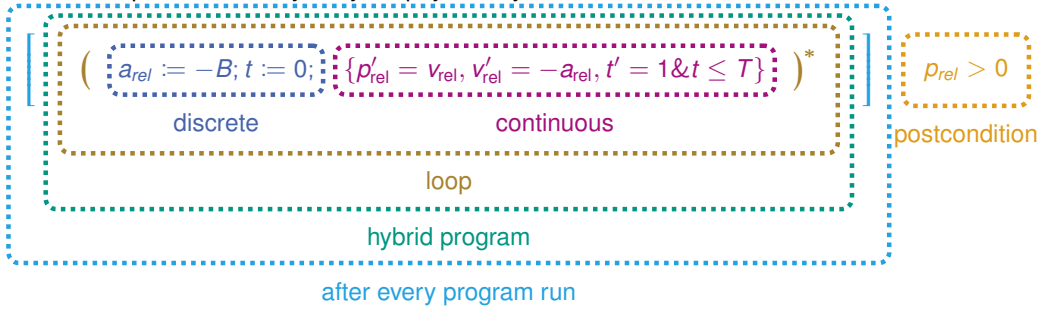
We will use $d\mathcal{L}$ as the tool to prove the safety of cyber-physical systems.



[Platzer 2008]

Differential Dynamic Logic by Example

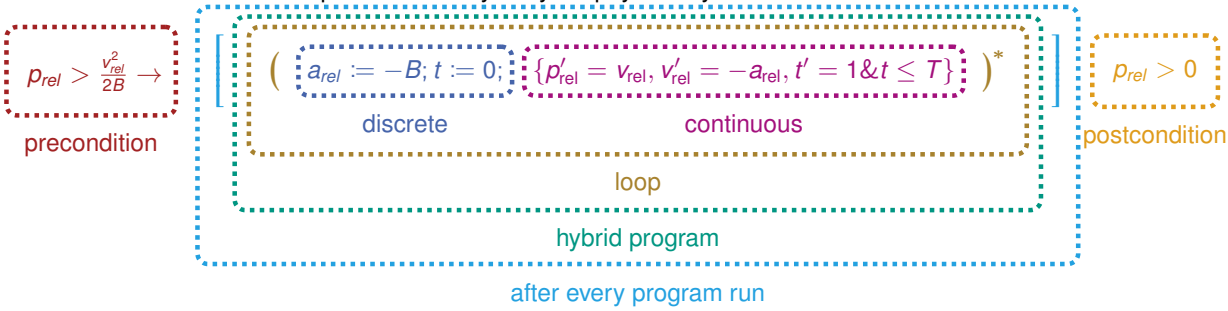
We will use $d\mathcal{L}$ as the tool to prove the safety of cyber-physical systems.



[Platzer 2008]

Differential Dynamic Logic by Example

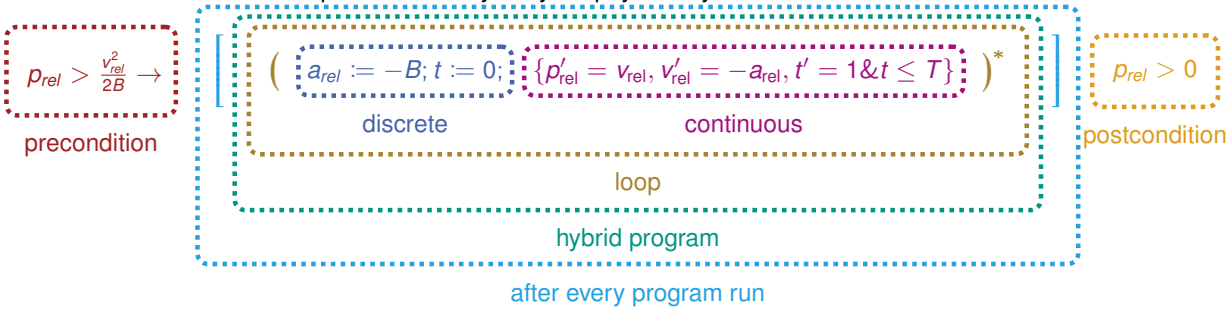
We will use $d\mathcal{L}$ as the tool to prove the safety of cyber-physical systems.



[Platzer 2008]

Differential Dynamic Logic by Example

We will use $d\mathcal{L}$ as the tool to prove the safety of cyber-physical systems.



We can prove safety through a proof calculus.

[Platzer 2008]

ModelPlex

Input: Safe Control Envelope in $d\mathcal{L}$

$$p_{\text{rel}} > \frac{v_{\text{rel}}^2}{2B} \rightarrow \left[\left(a_{\text{rel}} := -B; t := 0; \{p'_{\text{rel}} = 0, v'_{\text{rel}} = 0, t' = 1 \& t \leq T\}\right)^*\right] p_{\text{rel}} > 0$$

ModelPlex

Input: Safe Control Envelope in $d\mathcal{L}$ Controller/Envelope Plant

$$p_{\text{rel}} > \frac{v_{\text{rel}}^2}{2B} \rightarrow \left[\left(a_{\text{rel}} := -B; t := 0; \{p'_{\text{rel}} = 0, v'_{\text{rel}} = 0, t' = 1 \& t \leq T\}\right)^*\right] p_{\text{rel}} > 0$$

ModelPlex

Input: Safe Control Envelope in $d\mathcal{L}$ Controller/Envelope Plant

$$p_{\text{rel}} > \frac{v_{\text{rel}}^2}{2B} \rightarrow \left[\left(a_{\text{rel}} := -B; t := 0; \{p'_{\text{rel}} = 0, v'_{\text{rel}} = 0, t' = 1 \& t \leq T\}\right)^*\right] p_{\text{rel}} > 0$$

ModelPlex creates a **controller monitor formula**:

$$a_{\text{rel}}^+ = -B$$

Satisfaction during concrete run implies correct controller behavior.
Equally applicable for more complicated controllers (nondeterminism, conditions, ...)

[Mitsch and Platzer 2016]

ModelPlex

Input: Safe Control Envelope in $d\mathcal{L}$

$$p_{\text{rel}} > \frac{v_{\text{rel}}^2}{2B} \rightarrow \left[\left(a_{\text{rel}} := -B \cup \left(a_{\text{rel}} := *; ?\left(|a_{\text{rel}}| \leq B \wedge p_{\text{rel}} > 10^3\right)\right); \ \ldots\right)^*\right] p_{\text{rel}} > 0$$

ModelPlex creates a **controller monitor formula**:

Satisfaction during concrete run implies correct controller behavior.
Equally applicable for more complicated controllers (nondeterminism, conditions, ...)

[Mitsch and Platzer 2016]

ModelPlex

Input: Safe Control Envelope in $d\mathcal{L}$

$$p_{\text{rel}} > \frac{v_{\text{rel}}^2}{2B} \rightarrow \left[\left(a_{\text{rel}} := -B \cup \left(a_{\text{rel}} := *; ?\left(|a_{\text{rel}}| \leq B \wedge p_{\text{rel}} > 10^3\right)\right); \ \ldots\right)^*\right] p_{\text{rel}} > 0$$

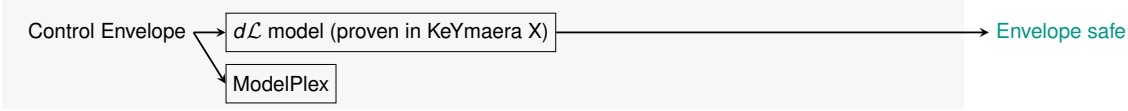
ModelPlex creates a **controller monitor formula**:

$$a_{\text{rel}}^+ = -B \vee \left|a_{\text{rel}}^+\right| \leq B \wedge p_{\text{rel}} > 10^3$$

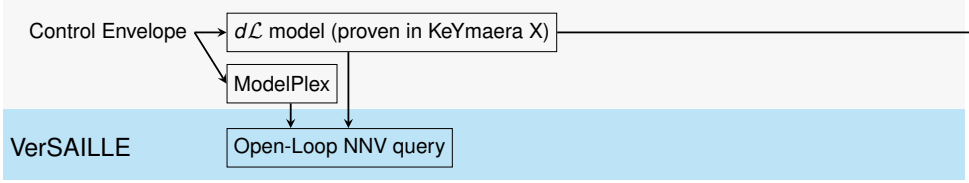
Satisfaction during concrete run implies correct controller behavior.
Equally applicable for more complicated controllers (nondeterminism, conditions, ...)

[Mitsch and Platzer 2016]

Contribution



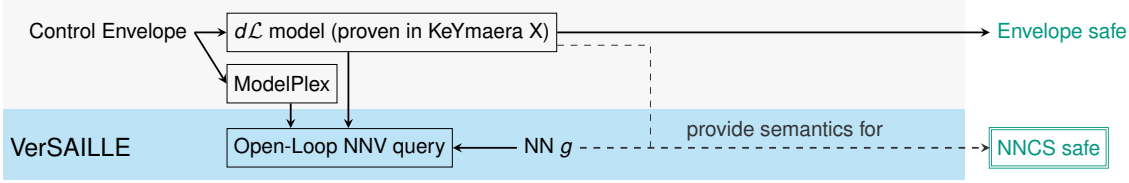
Contribution



VerSAILLE

Rigorous **infinite-time** horizon **safety** for **continuous-time** NNCS via $d\mathcal{L}$ & NN verification

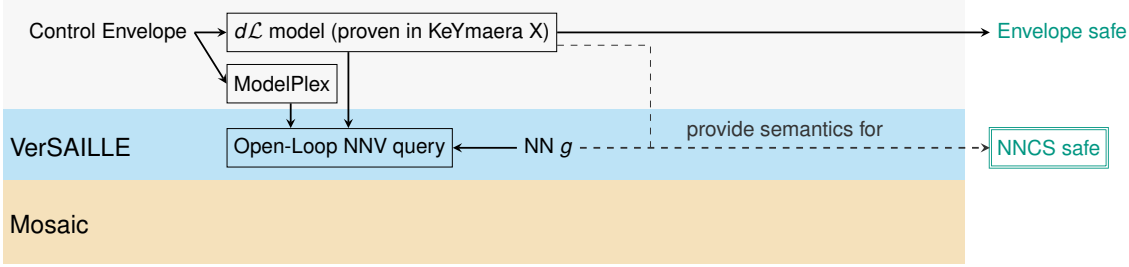
Contribution



VerSAILLE

Rigorous **infinite-time** horizon **safety** for **continuous-time** NNCS via $d\mathcal{L}$ & NN verification

Contribution



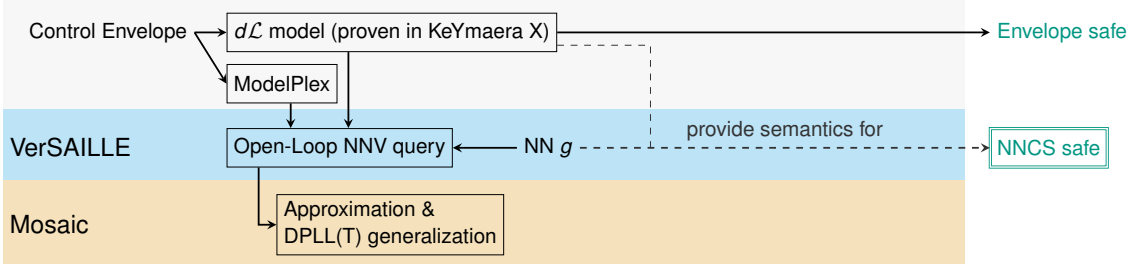
VerSAILLE

Rigorous **infinite-time** horizon **safety** for **continuous-time** NNCS via $d\mathcal{L}$ & NN verification

Mosaic

Sound, complete and efficient NN verification for **polynomial** constraints with **arbitrary propositional structure**.

Contribution



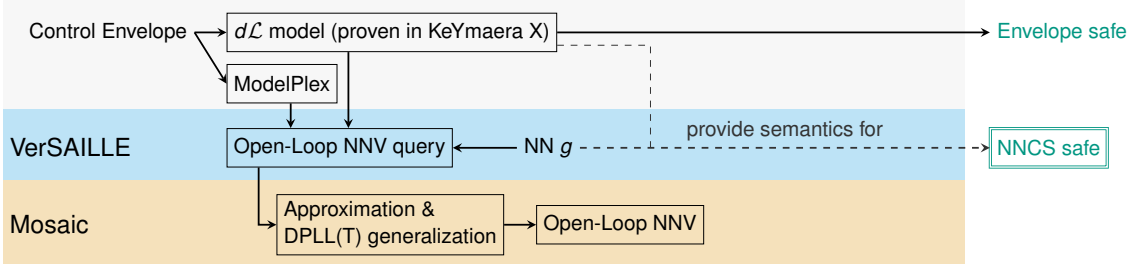
VerSAILLE

Rigorous **infinite-time** horizon **safety** for **continuous-time** NNCS via $d\mathcal{L}$ & NN verification

Mosaic

Sound, complete and efficient NN verification for **polynomial** constraints with **arbitrary propositional structure**.

Contribution



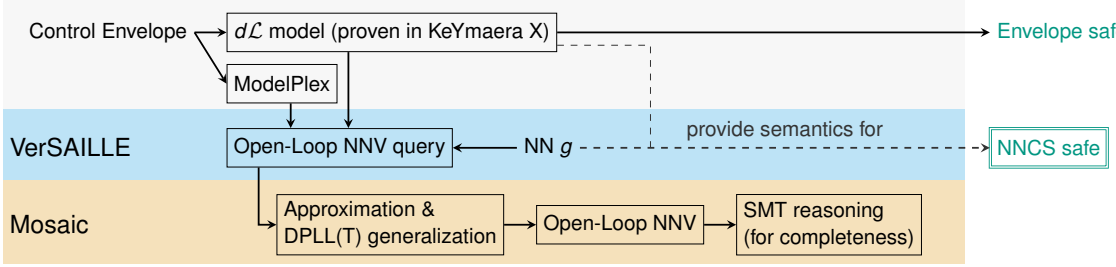
VerSAILLE

Rigorous **infinite-time** horizon **safety** for **continuous-time** NNCS via $d\mathcal{L}$ & NN verification

Mosaic

Sound, complete and efficient NN verification for **polynomial** constraints with **arbitrary propositional structure**.

Contribution



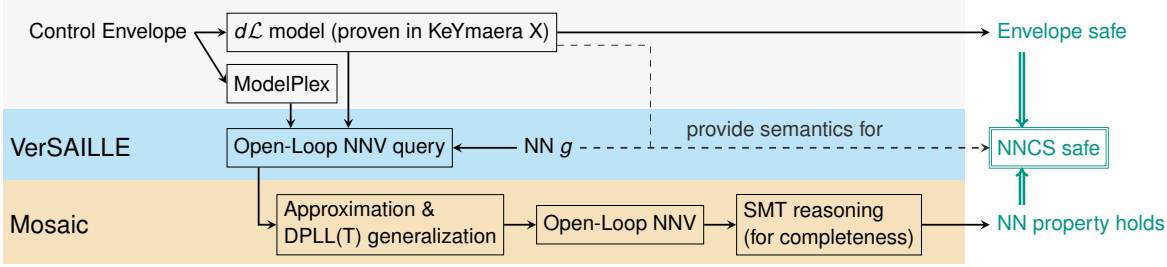
VerSAILLE

Rigorous **infinite-time** horizon **safety** for **continuous-time** NNCS via $d\mathcal{L}$ & NN verification

Mosaic

Sound, complete and efficient NN verification for **polynomial** constraints with **arbitrary propositional structure**.

Contribution



VerSAILLE

Rigorous **infinite-time** horizon **safety** for **continuous-time** NNCS via $d\mathcal{L}$ & NN verification

Mosaic

Sound, complete and efficient NN verification for **polynomial** constraints with **arbitrary propositional structure**.

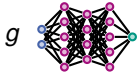
Verifiably Safe AI via Logically Linked Envelopes

$$\left(\alpha_{\text{ctrl}} \;;\; \alpha_{\text{plant}}\right)^{*} \text{Safe}$$

g 

Verifiably Safe AI via Logically Linked Envelopes

$$\left(\alpha_{\text{ctrl}} \; ; \alpha_{\text{plant}}\right)^* \text{Safe}$$



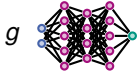
$$\left(\underbrace{\alpha_g}_{} \; ; \alpha_{\text{plant}}\right)^*$$

Nondeterministic Mirror:

$\alpha_g := \text{mirror}\,(g)$

Verifiably Safe AI via Logically Linked Envelopes

$\text{invariant} \wedge \neg\text{monitor}$ ← VerSAILLE — $\left(\alpha_{\text{ctrl}} \;;\; \alpha_{\text{plant}}\right)^*$ Safe

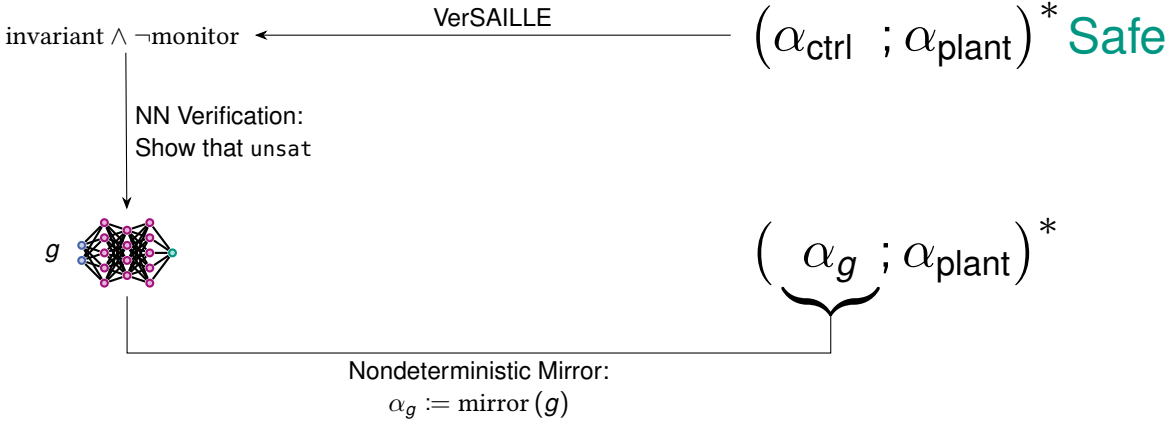


$\left(\underbrace{\alpha_g}_{} \;;\; \alpha_{\text{plant}}\right)^*$

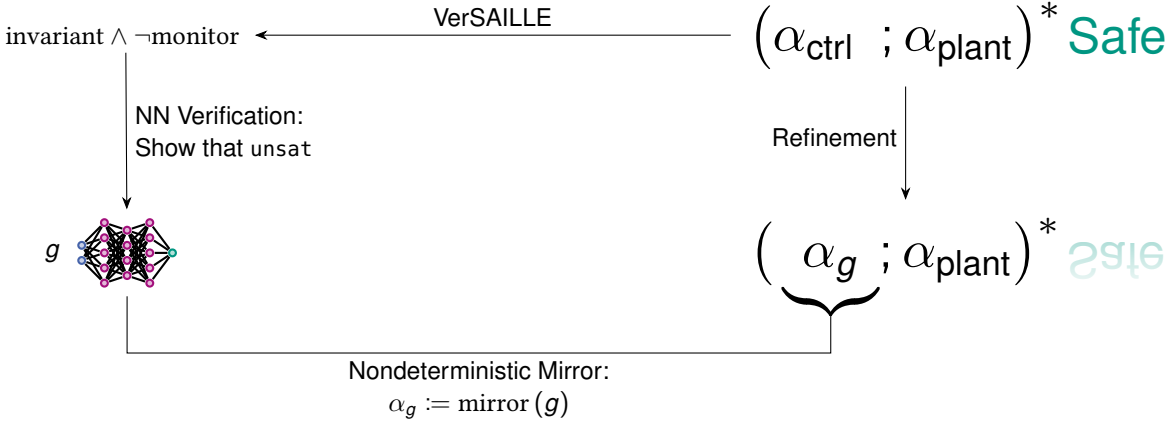
Nondeterministic Mirror:

$\alpha_g := \text{mirror}\,(g)$

Verifiably Safe AI via Logically Linked Envelopes



Verifiably Safe AI via Logically Linked Envelopes



Verifiably Safe AI via Logically Linked Envelopes

Karlsruhe Institute of Technology

Theorem (Soundness)

Assume:

- *g is a (piece-wise Noetherian) neural network*
- $C \equiv \left(\phi \rightarrow \left[(\alpha_{ctrl}; \alpha_{ctrl})^*\right] \psi\right)$ *is a valid d$\mathcal{L}$ contract*
- controller *is a controller monitor (ModelPlex)*
- invariant *is an inductive invariant*

Verifiably Safe AI via Logically Linked Envelopes

Karlsruhe Institute of Technology

Theorem (Soundness)

Assume:

- *g is a (piece-wise Noetherian) neural network*
- $C \equiv \left(\phi \rightarrow \left[(\alpha_{ctrl}; \alpha_{ctrl})^*\right] \psi\right)$ *is a valid d$\mathcal{L}$ contract*
- $\mathrm{controller}$ *is a controller monitor (ModelPlex)*
- $\mathrm{invariant}$ *is an inductive invariant*

If an NN Verifier returns `unsat` *for the query* $p \equiv (\mathrm{invariant} \wedge \neg \mathrm{controller})$ *on g*

Verifiably Safe AI via Logically Linked Envelopes

Karlsruhe Institute of Technology

Theorem (Soundness)

Assume:

- *g is a (piece-wise Noetherian) neural network*
- $C \equiv \left(\phi \rightarrow \left[(\alpha_{ctrl}; \alpha_{ctrl})^*\right] \psi\right)$ *is a valid d$\mathcal{L}$ contract*
- $\mathrm{controller}$ *is a controller monitor (ModelPlex)*
- $\mathrm{invariant}$ *is an inductive invariant*

If an NN Verifier returns `unsat` *for the query* $p \equiv (\mathrm{invariant} \wedge \neg \mathrm{controller})$ *on g*
Then $\phi \rightarrow \left[(\alpha_g; \alpha_{ctrl})^*\right] \psi$ *is valid.*

Verifiably Safe AI via Logically Linked Envelopes

Karlsruhe Institute of Technology

Theorem (Soundness)

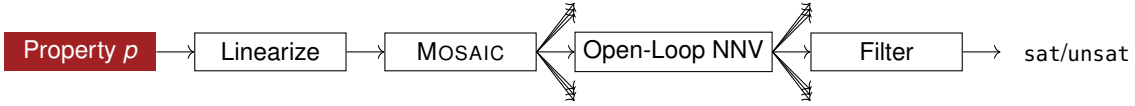
Assume:

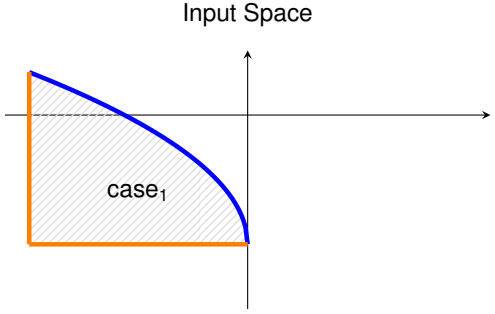
- *g is a (piece-wise Noetherian) neural network*
- $C \equiv \left(\phi \rightarrow \left[(\alpha_{ctrl}; \alpha_{ctrl})^*\right] \psi\right)$ *is a valid d$\mathcal{L}$ contract*
- $\mathrm{controller}$ *is a controller monitor (ModelPlex)*
- $\mathrm{invariant}$ *is an inductive invariant*

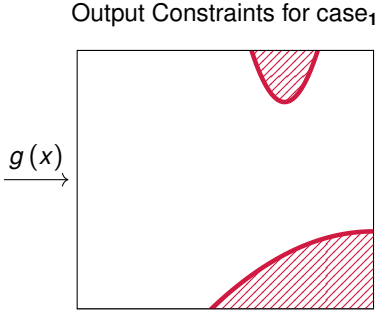
If an NN Verifier returns `unsat` *for the query* $p \equiv (\mathrm{invariant} \wedge \neg \mathrm{controller})$ *on g*
Then $\phi \rightarrow \left[(\alpha_g; \alpha_{ctrl})^*\right] \psi$ *is valid.*

How can we verify the property $\mathrm{invariant} \wedge \neg \mathrm{controller}$ in practice?

Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

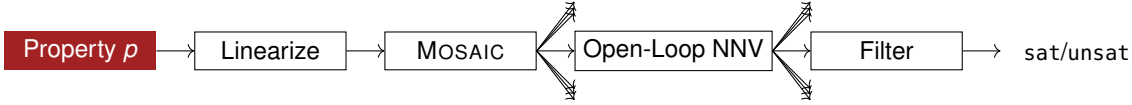


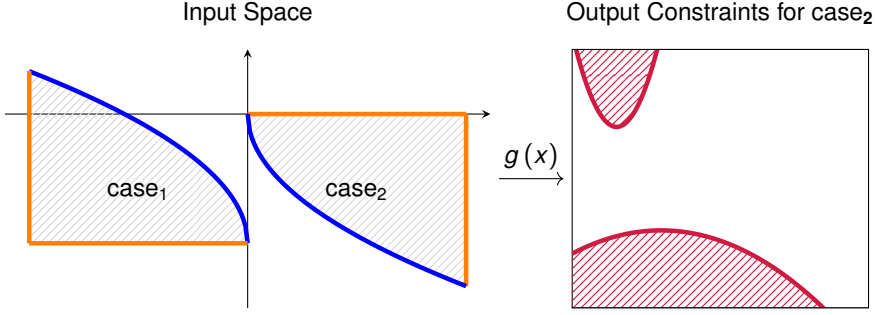




- Piece-wise linear NNs
- Nonlinear Constraints

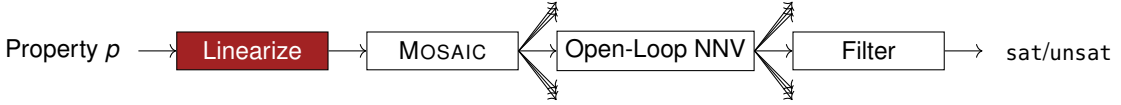
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

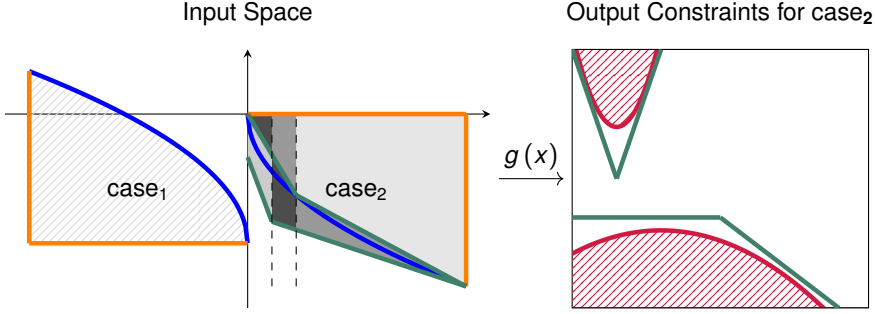




- Piece-wise linear NNs
- Nonlinear Constraints
- Arbitrary propositional structure

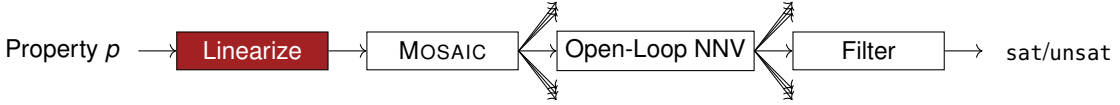
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

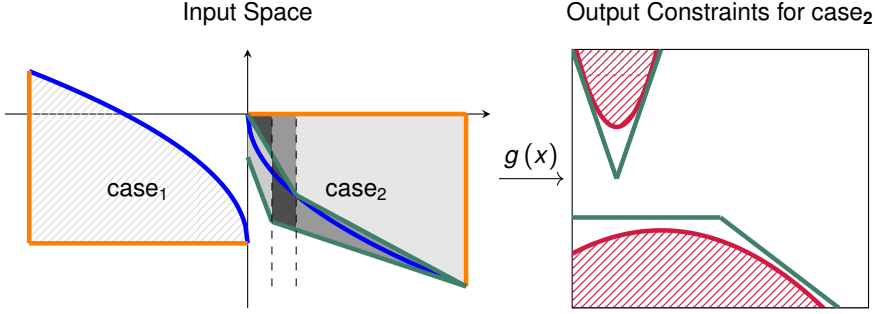




- Approximation (Sidrane et al. 2022)

Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

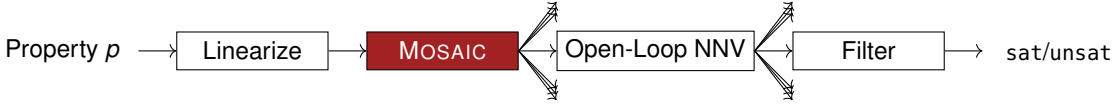


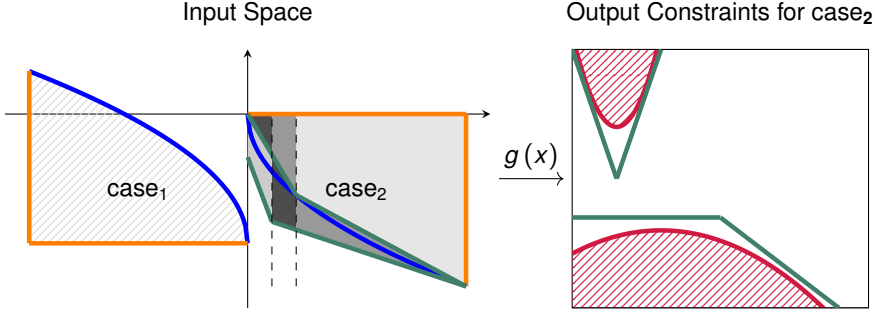


- Approximation (Sidrane et al. 2022)
- **Approximation added to formula:**

$$p \wedge c \rightarrow c_o$$
$$\wedge\, c_u \rightarrow c$$

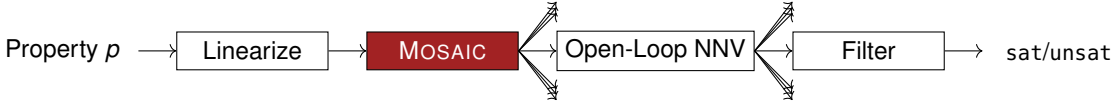
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

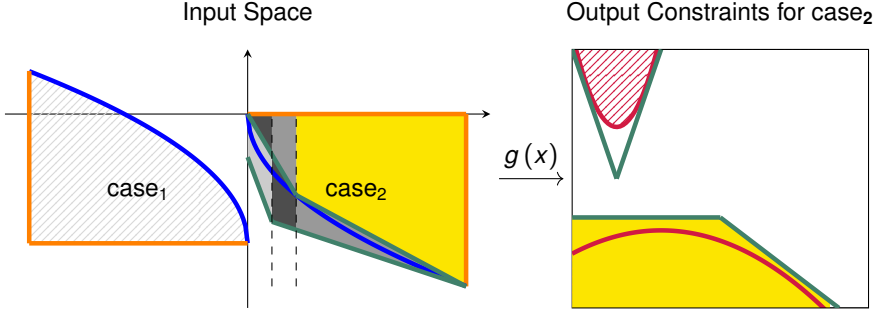




- DPLL(T) enumerates **all conjunctions**

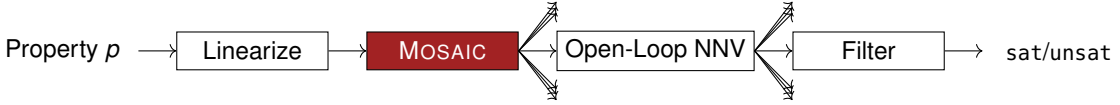
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

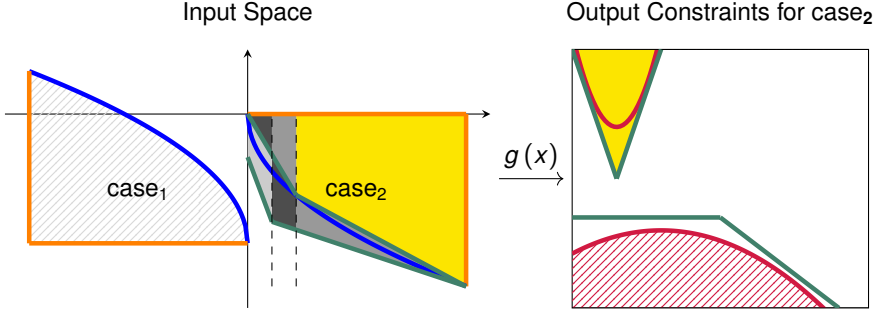




- DPLL(T) enumerates **all conjunctions**

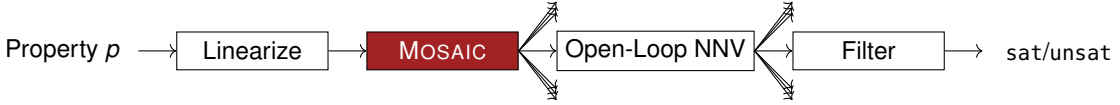
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

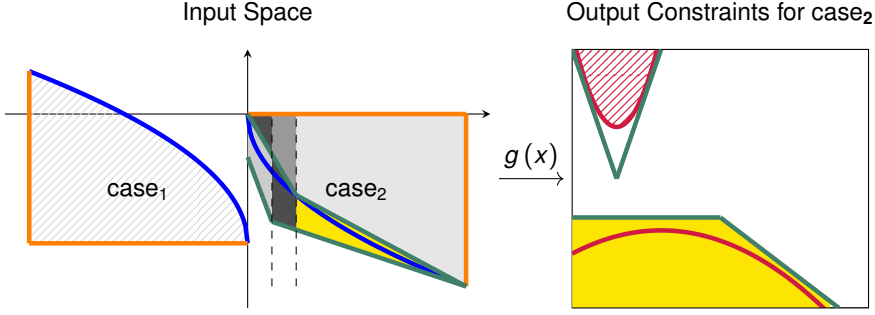




- DPLL(T) enumerates **all conjunctions**

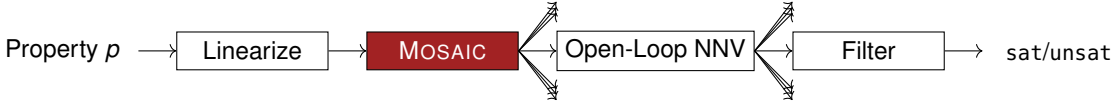
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

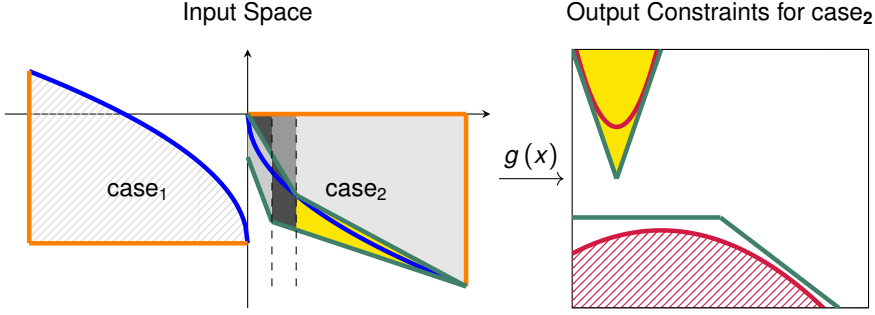




- DPLL(T) enumerates **all conjunctions**

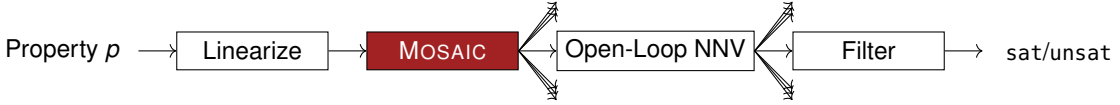
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

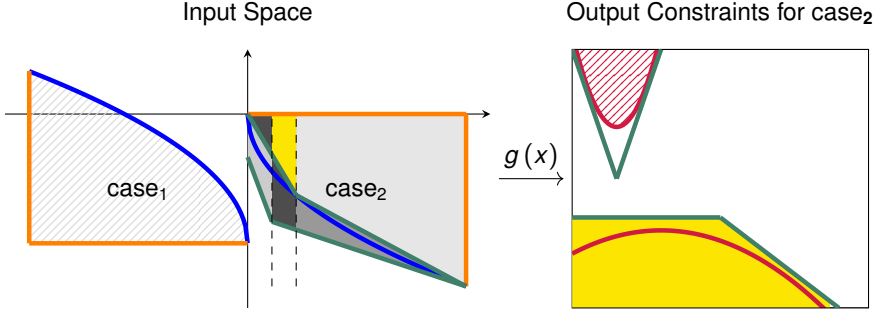




- DPLL(T) enumerates **all conjunctions**

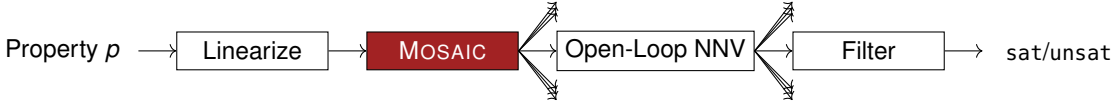
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

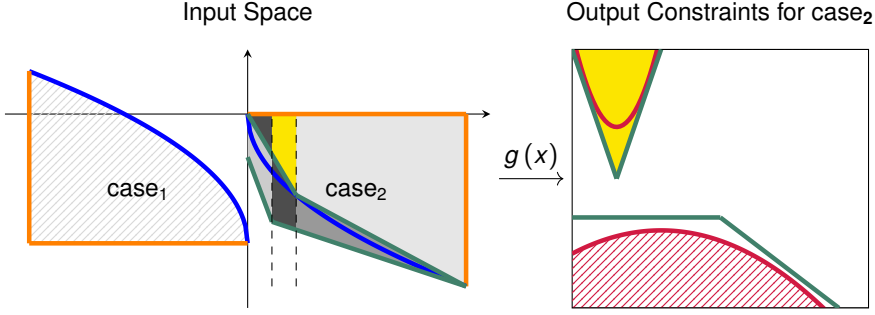




- DPLL(T) enumerates **all conjunctions**

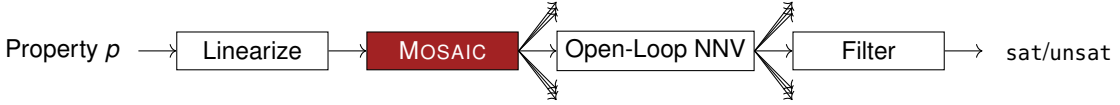
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

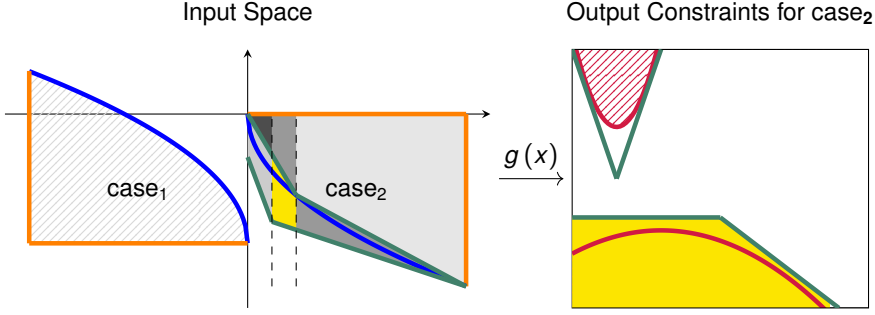




- DPLL(T) enumerates **all conjunctions**

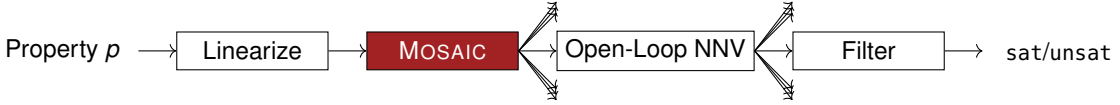
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

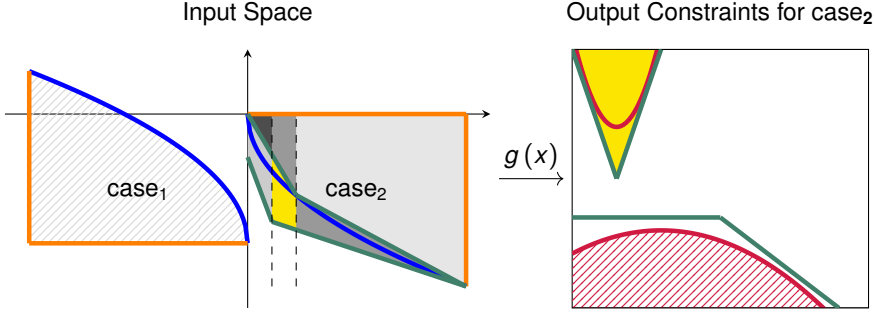




- DPLL(T) enumerates **all conjunctions**

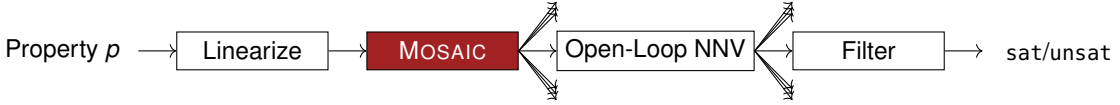
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

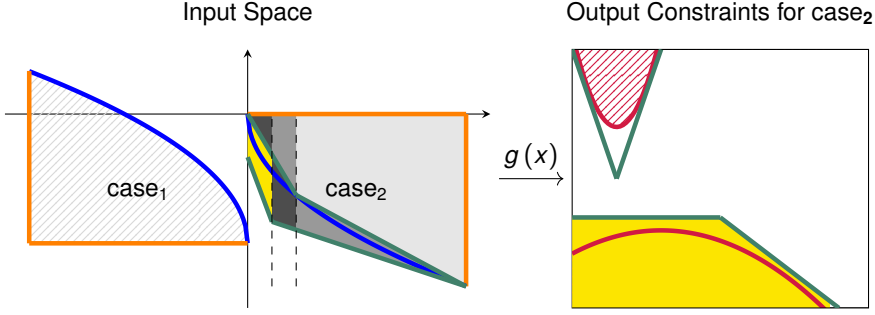




- DPLL(T) enumerates **all conjunctions**

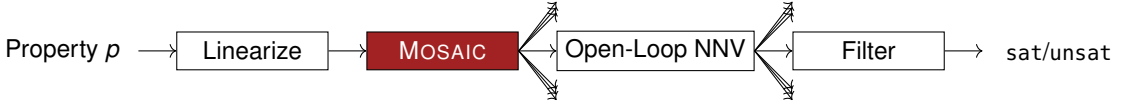
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

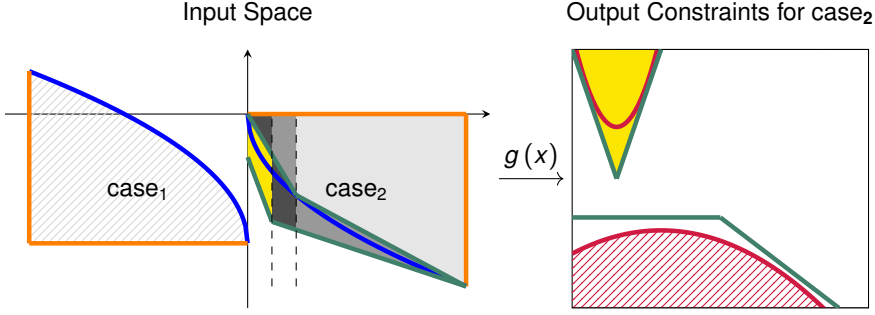




- DPLL(T) enumerates **all conjunctions**

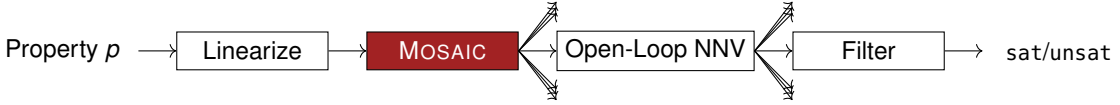
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

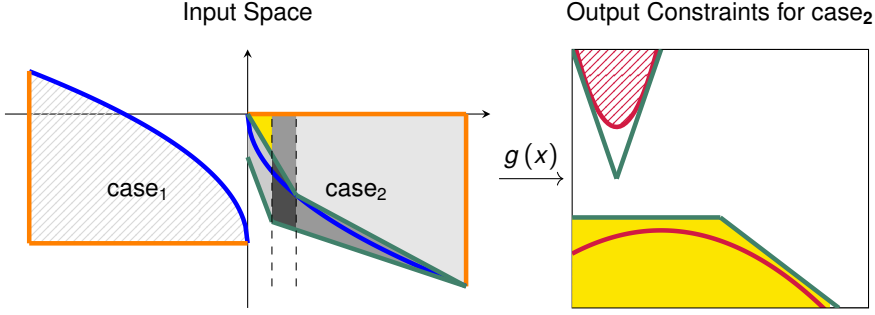




- DPLL(T) enumerates **all conjunctions**

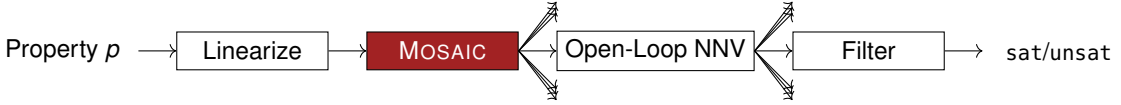
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

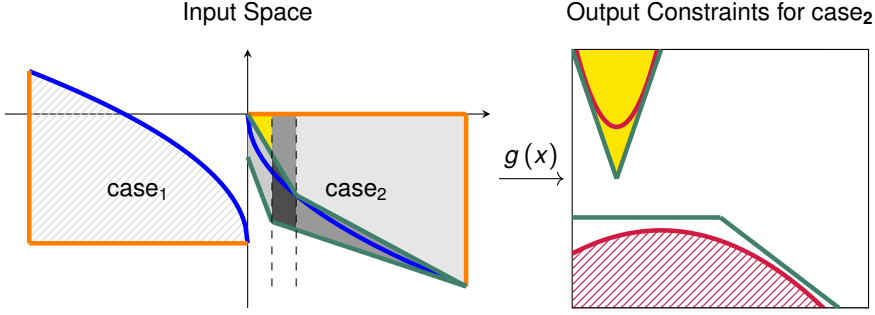




- DPLL(T) enumerates **all conjunctions**

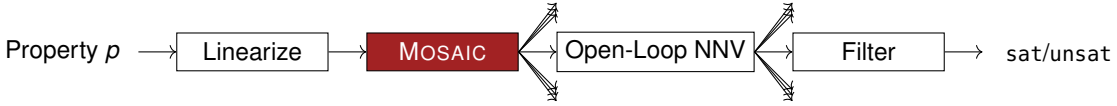
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

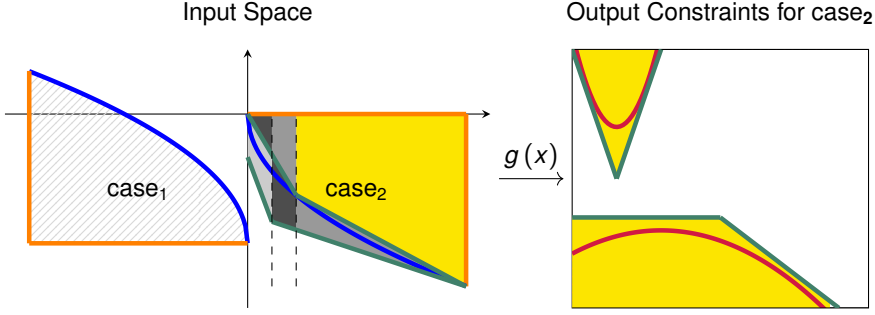




- DPLL(T) enumerates **all conjunctions**

Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

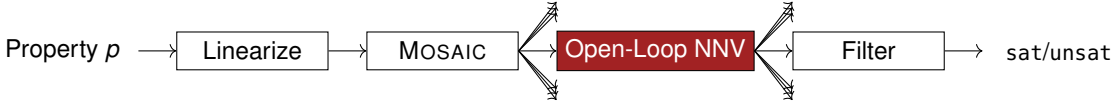


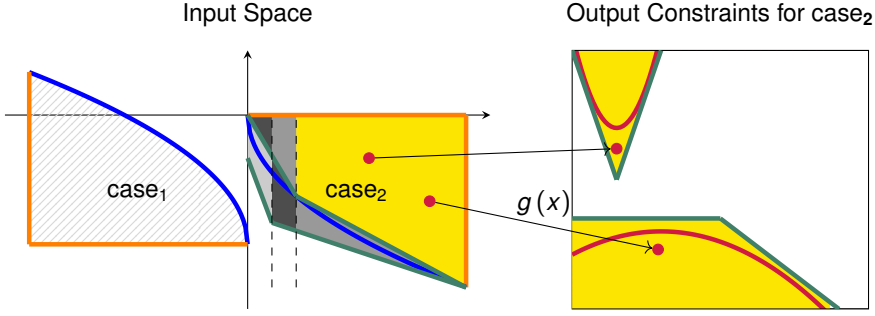


- DPLL(T) enumerates **all conjunctions**
- Mosaic:
  Conjunction over the input space
  Disjunction over the output space

$\Rightarrow$ **Minimality Guarantee**

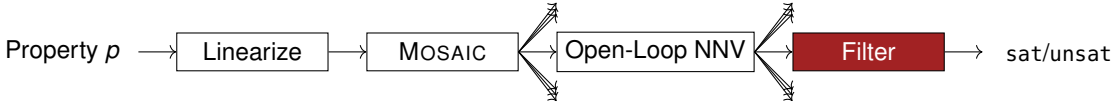
Mosaic: Efficient and Complete NN Verification for Nonlinear Properties

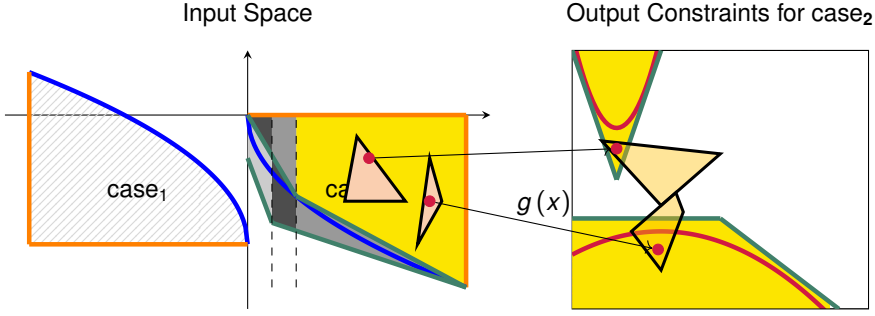




- Off-the-shelf tools
- May produce spurious counterexamples

Mosaic: Efficient and Complete NN Verification for Nonlinear Properties





- Exploit **linear** regions
- Only check nonlinear constraints where necessary

⇒ **Complete procedure**

Evaluation: Overview

- Implementation of Mosaic for ReLU NNs in Julia
  Uses NNEnum, PicoSAT and Z3

Bak and Tran 2022
Biere 2008
Jovanovic and Moura 2012

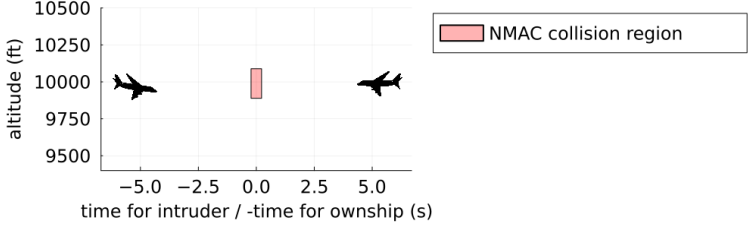
- Application of VerSAILLE & Mosaic to multiple case studies:
  - Adaptive Cruise Control and Zeppelin steering
  - **Vertical Airborne Collision Avoidance**
- Comparison to State of the Art tools

Evaluation: Overview

- Implementation of Mosaic for ReLU NNs in Julia
  Uses NNEnum, PicoSAT and Z3

Bak and Tran 2022
Biere 2008
Jovanovic and Moura 2012

- Application of VerSAILLE & Mosaic to multiple case studies:
  - Adaptive Cruise Control and Zeppelin steering
  - **Vertical Airborne Collision Avoidance**
- Comparison to State of the Art tools

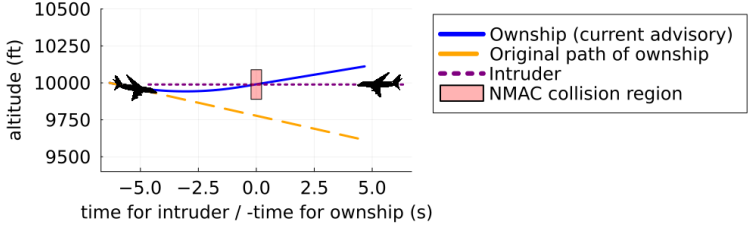


Evaluation: Overview

- Implementation of Mosaic for ReLU NNs in Julia
  Uses NNEnum, PicoSAT and Z3

Bak and Tran 2022
Biere 2008
Jovanovic and Moura 2012

- Application of VerSAILLE & Mosaic to multiple case studies:
  - Adaptive Cruise Control and Zeppelin steering
  - **Vertical Airborne Collision Avoidance**
- Comparison to State of the Art tools



Evaluation: Vertical Airborne Collision Avoidance

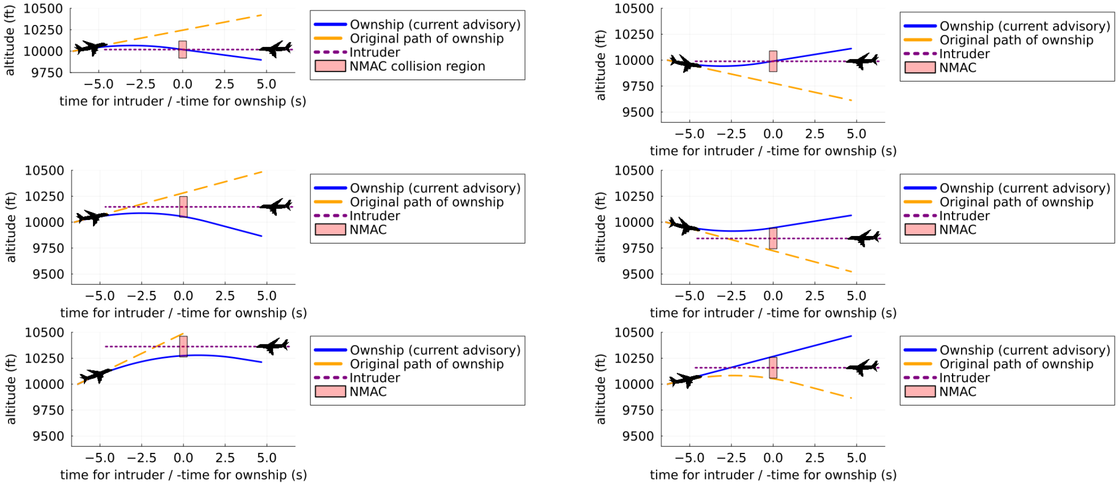
- NNs by Julian and Kochenderfer 2019
- $d\mathcal{L}$ formalization by Jeannin et al. 2017: Control Envelope and Loop Invariants
- This Analysis:
  - Exclude Clear-of-Conflict
  - Intruder in Level Flight
- Choice of NN dependent on prior state
  $\Rightarrow$ specialize invariant to current advisory

Evaluation: Vertical Airborne Collision Avoidance

- NNs by Julian and Kochenderfer 2019
- $d\mathcal{L}$ formalization by Jeannin et al. 2017: Control Envelope and Loop Invariants
- This Analysis:
  - Exclude Clear-of-Conflict
  - Intruder in Level Flight
- Choice of NN dependent on prior state
  $\Rightarrow$ specialize invariant to current advisory
- **Exhaustive characterization** of unsafe areas
- Heuristic search for unsafe trajectories in unsafe areas

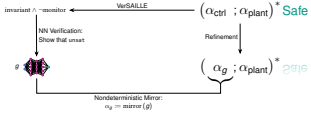
Prev. Adv.	Status	Time	CE regions	First CE
DNC	**safe**	0.35 h	—	—
DND	**safe**	0.28 h	—	—
DES1500	**unsafe**	5.45 h	49,428	0.04 h
CL1500	**unsafe**	5.18 h	34,658	0.08 h
SDES1500	**unsafe**	4.05 h	5,360	0.97 h
SCL1500	**unsafe**	4.89 h	11,323	0.36 h
SDES2500	**unsafe**	3.66 h	5,259	1.39 h
SCL2500	**unsafe**	4.45 h	7,846	0.53 h

Vertical Airborne Collision Avoidance: Trajectories



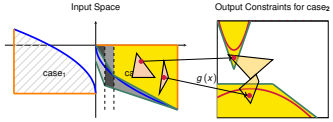
Summary

VerSAILLE



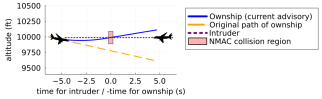
Infinite-time safety based on $d\mathcal{L}$ control envelopes

Mosaic



Nonlinear Properties with Arbitrary Propositional Structure

Case Study: Vertical CAS



Exhaustive characterization of unsafe regions

Future Work

- Further Case Studies
- Proof Certificates
- Further Engineering

Paper
arXiv:2402.10998

GitHub (Tool)
samysweb/NCubeV

References I

[1] Stanley Bak and Hoang-Dung Tran. “Neural Network Compression of ACAS Xu Early Prototype Is Unsafe: Closed-Loop Verification Through Quantized State Backreachability”. In: *NASA Formal Methods - 14th International Symposium, NFM 2022, Pasadena, CA, USA, May 24-27, 2022, Proceedings*. Ed. by Jyotirmoy V. Deshmukh, Klaus Havelund, and Ivan Perez. Vol. 13260. LNCS. Springer, 2022, pp. 280–298. DOI: 10.1007/978-3-031-06773-0_15.

[2] Armin Biere. “PicoSAT Essentials”. In: *J. Satisf. Boolean Model. Comput.* 4.2-4 (2008), pp. 75–97. DOI: 10.3233/sat190039.

[3] Jean-Baptiste Jeannin et al. “A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system”. In: *Int. J. Softw. Tools Technol. Transf.* 19.6 (2017), pp. 717–741. DOI: 10.1007/s10009-016-0434-1.

[4] Dejan Jovanovic and Leonardo de Moura. “Solving non-linear arithmetic”. In: *ACM Commun. Comput. Algebra* 46.3/4 (2012), pp. 104–105. DOI: 10.1145/2429135.2429155.

References II

[5] Kyle D. Julian and Mykel J. Kochenderfer. “Guaranteeing Safety for Neural Network-Based Aircraft Collision Avoidance Systems”. In: *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. 2019, pp. 1–10. DOI: 10.1109/DASC43569.2019.9081748.

[6] Stefan Mitsch and André Platzer. “ModelPlex: verified runtime validation of verified cyber-physical system models”. In: *Formal Methods Syst. Des.* 49.1-2 (2016), pp. 33–74. DOI: 10.1007/s10703-016-0241-z. URL: https://doi.org/10.1007/s10703-016-0241-z.

[7] André Platzer. “Differential Dynamic Logic for Hybrid Systems”. In: *J. Autom. Reason.* 41.2 (2008), pp. 143–189. DOI: 10.1007/s10817-008-9103-8. URL: https://doi.org/10.1007/s10817-008-9103-8.

[8] Chelsea Sidrane et al. “OVERT: An algorithm for safety verification of neural network control policies for nonlinear systems”. In: *Journal of Machine Learning Research* 23.117 (2022), pp. 1–45.

References III

[9] Samuel Teuber, Stefan Mitsch, and André Platzer. *Provably Safe Neural Network Controllers via Differential Dynamic Logic*. 2024. arXiv: 2402.10998 [eess.SY]. URL: https://arxiv.org/abs/2402.10998.

Backup

Complexity of Mosaic

We assume:

- An NN with N ReLU nodes
- A property with M atomic constraints and I input variables

Then we get the following complexities:

- M atomic constraints: $\mathcal{O}\left(2^M\right)$ NNV queries
- Naive encoding via SMT for N ReLU nodes: $\mathcal{O}\left(2^{2^{N+I}}\right)$
- With Mosaic: $\mathcal{O}\left(2^{N+2^I}\right)$

Overall:

$$\mathcal{O}\left(2^{M+2^{N+I}}\right) \text{ vs. } \mathcal{O}\left(2^{M+N+2^I}\right)$$

References

Backup

Evaluation: Vertical Airborne Collision Avoidance

Prev. Adv.	Status	Time	CE regions	First CE
DNC	**safe**	0.35 h	—	—
DND	**safe**	0.28 h	—	—
DES1500	**unsafe**	5.45 h	49,428	0.04 h
CL1500	**unsafe**	5.18 h	34,658	0.08 h
SDES1500	**unsafe**	4.05 h	5,360	0.97 h
SCL1500	**unsafe**	4.89 h	11,323	0.36 h
SDES2500	**unsafe**	3.66 h	5,259	1.39 h
SCL2500	**unsafe**	4.45 h	7,846	0.53 h

Evaluation: Comparison with SMT

Evaluated on two Adaptive Cruise Control properties (one satisfiable, one unsatisfiable).
NNs with 256 ReLU nodes.

Tool	`ACC_Large`		`ACC_Large` retrained	
	Status	Time	Status	Time
Mathematica	MO	—	MO	—
dReal	TO	—	TO	—
Z3	unknown	510s	unknown	1793s
Z3++	unknown	2550s	unknown	2269s
cvc5	TO	—	TO	—
MathSAT	TO	—	TO	—
ours	**sat**	**87s**	**unsat**	**124s**

References

Backup

Evaluation: Comparison with Closed-Loop Techniques

Attempt to prove bounded safety on part of Adaptive Cruise Control NN:

Tool	Nonlinearities	Evaluated Configurations	Time (s)	Share of State Space	Result
NNV	no	4	711	0.009%	**safe for 0.1s**
JuliaReach	no	4	—	0.009%	unknown
CORA	yes	10	—	0.009%	unknown
POLAR	poly. Zono.	12	—	0.009%	unknown
ours	polynomial	1	**124**	100.000%	**safe for** ∞

References

Backup

Evaluation: Comparison with Closed-Loop Techniques

Attempt to prove bounded safety on part of Adaptive Cruise Control NN:

Tool	Nonlinearities	Evaluated Configurations	Time (s)	Share of State Space	Result
NNV	no	4	711	0.009%	**safe for 0.1s**
JuliaReach	no	4	—	0.009%	unknown
CORA	yes	10	—	0.009%	unknown
POLAR	poly. Zono.	12	—	0.009%	unknown
ours	polynomial	1	**124**	100.000%	**safe for** ∞

Infinte-time horizon: k-induction?

Conceptual comparison to NNV

- Attempted to show invariance w.r.t. nonlinear loop invariant
- Overapproximation **can lead to wrong results!**

References

Backup

Evaluation: Performance of Mosaic

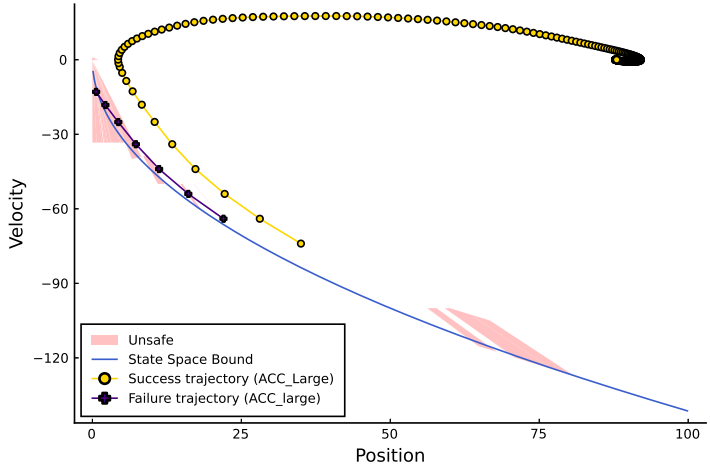
- DNNV: No support for nonlinear properties
- Comparing total/feasible propositional conjunctions

Property	# Conjunctions	**# Queries**	# Feasible Conjunctions	# SMT calls
ACC	2.4k	20	86	261
ACC (Fallback)	5.1k	15	72	235
ACAS (DNC)	117.5M	1.7k	9.9k	11.4k
ACAS (DND)	88.9M	1.8k	10.4k	12.0k
ACAS (DES1500)	451.3B	12.5k	58.8k	66.4k
ACAS (CLI1500)	374.4B	13.1k	62.5k	70.4k
ACAS (SDES1500)	9.1T	18.6k	64.1k	75.8k
ACAS (SCLI1500)	18.2T	21.8k	76.0k	88.5k
ACAS (SDES2500)	39.0T	19.0k	66.7k	78.5k
ACAS (SCLI2500)	19.4T	18.6k	67.7k	79.8k

References

Backup

Evaluation: Adaptive Cruise Control



References

Backup