Neural Model Checking

Mirco Giacobbe et al.

presenter: Byoungho Son

Intro

- **LTL verification** with proof certificates
- proof certificates represented as **Neural Networks**
- verification done on hardware designs
- unsupervised / sound / practical

Some Notations

- $X_{\mathcal{M}}$: set of (bit-vector) variables
  - $\mathrm{reg}\, X_{\mathcal{M}} \subseteq X_{\mathcal{M}}$: register variables
  - $\mathrm{inp}\, X_{\mathcal{M}} \subseteq X_{\mathcal{M}}$: input variables

- $\mathrm{obs}\, X_{\mathcal{M}} \subseteq X_{\mathcal{M}}$: observable variables for atomic props in LTL

- $\mathrm{Update}_{\mathcal{M}}$: relation over $X_{\mathcal{M}}$ and $\mathrm{reg}\, X'_{\mathcal{M}}$

- $\boldsymbol{s} \in S$: state (valuation over $X_{\mathcal{M}}$)

Recall: Automata-Theoretic LTL MC

- how to check $\mathcal{M} \models \Phi$?

- 1) build an NBA for $\mathcal{M} \parallel \mathcal{A}_{\neg\Phi}$
  - states of this NBA: (s, q)

- 2) check if final (or fair) states in F can be visited **i.o.**
  - can -> return false (+ counterexample: lasso)
  - cannot -> return true (+ **proof certificate: ranking function**)

Ranking Functions as Proofs

A ranking function takes a state of the **synchronous product** as input, and outputs a rank

i.e. ranking function $V: \operatorname{reg} S \times Q \to R$ where $(R, \prec)$: well-founded

★ Conditions for ranking functions:

$$(\boldsymbol{s}, \boldsymbol{q}) \to_{\mathcal{M} \| \mathcal{A}_{\neg\Phi}} (\boldsymbol{s}', \boldsymbol{q}') \implies V(\operatorname{reg} \boldsymbol{s}, \boldsymbol{q}) \succeq V(\operatorname{reg} \boldsymbol{s}', \boldsymbol{q}') \tag{1}$$

$$(\boldsymbol{s}, \boldsymbol{q}) \to_{\mathcal{M} \| \mathcal{A}_{\neg\Phi}} (\boldsymbol{s}', \boldsymbol{q}') \wedge \boldsymbol{q} \in F \implies V(\operatorname{reg} \boldsymbol{s}, \boldsymbol{q}) \succ V(\operatorname{reg} \boldsymbol{s}', \boldsymbol{q}') \tag{2}$$

Ranking Functions as NN's

Ranking functions are represented by neural networks:

$$V(r, q) \equiv \bar{V}(r; \theta_q)$$

where,

- $\bar{V} : \mathbb{R}^n \times \Theta \to \mathbb{R}$
- $n = |\operatorname{reg} X_{\mathcal{M}}|$

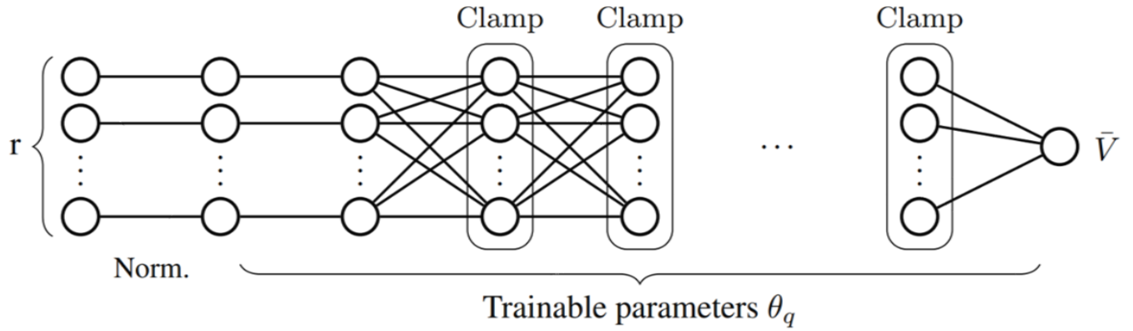
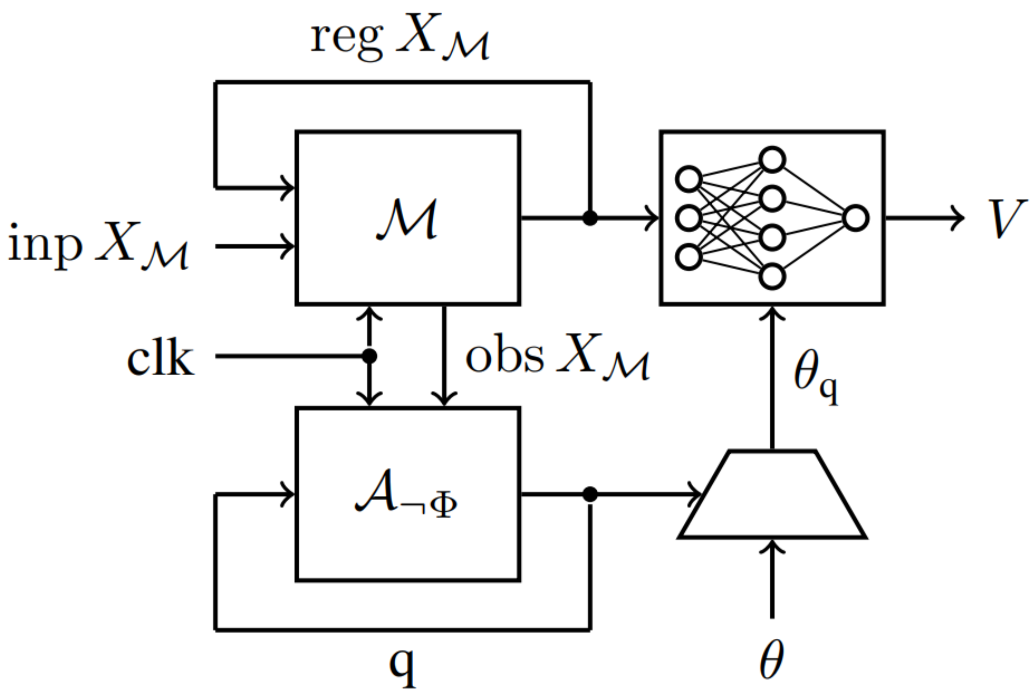
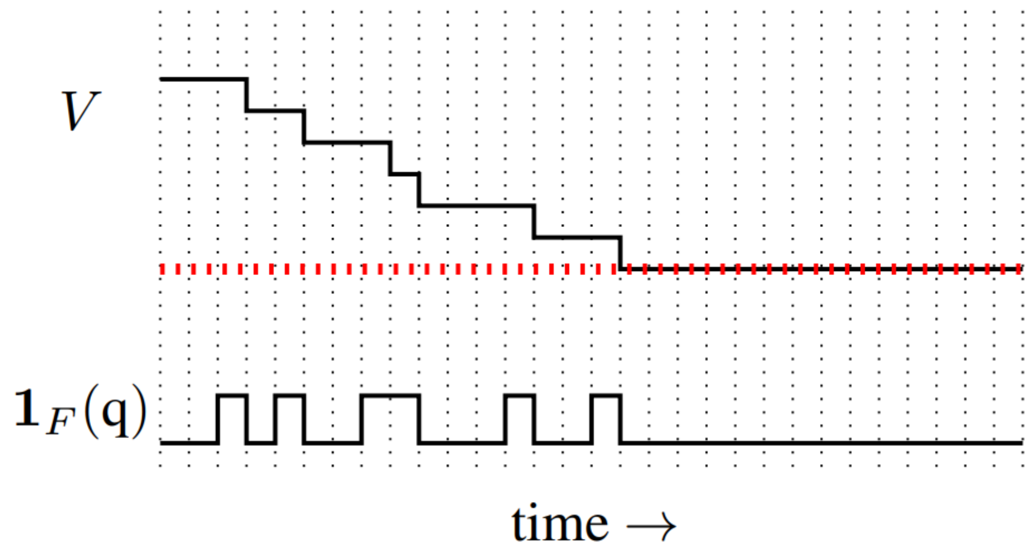


Figure 3: Neural ranking function architecture

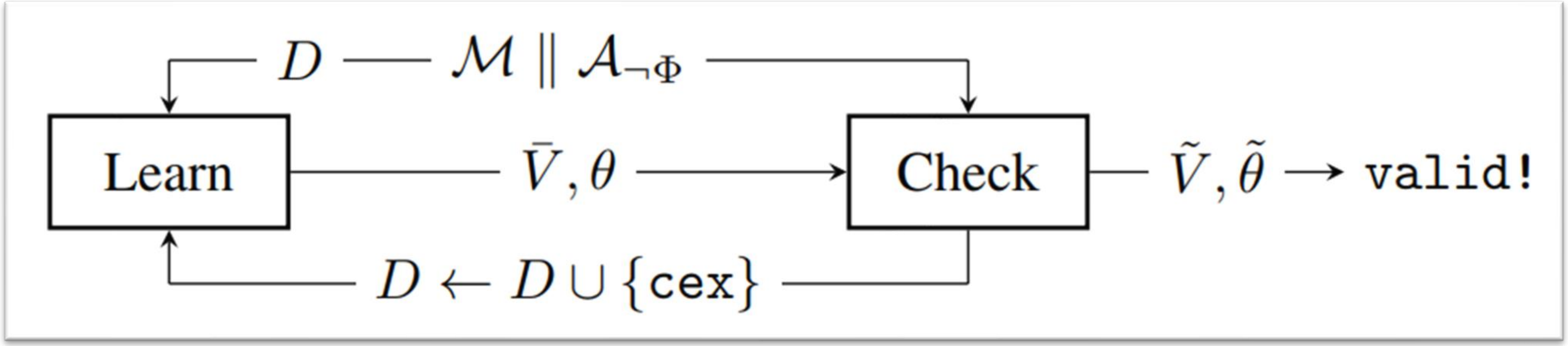
Synchronous Products & Ranking Functions





★ Learn-and-Check Procedure

Task: verify $L_{\mathcal{M}} \subseteq L_{\Phi}$



Learning

To train $\bar{V}$, for each quadruple $(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{r}', \boldsymbol{q}') \in D$, minimize the following:

$$\mathcal{L}_{\text{Rank}}(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{r}', \boldsymbol{q}'; \theta) = \text{ReLU}(\bar{V}(\boldsymbol{r}'; \theta_{\boldsymbol{q}'}) - \bar{V}(\boldsymbol{r}; \theta_{\boldsymbol{q}}) + \epsilon \cdot \mathbf{1}_F(\boldsymbol{q})).$$

Suppose loss = 0.

- if $\boldsymbol{q} \notin F$, $\quad \bar{V}(\boldsymbol{r}; \theta_{\boldsymbol{q}}) \geq \bar{V}(\boldsymbol{r}'; \theta_{\boldsymbol{q}'})$
- if $\boldsymbol{q} \in F$, $\quad \bar{V}(\boldsymbol{r}; \theta_{\boldsymbol{q}}) \geq \bar{V}(\boldsymbol{r}'; \theta_{\boldsymbol{q}'}) + \epsilon$

Overall, the training ensures the following total loss function to take **value zero**:

$$\mathcal{L}(D; \theta) = \mathbb{E}_{(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{r}', \boldsymbol{q}') \in D}[\mathcal{L}_{\text{Rank}}(\boldsymbol{r}, \boldsymbol{q}, \boldsymbol{r}', \boldsymbol{q}'; \theta)]$$

Learning

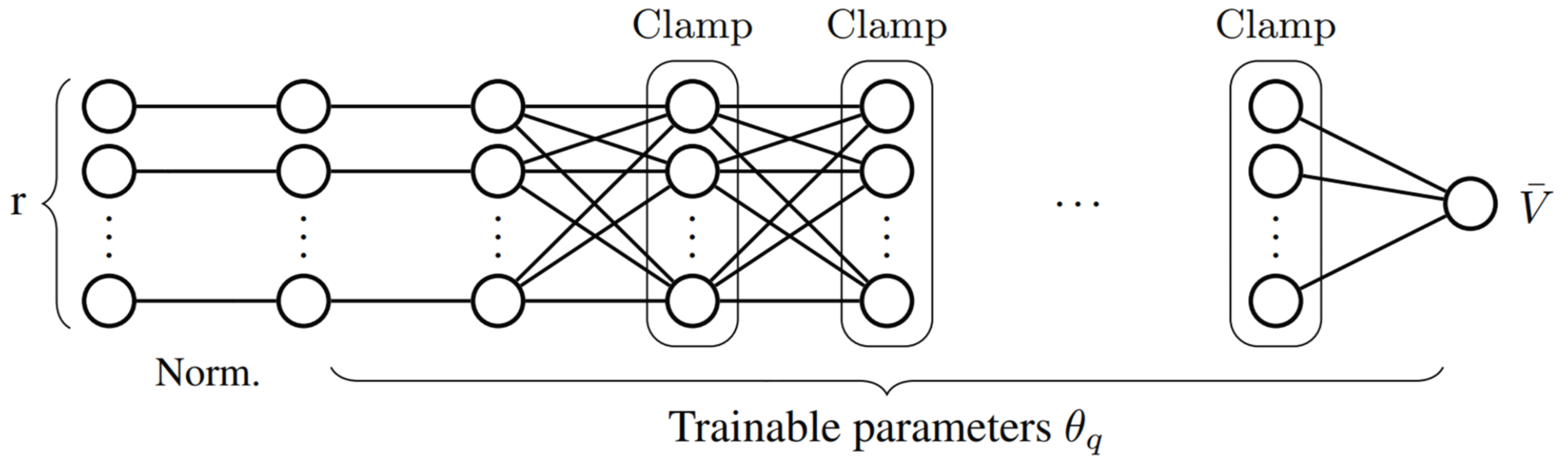


Figure 3: Neural ranking function architecture

$$\mathrm{Clamp}(x; u) = \max(0, \min(x, u))$$

Learning

- Suppose training resulted in *zero* loss
- What do we have?

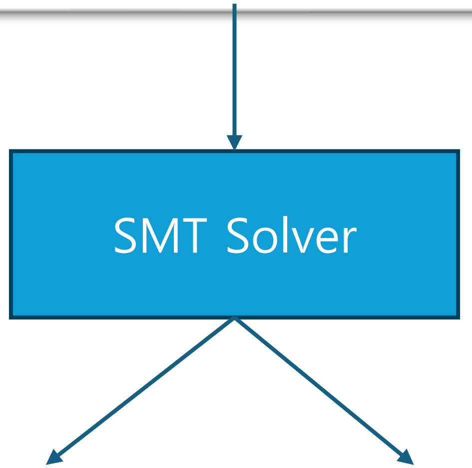
- Conditions (1) & (2) hold for the dataset D
- But not necessarily for *any* transitions in $\rightarrow_{\mathcal{M}\|\mathcal{A}_{\neg\Phi}}$

- So we need the "checking" phase

Checking

(For efficiency reasons, ranking functions are quantized as integers)

$$\bigvee_{q,q' \in Q} \mathrm{Update}_{\mathcal{M} \| \mathcal{A}_{\neg\Phi}}(s, q, r', q') \wedge \tilde{V}(\mathrm{reg}\, s; \tilde{\boldsymbol{\theta}}_q) - \mathbf{1}_F(q) < \tilde{V}(r'; \tilde{\boldsymbol{\theta}}_{q'})$$



a satisfying assignment

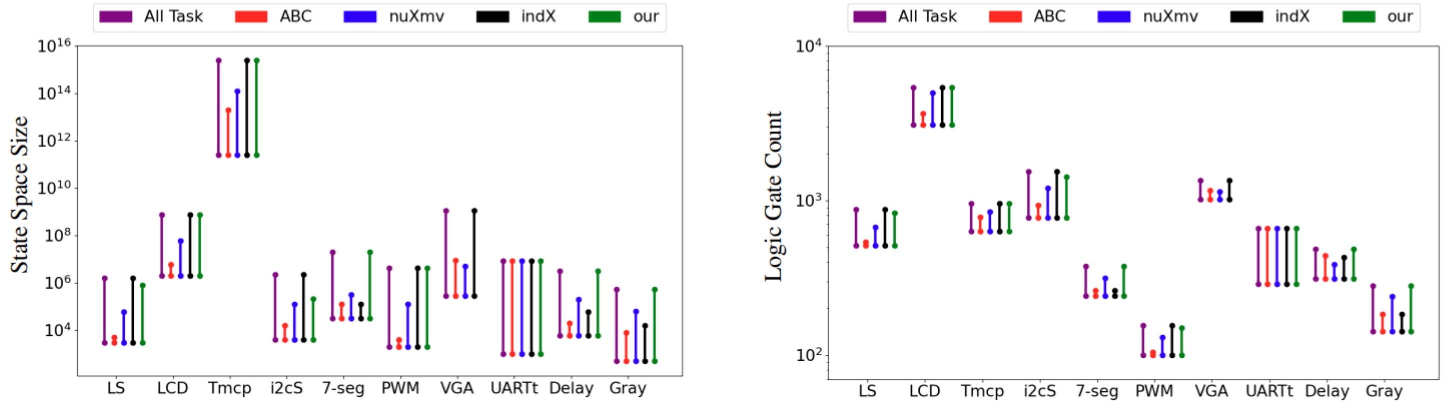
$s \in S, r' \in \mathrm{reg}\, S$

UNSAT
(verified)

Evaluation

Table 1: Number of verification task completed by academic and industrial tool, per design

	LS	LCD	Tmcp	i2cS	7-Seg	PWM	VGA	UARTt	Delay	Gray	Total
Tasks	16	14	17	20	30	12	10	10	32	33	194
ABC	2	3	7	3	8	2	3	**10**	6	13	57
nuXmv	8	9	12	10	10	7	3	**10**	24	24	117
→ our	15	**14**	**17**	**18**	**30**	11	0	**10**	**32**	**33**	180
Ind. X	**16**	**14**	**17**	**18**	18	**12**	**10**	**10**	19	22	156
Ind. Y	0	0	0	0	0	0	0	0	0	0	0



* timeout = 5h

Figure 5: Solved tasks in terms of state space size and logic gate count (log scale)