SYNTHESIS AND VERIFICATION OF STRING STABLE CONTROL FOR INTERCONNECTED SYSTEMS VIA NEURAL sISS CERTIFICATE

Jingyuan Zhou[1] Haoze Wu[2], Longhao Yan[1], Kaidi Yang[1]*
National University of Singapore[1], Amherst College[2]
jingyuanzhou@u.nus.edu, hwu@amherst.edu,
longhao.yan@u.nus.edu, kaidi.yang@nus.edu.sg

ABSTRACT

Large-scale interconnected systems require robust control strategies to ensure string stability, which is crucial for system safety and efficiency. Although learning-based controllers such as reinforcement learning (RL) have demonstrated significant potential in managing complex control scenarios, the lack of interpretability makes it difficult to provide formal string stability guarantees. To address this gap, we propose a novel verification and synthesis framework that integrates scalable input-to-state string stability (sISS) with neural network verification to formally guarantee string stability in interconnected systems. Our contributions are three-fold: (1) we reformulate the string stability analysis as a neural network verification problem by incorporating neural sISS certificates; (2) we develop a counterexample-guided training framework that synthesizes neural network-based controllers satisfying sISS constraints with minimal degradation in control performance; and (3) we validate our approach in an RL-based mixed-autonomy vehicle platooning scenario. Numerical simulations show that the refined RL controller guarantees sISS while preserving the RL policy's performance.

1 INTRODUCTION

With the rapid development of sensing, communication, and control technologies, large-scale interconnected systems have found wide applications in fields such as power systems (Gurrala & Sen, 2010), intelligent transportation systems (Zhou et al., 2024b; Zhou & Yang, 2024; Zhou et al., 2024a), and industrial process control systems (Zhu & Henson, 2002). A critical challenge in these applications is ensuring *string stability*, which prevents local disturbances from propagating or amplifying throughout the network (Feng et al., 2019). For instance, in vehicle platooning or multi-robot formations, a loss of string stability can lead to significant performance degradation and even safety risks.

Control strategies for interconnected systems can be divided into model-based and learning-based approaches. Model-based controllers, such as linear feedback control (Wang et al., 2021), model predictive control (Gratzer et al., 2022), and sliding mode control (Guo et al., 2016), enable direct analysis of string stability using time-domain (Lyapunov-based) or frequency-domain (transfer function) techniques (Feng et al., 2019). However, these model-based methods often rely on accurate system models and can become less effective or overly conservative when dealing with complex or uncertain dynamics in large-scale interconnected systems. To address these limitations, learning-based controllers, in particular, neural network-based controllers, have gained increasing popularity due to their ability to handle complex dynamics and uncertain environments in interconnected systems (Cheng et al., 2019; Li et al., 2021; Zhou et al., 2024b). However, due to the black-box nature of neural networks, it is challenging to provide formal guarantees for string stability. For existing neural network-based controllers, string stability is often treated as a soft constraint in the training

*corresponding author

process (e.g., incorporated into the loss or reward function) without offering a rigorous theoretical guarantee.

To the best of our knowledge, few studies have addressed the challenge of verifying and ensuring string stability in neural network-based controllers. The only notable attempt is the Estimation-Approximation-Derivation-Calculation framework proposed by Zhang et al. (2024) that approximates learning-based car-following controllers as linear models and then uses transfer function analysis to assess string stability. However, this approach suffers from significant approximation errors, making it unsuitable for rigorous string stability verification. It is worth noting that several works seek to provide formal guarantees for *local stability* rather than string stability. For instance, Dai et al. (2021) proposes a Lyapunov-based approach to ensure local stability by designing neural network controllers with a verifiable Lyapunov certificate. Building on this, Yang et al. (2024) introduces a framework for verifying neural control under both state and output feedback, further extending Lyapunov-based guarantees. Moreover, Mandal et al. (2024a;b) leverage Lyapunov barrier certificates to formally verify deep reinforcement learning controllers, demonstrating safe and reliable training for aerospace applications. However, these works on local stability cannot be readily extended to ensure string stability in interconnected systems. Local stability guarantees that small perturbations around an equilibrium for an individual agent decay, whereas string stability ensures that disturbances do not amplify as they propagate along a chain of agents. Thus, local stability alone is insufficient for analyzing the cumulative effects of inter-agent disturbance propagation inherent in string stability. Consequently, formally guaranteeing string stability within interconnected systems with neural network controllers remains an open and challenging problem.

Statement of Contribution. To bridge the research gap, we propose a verification and synthesis framework for learning a neural controller with a formal string stability guarantee. Our contributions are three-fold. First, we formulate the string stability analysis for heterogeneous interconnected systems with learning-based controllers as a neural network verification problem, whereby a notable notion for string stability, i.e., scalable input-to-state string stability (sISS) (Silva et al., 2024; Qiu et al., 2024), is verified. To the best of our knowledge, this is the first work that formally verifies string stability for interconnected systems under neural network-based control. Second, we synthesize a neural network-based controller with a formal sISS guarantee through a counterexample-guided training process, where a controller is initially trained with Reinforcement Learning (RL) algorithms and then fine-tuned using the counterexamples identified by the verification framework. This approach ensures the satisfaction of sISS constraints with minimum degradation in the control performance of the original RL-based policy. Third, we validate our framework in a vehicle platooning scenario. Simulation results show that the refined neural controller achieves sISS while preserving the RL control performance.

2 PROBLEM STATEMENT

Consider an interconnected system of N agents, indexed by the set $\mathcal{N} = \{1, \ldots, N\}$. The topology of agents is represented by an adjacency matrix $G \in \{0,1\}^{N\times N}$, where each element $G_{i,j} = 1$ if agent j is coupled to agent i and $G_{i,j} = 0$ otherwise. Let $\mathcal{N}_i = \{j \mid G_{i,j} = 1\} \subseteq \mathcal{N}$ denote the set of neighbors of agent i. The state of agent $i \in \mathcal{N}$ updates according to the dynamics

$$\dot{x}_i = f_i\big(x_i, \{x_j\}_{j\in\mathcal{N}_i}, u_i, d_i\big), \tag{1}$$

where $x_i \in \mathbb{R}^{n_i}$ is the state of agent i, and d_i is the external disturbance affecting agent i. The control input for agent i is determined by a neural network-based controller trained from RL, written as:

$$u_i = \pi_i\big(x_i, \{x_j\}_{j\in\mathcal{N}_i}\big). \tag{2}$$

The overarching goal of this work is to design and verify controllers for such an interconnected system so that string stability, as defined in Def. 1, is guaranteed. Intuitively, string stability requires that any disturbance *does not amplify* as it propagates through the remaining agents.

Definition 1 (Scalable Input-to-State Stability (Silva et al., 2024)) *The system Eq. (1) is sISS if there exists a class-$\mathcal{KL}$ function β and a class-$\mathcal{K}_\infty$ function γ such that, for any $N \in \mathbb{N}$, any initial conditions $x_i(0)$, and any disturbance d_i, the inequality*

$$\max_{i\in\mathcal{N}} |x_i(t)|_2 \le \beta\Big(\max_{i\in\mathcal{N}} |x_i(t_0)|_2, t - t_0\Big) + \gamma\Big(\max_{i\in\mathcal{N}} \|d_i\|_{\mathcal{L}_\infty}\Big) \tag{3}$$

is verified for any $t \geq 0$ and $i \in \mathcal{N}$.

Def. 1 quantifies how local disturbances and initial errors are attenuated in time by providing an upper bound for the disturbances of each agent. The first term on the right-hand side (RHS) captures the transient behavior of the system's response over time, where class-$\mathcal{KL}$ function β is a function that is a class-$\mathcal{K}$ function in its first argument (i.e., strictly increasing and zero at origin) and decays to zero as its second argument tends to infinity. The second term of the RHS captures the impact of the magnitude of the disturbance on the system's state, where class-$\mathcal{K}_\infty$ function γ is an unbounded strictly increasing function with $\gamma(0) = 0$.

A sufficient condition for the sISS is given in Theorem 1 (Silva et al., 2024) as follows.

Theorem 1 (sISS Vector Lyapunov Functions (Silva et al., 2024)) *Consider an interconnected system in the form of Eq. (1). Assume that for each agent $i \in \mathcal{N}$, there exists a local positive definite function $V_i : \mathbb{R}^{n_i} \to \mathbb{R}_+$, which verifies $\alpha_1\big(\|x_i\|_2\big) < V_i(x_i) < \alpha_2\big(\|x_i\|_2\big)$ for some class-$\mathcal{K}_\infty$ functions α_1, α_2. For all states $\{x_j\}_{j\in\mathcal{N}_i} \in \prod_{j\in\mathcal{N}_i} \mathbb{R}^{n_j}$, the inequality*

$$\dot{V}_i(x_i) = \nabla_{x_i} V_i\big(x_i(t)\big)^\top \dot{x}_i(t) \leq -\, a_{i,i}\, V_i(x_i) + \sum_{j\in\mathcal{N}_i} a_{i,j}\, V_j(x_j) + h_i\, |d_i|^2 \tag{4}$$

holds for any trajectory consistent for Eq. (1) with $a_{i,i} > 0$, $a_{i,j} \geq 0$, where $h_i \in \mathbb{R}$ represents the weight for the external disturbance applied to the i-th system. If for some $c > 0$, matrix $A \in \mathbb{R}^{(N+1)\times(N+1)}$ collecting the constants $a_{i,i}$ and $a_{i,j}$ satisfies the negative diagonal dominance condition, i.e.,

$$-a_{i,i} + \sum_{j\in\mathcal{N}_i} a_{i,j} \leq -c < 0\,, \tag{5}$$

then the equilibrium point of Eq. (1) satisfies sISS (see Definition 1), and $\mathcal{V} = [\,V_1, \ldots, V_n\,]^\top$ is an sISS vector Lyapunov function.

The sufficient condition represented by Theorem 1 can be used to verify the sISS property of an interconnected system with neural network-based controllers. Specifically, we aim to construct an sISS vector Lyapunov function such that each local function V_i is bounded by class-$\mathcal{K}_\infty$ functions, as in Eq. (4), where the local coupling coefficients $a_{i,j}$ (with $i, j \in \{0, 1, \ldots, N\}$) captures the influence of subsystem j on subsystem i. To this end, in the next section, we propose a verification and synthesis framework to learn an sISS-guaranteed neural network-based controller that satisfies this condition.

3 Methodology

Fig.1 demonstrates our verification and synthesis framework for learning a neural network-based controller with a formal sISS guarantee. This framework consists of two components: (i) a verification component that verifies the sufficient condition for sISS as stated in Theorem 1 (see Section 3.1), and (ii) a synthesis component that utilizes the counterexamples identified during verification to fine-tune a neural network-based controller, originally trained by RL, to ensure sISS (see Section 3.2).

3.1 Verification formulation

In this subsection, we certify sISS for interconnected systems with neural network-based controllers via neural network verification. Specifically, we first characterize the system dynamics and Lyapunov functions with neural networks and then verify the sufficient conditions for sISS as stated in Theorem 1.

Since each agent $i \in \mathcal{N}$ is trained with an actor-critic RL algorithm, we characterize its system dynamics, control policy characterized by the Actor Network, Q function characterized by the Critic Network, and Lyapunov function in discrete-time settings as neural networks $\phi_{\text{dyn},i}$, ϕ_{π_i}, ϕ_{Q_i},ϕ_{V_i},

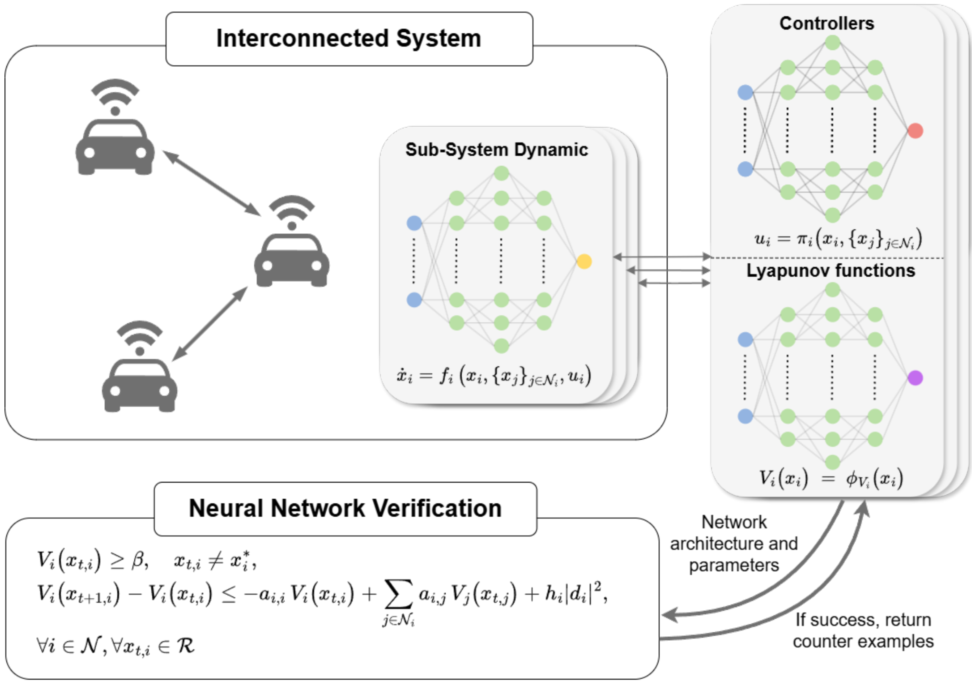


Figure 1: Framework of string stability verification for the interconnected system.

respectively, written as:

$$x_{t+1,i} = f_i(x_{t,i}, u_{t,i}, d_{t,i}) = \phi_{\mathrm{dyn},i}(x_{t,i}, u_{t,i}, d_{t,i}), \tag{6}$$

$$u_{t,i} = \pi_i(x_i, \{x_j\}_{j\in\mathcal{N}_i}) = \mathrm{clamp}\Big(\phi_{\pi_i}(x_i, \{x_j\}_{j\in\mathcal{N}_i}),\, u_{\min},\, u_{\max}\Big), \tag{7}$$

$$Q_i\left(x_{t,i}, \pi_i(x_{t,i})\right) = \phi_{Q_i}\left(x_{t,i}, \pi_i\right) \tag{8}$$

$$V_i(x_{t,i}) = \phi_{V_i}(x_{t,i}) - \phi_{V_i}(x^*_{t,i}), \tag{9}$$

where x^* represents the equilibrium state, $u_{\min}$ and $u_{\max}$ are the controller bounds, the clamp function ensures that the output of ϕ_{π_i} is restricted within the range $[u_{\min}, u_{\max}]$, and the Q-function $Q_i\left(x_{t,i}, \pi_i\right)$ evaluates control performance, where a higher Q-value corresponds to a better cumulative reward. Moreover, we represent the pre-trained RL policy, which requires further verification and fine-tuning, as $\pi_{i,\mathrm{ori}}$.

According to Theorem 1, we aim to verify the following sISS sufficient conditions:

$$V_i(x_{t,i}) \geq \beta, \quad x_{t,i} \neq x_i^*,\ \forall i \in \mathcal{N},\ \forall t \tag{10}$$

$$V_i(x_{t+1,i}) - V_i(x_{t,i}) + a_{i,i}\, V_i(x_{t,i}) - \sum_{j\in\mathcal{N}_i} a_{i,j}\, V_j(x_{t,j}) - h_i|d_{t,i}|^2 \leq 0,\ \forall i \in \mathcal{N},\ \forall t. \tag{11}$$

In our distributed verification scheme, each agent $i \in \mathcal{N}$ independently verifies that both Eq. (10) and Eq. (11) hold. The global sISS certificate is then constructed as:

$$\bigwedge_{i\in\mathcal{N}} \Big[\text{Eq. } (10) \wedge \text{Eq. } (11)\Big]. \tag{12}$$

This distributed formulation enables us to identify counterexamples for each agent's controller using verification tools such as Marabou (Wu et al., 2024) in parallel, thereby facilitating scalable and efficient verification of large-scale interconnected systems.

3.2 SYNTHESIS OF sISS GUARANTEED CONTROLLERS

This subsection presents a counterexample-guided training process for the synthesis of sISS-guaranteed neural network-based controllers in interconnected systems. This training process leverages the counterexamples found in Section 3.1 to fine-tune the neural network-based controller while simultaneously identifying the Lyapunov function candidates that satisfy sISS conditions.

As Eq. (3) serves as a sufficient condition for sISS with any coupling matrix A satisfying Eq. (5), we simultaneously search for the coupling matrix and vector Lyapunov candidates to facilitate training. Such a treatment expands the search space and thus can increase the probability of successfully identifying a valid vector Lyapunov function. To this end, we represent the coupling matrix A by combining a learnable matrix A_{pure} and adjacency matrix G:

$$A = \mathrm{ReLU}(A_{\text{pure}}) \circ G, \tag{13}$$

where the elementwise product $\circ$ ensures that $a_{i,j} = 0$ when $G_{i,j} = 0$, and the ReLU activation guarantees $a_{i,j} \geq 0$. The matrix A_{pure} is then trained together with Lyapunov functions and controllers to identify matrix A. Moreover, to ensure string stability as stated in Theorem 1, we require the negative diagonal dominance condition Eq. (5) for matrix A, which is integrated into the loss function.

Then, we formulate the training process of the coupling matrix, the vector Lyapunov function, and the RL-based controller as the following optimization problem:

$$\begin{aligned}
\min_{\{\pi_i\}_{i\in\mathcal{N}},\{V_i\}_{i\in\mathcal{N}},A} \quad & \sum_i \sigma_{\text{o},1} \|\pi_i(x_{t,i}) - \pi_{i,\text{ori}}(x_{t,i})\|^2 + \\
& \sigma_{\text{o},2} \left(Q_i\left(x_{t,i}, \pi_{i,\text{ori}}(x_{t,i})\right) - Q_i\left(x_{t,i}, \pi_i(x_{t,i})\right)\right) && (14)\\
\text{s.t.} \quad & V_i(x_{t,i}) \geq \beta, \quad x_{t,i} \neq x_i^*, && (15)\\
& V_i(x_{t+1,i}) - V_i(x_{t,i}) \leq -a_{i,i}\, V_i(x_{t,i}) + \sum_{j\in\mathcal{N}_i} a_{i,j}\, V_j(x_{t,j}) + h_i|d_{t,i}|^2, && (16)\\
& -a_{i,i} + \sum_{j\in\mathcal{N}_i} a_{i,j} \leq -c, && (17)\\
& \forall i \in \mathcal{N}, \forall x_{t,i} \in \mathcal{R} && (18)
\end{aligned}$$

where the term $\|\pi_i(x_{t,i}) - \pi_{i,\text{ori}}(x_{t,i})\|^2$ in the objective function ensures that the output of the newly synthesized controller $\pi_i(x_{t,i})$ remains close to that of the original RL-trained controller $\pi_{i,\text{ori}}(x_{t,i})$, and the term $\left(Q_i\left(x_{t,i}, \pi_{i,\text{ori}}(x_{t,i})\right) - Q_i\left(x_{t,i}, \pi_i(x_{t,i})\right)\right)$ encourages the new controller to achieve higher Q-values, thereby implicitly enhancing control performance. The coefficients $\sigma_{\text{o},1}$ and $\sigma_{\text{o},2}$ are the weighting coefficients. Constraints Eq. (15) and Eq. (16) enforce the sISS conditions, while constraint Eq. (17) imposes the coupling matrix requirement derived from the negative diagonal dominance condition. $\mathcal{R}$ is reachable set for all the agents.

Since the optimization problem cannot be solved in closed form, we reformulate it using the following loss function:

$$L_{\text{o},1} = \frac{\sum_{i\in\mathcal{N}} \|\pi_i(x_{t,i}) - \pi_{i,\text{ori}}(x_{t,i})\|^2}{|\mathcal{N}|}, \tag{19}$$

$$L_{\text{o},2} = \frac{\sum_{i\in\mathcal{N}} \mathrm{ReLU}\left(Q_i\left(x_{t,i}, \pi_{i,\text{ori}}(x_{t,i})\right) - Q_i\left(x_{t,i}, \pi_i(x_{t,i})\right) + \epsilon_{\text{o},2}\right)}{|\mathcal{N}|}, \tag{20}$$

$$L_{\text{p}} = \frac{\sum_{i\in\mathcal{N}} \mathrm{ReLU}\left(\beta - V_i(x_{t,i}) + \epsilon_{\text{p}}\right)}{|\mathcal{N}|}, \tag{21}$$

$$L_{\text{d}} = \frac{\sum_{i\in\mathcal{N}} \mathrm{ReLU}\left(V_i(x_{t+1,i}) - V_i(x_{t,i}) + a_{i,i}V_i(x_{t,i}) - \sum_{j\in\mathcal{N}_i} a_{i,j}\, V_j(x_{t,j}) - h_i|d_{t,i}|^2 + \epsilon_{\text{d}}\right)}{|\mathcal{N}|}, \tag{22}$$

$$L_{\text{v}} = \frac{\sum_{i\in\mathcal{N}} \mathrm{ReLU}\left(-a_{i,i} + \sum_{j\in\mathcal{N}_i} a_{i,j} + c + \epsilon_{\text{v}}\right)}{|\mathcal{N}|}, \tag{23}$$

$$L(A,\pi,V) = \sigma_{\text{o},1}L_{\text{o},1} + \sigma_{\text{o},2}L_{\text{o},2} + \sigma_{\text{p}}L_{\text{p}} + \sigma_{\text{d}}L_{\text{d}} + \sigma_{\text{v}}L_{\text{v}} \tag{24}$$

where the coefficients $\sigma_{\mathrm{o},1}, \sigma_{\mathrm{o},2}, \sigma_{\mathrm{p}}, \sigma_{\mathrm{d}}, \sigma_{\mathrm{v}}$ are weighting factors that balance the different objectives in the loss function. The values ϵ_{p}, $\epsilon_{\mathrm{o},2}$, ϵ_{d} and ϵ_{v} act as margins to ensure that we can adopt a more conservative policy to satisfy the sISS conditions.

With the training and verification formulation, we use a counterexample-guided inductive synthesis (CEGIS) loop to obtain a fully verified controller, coupling matrix, and certificate as in Algorithm 1. At each CEGIS iteration, we jointly train $\{V_i\}_{i\in\mathcal{N}}$, $\{\pi_i\}_{i\in\mathcal{N}}$, and A until the loss Eq. (24) converges and then use a neural network verifier (e.g. Marabou (Wu et al., 2024)) to identify counterexamples. If the verifier identifies counterexamples violating constraints Eq. (10) or Eq. (11), we sample points in the proximity of the counterexample and use these to augment the training data. By sampling multiple nearby points, we learn smooth behavior for a localized neighborhood instead of overfitting to a specific point. This process is repeated iteratively until no counterexamples are found, at which point we are guaranteed to have produced a fully verified controller, coupling matrix, and Lyapunov certificates.

Algorithm 1 Counterexample-Guided Inductive Synthesis Loop for sISS Certificate

Require: Initial controller parameters $\{\pi_i\}_{i\in\mathcal{N}}$, initial dataset $\mathcal{D}$ containing state-action pairs, Lyapunov function parameters $\{V_i\}_{i\in\mathcal{N}}$, an initial coupling matrix A, and a neural verifier (e.g., Marabou).

Ensure: Verified controllers $\{\pi_i\}_{i\in\mathcal{N}}$, Lyapunov functions $\{V_i\}_{i\in\mathcal{N}}$, and coupling matrix A ensuring stability conditions.

Initialize: Load controller parameters, initialize Lyapunov function parameters, and set an initial coupling matrix.
repeat
  Train controllers $\{\pi_i\}_{i\in\mathcal{N}}$, coupling matrix A and Lyapunov functions $\{V_i\}_{i\in\mathcal{N}}$ by minimizing the loss function Eq. (24) using gradient descent.
  Evaluate the stability conditions by verifying the sISS constraints Eq. (10) and Eq. (11) using the neural verifier (Marabou).
  if counterexamples violating the constraints are found **then**
    Identify the violating states and generate new samples in their proximity to refine training.
    Augment dataset $\mathcal{D}$ with these new samples and retrain the controllers and Lyapunov functions.
  else
    Terminate as the obtained controllers, Lyapunov functions, and coupling matrix satisfy the stability conditions.
  end if
until no counterexamples are found after verification.
return Verified controllers $\{\pi_i\}_{i\in\mathcal{N}}$, Lyapunov functions $\{V_i\}_{i\in\mathcal{N}}$, and coupling matrix A ensuring system stability.

4 Numerical Simulation

In this section, we present numerical simulations to evaluate the performance of the proposed control framework. Section 4.1 introduces the training setting of the proposed method. Section 4.2 presents the simulation results.

4.1 Training setting

We evaluate our proposed method in a mixed-autonomy platoon environment (Zhou et al., 2024b), where there are one leading vehicle, one connected and automated vehicle (CAV), and one following human-driven vehicle (HDV). The details of the scenario are described in Appendix A. The parameters of the training are given in Table 1.

Figure 2 and Table 2 show the training results. Figure 2 depicts the training loss for each iteration, each initialized with the policy from the previous iteration. The losses converge to lower values as iterations progress, indicating improved policy performance. Table 2 shows that the number of

Table 1: Parameter setting

c	$\epsilon_{o,2}$	ϵ_p	ϵ_d	ϵ_v	β	$\sigma_{o,1}$	$\sigma_{o,2}$	σ_p	σ_d	σ_v
0.05	-10	1e-2	1e-2	1e-4	0.01	5e-4	1e-4	10	5	100

counterexamples drops significantly across the three iterations and ultimately to zero, indicating that each refinement step successfully integrates discovered counterexamples and reduces violations. The calculation time also slightly decreases (37s to 30s) because verifying a more robust policy involves fewer search steps for potential violations.

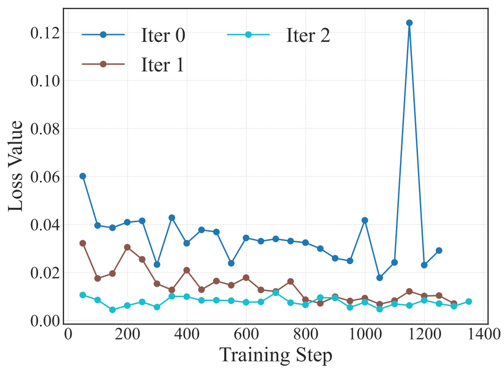


Figure 2: Training Loss over Time

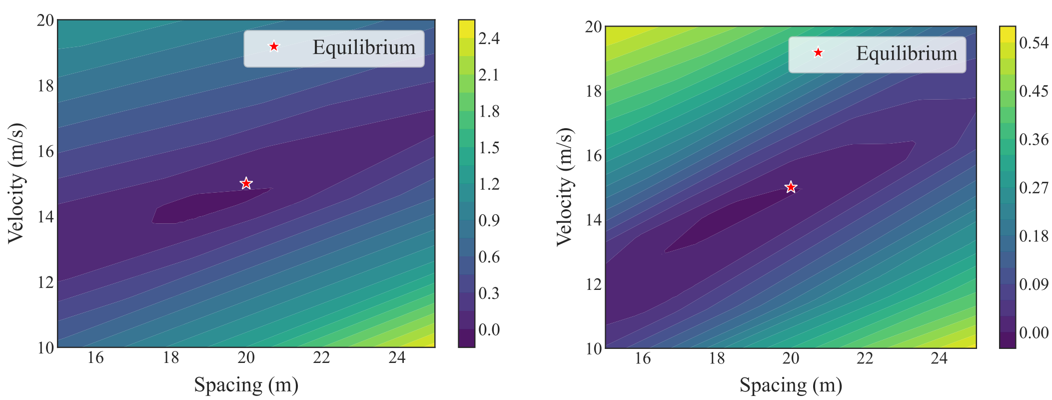
Table 2: Number of Counter Examples and Calculation Time per Iteration

Iteration	1	2	3
# Counter Examples	21	9	0
Time (s)	37	32	30

4.2 Simulation results

We present the simulation results for string stability verified by the sISS certificate and control performance.

String stability analysis: Figure 3 provides a visualization of the Lyapunov function for both CAV and HDV. The contour plots (a) and (b) offer a top-down view of the Lyapunov functions, where darker (lower) values cluster around the equilibrium point. These results demonstrate that the controller's learned Lyapunov function remains strictly positive away from equilibrium and decreases as the states approach it. This empirically validates the string stability of the mixed-autonomy platoon.



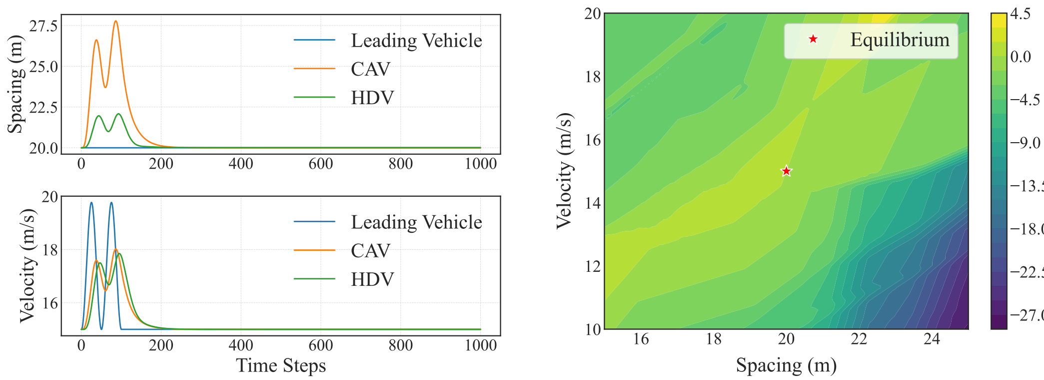
(a) Lyapunov Contour for CAV

(b) Lyapunov Contour for HDV

Figure 3: Lyapunov Function Visualization

Control performance analysis: Figure 4 (a) illustrates the response of both the CAV and the HDV to the leading vehicle's velocity disturbance under the fine-tuned CAV controller. The vehicle trajectories show that the CAV and HDV achieve stable spacing and velocity profiles over time and the system effectively mitigates disturbance propagation. Meanwhile, Figure 4 (b) displays a contour plot of the Q-value differences between the refined and original controllers across various spacing–velocity pairs. Notably, the region around the equilibrium exhibits higher Q-values for the refined controller, indicating its better performance as evaluated by the Q function in that critical area.

Overall, these results confirm that our approach not only rigorously verifies string stability through the learned Lyapunov function but also maintains robust control performance, ensuring both stability and efficiency in mixed-autonomy platoons.



(a) Vehicle Trajectory (b) Q-Value Difference Contour

Figure 4: Control performance analysis.

5 Conclusion

In this work, we present a novel verification and synthesis framework that formally guarantees string stability for interconnected systems with neural network-based controllers. By reformulating string stability analysis as a neural network verification problem and incorporating scalable input-to-state string stability (sISS) certificates, our approach addresses the limitations of traditional methods that only ensure local stability. We further develop a counterexample-guided training process that fine-tunes an RL-based controller to satisfy sISS constraints with minimal performance degradation. Validation in a mixed-autonomy vehicle platooning scenario demonstrates that the refined controller not only preserves the control performance of the original RL policy but also rigorously mitigates disturbance propagation, ensuring system string stability and efficiency. Future work will apply this string stability analysis framework to a broader range of scenarios, such as power system networks, multi-robot formations, and industrial process control systems.

References

Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3387–3395, 2019.

Hongkai Dai, Benoit Landry, Lujie Yang, Marco Pavone, and Russ Tedrake. Lyapunov-stable neural-network control. *Robotics: Science and Systems*, 2021.

Shuo Feng, Yi Zhang, Shengbo Eben Li, Zhong Cao, Henry X Liu, and Li Li. String stability for vehicular platoon control: Definitions and analysis methods. *Annual Reviews in Control*, 47: 81–97, 2019.

Alexander L Gratzer, Sebastian Thormann, Alexander Schirrer, and Stefan Jakubek. String stable and collision-safe model predictive platoon control. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):19358–19373, 2022.

Xianggui Guo, Jianliang Wang, Fang Liao, and Rodney Swee Huat Teo. Distributed adaptive integrated-sliding-mode controller synthesis for string stability of vehicle platoons. *IEEE Transactions on Intelligent Transportation Systems*, 17(9):2419–2429, 2016.

Gurunath Gurrala and Indraneel Sen. Power system stabilizers design for interconnected power systems. *IEEE Transactions on power systems*, 25(2):1042–1051, 2010.

Meng Li, Zehong Cao, and Zhibin Li. A reinforcement learning-based vehicle platoon control strategy for reducing energy consumption in traffic oscillations. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5309–5322, 2021.

Udayan Mandal, Guy Amir, Haoze Wu, Ieva Daukantas, Fletcher Lee Newell, Umberto Ravaioli, Baoluo Meng, Michael Durling, Kerianne Hobbs, Milan Ganai, et al. Safe and reliable training of learning-based aerospace controllers. In *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, pp. 1–10. IEEE, 2024a.

Udayan Mandal, Guy Amir, Haoze Wu, Ieva Daukantas, Fletcher Lee Newell, Umberto J Ravaioli, Baoluo Meng, Michael Durling, Milan Ganai, Tobey Shim, et al. Formally verifying deep reinforcement learning controllers with lyapunov barrier certificates. *arXiv preprint arXiv:2405.14058*, 2024b.

Meng Qiu, Di Liu, Simone Baldi, Guodong Yin, Wenwu Yu, and Ming Cao. Scalable input-to-state stability of heterogeneous systems with applications in vehicle platooning. *IEEE Transactions on Intelligent Vehicles*, 2024.

Guilherme Fróes Silva, Alejandro Donaire, Richard Middleton, Aaron McFadyen, and Jason Ford. Scalable input-to-state stability of nonlinear interconnected systems. *IEEE Transactions on Automatic Control*, 2024.

Jiawei Wang, Yang Zheng, Chaoyi Chen, Qing Xu, and Keqiang Li. Leading cruise control in mixed traffic flow: System modeling, controllability, and string stability. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):12861–12876, 2021.

Haoze Wu, Omri Isac, Aleksandar Zeljić, Teruhiro Tagomori, Matthew Daggitt, Wen Kokke, Idan Refaeli, Guy Amir, Kyle Julian, Shahaf Bassan, et al. Marabou 2.0: a versatile formal analyzer of neural networks. In *International Conference on Computer Aided Verification*, pp. 249–264. Springer, 2024.

Lujie Yang, Hongkai Dai, Zhouxing Shi, Cho-Jui Hsieh, Russ Tedrake, and Huan Zhang. Lyapunov-stable neural control for state and output feedback: A novel formulation. In *Forty-first International Conference on Machine Learning*, 2024.

Xiaohui Zhang, Jie Sun, Zuduo Zheng, and Jian Sun. On the string stability of neural network-based car-following models: A generic analysis framework. *Transportation research part C: emerging technologies*, 160:104525, 2024.

Jingyuan Zhou and Kaidi Yang. A parameter privacy-preserving strategy for mixed-autonomy platoon control. *Transportation Research Part C: Emerging Technologies*, 169:104885, 2024.

Jingyuan Zhou, Longhao Yan, Jinhao Liang, and Kaidi Yang. Enforcing cooperative safety for reinforcement learning-based mixed-autonomy platoon control. *arXiv preprint arXiv:2411.10031*, 2024a.

Jingyuan Zhou, Longhao Yan, and Kaidi Yang. Enhancing system-level safety in mixed-autonomy platoon via safe reinforcement learning. *IEEE Transactions on Intelligent Vehicles*, pp. 1–13, 2024b.

Guang-Yan Zhu and Michael A Henson. Model predictive control of interconnected linear and nonlinear processes. *Industrial & engineering chemistry research*, 41(4):801–816, 2002.

A SYSTEM DYNAMICS OF MIXED-AUTONOMY PLATOONS

We consider a mixed-autonomy platoon consisting of both CAVs in the set $\Omega_{\mathcal{C}}$ and HDVs in the set $\Omega_{\mathcal{H}}$. Let $n = |\Omega_{\mathcal{C}} \cup \Omega_{\mathcal{H}}|$ be the total number of vehicles, and $m = |\Omega_{\mathcal{C}}|$ denote the number of CAVs, with $|\cdot|$ representing the cardinality of a set.

Each vehicle's dynamic is described by second-order ordinary differentiable equations:

$$\dot{s}_i(t) = v_{i-1}(t) - v_i(t), \quad i \in \Omega \tag{25}$$

$$\dot{v}_i(t) = \begin{cases} u_i(t), & i \in \Omega_{\mathcal{C}} \\ \mathbb{F}_i\left(s_i(t), v_i(t), v_{i-1}(t)\right), & i \in \Omega_{\mathcal{H}} \end{cases} \tag{26}$$

where $s_i(t)$ and $v_i(t)$ represent the spacing and velocity of vehicle i, respectively. For an HDV $i \in \Omega_{\mathcal{H}}$, the acceleration is governed by an unknown car-following model $\mathbb{F}_i(\cdot)$, which depends on the spacing $s_i(t)$, the HDV's own velocity $v_i(t)$, and the preceding vehicle's velocity $v_{i-1}(t)$.

We then rewrite the longitudinal dynamics of the mixed-autonomy platoon Eq. (25)-Eq. (26) into the matrix form as:

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), v_0(t)) + B\boldsymbol{u}(t), \tag{27}$$

where $\boldsymbol{x}(t) = \left[s_1(t), v_1(t), \ldots, s_n(t), v_n(t)\right]^\top \in \mathbb{R}^{2n}$ represents the states of all n vehicles. The function $f(\cdot)$ encompasses the dynamics of both HDVs and CAVs, while $v_0(t)$ denotes the speed of the leading vehicle. The matrix $B \in \mathbb{R}^{2n \times m}$ incorporates the CAV inputs $\boldsymbol{u}(t) \in \mathbb{R}^m$, where $\boldsymbol{e}_{2n}^{2i}$ is a vector of length $2n$ with 1 in the $2i$-th position and 0 elsewhere.

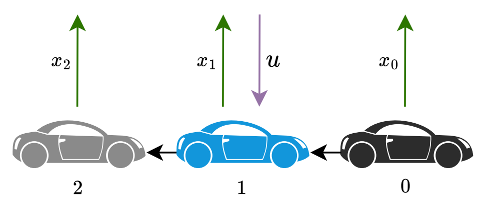


Figure 5: Mixed-autonomy platoon environment

We next introduce the adjacency matrix G, which encodes how vehicles share information. HDVs rely on onboard sensors to measure the states of preceding vehicles, whereas CAVs can also communicate with surrounding vehicles. As in Figure 5, in our mixed-autonomy platoon with one leading vehicle, one CAV, and one following HDV, the adjacency matrix $G \in \{0,1\}^{3\times 3}$ is specified as:

$$G = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \tag{28}$$

where $G_{i,j} = 1$ means that vehicle j directly provides information to vehicle i. In this example, $G_{2,3} = 1$ indicates that the HDV (vehicle 3) influences the CAV (vehicle 2), aligning with the CAV's V2V communication links.