

# Stability Verification for Switched Systems using Neural Multiple Lyapunov Functions

Junyue Huang, Shaoyuan Li and Xiang Yin

**Abstract**— Stability analysis of switched systems, characterized by multiple operational modes and switching signals, is challenging due to their nonlinear dynamics. While frameworks such as multiple Lyapunov functions (MLF) provide a foundation for analysis, their computational applicability is limited for systems without favorable structure. This paper investigates stability analysis for switched systems under state-dependent switching conditions. We propose neural multiple Lyapunov functions (NMLF), a unified framework that combines the theoretical guarantees of MLF with the computational efficiency of neural Lyapunov functions (NLF). Our approach leverages a set of tailored loss functions and a counter-example guided inductive synthesis (CEGIS) scheme to train neural networks that rigorously satisfy MLF conditions. Through comprehensive simulations and theoretical analysis, we demonstrate NMLF's effectiveness and its potential for practical deployment in complex switched systems.

## I. INTRODUCTION

Switched systems, composed of multiple subsystems and a switching signal that controls transitions between them, are widely used in engineering applications such as bipedal walking robots [1], power electronics [2], and automotive control [3]. A fundamental challenge in analyzing these systems lies in *stability verification*, since the complex interaction between subsystem dynamics and switching mechanisms can potentially lead to unstable behavior. Ensuring stability is crucial for reliable operation and for preventing performance degradation or system failure.

Over the past decades, numerous methodologies have been developed for stability verification of switched systems; see, e.g., [4]–[8]. A particularly prominent approach is the multiple Lyapunov functions (MLF) method [9]–[11], which has emerged as an effective tool for stability analysis. The MLF framework employs subsystem-specific Lyapunov functions and establishes conditions to guarantee stability under switching. Compared to common Lyapunov function approaches, MLF provides greater flexibility for systems lacking a universal Lyapunov function, while accommodating broader classes of switching signals.

In general, constructing Lyapunov functions for arbitrary systems is a very challenging task. Analytical methods are only applicable to a restricted class of systems with favorable structural properties. While numerical approaches, such as sum-of-squares (SOS) programming [12]–[14], offer

a more general computational framework, they still face scalability limitations due to computational complexity in high-dimensional systems. In recent years, Neural Lyapunov Functions (NLFs) have emerged as a powerful alternative for stability analysis [15], [16]. By leveraging neural networks to approximate Lyapunov functions, NLFs can capture complex stability properties that traditional analytical methods often fail to address. These data-driven approaches enable stability verification for nonlinear systems with high-dimensional state spaces and even unknown dynamics. Moreover, NLF-based methods provide key advantages including fully-automated synthesis, scalability, and adaptability to diverse system configurations.

In this paper, we propose a new stability verification method for switched systems that combines the advantages of multiple Lyapunov functions and neural Lyapunov functions. We consider a general state-dependent switching setting, where mode transitions are permitted within specific regions associated with each switching behavior. Our approach introduces a Neural Multiple Lyapunov Function (NMLF) framework to rigorously verify stability in state-dependent switched systems. The framework employs a set of carefully designed loss functions along with a Counter-Example Guided Inductive Synthesis (CEGIS) scheme to train neural networks that satisfy MLF conditions. Through extensive simulations and theoretical analysis, we demonstrate the effectiveness of our method and its potential for practical applications in complex switched systems.

### A. Related Works

As mentioned above, some recent works in the literature have explored the use of neural Lyapunov functions for stability verification and controller synthesis of dynamic systems. For instance, [15]–[22] has demonstrated how NLF can effectively verify stability while simultaneously maximizing the region of attraction (ROA). In contrast, [23], [24] focuses on reducing computational complexity while ensuring stability through the use of NLF. However, these works primarily focus on a non-switched nonlinear systems, and their applicability to hybrid systems, particularly region-based switching systems, remains limited. Our research aims to extend these efforts by applying NLF to the stability verification of hybrid systems with state constraints.

Recent studies have explored the integration of neural networks with multiple Lyapunov functions, including [25]–[27]. However, these works primarily employ neural networks for adaptive controller synthesis rather than direct Lyapunov function approximation for stability verification. Notably,

This work was supported by the National Natural Science Foundation of China (62573291, 62173226) and Science Center Program of National Natural Science Foundation of China under Grant 62188101.

Junyue Huang, Shaoyuan Li and Xiang Yin are with the School of Automation and Intelligent Sensing, Shanghai 200240, China. e-mail: {hyy-564904993, syli, yinxiang}@sjtu.edu.cn

existing approaches do not specifically address the unique requirements of switched systems in their neural network training procedures. Our work addresses this gap by developing a tailored framework that combines neural networks with the MLF methodology, enabling direct approximation of Lyapunov functions for switched system stability verification.

## II. PRELIMINARIES

In this work, we consider the stability verification problem for switched systems in both continuous-time setting and discrete-time setting. Specifically, for the continuous-time setting, the system dynamic is given by:

$$\dot{x} = f_i(x), \quad i \in Q = \{1, 2, \dots, N\}, \quad (1)$$

where  $x \in \mathbb{R}^n$  represents the system state,  $Q$  is the set of  $N$  system modes and each  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a vector field corresponding to the  $i$ -th mode of the system. We assume that each function  $f_i(x)$  is locally Lipschitz continuous, which ensure the existence and uniqueness of solutions.

We also consider the discrete-time setting, and the system dynamic is given by

$$x_{k+1} = f_i(x_k), \quad i \in Q = \{1, 2, \dots, N\}, \quad (2)$$

where  $x_k \in \mathbb{R}^n$  denotes the system state at time step  $k$ , and each  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a mapping corresponding to the  $i$ -th mode.

To analyze the stability of the switched systems, we first recall the notion of asymptotic stability.

**Definition 1** (Asymptotic Stability). A switched system is said to be *asymptotically stable* at an equilibrium point  $x^*$  in the region  $\mathcal{D}$  if, for every initial condition  $x(0) \in \mathcal{D}$ , the corresponding solution (with  $t \geq 0$  for continuous-time systems or  $k \geq 0$  for discrete-time systems) satisfies  $\lim_{t \rightarrow \infty} x(t) = x^*$  (for continuous-time systems) or  $\lim_{k \rightarrow \infty} x_k = x^*$  (for discrete-time systems).

In order to analyze stability of such switched systems, we employ the tool of multiple Lyapunov functions (MLFs), which allow for stability analysis under arbitrary or constrained switching rules. Given a continuously differentiable function  $V : \mathbb{R}^n \rightarrow \mathbb{R}$ , the Lie derivative of  $V$  along the vector field  $f_i(x)$  is defined as:

$$L_{f_i} V(x) = \left. \frac{d}{dt} V(x) \right|_{\dot{x}=f_i(x)} = \nabla V(x)^\top f_i(x). \quad (3)$$

Then the MLFs are formally defined as follows.

**Definition 2** (Multiple Lyapunov Functions). Given a switched system either in the continuous-time setting of form (1) or in the discrete-time setting of form (2), we say  $\{V_i(x)\}_{i \in Q}$  is a multiple Lyapunov function with respect to region  $\mathcal{D}$  and equilibrium point  $x^*$ , if the following conditions hold:

- 1) **Mode-wise Lyapunov Function Properties:** For each mode  $i$ , we have  $V_i(x^*) = 0$  and  $V_i(x) > 0$ ,  $\forall x \in \mathcal{D} \setminus \{x^*\}$ . Furthermore, in the continuous-time case,  $V_i$  is continuously differentiable and its derivative along

$f_i(x)$  satisfies  $L_{f_i} V_i(x) = \nabla V_i(x)^\top f_i(x) < 0$ ,  $\forall x \in \mathcal{D} \setminus \{x^*\}$ . In the discrete-time case, the corresponding condition is  $V_i(f_i(x)) - V_i(x) < 0$ ,  $\forall x \in \mathcal{D} \setminus \{x^*\}$ , and no differentiability assumption is required.

- 2) **Switching Decrease Properties:** Whenever the system switches from mode  $i$  to mode  $j$ , the multiple Lyapunov function satisfy  $V_j(x) < V_i(x)$ .

Then we have the following theorem for MLFs [28].

**Theorem 1.** For a switched system, if one can find a MLF  $\{V_i(x)\}_{i \in Q}$  with respect to region  $\mathcal{D}$  and equilibrium point  $x^*$ , then the switched system is asymptotically stable in  $\mathcal{D}$ . That is, for any initial state  $x(0) \in \mathcal{D}$  (or  $x_0 \in \mathcal{D}$  in the discrete case), the corresponding solution converges to  $x^*$  as  $t \rightarrow \infty$  (or as  $k \rightarrow \infty$  in the discrete-time case).

**Remark 1.** In some situations, the mode-wise Lyapunov function properties are enforced only on  $\mathcal{D} \setminus B_{\epsilon_b}(x^*)$ , i.e., by excluding a ball of radius  $\epsilon_b > 0$  centered at the equilibrium  $x^*$ . In this case, one can conclude that every trajectory from an initial state  $x(0) \in \mathcal{D}$  (or  $x_0 \in \mathcal{D}$  in the discrete case) will ultimately enter and remain within the ball  $B_{\epsilon_b}(x^*)$ . In other words, even though the Lyapunov conditions are not verified arbitrarily close to  $x^*$ , the system is practically stable in the sense that its state converges to an  $\epsilon_b$ -neighborhood of the equilibrium. This result is particularly useful in numerical implementations, as it mitigates pathological issues such as arithmetic underflow near  $x^*$  while still guaranteeing a meaningful stability property.

## III. PROBLEM FORMULATION

In the literature, the stability verification problem has been studied under various switching rules, such as arbitrary switching and language-constrained switching. Here, we consider a class of *state-dependent switching constraints*, where mode transitions are restricted to specific regions of the state space.

Formally, we consider a collection of pairwise disjoint switching regions  $\{\mathcal{D}_{ij}\}_{(i,j) \in Q \times Q}$ , where each  $\mathcal{D}_{ij} \subseteq \mathcal{D}$ . The switching rule is given by: When the system is in mode  $i$  and the state  $x$  enters  $\mathcal{D}_{ij}$ , it may either:

- Switch from mode  $i$  to  $j$ , or
- Remain in mode  $i$ .

The choice between staying or switching is nondeterministic whenever the switching condition is met. Our goal is to verify stability under all possible switching scenarios, which is formulated as follows.

**Problem 1** (Stability Verification under State-Dependent Switchings). Given a switched system either in the continuous-time setting of form (1) or in the discrete-time setting of form (2), with state-dependent switching constraints, design a computational framework to construct a multiple Lyapunov function  $\{V_i : \mathbb{R}^n \rightarrow \mathbb{R}\}_{i \in Q}$  satisfying the Mode-wise Lyapunov Properties enforced only on  $\mathcal{D} \setminus B_{\epsilon_b}(x^*)$  and switching decrease properties.

In the problem formulation above, we adopt a state-dependent switching framework motivated by both practical considerations and theoretical concerns:

- First, the state-dependent switching framework generalizes arbitrary switching. Particularly, by setting each  $\mathcal{D}_{ij} = \mathcal{D}$ , we recover arbitrary switching as a special case. However, real-world systems typically exhibit physical constraints that prevent truly arbitrary switching as mode transitions can only occur when specific state-dependent predicates are satisfied.
- As our analysis will demonstrate, arbitrary switching may induce *Lyapunov mode collapse* in our learning-based framework. This phenomenon occurs when transition compatibility conditions ( $V_j(x) < V_i(x)$ ) must hold across unconstrained switching regions, forcing distinct Lyapunov functions  $\{V_i\}$  to adopt similar geometric structures. The conflicting gradient directions during neural network training effectively paralyze the learning process. Our Neural MLF framework avoids this degeneracy by incorporating state-dependent switching constraints, enabling stable learning of distinct Lyapunov functions while maintaining training efficiency.

#### IV. METHODOLOGY

In this work, we propose a neural framework for synthesizing Lyapunov functions that certify the stability of state-constrained switching systems. The problem is addressed in two parts: (i) constructing mode-wise candidate Lyapunov functions with tailored loss terms to enforce mode-specific stability properties, and (ii) imposing switching decrease conditions to ensure stability across mode transitions. To guarantee these requirements, we employ a Counter-Example Guided Inductive Synthesis (CEGIS) framework, where an SMT solver verifies candidates and provides counterexamples for iterative refinement. The methodology is detailed in the following subsections.

##### A. Mode-Wise NLFs and Mode-Wise Losses

For each mode  $i$ , we construct a candidate Lyapunov function of the form

$$V_{i,\theta}(x) = v_{i,\theta}(x) - v_{i,\theta}(0), \quad (4)$$

where  $v_{i,\theta}(\cdot)$  is a multilayer perceptron (MLP) with parameter vector  $\theta$ . This formulation guarantees that  $V_{i,\theta}(0) = 0$ , a necessary property for any Lyapunov function candidate.

To enforce that the candidate function  $V_{i,\theta}(x)$  satisfies the remaining **mode-wise Lyapunov function properties** over the entire working domain  $\mathcal{D}$ , we introduce the following loss function.

**Definition 3 (Mode-Wise Loss Functions).** *Let  $\rho$  be a sampling distribution defined on  $\mathcal{D}$ . The mode-wise loss function for mode  $i$  is defined by:*

- *Continuous-time case:*

$$\mathcal{L}_\rho^{\text{ct}}(\theta; i) = \mathbb{E}_{x \sim \rho(\mathcal{D})} \left[ \max \left\{ 0, L_{f_i} V_{i,\theta}(x) + \epsilon \right\} \right] + \alpha \mathbb{E}_{x \sim \rho(\mathcal{D})} \left[ \max \left\{ 0, \epsilon - V_{i,\theta}(x) \right\} \right]. \quad (5)$$

- *Discrete-time case:*

$$\mathcal{L}_\rho^{\text{dt}}(\theta; i) = \mathbb{E}_{x_k \sim \rho(\mathcal{D})} \left[ \max \left\{ 0, V_{i,\theta}(f_i(x_k)) - V_{i,\theta}(x_k) + \epsilon \right\} \right] + \alpha \mathbb{E}_{x_k \sim \rho(\mathcal{D})} \left[ \max \left\{ 0, \epsilon - V_{i,\theta}(x_k) \right\} \right]. \quad (6)$$

Note that, the above defined loss functions depend on the distribution  $\rho$  and the expectation is difficult to compute in general. In practice, the loss is computed in a data-driven (empirical) fashion by sampling  $N$  points  $\{x_j\}_{j=1}^N$  from  $\mathcal{D}$  according to  $\rho$ , leading to the following **empirical loss functions** defined as follows:

- *Continuous-time case:*

$$\mathcal{L}_{N,\rho}^{\text{ct}}(\theta; i) = \frac{1}{N} \sum_{j=1}^N \left[ \max \left\{ 0, L_{f_i} V_{i,\theta}(x_j) + \epsilon \right\} \right] + \alpha \frac{1}{N} \sum_{j=1}^N \left[ \max \left\{ 0, \epsilon - V_{i,\theta}(x_j) \right\} \right]. \quad (7)$$

- *Discrete-time case:*

$$\mathcal{L}_{N,\rho}^{\text{dt}}(\theta; i) = \frac{1}{N} \sum_{j=1}^N \left[ \max \left\{ 0, V_{i,\theta}(f_i(x_j)) - V_{i,\theta}(x_j) + \epsilon \right\} \right] + \alpha \frac{1}{N} \sum_{j=1}^N \left[ \max \left\{ 0, \epsilon - V_{i,\theta}(x_j) \right\} \right]. \quad (8)$$

Intuitively, minimizing either  $\mathcal{L}_\rho^{\text{ct}}(\theta; i)$  or its empirical counterpart  $\mathcal{L}_{N,\rho}^{\text{ct}}(\theta; i)$  ensures that the candidate Lyapunov function satisfies  $L_{f_i} V_{i,\theta}(x) \leq -\epsilon$  and  $V_{i,\theta}(x) \geq \epsilon$ ,

for all  $x \in \mathcal{D}$  (except at the equilibrium  $x = 0$ ), thereby enforcing the desired Lyapunov conditions for mode  $i$ . Similarly, minimizing  $\mathcal{L}_\rho^{\text{dt}}(\theta; i)$  or  $\mathcal{L}_{N,\rho}^{\text{dt}}(\theta; i)$  ensures the discrete-time system satisfies  $V_{i,\theta}(f_i(x_k)) \leq V_{i,\theta}(x_k) - \epsilon$  and  $V_{i,\theta}(x_k) \geq \epsilon$ , for all  $x_k \in \mathcal{D}$ , thus enforcing the Lyapunov conditions in the discrete-time case.

##### B. Switching Losses

In region  $\mathcal{D}_{ij}$ , where the system is allowed to switch from mode  $i$  to mode  $j$ , the desired property is that

$$V_{i,\theta}(x) > V_{j,\theta}(x), \quad \forall x \in \mathcal{D}_{ij}.$$

To enforce this requirement during the training process, we introduce the following loss function.

**Definition 4 (Switching Loss Functions).** *Let  $\rho$  be a sampling distribution defined on  $\mathcal{D}$  and  $\epsilon > 0$  be a prescribed margin parameter. For a given margin  $\epsilon > 0$ , the switching loss is defined by:*

- *Continuous-time case:*

$$\mathcal{L}_\rho^{(i,j),\text{ct}}(\theta) = \mathbb{E}_{x \sim \rho(\mathcal{D}_{ij})} \left[ \max \left\{ 0, V_{j,\theta}(x) - V_{i,\theta}(x) + \epsilon \right\} \right]. \quad (9)$$

- *Discrete-time case:*

$$\mathcal{L}_\rho^{(i,j),\text{dt}}(\theta) = \mathbb{E}_{x_k \sim \rho(\mathcal{D}_{ij})} \left[ \max \left\{ 0, V_{j,\theta}(f_j(x_k)) - V_{i,\theta}(x_k) + \epsilon \right\} \right]. \quad (10)$$

Similar to 3, we can define an empirical version of switching decrease functions by sampling  $N$  points  $\{x_k\}_{k=1}^N$  from  $\mathcal{D}_{ij}$  according to  $\rho$ .

- Continuous-time case:

$$\mathcal{L}_{N,\rho}^{(i,j),\text{ct}}(\theta) = \frac{1}{N} \sum_{k=1}^N \max\left\{0, V_{j,\theta}(x_k) - V_{i,\theta}(x_k) + \epsilon\right\}. \quad (11)$$

- Discrete-time case:

$$\mathcal{L}_{N,\rho}^{(i,j),\text{dt}}(\theta) = \frac{1}{N} \sum_{k=1}^N \max\left\{0, V_{j,\theta}(f_j(x_k)) - V_{i,\theta}(x_k) + \epsilon\right\}. \quad (12)$$

### C. Verification and Counter-Example Generations

In our work, we adopt a Counter-Example Guided Inductive Synthesis (CEGIS) framework to ensure that the candidate Lyapunov functions satisfy the desired properties. Let  $x^*$  denote the equilibrium point. To avoid pathological numerical issues such as arithmetic underflow near  $x^*$ , we exclude a ball of radius  $\epsilon_b > 0$  centered at  $x^*$ . Specifically, we define the verification region as  $\mathcal{D}_V = \{x \in \mathcal{D} \mid \|x - x^*\| \geq \epsilon_b\}$ .

Within  $\mathcal{D}_V$ , the candidate Lyapunov functions must satisfy the mode-wise Lyapunov properties and the switching decrease properties. We encapsulate the failure of these properties through the violation predicate  $\Phi(x)$ . For  $x \in \mathcal{D}_V$ , we define:

- **Continuous-time case:**

$$\Phi(x) := \left( \bigvee_{i \in \mathcal{I}} \left\{ V_i(x) \leq 0 \vee L_{f_i} V_i(x) \geq 0 \right\} \right) \vee \left( \bigvee_{(i,j)} \left[ x \in \mathcal{D}_{ij} \wedge \{V_j(x) \geq V_i(x)\} \right] \right). \quad (13)$$

- **Discrete-time case:**

$$\Phi(x) := \left( \bigvee_{i \in \mathcal{I}} \left\{ V_i(x) \leq 0 \vee [V_i(f_i(x)) - V_i(x) \geq 0] \right\} \right) \vee \left( \bigvee_{(i,j)} \left[ x \in \mathcal{D}_{ij} \wedge \{V_j(x) \geq V_i(x)\} \right] \right). \quad (14)$$

That is, for every  $x \in \mathcal{D}_V$ , if there exists some mode  $i$  for which either  $V_i(x) \leq 0$  or the corresponding Lyapunov decrease condition (i.e.,  $L_{f_i} V_i(x) < 0$  in the continuous case or  $V_i(f_i(x)) - V_i(x) < 0$  in the discrete case) is violated, then  $\Phi(x)$  evaluates to true. In addition, for any switching region  $\mathcal{D}_{ij}$ , if there exists  $x \in \mathcal{D}_{ij}$  such that  $V_j(x) \geq V_i(x)$ , then  $\Phi(x)$  is true.

An SMT solver is employed to search for any  $x_e \in \mathcal{D}_V$  that satisfies  $\Phi(x_e)$ . If such a counter-example  $x_e$  is found, it is incorporated into the training set for further refinement of the candidate Lyapunov functions. Otherwise, if no counter-example exists, the candidate is deemed to satisfy the required properties. This verification step is executed after each iteration within our CEGIS loop, ensuring that the candidate Lyapunov functions conform to the mode-wise and switching

decrease properties (with the discrete-time case enforcing  $x_{k+1} = f_j(x_k)$  upon mode switching).

As discussed in [15], the main purpose of excluding the region  $B_{\epsilon_b}(x^*)$  from the verification process is to mitigate numerical sensitivity issues (e.g., arithmetic underflow) while preserving the properties of the Lyapunov level sets and the regions of attraction outside this excluded ball.

### D. Training Algorithm

Combining the two components described above, we define the overall empirical loss function  $\mathcal{L}$  as follows. For the continuous-time case, the overall loss is given by

$$\mathcal{L}^{\text{ct}}(\theta) = \sum_i \mathcal{L}_{N,\rho}^{\text{ct}}(\theta; i) + \beta \sum_{(i,j)} \mathcal{L}_{N_{ij},\rho}^{(i,j),\text{ct}}(\theta), \quad (15)$$

where  $\mathcal{L}_{N,\rho}^{\text{ct}}(\theta; i)$  is the empirical loss for mode  $i$  (see Equation (7) in the previous section) and  $\mathcal{L}_{N_{ij},\rho}^{(i,j),\text{ct}}(\theta)$  is the empirical switching loss for the mode pair  $(i, j)$  defined over  $\mathcal{D}_{ij}$ . In the discrete-time case, the overall loss is defined analogously:

$$\mathcal{L}^{\text{dt}}(\theta) = \sum_i \mathcal{L}_{N,\rho}^{\text{dt}}(\theta; i) + \beta \sum_{(i,j)} \mathcal{L}_{N_{ij},\rho}^{(i,j),\text{dt}}(\theta), \quad (16)$$

with the additional requirement that when switching from mode  $i$  to mode  $j$  we have  $x_{k+1} = f_j(x_k)$ .

The overall training procedure is summarized in Algorithm 1. In brief, we sample states from  $\mathcal{D}$  and the switching regions  $\mathcal{D}_{ij}$  according to the distribution  $\rho$ , compute the overall empirical loss  $\mathcal{L}(\theta)$ , and update the network parameters  $\theta$  via gradient descent. Subsequently, as detailed in Section IV-C, an SMT solver is employed to verify that the candidate Lyapunov functions satisfy both the mode-wise Lyapunov properties and the switching decrease conditions over the verification region  $\mathcal{D}_V = \{x \in \mathcal{D} \mid \|x - x^*\| \geq \epsilon_b\}$ , (with  $x_{k+1} = f_j(x_k)$  enforced in the discrete-time case). If a counterexample is found, it is incorporated into the training set and the process is repeated until no counterexamples exist within the specified tolerance.

This training algorithm integrates the empirical loss from both the individual mode conditions and the switching requirements, followed by a verification step via an SMT solver. The incorporation of CEGIS ensures that the training progressively refines the candidate Lyapunov functions until the desired conditions are satisfied. According to remark 1, if we can obtain a set of neural multiple Lyapunov functions from algorithm 1, then for any initial state  $x(0) \in \mathcal{D}$  (or  $x_0 \in \mathcal{D}$  in the discrete case), the corresponding solution will finally enter and remain within  $\mathcal{B}_{\epsilon_b}$ .

## V. EXPERIMENTAL RESULTS

We demonstrate the correctness and efficiency of Algorithm 1 across a variety of experiments. All experiments were conducted on a single NVIDIA GeForce RTX 3090 GPU. Furthermore, in the verification step, we employ the SMT solver *dReal* to solve nonlinear, non-polynomial disjunctive constraint systems. All codes are available at [https://github.com/JunyueHuang/Neural\\_MLF](https://github.com/JunyueHuang/Neural_MLF).

---

**Algorithm 1** Training Algorithm for Mode-wise Neural Lyapunov Functions (CEGIS)

---

- 1: **Input:**  $\theta_0$ , sample sets  $S \subset \mathcal{D}$ ,  $\{S_{ij}\}$  for switching regions  $\mathcal{D}_{ij}$ , tolerance  $\delta > 0$
  - 2:  $\theta \leftarrow \theta_0$
  - 3: **repeat**
  - 4: For each mode  $i$ , compute  $\mathcal{L}_{N,\rho}^*(\theta; i)$  on  $S$ ,  $* \in \{\text{ct}, \text{dt}\}$
  - 5: For each mode pair  $(i, j)$  with switching region  $\mathcal{D}_{ij}$ , compute  $\mathcal{L}_{N_{ij},\rho}^{(i,j),*}(\theta)$  on  $S_{ij}$
  - 6: Update  $\theta \leftarrow \theta - \eta \nabla_{\theta} [\sum_i \mathcal{L}_{N,\rho}^*(\theta; i) + \beta \sum_{(i,j)} \mathcal{L}_{N_{ij},\rho}^{(i,j),*}(\theta)]$
  - 7: **Verification:** use SMT solver (Sec. IV-C) to check Lyapunov and switching conditions over  $\mathcal{D}_V$ ; if counterexamples  $X^*$  found, add them to  $S$  or  $S_{ij}$  accordingly
  - 8: **until** no counterexamples in  $\mathcal{D}_V$  within  $\delta$
  - 9: **Output:**  $\theta$
- 

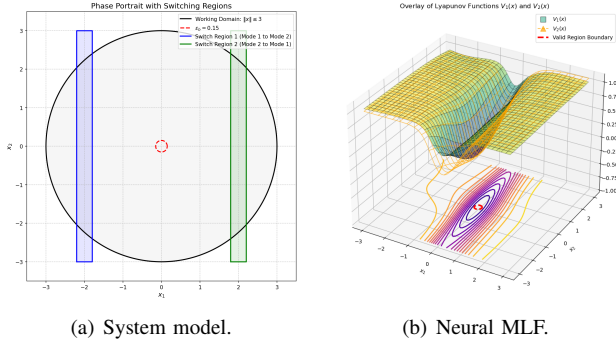


Fig. 1: Experiment in Section V.A.

### A. Pendulum Systems

In the first experiment, we consider a nonlinear continuous-time pendulum system operating on a two-dimensional state  $x = [s, v]^T$  within the working domain  $\mathcal{D} = \{x \in \mathbb{R}^2 : \|x\| \leq 3\}$ . The system dynamic is given by  $\dot{s} = v$ ,  $\dot{v} = \frac{-mGL \sin(s) - bv}{mL^2}$ , where  $s$  and  $v$  denote the angular position and velocity, respectively, and  $m$ ,  $G$ , and  $L$  represent the mass, gravitational constant, and pendulum length.

We consider two operating modes of the system:

- **mode 1** with damping  $b = 0.1$ ; and
- **mode 2** with damping  $b = 0.3$ .

Switching regions are defined as: from mode 1 to mode 2, we have  $\mathcal{D}_{12} = [-2.2, -1.8, -3, 3]$ ; and from mode 2 to mode 1, we have  $\mathcal{D}_{21} = [1.8, 2.2, -3, 3]$  (see Fig. 1(a)). Furthermore, a ball of radius  $\epsilon_b = 0.15$  centered at the equilibrium is excluded during verification to mitigate numerical issues.

By applying our training algorithm, we successfully synthesize a multiple neural Lyapunov functions. The values of these two functions are shown graphically in Fig. 1(b), which satisfy the prescribed conditions and therefore, verify the stability of the system.

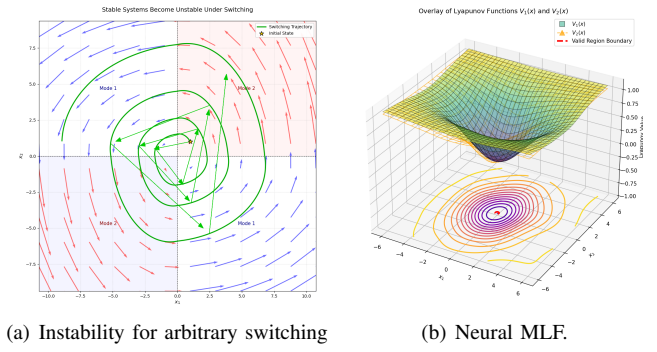


Fig. 2: Experiment in Section V.B.

### B. No Common Lyapunov Function Exists

In the second experiment, we consider two linear systems with dynamics  $\dot{x} = A_i x$ ,  $i = 1, 2$ , where

$$A_1 = \begin{bmatrix} -0.2 & -2.0 \\ 1.0 & -0.2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -0.2 & -1.0 \\ 2.0 & -0.2 \end{bmatrix}.$$

The working domain is defined as  $\mathcal{D} = \{x \in \mathbb{R}^2 \mid \|x\| \leq 6\}$ , with a circular region of radius  $\epsilon_b = 0.5$  (centered at the origin) excluded. Note that, under arbitrary switching between modes without state-constraint, the overall system becomes unstable (see Fig. 2(a)), indicating that no common Lyapunov function exists.

Here, we further consider a state-dependent switching setting defined by:

- from mode 1 to mode 2, the switching region is  $\mathcal{D}_{12} = [-0.2, 0.2, -2.2, -1.8]$ ; and
- from mode 2 to mode 1, the switching region is  $\mathcal{D}_{21} = [1.8, 2.2, -0.2, 0.2]$ .

By applying our training algorithm, we can still synthesize a multiple neural Lyapunov function as shown graphically in Fig. 2(b). Therefore, we can claim that this system, which is not stable under arbitrary switching, is stable under the given state-dependent switching setting.

### C. Multi-Mode-Switching

Finally, we consider experiments for switched systems with more than two modes.

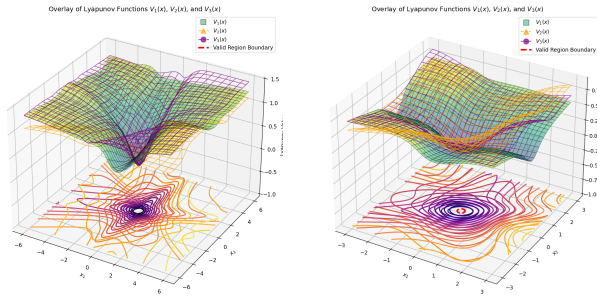
1) *Discrete-Time Case:* We consider a switched system with dynamics  $\dot{x} = A_i x$ ,  $i = 1, 2, 3$ , where

$$A_1 = \begin{bmatrix} 0.6 & -0.1 \\ 0.2 & 0.8 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.6 & 0.1 \\ 0 & 0.6 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0.7 & 0.1 \\ -0.3 & 0.7 \end{bmatrix}.$$

The working domain is defined as  $\mathcal{D} = \{x \in \mathbb{R}^2 \mid \|x\| \leq 6\}$ , with a circular region of radius  $\epsilon_b = 0.5$  (centered at the origin) excluded. The switching regions are defined as:

- from mode 1 to mode 2,  $\mathcal{D}_{12} = [-2.5, -1.5, -0.5, 0.5]$ ;
- from mode 2 to mode 3,  $\mathcal{D}_{23} = [1.5, 2.5, -0.5, 0.5]$ ; and
- from mode 3 to mode 1,  $\mathcal{D}_{31} = [-0.5, 0.5, -2.5, -1.5]$ .

The switching yields overall stable behavior as our learning algorithm successfully finds a multiple neural Lyapunov function as shown graphically in Fig. 3(a).



(a) Neural MLF for discrete case. (b) Neural MLF for continuous case.

Fig. 3: Experiment in Section V.C.

2) *Continuous Case*: Finally, we consider a nonlinear system with three switching modes defined as:

$$\dot{x} = f(x) = \begin{cases} [-s + 2s^2v, -v]^T & \text{(Mode 1)} \\ [-s, -2v + 0.1sv^2]^T & \text{(Mode 2)} \\ [-3s - 0.1sv^3, -v]^T & \text{(Mode 3)} \end{cases}$$

where  $x = [s, v]^T \in \mathcal{D} = \{x \in \mathbb{R}^2 : \|x\| \leq 3\}$ . The switching regions are defined as:

- from mode 1 to mode 2,  $\mathcal{D}_{12} = [-0.5, 0.5, 1.5, 2.5]$ ;
- from mode 2 to mode 3,  $\mathcal{D}_{23} = [1.5, 2.5, -0.5, 0.5]$ ; and
- from mode 3 to mode 1,  $\mathcal{D}_{31} = [-0.5, 0.5, -2.5, -1.5]$ .

As shown in Fig. 3(b), excluding a circular region  $\epsilon_b = 0.15$  around the origin, the switching yields overall stable behavior.

## VI. CONCLUSION

This paper presented a neural multiple Lyapunov function (NMLF) framework for stability analysis of switched systems with state-dependent switching, combining the theoretical guarantees of multiple Lyapunov functions with the approximation power of neural networks. We developed a practical training algorithm and validated its effectiveness through case studies. Future work includes extending the framework to broader classes of switched systems, such as those with language-constrained switching, and adapting it for control synthesis to jointly design control laws and switching strategies for stabilization.

## REFERENCES

- [1] S. Veer and I. Poulakakis, “Switched systems with multiple equilibria under disturbances: Boundedness and practical stability,” *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2371–2386, 2020.
- [2] L. Fribourg and R. Soulat, *Control of switching systems by invariance analysis: application to power electronics*. John Wiley & Sons, 2013.
- [3] A. N. Vargas, E. F. Costa, L. Acho, and J. B. do Val, “Switching stochastic nonlinear systems with application to an automotive throttle,” *IEEE Transactions on Automatic Control*, vol. 63, no. 9, pp. 3098–3104, 2018.
- [4] A. A. Ahmadi, R. M. Jungers, P. A. Parrilo, and M. Roozbehani, “Joint spectral radius and path-complete graph Lyapunov functions,” *SIAM Journal on Control and Optimization*, vol. 52, no. 1, pp. 687–717, 2014.
- [5] Y. Wang, N. Roohi, G. E. Dullerud, and M. Viswanathan, “Stability analysis of switched linear systems defined by regular languages,” *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2568–2575, 2016.

- [6] M. Fiacchini, M. Jungers, and A. Girard, “Stabilization and control Lyapunov functions for language constrained discrete-time switched linear systems,” *Automatica*, vol. 93, pp. 64–74, 2018.
- [7] S. Yang, Y. Chen, X. Yin, and R. Mangharam, “Learning local control barrier functions for safety control of hybrid systems,” *arXiv:2401.14907*, 2024.
- [8] X. Yin, B. Gao, and X. Yu, “Formal synthesis of controllers for safety-critical autonomous systems: Developments and challenges,” *Annual Reviews in Control*, vol. 57, p. 100940, 2024.
- [9] M. S. Branicky, “Multiple Lyapunov functions and other analysis tools for switched and hybrid systems,” *IEEE Transactions on automatic control*, vol. 43, no. 4, pp. 475–482, 1998.
- [10] Z. She and B. Xue, “Discovering multiple Lyapunov functions for switched hybrid systems,” *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 3312–3340, 2014.
- [11] L. Long, “Multiple Lyapunov functions-based small-gain theorems for switched interconnected nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3943–3958, 2017.
- [12] A. Papachristodoulou and S. Prajna, “On the construction of Lyapunov functions using the sum of squares decomposition,” in *IEEE Conference on Decision and Control (CDC)*, vol. 3, pp. 3482–3487, 2002.
- [13] W. Tan and A. Packard, “Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming,” *IEEE Transactions on Automatic Control*, vol. 53, no. 2, pp. 565–571, 2008.
- [14] Y. Chen, C. Shang, X. Huang, and X. Yin, “Data-driven safe controller synthesis for deterministic systems: A posteriori method with validation tests,” in *IEEE Conference on Decision and Control (CDC)*, pp. 7988–7993, 2023.
- [15] Y.-C. Chang, N. Roohi, and S. Gao, “Neural Lyapunov control,” *Advances in neural information processing systems*, vol. 32, 2019.
- [16] A. Abate, D. Ahmed, M. Giacobbe, and A. Peruffo, “Formal synthesis of Lyapunov neural networks,” *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 773–778, 2020.
- [17] S. M. Richards, F. Berkenkamp, and A. Krause, “The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems,” in *Conference on robot learning*, pp. 466–476, PMLR, 2018.
- [18] J. Wang and M. Fazlyab, “Actor-critic physics-informed neural Lyapunov control,” *IEEE Control Systems Letters*, vol. 8, pp. 1751–1756, 2024.
- [19] A. Shirin, M. Martínez-Ramón, and R. Fierro, “Kernel machine to estimate a Lyapunov function and region of attraction (roa) for nonlinear systems,” *IEEE Access*, vol. 11, pp. 59652–59660, 2023.
- [20] H. Dai, B. Landry, L. Yang, M. Pavone, and R. Tedrake, “Lyapunov-stable neural-network control,” in *Robotics: Science and Systems (RSS)*, 2021.
- [21] R. C. B. Rego and F. M. U. de Araújo, “Learning-based robust neuro-control: A method to compute control Lyapunov functions,” *International Journal of Robust and Nonlinear Control*, vol. 32, no. 5, pp. 2644–2661, 2022.
- [22] N. Gaby, F. Zhang, and X. Ye, “Lyapunov-net: A deep neural network architecture for Lyapunov function approximation,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 2091–2096, 2022.
- [23] P. Samanipour and H. A. Poonawala, “Automated stability analysis of piecewise affine dynamics using vertices,” in *2023 59th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1–8, 2023.
- [24] A. Jena, T. Huang, S. Sivaranjani, D. Kalathil, and L. Xie, “Distributed learning of neural Lyapunov functions for large-scale networked dissipative systems,” 2022.
- [25] B. Niu, Y. Liu, W. Zhou, H. Li, P. Duan, and J. Li, “Multiple Lyapunov functions for adaptive neural tracking control of switched nonlinear nonlower-triangular systems,” *IEEE Transactions on Cybernetics*, vol. 50, no. 5, pp. 1877–1886, 2019.
- [26] L. Long, “Multiple Lyapunov functions-based adaptive neural network tracking control of uncertain switched nonlinear systems,” *International Journal of Robust and Nonlinear Control*, vol. 29, no. 13, pp. 4577–4593, 2019.
- [27] D. Yang, G. Zong, Y. Liu, and C. K. Ahn, “Adaptive neural network output tracking control of uncertain switched nonlinear systems: An improved multiple Lyapunov function method,” *Information Sciences*, vol. 606, pp. 380–396, 2022.
- [28] M. Branicky, “Multiple Lyapunov functions and other analysis tools for switched and hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 475–482, 1998.