Learning Provable Neural Network Observer for Uncertain Dynamical Systems

Zhangyi Wang[1] **Jiaxu Liu**[2,*] **Song Chen**[3,*] **Chao Xu**[1] **Shengze Cai**[1,*]
[1]College of Control Science and Engineering, Zhejiang University, China
[2]School of Science, Huzhou Normal University, China
[3]Department of Mathematics, National University of Singapore, Singapore

Abstract

In many safety-critical applications, control of uncertain dynamical systems relies on observers that estimate states and external disturbances. Neural network observers can improve estimation accuracy, but certifying their Lyapunov stability via Linear Matrix Inequality (LMI) constraints leads to large-scale semidefinite programs (SDPs) that are difficult to solve for large networks. To overcome this scalability bottleneck, we propose a novel two-stage training framework for provably stable neural network observers. Our approach decouples the optimization into **a point-guided Lyapunov pre-training phase**, which rapidly achieves high estimation accuracy and local stability over sampled states, followed by **an LMI fine-tuning phase** that efficiently satisfies a strict global Lyapunov stability certificate. We provide formal theoretical guarantees for local stability radii and probabilistic coverage over a prescribed compact error-state domain under specified regularity and sampling assumptions. Experiments on nonlinear control benchmarks and X-29 aircraft ablations show that our LMI-certified neural network observers train significantly faster than direct LMI-based methods and generalize robustly across diverse systems, achieving improved tracking accuracy over a range of observer baselines. The code is available at `https://github.com/Berry-Myon/LearningNeuralNetworkObserver`.

1 Introduction

The control and observation of real-world dynamical systems are fundamentally challenged by the presence of external disturbances and unmodeled dynamics, which often appear as complex black-box effects in practical engineering Banaszuk et al. [2011], Das et al. [2025]. This challenge is particularly critical in safety-critical domains such as plasma magnetic control De Tommasi [2019], Tartaglione et al. [2022] and high-precision trajectory tracking for autonomous aerial vehicles Shi et al. [2019], Tian et al. [2023], Alqudsi [2024]. As demonstrated in recent studies on Neural Lander Shi et al. [2019], accurately estimating state-dependent uncertainties is essential for maintaining stability in unpredictable environments. Therefore, developing robust state and disturbance estimation techniques is critical for safely tracking desired trajectories and maintaining overall system stability in such environments Alan et al. [2022], Izadi and Faieghi [2024], Wang and Xu [2023].

In uncertain dynamical systems, many observer designs for state estimation have been developed with rigorous theoretical guarantees. Classical approaches such as Sliding Mode Observers (SMO) Xiong and Saif [2001], Tan and Edwards [2001], Shtessel et al. [2014], Extended State Observers (ESO) Talole et al. [2009], Guo and Zhao [2011], Pu et al. [2015], and H_∞-based filters Karimi [2009], Zhao and Mili [2018, 2019] provide robustness against disturbances and modeling errors, but typically rely on linear approximations or specific structural assumptions on the uncertainty. As a result, their performance degrades when faced with strong nonlinearities or significant model-plant mismatch. In parallel, machine learning has increasingly been applied to estimation and

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

control Deisenroth and Rasmussen [2011], Berkenkamp et al. [2017], Chow et al. [2018], Hewing et al. [2020], with neural network observers emerging as a powerful tool for capturing complex, unstructured uncertainties Chen et al. [2017], Hou et al. [2022], Yang et al. [2024]. Most relevant to our setting, Chen et al. [2023] introduced a neural observer for uncertain nonlinear systems that augments the observer dynamics with a neural network term and derives a Lyapunov stability certificate through a Linear Matrix Inequality (LMI) condition on the network parameters, thereby showing that expressive neural observers can be made provably stable. Complementary analyses based on quadratic constraints and semidefinite programming (SDP) relaxations have provided tractable stability conditions for dynamical systems with neural network controllers Yin et al. [2021], Fazlyab et al. [2019], establishing an important analytical foundation for certifying neural architectures in feedback loops. Dawson et al. [2023] further surveyed neural Lyapunov, barrier, and contraction certificate methods, emphasizing both the promise of learned certificates and the practical difficulty of scaling formal guarantees to complex neural models. However, current neural network controller and observer frameworks face a critical computational bottleneck: certifying Lyapunov stability typically requires solving LMIs A. Mozelli and S. Adriano [2019], Fazlyab et al. [2019], Pauli et al. [2022]. For high-dimensional systems or large-scale networks, the resulting SDP tasks become extremely slow and often numerically intractable Nie and Wang [2012], Fazlyab et al. [2019], Pauli et al. [2022].

In response, related machine learning work has pursued direct Lyapunov-guided learning of certified neural controllers Chang et al. [2019], Zhou et al. [2022], Wu et al. [2023], while recent neural observer approaches often avoid direct end-to-end certificate-constrained optimization and instead rely on falsification, mixed-integer or satisfiability modulo theories (SMT) verification, or post-hoc certification Yang et al. [2024]. Inspired by the expressive power of deep learning and the need for scalable certification, this paper proposes a new framework to resolve this conflict.

Main Contributions. Our work builds on classical Lyapunov stability analysis and LMI-based certification techniques for nonlinear dynamical systems, and develops a general training framework for large-scale, LMI-constrained learning architectures. Specializing this framework to neural network observers for uncertain systems, our main contributions are:

- **A scalable two-stage training framework (Section 3.1).** We propose a two-stage pipeline that decouples expressive observer learning from global stability certification: Stage I performs point-guided Lyapunov pre-training over sampled error states to quickly learn a high-capacity neural network observer with strong empirical estimation performance, while Stage II starts from the above warm initialization and applies a lightweight LMI fine-tuning step to recover a strict global Lyapunov stability certificate. By avoiding full-scale LMI-constrained optimization throughout training, this framework makes LMI-certified neural network observers practical for deep, high-capacity architectures beyond the reach of standard SDP-based methods.
- **Point-guided Lyapunov pre-training with theoretical characterization (Section 3.2).** We introduce a sampling-based Lyapunov loss that enforces strict Lyapunov decrease at sampled error states, thereby giving Stage I a stability-oriented objective that provides a Lyapunov-based warm start for LMI certification. Under mild Lipschitz assumptions on the observer nonlinearity, we show that this sampled decrease condition induces explicit local stability neighborhoods around each sample and yields a probabilistic coverage interpretation over a prescribed compact error-state domain under a sampling distribution with full support on that domain. These results characterize the local stability structure promoted by point-guided pre-training before the final global LMI certification step (Theorems 1 and 2).
- **Comprehensive empirical validation (Section 4 and Appendix E).** We evaluate the proposed framework on a quadrotor unmanned aerial vehicle (Quad-UAV) under ground effect and an autonomous underwater vehicle (AUV) subject to complex fluid disturbances, comparing against data-driven and classical ESO-based baselines. We report X-29 aircraft training-strategy ablations in Section 4.1 and additional capacity and robustness results in Appendix E. Across these settings, the final trained observers retain formal stability certificates while improving tracking and disturbance rejection.

2 Background and Motivation

This section introduces the uncertain-system setting, the neural network observer studied in this paper, and the Lyapunov stability-certification bottleneck that motivates our two-stage framework.

Consider a general uncertain nonlinear dynamical system subject to external disturbances and unmodeled dynamics, represented as follows:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B_\omega \mathcal{K}(x, d, t), \\ y(t) = Cx(t), \end{cases} \tag{1}$$

where $x(t) \in \mathbb{R}^{n_{in}}$ is the system state, $u(t) \in \mathbb{R}^{n_u}$ is the control input, $y(t) \in \mathbb{R}^{n_{out}}$ is the measured output, $\mathcal{K}(x, d, t)$ encapsulates both the nonlinear dynamics and the external disturbances, and the system matrices A, B, C, B_ω are known and fixed. Since the internal state is often difficult to measure directly, state observation is essential in uncertain systems, and many control laws are built on the observed state. For this reason, the observer is designed to provide accurate state and disturbance estimation so that the closed-loop system can track a desired trajectory $x_d(t)$ under the uncertainty induced by $\mathcal{K}(x, d, t)$.

Following Guo and Zhao [2011], Han [2009], Chen et al. [2023], a neural network observer is designed for augmented-state estimation using the output estimation error as feedback:

$$\begin{cases} \dot{\hat{x}}_1(t) = A\hat{x}_1(t) + Bu(t) + B_\omega \hat{x}_2(t) + \pi_{\theta_1}(\epsilon^{-1}(y(t) - \hat{y}(t))), \\ \dot{\hat{x}}_2(t) = \epsilon^{-1}\pi_{\theta_2}(\epsilon^{-1}(y(t) - \hat{y}(t))), \\ \hat{y}(t) = C\hat{x}_1(t), \end{cases} \tag{2}$$

where $\hat{x}_1(t)$ and $\hat{x}_2(t)$ denote the observer outputs for state estimation and disturbance estimation, respectively. The function $\pi_\theta = [\pi_{\theta_1}^\top, \pi_{\theta_2}^\top]^\top$ is a neural network parameterized by θ, and $\epsilon > 0$ is a tunable scaling factor. A concrete residual-network parameterization of π_θ is provided in Appendix B.

Defining the scaled state estimation error and the lumped disturbance estimation error as $\eta_1 = \epsilon^{-1}(x - \hat{x}_1)$ and $\eta_2 = \mathcal{K}(x, d, t) - \hat{x}_2$, respectively, the observer induces the error dynamics

$$\dot{\eta}(t) = A_\epsilon \eta(t) - \nu(\eta, \theta) + \epsilon w(t), \tag{3}$$

where $\eta = [\eta_1^\top, \eta_2^\top]^\top$, $A_\epsilon = \begin{bmatrix} \epsilon A & B_\omega \\ O & O \end{bmatrix}$, $\nu^\top(\eta, \theta) = \begin{bmatrix} \pi_{\theta_1}(\eta_1) \\ \pi_{\theta_2}(\eta_1) \end{bmatrix}$, and $w(t) \triangleq \frac{d}{dt}\mathcal{K}(x, d, t)$ denotes the time derivative of the lumped disturbance term. This architecture allows the observer to capture high-frequency uncertainties that linear Luenberger-type observers fail to model.

Given the observer error dynamics above, the next question is how to guarantee that the estimation errors decay despite the unknown disturbance term. A natural route is to study a quadratic Lyapunov function for the error system and derive a sufficient decrease condition that can be enforced during training. Specifically, by differentiating $V(\eta) = \eta^\top P \eta$ along the error dynamics and using the disturbance bound, we obtain the pointwise inequality in Eq. (4), leading to the following criterion.

Lemma 1 (Khalil [2002]). Consider the quadratic Lyapunov candidate $V(\eta) = \eta^\top P \eta$ with a symmetric positive definite matrix $P \succ O$. Assume that the lumped disturbance derivative is bounded such that $\|w(t)\| \le M_w$. If there exists a margin $\rho > 0$ such that

$$\eta^\top (A_\epsilon^\top P + P A_\epsilon + \rho P)\eta - 2\nu^\top(\eta, \theta) P \eta + 2\epsilon M_w \|\eta^\top P\| \le 0, \tag{4}$$

holds for all η in the considered domain, then the time derivative of V satisfies $\dot{V} \le -\rho V$, establishing the desired Lyapunov stability of the error system.

Lemma 1 shows that the desired Lyapunov stability property can be guaranteed if the pointwise inequality in Eq. (4) holds over the domain of interest. However, directly checking this condition for all η is difficult because experiments or sampled evaluations cannot exhaust a continuous domain. Prior work therefore replaces the pointwise condition with a sufficient global matrix inequality in the network parameters, stated next.

Lemma 2 (Chen et al. [2023]). The Lyapunov stability condition in Lemma 1 can be enforced via an LMI

$$\mathcal{H}(\theta) \triangleq \hat{R}_\pi^\top(\theta) \begin{bmatrix} A_\epsilon^\top P + P A_\epsilon & -P \\ -P & O \end{bmatrix} \hat{R}_\pi(\theta) + \hat{R}_\xi^\top(\theta)\, \Psi^\top M \Psi\, \hat{R}_\xi(\theta) \prec O, \tag{5}$$

where matrices $\hat{R}_\pi(\theta)$ and $\hat{R}_\xi(\theta)$ are constructed from the neural network weights as detailed in Appendix B, Ψ is the sector-boundedness matrix associated with the activation function, and

$M = \begin{bmatrix} O & \Lambda \\ \Lambda & O \end{bmatrix}$ is the gain matrix with Λ being a positive definite diagonal matrix. For the `tanh` activation, $\Psi = \begin{bmatrix} I & -I \\ O & I \end{bmatrix}$.

Therefore, adopting the above neural network observer for observer-based control and tracking design inevitably requires solving the LMI in Eq. (5) to satisfy the condition in Lemma 2. Although this condition provides a rigorous global Lyapunov stability certificate, solving it for high-capacity neural network observers is extremely challenging in practice.

The difficulty comes from three related aspects. *First*, the estimation performance of neural network observers depends strongly on the expressive power of π_θ: in complex aerospace or fluid dynamics systems, small-scale networks often fail to capture high-frequency uncertainties, whereas increasing network width and depth rapidly enlarges the parameter dimension of θ. *Second*, standard LMI solvers such as interior-point methods scale poorly with the size of the matrix variables, and in Eq. (5) the dimension of $\mathcal{H}(\theta)$ grows directly with the number of neurons and layers, making the resulting SDP computationally prohibitive for high-capacity networks. *Third*, enforcing Eq. (5) directly during gradient-based training is itself numerically unstable and slow, because it turns the optimization over θ into a highly non-convex constrained problem.

Research Objective. How can we train a *high-capacity neural network observer* that still satisfies the *global LMI certificate* in Lemma 2, without paying the *prohibitive cost of solving large-scale SDPs* throughout training? This tension between expressive estimation and scalable certification is exactly what motivates the two-stage strategy developed in Section 3. Once the LMI-constrained Lyapunov certification problem for neural network observers is recast from large-scale SDP solving into a trainable optimization problem, the same framework can be readily extended to broader neural-network-based control systems that require scalable stability certificates.

3 Main Results

In this section, we first present the proposed two-stage training framework for scalable, stability-certified neural network observers, and then provide the associated Lyapunov-based convergence and stability analysis.

3.1 Method Development

The core idea is to decouple *expressive observer learning* from *global stability certification*. We first use point-guided pre-training, then optimize under the full LMI constraint from this initialization.

Stage I: Point-Guided Lyapunov Pre-Training. The first stage learns a high-capacity observer that is locally stable around many sampled error states, without solving a large-scale LMI or SDP.

Let $\mathcal{X}$ be a prescribed compact domain in the error state space and let $\mathcal{D} = \{\eta_i\}_{i=1}^N$ be error states sampled from $\mathcal{X}$ (e.g., from a truncated Gaussian or uniform distribution). We consider the quadratic Lyapunov candidate $V(\eta) = \eta^\top P \eta$ with $P \succ O$ and define the computable Lyapunov upper bound

$$F_V(\eta, \theta) \triangleq \eta^\top (A_\epsilon^\top P + P A_\epsilon + \rho P)\eta - 2\nu(\eta, \theta)^\top P\eta + 2\epsilon M_w \|P\eta\|, \tag{6}$$

which satisfies $\dot{V}(\eta) + \rho V(\eta) \leq F_V(\eta, \theta)$ by Lemma 1. Here, $\rho > 0$ is a desired decay rate and M_w bounds the lumped disturbance derivative. This upper bound suggests a surrogate objective for Stage I that directly penalizes violations of the Lyapunov decrease condition at sampled error states. These pointwise evaluations reduce the computational cost associated with enforcing the global LMI throughout high-capacity network training. In this way, the training objective already encourages the observer toward a stability-consistent region while preserving the efficiency and flexibility of standard gradient-based optimization. Based on this idea, we introduce the *point-guided Lyapunov loss*

$$\mathcal{L}_{\text{point}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left[F_V(\eta_i, \theta) + \xi \right]_+ + \gamma \|\theta\|^2, \tag{7}$$

where $[\cdot]_+ = \max(0, \cdot)$ is the ReLU operation, $\xi > 0$ is a strict Lyapunov margin, and $\gamma > 0$ is a regularization coefficient. As shown formally in Lemma 3, once the data-dependent violation term vanishes, the Lyapunov decrease condition is strictly satisfied at all sampled points.

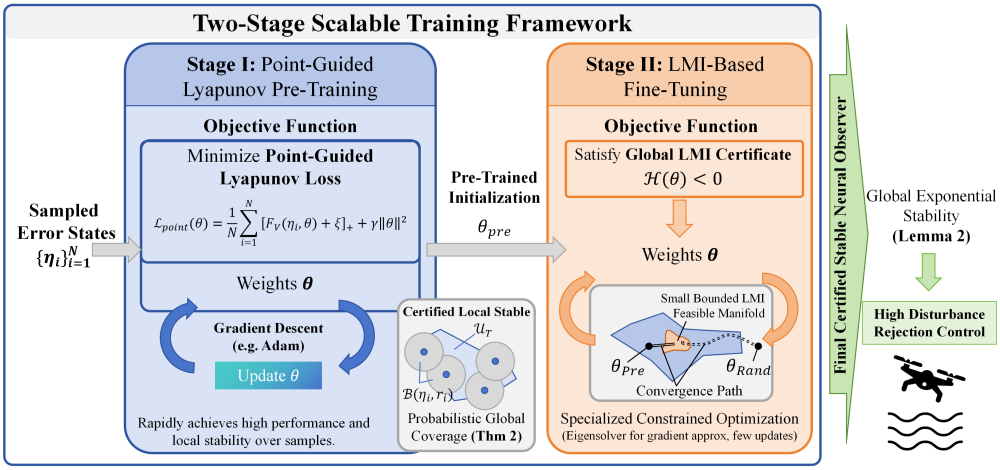


Figure 1: **Two-stage training framework for neural network observers.** Stage I uses sampled error states for point-guided Lyapunov pre-training, and Stage II applies lightweight LMI fine-tuning to obtain an observer satisfying the global LMI-based Lyapunov stability certificate.

In Stage I, illustrated on the left of Figure 1, we sample error states $\{\eta_i\}$ from $\mathcal{X}$ and minimize $\mathcal{L}_{\text{point}}(\theta)$ by stochastic gradient descent to obtain a pre-trained parameter θ_{pre}. The purpose of this pre-training stage is to cheaply inject Lyapunov structure into the observer before imposing the full certificate: it quickly pushes the parameters away from clearly unstable regions and toward a near-feasible one using only pointwise evaluations. As a result, Stage I preserves the scalability of unconstrained neural network training, exploits the expressive power of deep architectures, and provides a much better warm start for the subsequent LMI-based refinement than random initialization, all without solving any SDP.

Stage II: LMI-Based Fine-Tuning. Stage I promotes stability over sampled states, and Stage II enforces the global LMI condition in Lemma 2. The second stage, depicted on the right of Figure 1, therefore starts from θ_{pre} and performs a fine-tuning phase to strictly satisfy the LMI $\mathcal{H}(\theta) \prec O$, thereby recovering the global LMI-based Lyapunov certificate from Lemma 2 for the error system. In the LMI formulation of Chen et al. [2023], once this condition is satisfied, the observation error approaches zero as the gain ϵ is chosen sufficiently small. We formulate the fine-tuning problem as a penalty-based optimization initialized at θ_{pre}:

$$\min_{\theta} \ \mathcal{L}_{\text{LMI}}(\theta) = \phi\big(\lambda_{\max}(\mathcal{H}(\theta))\big) + \beta\|\theta\|^2, \tag{8}$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue, $\phi(\cdot)$ is a monotonically increasing penalty function that heavily penalizes positive eigenvalues (i.e., LMI violations), and $\beta > 0$ is a regularization coefficient.

In Stage II, we initialize at θ_{pre} and fine-tune the network with the LMI penalty until $\mathcal{H}(\theta) \prec O$. Empirically, Stage I often places the parameters closer to the feasible region than random initialization, which can reduce the number of updates needed in this stage.

Computational Advantages and Overall Pipeline. Our two-stage framework, summarized schematically in Figure 1, directly addresses the main scalability bottleneck of prior LMI-based neural network observer training. Stage I uses an inexpensive, sample-based Lyapunov loss to quickly shape the parameters into a stability-oriented region, thereby exploiting the expressive power of deep networks without solving any SDPs. Stage II then starts from θ_{pre} and optimizes the LMI penalty to obtain a strictly feasible θ. As shown empirically in Section 4.1, this warm start can reduce the practical optimization burden of the final certification step.

3.2 Convergence and Stability Analysis

We now analyze the Lyapunov structure induced by the two-stage method. The analysis proceeds in three steps to match the sample-based design of Stage I. *First,* Lemma 3 establishes the direct consequence of minimizing the point-guided loss: once the sample-wise violation term vanishes,

strict Lyapunov decrease holds at all sampled points. *Second,* Theorem 1 and Corollary 1 extend these pointwise guarantees to the regions between samples by showing that each successfully trained sample induces a nontrivial certified neighborhood with a strictly positive lower bound on the local stability radius. *Third,* Theorem 2 gives a probabilistic coverage interpretation of these local regions over the prescribed compact error-state domain under a sampling distribution with full support. The analysis characterizes Stage I geometry. Finite-time optimization and sample-complexity guarantees for training success remain open. Finally, we connect the point-guided loss with the LMI certificate and explain why Stage I can provide a useful initialization for Stage II in practice.

3.2.1 Point-Guided Lyapunov Optimization

According to Lemma 1, if $F_V(\eta, \theta) \leq 0$ holds for all η in the considered domain, then the time derivative of $V(\eta)$ satisfies $\dot{V} \leq -\rho V$, establishing the desired Lyapunov stability of the error system. The point-guided loss in Eq. (7) enforces a strict version of this condition at sampled points.

Lemma 3. Let θ^* be any parameter such that the data-dependent term of $\mathcal{L}_{\text{point}}$ vanishes, i.e., $\frac{1}{N}\sum_{i=1}^{N}\left[F_V(\eta_i, \theta^*) + \xi\right]_+ = 0$. Then, for every sampled point $\eta_i \in \mathcal{D}$, the Lyapunov decrease condition is strictly satisfied: $F_V(\eta_i, \theta^*) \leq -\xi < 0$.

Proof. The proof is provided in Appendix C.1. □

3.2.2 Local Stability Radius and Probabilistic Coverage

Lemma 3 establishes Lyapunov decrease at sampled points. Since neural networks define continuous mappings, this property extends to a neighborhood under mild regularity conditions.

Assumption 1 (Lipschitz Continuity). The neural network $\nu(\eta, \theta)$ is Lipschitz continuous with respect to η on the compact domain $\mathcal{X}$, with Lipschitz constant L_ν, i.e., $\|\nu(\eta_1) - \nu(\eta_2)\| \leq L_\nu \|\eta_1 - \eta_2\|$ for all $\eta_1, \eta_2 \in \mathcal{X}$.

Remark 1. Assumption 1 is used in the analysis of nonlinear observers and learning-based controllers. See, e.g., Khalil [2002], Chen et al. [2023], Fazlyab et al. [2019], Zhang and Fan [2024]. Our analysis uses global Lipschitz continuity of the observer nonlinearity on $\mathcal{X}$, with no Lipschitz smoothness requirement. This regularity condition applies to a broad class of practical observer nonlinearities.

Under Assumption 1, we obtain the following explicit local stability radius.

Theorem 1. Let $Q = A_\epsilon^\top P + P A_\epsilon + \rho P$ and fix the trained parameter θ. Under Assumption 1, suppose the Lyapunov condition is strictly satisfied at a sampled point η_s such that $F_V(\eta_s, \theta) \leq -\xi < 0$. Then there exists a continuous stability neighborhood $\mathcal{B}(\eta_s, r_s) \cap \mathcal{X}$ within which $F_V(\eta, \theta) \leq 0$. When $C_2 > 0$, the radius r_s can be chosen as the positive root of the quadratic equation:

$$C_2 r_s^2 + C_1(\eta_s) r_s + F_V(\eta_s, \theta) = 0, \tag{9}$$

where $C_1(\eta_s)$ and C_2 are nonnegative coefficients whose explicit expressions, including the degenerate case $C_2 = 0$, are provided in Appendix C.2.

Proof. The proof is provided in Appendix C.2. □

Moreover, these radii admit a strictly positive uniform lower bound.

Corollary 1. Assume $F_V(\eta_s, \theta) \leq -\xi < 0$ for all sampled points $\eta_s \in \mathcal{D} \subset \mathcal{X}$ and Assumption 1 holds. Then there exists a constant $\underline{r} > 0$, independent of η_s, such that $r_{\min} \coloneqq \min_{1 \leq s \leq N} r_s \geq \underline{r} > 0$.

Proof. The explicit expression of $\underline{r}$ and its derivation are provided in Appendix C.3. □

These results allow us to interpret the domain-level effect of point-wise training. We call a sampled point η_i *successful* if the trained parameter θ satisfies the strict sampled Lyapunov condition $F_V(\eta_i, \theta) \leq -\xi$. Let $U_T = \bigcup_{i=1}^{T} \left(\mathcal{B}(\eta_i, r_i) \cap \mathcal{X}\right)$ denote the union of the certified local stable regions generated by T successful samples inside the prescribed domain.

Theorem 2. Let $\mathcal{X} \subset \mathbb{R}^{n_\eta}$ be a compact domain in the error-state space of η, where $n_\eta = \dim(\eta)$. Let μ be a probability measure supported on $\mathcal{X}$ such that, for every $z \in \mathcal{X}$ and every $\varrho > 0$, $\mu(\mathcal{B}(z, \varrho) \cap \mathcal{X}) > 0$. Assume $\eta_1, \ldots, \eta_T$ are independent successful samples drawn from μ, and each

sample admits a certified neighborhood radius $r_i \geq r_{\min} > 0$. Then there exist constants $N_{\mathrm{cov}} < \infty$ and $q > 0$, depending only on $\mathcal{X}$, μ, and $r_{\min}$, such that

$$\mathbb{P}(\mathcal{X} \subset U_T) \geq 1 - N_{\mathrm{cov}}(1-q)^T. \tag{10}$$

In particular, as the number of successful samples T approaches infinity, the probability that the certified stable region U_T completely covers the error-state domain $\mathcal{X}$ approaches 1:

$$\lim_{T\to\infty} \mathbb{P}(\mathcal{X} \subset U_T) = 1. \tag{11}$$

Proof. The proof is provided in Appendix C.4. □

A finite-sample consequence of Theorem 2 is provided in Corollary 2 in Appendix C.4.

Remark 2. Theorems 1 and 2 jointly show that enforcing a strict Lyapunov decrease at finitely many sampled error states induces nontrivial local stable neighborhoods around each sample and yields a probabilistic coverage interpretation over the prescribed compact error-state domain used in Stage I. This can be viewed as a sample-driven expansion of a *trusted* stable region for pre-training. Related learning-based control studies, e.g., Chang et al. [2019], Zhou et al. [2022], Zhang and Fan [2024], typically use stronger regularity assumptions such as Lipschitz smoothness. Our analysis derives explicit local radii and a finite-sample coverage bound using Lipschitz continuity of the observer nonlinearity and a full-support sampling assumption. This formulation leads to simpler and more practical regularity conditions. More broadly, this paradigm can also be extended to other stability-certified learning-for-control problems, such as neural network controller synthesis, e.g. Gu et al. [2022], Wu et al. [2023], Yang et al. [2024]. Appendix D gives one example with recurrent neural network controllers.

Remark 3. Let $\Theta_{\mathrm{LMI}} = \{\theta : \mathcal{H}(\theta) \prec O\}$ and $\Theta_{\mathrm{point}} = \{\theta : F_V(\eta, \theta) < 0,\ \forall \eta \in \mathcal{X}\}$. Under the assumptions and certificate variables of Lemma 2, any $\theta \in \Theta_{\mathrm{LMI}}$ also belongs to Θ_{point}, so $\Theta_{\mathrm{LMI}} \subseteq \Theta_{\mathrm{point}}$. In practice, Stage I provides a stability-oriented initialization by promoting pointwise Lyapunov decrease over sampled error states, which can move the parameters closer to the certified set. The global LMI is imposed during Stage II, which starts from θ_{pre} and empirically resolves the remaining LMI violations. Compared with random initialization, this warm start can reduce the practical cost of the final fine-tuning step.

4 Experiments

In this section, we evaluate the proposed framework against representative observer and learning-based control baselines on two nonlinear benchmarks: a Quad-UAV under aerodynamic ground effect and an AUV navigating through dynamic fluid vortices. These experiments emphasize comparison with existing methods and the resulting gains in tracking and disturbance rejection. We provide the X-29 aircraft training-strategy ablations in Section 4.1 and capacity and robustness results in Appendix E. All experiments were conducted on our computing server. See Appendix G.4 for details.

4.1 X-29 Training Efficiency and Ablation Study

We use the X-29 aircraft benchmark Bosworth [1992] to study training efficiency and the contribution of each training stage. The Large NN has 576 hidden neurons. The system matrices, observer architectures, and training settings are given in Appendix G.3. Additional capacity and robustness results are reported in Appendix E.

Table 1 reports the computational cost of enforcing LMI-based stability. Direct LMI solvers work only for tiny networks and become infeasible as capacity grows. For the Large NN, direct solving fails, while pure LMI gradient descent takes 2210.39 seconds. The proposed two-stage method reduces this to 895.31 seconds, giving a $2.5\times$ speedup while still satisfying the global LMI certificate.

Ablation Study: Effect of Point-Guided Pre-Training and LMI Fine-Tuning. We compare three training strategies for the Large NN: *Point-Only*, *LMI-Only*, and the proposed two-stage scheme.

Figure 2 and Table 2 show that Point-Only has the largest final RMSE and leaves LMI certification to Stage II. Its pre-trained parameters provide a favorable initialization that substantially accelerates subsequent LMI fine-tuning. The two-stage method therefore achieves the best final RMSE while reducing training time from 2210.39 seconds to 895.31 seconds.

Table 1: **Computational efficiency and scalability of the two-stage training framework.** Comparison of solving or training time across observer capacities and optimization methods.

Architecture	Total Neurons	Solving/Training Method	Time (s)
Linear	0	Pole Placement	<1
Tiny NN	9	Direct LMI Solver	<1
Small NN	128	Direct LMI Solver	Infeasible
Large NN	576	Direct LMI Solver	Failed
Large NN	576	LMI Gradient Descent (Only)	2210.39
Large NN	**576**	**Point & LMI Tuning (Ours)**	**895.31**

Table 2: **Ablation study on training strategy for the Large NN observer.** Comparison of training time and root mean squared estimation error (RMSE) for point-guided pre-training, pure LMI optimization, and the proposed two-stage Point & LMI Tuning method.

Training Method	Time (s)	RMSE (m)
Point-Guided (Only)	99.53	1.0317
LMI Gradient Descent (Only)	2210.39	0.2659
Point & LMI Tuning (Ours)	**895.31**	**0.2234**

4.2 Quad-UAV under Ground Effect

We next evaluate the method on a Quad-UAV subject to aerodynamic ground effect, a nonlinear setting with strong state-dependent disturbances.

The 12-dimensional state is $x = [p, v, \Omega, \omega]^\top \in \mathbb{R}^{12}$, and the nonlinear dynamics with disturbance f_a are

$$\begin{aligned} \dot{p} &= v, & m\dot{v} &= m\vec{g} + R(f_u + f_a), \\ \dot{R}(\Omega) &= R(\Omega)S(\omega), & J\dot{\omega} &= J\omega \times \omega + \tau_\omega, \end{aligned} \tag{12}$$

where f_u and τ_ω denote thrust and body torques, and $S(\cdot)$ denotes the skew-symmetric matrix operator. Near the ground, rotor downwash generates a cushioning force known as ground effect, modeled as

$$f_G = \frac{g_2 R(z)}{h^2 + g_1} T_{\text{cmd}}, \tag{13}$$

where h is the altitude, $R(z)$ is the z-axis component of the rotation matrix, T_{cmd} is the nominal thrust command, and g_1, g_2 are aerodynamic constants Yang et al. [2025]. This term introduces strong nonlinear disturbances that degrade landing and take-off performance.

We compare three controllers: a *Basic Proportional-Integral-Derivative (PID)* baseline, *PID + Neural Lander* Shi et al. [2019], and *PID + Neural Network Observer* (ours). Neural Lander learns a task-specific feedforward compensator from 40,000 data points, whereas our method estimates f_a online and modifies the control thrust as $T'_{\text{cmd}} = T_{\text{cmd}} - \hat{f}_a$. Training details are provided in Appendix G.1, with observer-error results in Appendix F.1.

Figure 3 and Table 3 report vertical tracking performance for both landing and take-off. In landing, Basic PID incurs a large error of 2.2006 m, while Neural Lander and our observer reduce it to 0.5732 m and 0.1471 m, respectively.

Table 3: **Comparative analysis of tracking errors and data requirements for Quad-UAV control.** Here, "per task" means that once the environment or disturbance changes, new data must be collected and the model must be retrained. The label "once" means that the same trained parameters can be reused across changes in the environment as long as the underlying system is unchanged.

Method	Data Required	Training Time	Landing (m)	Take-off (m)
Basic PID	×	—	2.2006 ± 0.1456	0.1255 ± 0.0266
PID + Neural Lander (Shi et al. [2019])	✓	88.2 s (per task)	0.5732 ± 0.1188	0.2444 ± 0.0297
PID + Neural Network Observer (Ours)	×	**233.8 s (once)**	$\mathbf{0.1471 \pm 0.0602}$	$\mathbf{0.0002 \pm 0.0000}$

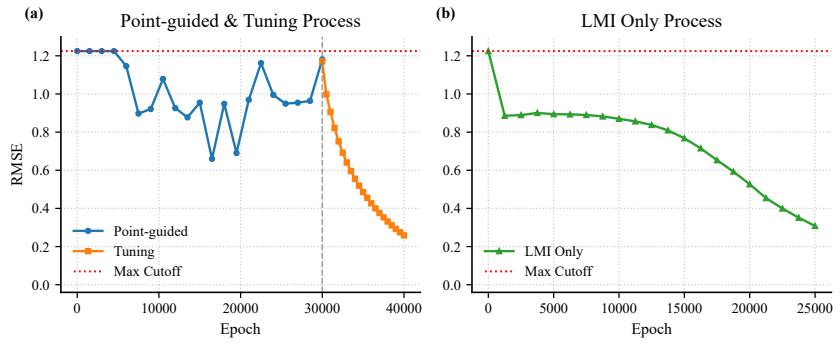


Figure 2: **Ablation on training strategy for the Large NN observer.** RMSE under *Point-Only*, *LMI-Only*, and the proposed two-stage scheme.

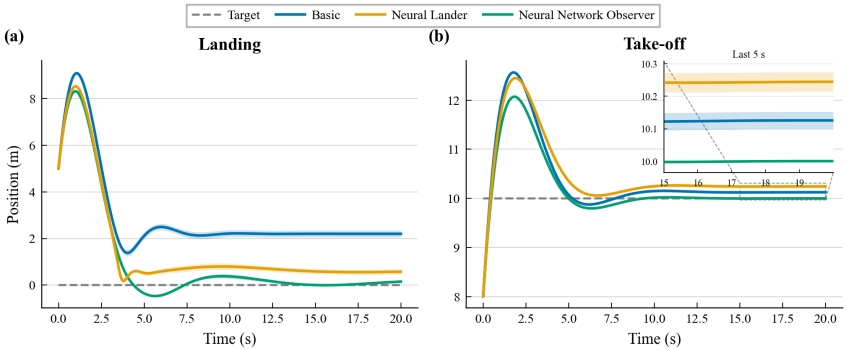


Figure 3: **Generalization performance on a Quad-UAV under ground effect.** Vertical trajectory tracking for landing (a) and take-off (b). Shaded regions denote the mean $\pm$ standard deviation (SD).

The take-off results further highlight generalization. Neural Lander degrades to 0.2444 m error, worse than the Basic PID baseline (0.1255 m), whereas the proposed observer achieves near-perfect tracking with 0.0002 m error after a single offline training stage. This suggests that the observer-based design generalizes across tasks more reliably than the task-specific baseline.

4.3 Disturbance Rejection in Fluid Environment

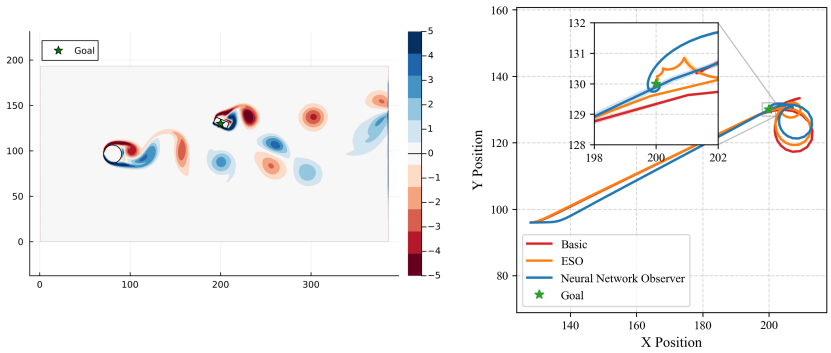
We further investigate the neural network observer in a highly complex and time-varying disturbance field, where an AUV navigates through a dynamic fluid environment. This scenario tests the ability of the method to handle strong, spatially and temporally varying disturbances arising from fluid-structure interactions.

The AUV is modeled as a non-holonomic agent confined to a 2D plane with state $x = [p_x, p_y, \psi, v, \omega]^\top$, representing 2D position, heading angle, linear velocity, and angular velocity. The dynamics are

$$\begin{aligned} \dot{p}_x &= v\cos(\psi), \quad \dot{p}_y = v\sin(\psi), \quad \dot{\psi} = \omega, \\ \dot{v} &= \frac{F + f_a}{m}, \quad \dot{\omega} = \frac{\tau + \tau_a}{J}, \end{aligned} \tag{14}$$

where $u = [F, \tau]^\top$ is the control input, m and J are the mass and moment of inertia, and f_a, τ_a denote unknown fluid-induced forces and torques. The goal is to track a prescribed reference trajectory while rejecting these disturbances.

The surrounding fluid is simulated using the WaterLily engine on a 384×192 grid. The flow has an upstream velocity $U = 10.0$ and Reynolds number $Re = 200$. A fixed cylindrical obstacle of diameter $D_1 = 20$ is placed at $(144, 96)$, generating a von Kármán vortex street in its wake. The AUV, modeled as a 16×8 rectangular body, starts at $(128, 96)$ and must traverse the vortex street to reach a target at $(200, 130)$, experiencing strong, unsteady hydrodynamic forces along the way.



(a) WaterLily Environment (b) X-Y Trajectories

Figure 4: **AUV navigation in a simulated von Kármán vortex street.** (a) The WaterLily fluid environment showing pressure fields and the AUV's path. (b) Trajectory tracking results for different controllers. Shaded regions denote the mean ± SD.

Table 4: **Disturbance rejection performance in dynamic fluid vortices.** Tracking error of different control configurations in the vortex environment.

Method	Tracking Error (m)
Basic NMPC	1.4932 ± 0.0199
NMPC + ESO (Guo and Zhao [2011])	0.0959 ± 0.0088
NMPC + Neural Network Observer (Ours)	$\mathbf{0.0499 \pm 0.0019}$

We compare three control configurations: a *Basic Nonlinear Model Predictive Control (NMPC)* controller without explicit disturbance compensation, an *ESO + NMPC* based on scheme Guo and Zhao [2011], and the proposed *Neural Network Observer + NMPC* framework. The NMPC implementation uses an interior-point optimizer Wächter and Biegler [2006]. All NMPC details and observer parameters are provided in Appendix G.2, with observer-error results in Appendix F.2. Performance is evaluated via the overall trajectory tracking error (mean ± SD) over multiple runs with randomized initial conditions or disturbance realizations.

Table 4 shows that adding ESO reduces the Basic NMPC error from 1.4932 ± 0.0199 m to 0.0959 ± 0.0088 m, while the proposed Neural Network Observer + NMPC further lowers it to 0.0499 ± 0.0019 m, a 48.0% reduction relative to ESO. Figure 4 confirms this improvement qualitatively: the neural observer tracks the reference through the vortex street more closely, indicating stronger disturbance rejection than the classical ESO baseline.

5 Conclusion

In this paper, we present a scalable, provably stable neural network observer framework for state and disturbance estimation in uncertain dynamical systems. To mitigate the computational intractability of enforcing global LMI constraints on high-capacity networks, we introduce a two-stage pipeline: point-guided Lyapunov pre-training followed by LMI fine-tuning. This approach permits the use of deep architectures while maintaining a final global LMI-based Lyapunov certificate. In addition, the Stage I analysis provides explicit local stability radii and a probabilistic coverage interpretation over a prescribed compact error-state domain under a full-support sampling assumption. Evaluations on Quad-UAV and AUV benchmarks indicate improved robustness and generalization under the tested conditions compared to data-driven and classical ESO-based baselines, while the X-29 ablations in Section 4.1 examine training efficiency and Appendix E reports additional capacity and robustness results. By satisfying stability requirements offline without requiring per-task disturbance datasets, our framework provides a practical means for deploying deep learning methods in safety-critical control settings under model uncertainty. More broadly, the same training-and-certification principle can be extended to neural network systems whose Lyapunov stability guarantees are imposed through LMI constraints.

Acknowledgments and Disclosure of Funding

This work was supported by FAST-LAB, College of Control Science and Engineering, Zhejiang University.

References

L. A. Mozelli and R. L. S. Adriano. On computational issues for stability analysis of LPV systems using parameter-dependent Lyapunov functions and LMIs. *International Journal of Robust and Nonlinear Control*, 29(10):3267–3277, 2019. doi: https://doi.org/10.1002/rnc.4528. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.4528`.

A. Alan, T. G. Molnar, E. Daş, A. D. Ames, and G. Orosz. Disturbance observers for robust safety-critical control with control barrier functions. *IEEE Control Systems Letters*, 7:1123–1128, 2022.

Y. Alqudsi. Advanced control techniques for high maneuverability trajectory tracking in autonomous aerial robots. In *2024 1st International Conference on Emerging Technologies for Dependable Internet of Things (ICETI)*, pages 1–8. IEEE, 2024.

A. Banaszuk, V. A. Fonoberov, T. A. Frewen, M. Kobilarov, G. Mathew, I. Mezic, A. Pinto, T. Sahai, H. Sane, A. Speranzon, and A. Surana. Scalable approach to uncertainty quantification and robust design of interconnected dynamical systems. *Annual Reviews in Control*, 35(1):77–98, 2011. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2011.03.005. URL `https://www.sciencedirect.com/science/article/pii/S136757881100006X`.

F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems 30*, 2017.

J. T. Bosworth. Linearized aerodynamic and control law models of the X-29A airplane and comparison with flight data. NASA Technical Memorandum, NASA/STI Accession number: 19920009932, Feb. 1992.

Y.-C. Chang, N. Roohi, and S. Gao. Neural Lyapunov control. In *Advances in Neural Information Processing Systems 32*, 2019.

B. Chen, H. Zhang, X. Liu, and C. Lin. Neural observer and adaptive neural control design for a class of nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9): 4261–4271, 2017.

S. Chen, S. Cai, T. Chen, C. Xu, and J. Chu. Neural observer with Lyapunov stability guarantee for uncertain nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8): 11527–11541, 2023.

Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh. A Lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems 31*, 2018.

A. Das, R. D. Yadav, S. Sun, M. Sun, S. Kaski, and W. Pan. Dronediffusion: Robust quadrotor dynamics learning with diffusion models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1604–1610. IEEE, 2025.

C. Dawson, S. Gao, and C. Fan. Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 39(3): 1749–1767, 2023.

G. De Tommasi. Plasma magnetic control in tokamak devices. *Journal of Fusion Energy*, 38(3): 406–436, 2019.

M. P. Deisenroth and C. E. Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. J. Pappas. Efficient and accurate estimation of Lipschitz constants for deep neural networks. In *Advances in Neural Information Processing Systems 32*, 2019.

F. Gu, H. Yin, L. El Ghaoui, M. Arcak, P. Seiler, and M. Jin. Recurrent neural network controllers synthesis with stability guarantees for partially observed systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5385–5394, 2022. doi: 10.1609/aaai.v36i5.20476.

B.-Z. Guo and Z.-l. Zhao. On the convergence of an extended state observer for nonlinear systems with uncertainty. *Systems & Control Letters*, 60(6):420–430, 2011.

J. Han. From PID to active disturbance rejection control. *IEEE Transactions on Industrial Electronics*, 56(3):900–906, 2009.

L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020.

S. Hou, C. Wang, Y. Chu, and J. Fei. Neural-observer-based terminal sliding mode control: Design and application. *IEEE Transactions on Fuzzy Systems*, 30(11):4800–4814, 2022.

M. Izadi and R. Faieghi. High-gain disturbance observer for robust trajectory tracking of quadrotors. *Control Engineering Practice*, 145:105854, 2024. ISSN 0967-0661. doi: https://doi.org/10.1016/j.conengprac.2024.105854. URL https://www.sciencedirect.com/science/article/pii/S0967066124000145.

H. R. Karimi. Robust H-infinity filter design for uncertain linear systems over network with network-induced delays and output quantization. *Modeling, Identification and Control*, 30(1):27, 2009.

H. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002. ISBN 9780130673893. URL https://books.google.com.tw/books?id=t_d1QgAACAAJ.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

J. Nie and L. Wang. Regularization methods for SDP relaxations in large-scale polynomial optimization. *SIAM Journal on Optimization*, 22(2):408–428, 2012. doi: 10.1137/110825844. URL https://doi.org/10.1137/110825844.

P. Pauli, N. Funcke, D. Gramlich, M. A. Msalmi, and F. Allgöwer. Neural network training under semidefinite constraints. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2731–2736, 2022. doi: 10.1109/CDC51059.2022.9992331.

Z. Pu, R. Yuan, J. Yi, and X. Tan. A class of adaptive extended state observers for nonlinear disturbed systems. *IEEE Transactions on Industrial Electronics*, 62(9):5858–5869, 2015.

G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung. Neural lander: Stable drone landing control using learned dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9784–9790. IEEE, 2019.

Y. Shtessel, C. Edwards, L. Fridman, A. Levant, et al. *Sliding mode control and observation*, volume 10. Springer, 2014.

S. E. Talole, J. P. Kolhe, and S. B. Phadke. Extended-state-observer-based control of flexible-joint system with experimental validation. *IEEE Transactions on Industrial Electronics*, 57(4):1411–1419, 2009.

C. P. Tan and C. Edwards. An LMI approach for designing sliding mode observers. *International Journal of Control*, 74(16):1559–1568, 2001.

G. Tartaglione, M. Ariola, W. Bie, L. E. Di Grazia, M. Mattei, and A. Mele. Plasma magnetic control for DEMO tokamak using MPC. In *2022 IEEE Conference on Control Technology and Applications (CCTA)*, pages 825–830. IEEE, 2022.

G. Tian, B. Li, Q. Zhao, and G. Duan. High-precision trajectory tracking control for free-flying space manipulators with multiple constraints and system uncertainties. *IEEE Transactions on Aerospace and Electronic Systems*, 60(1):789–801, 2023.

A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

Y. Wang and X. Xu. Disturbance observer-based robust control barrier functions. In *2023 American Control Conference (ACC)*, pages 3681–3687. IEEE, 2023.

J. Wu, A. Clark, Y. Kantaros, and Y. Vorobeychik. Neural Lyapunov control for discrete-time systems. In *Advances in Neural Information Processing Systems 36*, 2023.

Y. Xiong and M. Saif. Sliding mode observer for nonlinear uncertain systems. *IEEE Transactions on Automatic Control*, 46(12):2012–2017, 2001.

L. Yang, H. Dai, Z. Shi, C.-J. Hsieh, R. Tedrake, and H. Zhang. Lyapunov-stable neural control for state and output feedback: A novel formulation. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 56033–56046. PMLR, 2024. URL `https://proceedings.mlr.press/v235/yang24f.html`.

T. Yang, K. Chai, J. Ji, Y. Wu, C. Xu, and F. Gao. Ground-effect-aware modeling and control for multicopters. *IEEE/ASME Transactions on Mechatronics*, 2025.

H. Yin, P. Seiler, and M. Arcak. Stability analysis using quadratic constraints for systems with neural network controllers. *IEEE Transactions on Automatic Control*, 67(4):1980–1987, 2021.

S. Zhang and C. Fan. Learning to stabilize high-dimensional unknown systems using Lyapunov-guided exploration. In A. Abate, M. Cannon, K. Margellos, and A. Papachristodoulou, editors, *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*, volume 242 of *Proceedings of Machine Learning Research*, pages 52–67. PMLR, 15–17 Jul 2024. URL `https://proceedings.mlr.press/v242/zhang24a.html`.

J. Zhao and L. Mili. A decentralized H-infinity unscented Kalman filter for dynamic state estimation against uncertainties. *IEEE Transactions on Smart Grid*, 10(5):4870–4880, 2018.

J. Zhao and L. Mili. A theoretical framework of robust H-infinity unscented Kalman filter and its application to power system dynamic state estimation. *IEEE Transactions on Signal Processing*, 67(10):2734–2746, 2019.

R. Zhou, T. Quartz, H. De Sterck, and J. Liu. Neural Lyapunov control of unknown nonlinear systems with stability guarantees. In *Advances in Neural Information Processing Systems 35*, 2022.

A Limitations and Future Work

Our framework has several limitations. First, the Stage I analysis is carried out on a prescribed compact domain $\mathcal{X}$. The local-radius and probabilistic-coverage results in Theorems 1 and 2 characterize domain-dependent properties of pre-training. The scope of the final LMI certificate is determined separately by its certification assumptions. Finite-sample guarantees for point-guided training remain open, as do the required sample size and conditions for successful gradient-based certification.

Second, the method relies on a reasonably informative nominal model and on deployment conditions that are consistent with the offline certification assumptions. Significant model mismatch, severe noise, or out-of-distribution disturbances may therefore degrade both estimation accuracy and certificate usefulness. Finally, the two-stage pipeline improves scalability relative to direct LMI-constrained training. The offline certification step can remain costly for very large architectures or high-dimensional systems. The controller extension in Appendix D suggests broader applicability under the assumptions of the underlying controller LMI certificate and requires dedicated closed-loop validation. Future work will focus on less conservative and more scalable certificates, finite-sample analysis for Stage I, and deployment-aware mechanisms such as runtime monitoring, safe switching, and controller-side experiments.

B Structure of the Neural Network Observer

The neural term π_θ in the observer dynamics is implemented as a residual feedforward network, as illustrated in Figure 5. The network input is $x \in \mathbb{R}^{n_{\text{in}}}$, and the network output is $\pi_\theta(x) \in \mathbb{R}^{n_{\text{out}}}$. For an L-layer network with hidden widths $n_1, \dots, n_L$, the layer-wise propagation is defined as

$$\pi_\theta^{[0]}(x) = x, \qquad \pi_\theta^{[l]}(x) = \sigma\Big(W^{[l]}\pi_\theta^{[l-1]}(x)\Big), \quad l = 1, \dots, L, \tag{15}$$

where $W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$, $n_0 = n_{\text{in}}$, and $\sigma(\cdot)$ is the activation function. The observer network output is

$$\pi_\theta(x) = W^{[L+1]}\pi_\theta^{[L]}(x) + W^{[L+2]}\pi_\theta^{[0]}(x), \tag{16}$$

where $W^{[L+1]} \in \mathbb{R}^{n_{\text{out}} \times n_L}$ maps the final hidden layer to the output and $W^{[L+2]} \in \mathbb{R}^{n_{\text{out}} \times n_{\text{in}}}$ is the linear shortcut. The parameter set is

$$\theta = \{L, n_1, \dots, n_L, W^{[1]}, \dots, W^{[L+2]}\}. \tag{17}$$

The output vector is partitioned according to the observer dynamics into π_{θ_1} and π_{θ_2}.

The LMI lifting matrices $\hat{R}_\pi(\theta)$ and $\hat{R}_\xi(\theta)$ in Lemma 2 follow the neural-network isolation construction in Chen et al. [2023]. The stacked pre-activation vector and activation-output vector are

$$\xi_\sigma(x) = \begin{bmatrix} W^{[1]}\pi_\theta^{[0]}(x) \\ \vdots \\ W^{[L]}\pi_\theta^{[L-1]}(x) \end{bmatrix}, \qquad s_\sigma(x) = \begin{bmatrix} \pi_\theta^{[1]}(x) \\ \vdots \\ \pi_\theta^{[L]}(x) \end{bmatrix}. \tag{18}$$

With $n_\sigma = \sum_{l=1}^{L} n_l$, the residual network can be written as the linear relation

$$\begin{bmatrix} \pi_\theta(x) \\ \xi_\sigma(x) \end{bmatrix} = \begin{bmatrix} N_{\pi x} & N_{\pi s} \\ N_{\xi x} & N_{\xi s} \end{bmatrix} \begin{bmatrix} x \\ s_\sigma(x) \end{bmatrix}, \tag{19}$$

where

$$\begin{aligned} N_{\pi x} &= W^{[L+2]}, \qquad N_{\pi s} = \begin{bmatrix} O & \cdots & O & W^{[L+1]} \end{bmatrix}, \\ N_{\xi x} &= \begin{bmatrix} (W^{[1]})^\top & O & \cdots & O \end{bmatrix}^\top, \end{aligned} \tag{20}$$

and $N_{\xi s} \in \mathbb{R}^{n_\sigma \times n_\sigma}$ is the block lower-shift matrix whose $(l, l-1)$ block is $W^{[l]}$ for $l = 2, \dots, L$ and whose other blocks are zero. Therefore,

$$\hat{R}_\pi(\theta) = \begin{bmatrix} I & O \\ N_{\pi x} & N_{\pi s} \end{bmatrix}, \qquad \hat{R}_\xi(\theta) = \begin{bmatrix} N_{\xi x} & N_{\xi s} \\ O & I \end{bmatrix}. \tag{21}$$

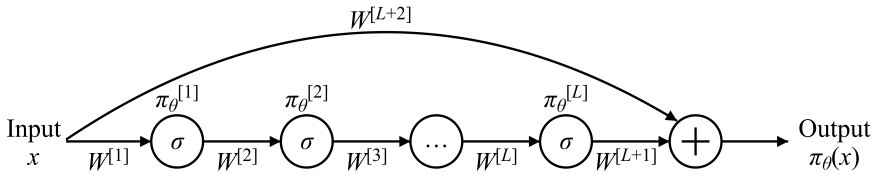


Figure 5: **Residual network parameterization for the neural observer.** The network uses hidden layers with the activation function σ and a linear shortcut $W^{[L+2]}$ from the input to the output.

In Lemma 2, this construction is applied to the observer nonlinearity $\nu(\eta, \theta)$, whose network input is the corresponding error-feedback signal represented as a fixed linear map of η. This linear map is absorbed into $W^{[1]}$ and $W^{[L+2]}$ when forming the N matrices. If π_{θ_1} and π_{θ_2} are implemented as separate subnetworks, the corresponding N matrices are obtained by vertical and block-diagonal stacking of the subnetwork matrices.

C Proof for Section 3

C.1 Proof of Lemma 3

Proof. If the data-dependent term of the loss is zero, we have

$$[F_V(\eta_i, \theta^*) + \xi]_+ = 0, \forall i \in \{1, \dots, N\}.$$

By the definition of the ReLU function, this requires

$$F_V(\eta_i, \theta^*) + \xi \leq 0,$$

and hence,

$$F_V(\eta_i, \theta^*) \leq -\xi.$$

□

C.2 Proof of Theorem 1

Proof. Let $\eta = \eta_s + \zeta \in \mathcal{X}$. Since $Q = A_\epsilon^\top P + P A_\epsilon + \rho P$ is symmetric, we evaluate the difference $F_V(\eta, \theta) - F_V(\eta_s, \theta)$ as follows:

$$\begin{aligned}
&F_V(\eta, \theta) - F_V(\eta_s, \theta) \\
&= \eta^\top Q \eta - \eta_s^\top Q \eta_s - 2\nu(\eta, \theta)^\top P \eta + 2\nu(\eta_s, \theta)^\top P \eta_s + 2\epsilon M_w(\|P\eta\| - \|P\eta_s\|) \\
&= 2\eta_s^\top Q \zeta + \zeta^\top Q \zeta - 2(\nu(\eta, \theta) - \nu(\eta_s, \theta))^\top P \eta_s - 2\nu(\eta_s, \theta)^\top P \zeta \\
&\quad - 2(\nu(\eta, \theta) - \nu(\eta_s, \theta))^\top P \zeta + 2\epsilon M_w(\|P\eta\| - \|P\eta_s\|) \\
&\leq 2\|Q\eta_s\| \, \|\zeta\| + \|Q\| \, \|\zeta\|^2 + 2\|P\eta_s\| \, \|\nu(\eta, \theta) - \nu(\eta_s, \theta)\| + 2\|P\nu(\eta_s, \theta)\| \, \|\zeta\| \\
&\quad + 2\|P\| \, \|\nu(\eta, \theta) - \nu(\eta_s, \theta)\| \, \|\zeta\| + 2\epsilon M_w \|P\| \, \|\zeta\| \\
&\leq (2\|Q\eta_s\| + 2\|P\eta_s\| L_\nu + 2\|P\nu(\eta_s, \theta)\| + 2\epsilon M_w \|P\|) \|\zeta\| + (\|Q\| + 2\|P\| L_\nu) \|\zeta\|^2.
\end{aligned}$$

Applying the triangle inequality, the Cauchy–Schwarz inequality, and the Lipschitz condition $\|\nu(\eta, \theta) - \nu(\eta_s, \theta)\| \leq L_\nu \|\zeta\|$, we obtain

$$F_V(\eta, \theta) - F_V(\eta_s, \theta) \leq C_1(\eta_s) \|\zeta\| + C_2 \|\zeta\|^2,$$

where $C_1(\eta_s) = 2\|Q\eta_s\| + 2\|P\eta_s\| L_\nu + 2\|P\nu(\eta_s, \theta)\| + 2\epsilon M_w \|P\|$ and $C_2 = \|Q\| + 2\|P\| L_\nu$.

Therefore, we have

$$F_V(\eta, \theta) \leq F_V(\eta_s, \theta) + C_1(\eta_s) \|\zeta\| + C_2 \|\zeta\|^2.$$

To ensure $F_V(\eta, \theta) \leq 0$, it suffices to require

$$C_2 \|\zeta\|^2 + C_1(\eta_s) \|\zeta\| + F_V(\eta_s, \theta) \leq 0.$$

Since $F_V(\eta_s, \theta) < 0$, the polynomial $h(r) = C_2 r^2 + C_1(\eta_s) r + F_V(\eta_s, \theta)$ satisfies $h(0) = F_V(\eta_s, \theta) < 0$. If $C_2 > 0$, then $h(r) \to +\infty$ as $r \to \infty$, and h has a unique positive root

$$r_s = \frac{-C_1(\eta_s) + \sqrt{C_1(\eta_s)^2 - 4C_2 F_V(\eta_s, \theta)}}{2C_2}.$$

If $C_2 = 0$ and $C_1(\eta_s) > 0$, choose

$$r_s = \frac{-F_V(\eta_s, \theta)}{C_1(\eta_s)}.$$

If $C_2 = 0$ and $C_1(\eta_s) = 0$, then $h(r) = F_V(\eta_s, \theta) < 0$ for all $r \geq 0$, so any positive r_s is admissible. In all cases, for every $\eta \in \mathcal{B}(\eta_s, r_s) \cap \mathcal{X}$, we have $h(\|\zeta\|) \leq 0$, which implies $F_V(\eta, \theta) \leq 0$. Since $\dot{V}(\eta) + \rho V(\eta) \leq F_V(\eta, \theta)$, the sufficient Lyapunov decrease condition also holds throughout $\mathcal{B}(\eta_s, r_s) \cap \mathcal{X}$. □

C.3 Proof of Corollary 1

Proof. By Theorem 1, each sampled point $\eta_s \in \mathcal{D}$ admits a certified radius. When $C_2 > 0$, this radius can be chosen as

$$r_s = \frac{-C_1(\eta_s) + \sqrt{C_1(\eta_s)^2 - 4C_2 F_V(\eta_s, \theta)}}{2C_2}. \tag{22}$$

Since $F_V(\eta_s, \theta) \leq -\xi < 0$ for all $\eta_s \in \mathcal{D}$,

$$r_s \geq \frac{-C_1(\eta_s) + \sqrt{C_1(\eta_s)^2 + 4C_2 \xi}}{2C_2}. \tag{23}$$

Because $\mathcal{X}$ is compact and $\nu(\cdot, \theta)$ is Lipschitz on $\mathcal{X}$, there exist constants $R_\eta, R_\nu < \infty$ such that $\|\eta_s\| \leq R_\eta$ and $\|\nu(\eta_s, \theta)\| \leq R_\nu$ for all $\eta_s \in \mathcal{X}$. Therefore,

$$C_1(\eta_s) = 2\|Q\eta_s\| + 2\|P\eta_s\| L_\nu + 2\|P\nu(\eta_s, \theta)\| + 2\epsilon M_w \|P\| \tag{24}$$

is uniformly bounded on $\mathcal{X}$. For example, we may take

$$\bar{C}_1 = 2\|Q\| R_\eta + 2\|P\| L_\nu R_\eta + 2\|P\| R_\nu + 2\epsilon M_w \|P\|, \tag{25}$$

so that $C_1(\eta_s) \leq \bar{C}_1$ for all $\eta_s \in \mathcal{X}$. Moreover, $C_2 = \|Q\| + 2\|P\| L_\nu$ is independent of η_s and satisfies $C_2 \geq 0$.

If $C_2 > 0$, define

$$g(c) \coloneqq \frac{-c + \sqrt{c^2 + 4C_2 \xi}}{2C_2}, \qquad c \geq 0. \tag{26}$$

Then

$$g'(c) = \frac{-1 + c/\sqrt{c^2 + 4C_2 \xi}}{2C_2} < 0, \tag{27}$$

so g is strictly decreasing on $[0, \infty)$. Hence, $C_1(\eta_s) \leq \bar{C}_1$ implies $g(C_1(\eta_s)) \geq g(\bar{C}_1)$, and all local stability radii admit the uniform lower bound

$$r_{\min} \geq \underline{r} \coloneqq \frac{-\bar{C}_1 + \sqrt{\bar{C}_1^2 + 4C_2 \xi}}{2C_2} > 0. \tag{28}$$

If $C_2 = 0$ and $\bar{C}_1 > 0$, then each radius with $C_1(\eta_s) > 0$ satisfies

$$r_s = \frac{-F_V(\eta_s, \theta)}{C_1(\eta_s)} \geq \frac{\xi}{C_1(\eta_s)} \geq \frac{\xi}{\bar{C}_1}, \tag{29}$$

while samples with $C_1(\eta_s) = 0$ allow any positive radius. Hence, we may take $\underline{r} = \xi / \bar{C}_1 > 0$ in this case. Finally, if $C_2 = 0$ and $\bar{C}_1 = 0$, then $C_1(\eta_s) = 0$ for all $\eta_s \in \mathcal{X}$, so any fixed positive radius is admissible for all samples. Therefore a strictly positive uniform lower bound exists in all cases. □

C.4 Proof of Theorem 2

Proof. Since $\mathcal{X}$ is compact, for $\bar{r} = r_{\min}/2$, there exists a finite $\bar{r}$-net $\{c_1, \ldots, c_{N_{\text{cov}}}\} \subset \mathcal{X}$ such that

$$\mathcal{X} \subset \bigcup_{k=1}^{N_{\text{cov}}} \mathcal{B}(c_k, \bar{r}).$$

By the full-support assumption on μ, each set $\mathcal{B}(c_k, \bar{r}) \cap \mathcal{X}$ has strictly positive probability. Define

$$q \coloneqq \min_{1 \le k \le N_{\text{cov}}} \mu(\mathcal{B}(c_k, \bar{r}) \cap \mathcal{X}) > 0.$$

For any fixed k, the probability that none of the T independent successful samples falls in $\mathcal{B}(c_k, \bar{r}) \cap \mathcal{X}$ is at most $(1-q)^T$. By the union bound, the probability that at least one of the N_{cov} covering balls contains no successful sample is at most $N_{\text{cov}}(1-q)^T$.

Let $\mathcal{E}$ be the event that every $\bar{r}$-ball $\mathcal{B}(c_k, \bar{r})$ contains at least one successful sample. Then

$$\mathbb{P}(\mathcal{E}) \ge 1 - N_{\text{cov}}(1-q)^T.$$

Conditioning on $\mathcal{E}$, consider any arbitrary point $\bar{\eta} \in \mathcal{X}$. By construction of the $\bar{r}$-net, there exists a center c_k such that $\|\bar{\eta} - c_k\| \le \bar{r} = r_{\min}/2$. Since $\mathcal{E}$ holds, there exists a successful sample $\eta_i \in \mathcal{B}(c_k, \bar{r})$, so $\|\eta_i - c_k\| \le r_{\min}/2$.

By the triangle inequality,

$$\|\bar{\eta} - \eta_i\| \le \|\bar{\eta} - c_k\| + \|c_k - \eta_i\| \le \frac{r_{\min}}{2} + \frac{r_{\min}}{2} = r_{\min}.$$

Hence, $\bar{\eta} \in \mathcal{B}(\eta_i, r_{\min}) \cap \mathcal{X} \subset U_T$. Since $\bar{\eta}$ was arbitrary, $\mathcal{X} \subset U_T$ on the event $\mathcal{E}$.

Therefore,

$$\mathbb{P}(\mathcal{X} \subset U_T) \ge \mathbb{P}(\mathcal{E}) \ge 1 - N_{\text{cov}}(1-q)^T.$$

Taking $T \to \infty$ gives

$$\lim_{T \to \infty} \mathbb{P}(\mathcal{X} \subset U_T) = 1,$$

which completes the proof. □

Corollary 2. Under the assumptions of Theorem 2, for any $\alpha \in (0,1)$, if

$$T \ge \frac{\log(N_{\text{cov}}/\alpha)}{\log\big(1/(1-q)\big)}, \tag{30}$$

then

$$\mathbb{P}(\mathcal{X} \subset U_T) \ge 1 - \alpha. \tag{31}$$

Proof. By Theorem 2, it suffices to require $N_{\text{cov}}(1-q)^T \le \alpha$, which is equivalent to the stated bound on T. □

D Extension to LMI-Certified Neural Network Controllers

The main paper develops the two-stage framework for neural network observers. The same training principle applies to neural network controllers whose closed-loop Lyapunov certificates are imposed through LMIs. A representative example is the recurrent neural network (RNN) controller synthesis framework of Gu et al. [2022], which considers a partially observed discrete-time plant

$$x_{k+1} = A_G x_k + B_G u_k, \qquad y_k = C_G x_k, \tag{32}$$

where the controller only observes y_k, not the full state x_k. The controller is therefore modeled as a dynamic RNN with hidden state ξ_k:

$$\begin{aligned} \xi_{k+1} &= A_K \xi_k + B_{K1} w_k + B_{K2} y_k, \\ u_k &= C_{K1} \xi_k + D_{K1} w_k + D_{K2} y_k, \\ v_k &= C_{K2} \xi_k + D_{K3} y_k, \qquad w_k = \phi(v_k), \end{aligned} \tag{33}$$

where v_k and w_k are the activation input and output, respectively, and ϕ is applied elementwise and described by sector or integral quadratic constraint (IQC) conditions.

The key technical step in Gu et al. [2022] is a loop transformation of the activation sector, yielding a transformed controller with parameters $\tilde{\theta}$ and a transformed nonlinearity $\tilde{\phi}$ satisfying a simple sector bound. With the augmented closed-loop state $\zeta_k = [x_k^\top, \xi_k^\top]^\top$, the transformed interconnection can be written abstractly as

$$\zeta_{k+1} = \mathcal{A}(\tilde{\theta})\zeta_k + \mathcal{B}(\tilde{\theta})z_k, \qquad v_k = \mathcal{C}(\tilde{\theta})\zeta_k + \mathcal{D}(\tilde{\theta})z_k, \qquad z_k = \tilde{\phi}(v_k). \tag{34}$$

Here z_k is the output of the transformed activation $\tilde{\phi}$. Combining the sector constraint for $\tilde{\phi}$ with a quadratic Lyapunov function $V(\zeta) = \zeta^\top P \zeta$ and the S-lemma yields a sequentially convex LMI certificate. For uncertain plants, Gu et al. [2022] augments the nominal dynamics with an IQC description of the uncertainty:

$$\begin{aligned} \zeta_{k+1} &= \mathcal{A}\zeta_k + \mathcal{B}_1 q_k + \mathcal{B}_2 z_k, \\ v_k &= \mathcal{C}_1\zeta_k + \mathcal{D}_1 q_k + \mathcal{D}_2 z_k, \\ r_k &= \mathcal{C}_2\zeta_k + \mathcal{D}_3 q_k + \mathcal{D}_4 z_k, \end{aligned} \tag{35}$$

where q_k is the uncertainty output and r_k is the IQC-filter output. For fixed matrices $\bar{P}$ and $\bar{\Lambda}$ from the previous iterate, the robust LMI is

$$\begin{bmatrix} R^\top \Gamma R & \begin{bmatrix} \mathcal{A} & \mathcal{B}_1 & \mathcal{B}_2 \\ \mathcal{C}_1 & \mathcal{D}_1 & \mathcal{D}_2 \end{bmatrix}^\top \\ \begin{bmatrix} \mathcal{A} & \mathcal{B}_1 & \mathcal{B}_2 \\ \mathcal{C}_1 & \mathcal{D}_1 & \mathcal{D}_2 \end{bmatrix} & \begin{bmatrix} Q_1 & O \\ O & Q_2 \end{bmatrix} \end{bmatrix} \succeq O, \tag{36}$$

where

$$\Gamma = \operatorname{diag}\big(\rho^2(2\bar{P} - \bar{P}^\top Q_1 \bar{P}), 2\bar{\Lambda} - \bar{\Lambda}^\top Q_2 \bar{\Lambda}, -M\big), \qquad R = \begin{bmatrix} I & O & O \\ O & O & I \\ \mathcal{C}_2 & \mathcal{D}_3 & \mathcal{D}_4 \end{bmatrix}. \tag{37}$$

Here $M \in \mathcal{M}$ is the IQC multiplier for the plant uncertainty, with $\mathcal{M}$ denoting the admissible convex multiplier set. This LMI is jointly convex in $(Q_1, Q_2, \tilde{\theta}, M)$ for fixed $(\bar{P}, \bar{\Lambda})$, where Q_1 and Q_2 are inverse-type variables associated with the Lyapunov matrix and sector multiplier. Its feasible set defines the robust convex inner approximation

$$\mathcal{C}_R(\bar{P}, \bar{\Lambda}) = \{\tilde{\theta} : \exists Q_1, Q_2,\ M \in \mathcal{M} \text{ s.t. (36)}\}, \tag{38}$$

which guarantees exponential stability with the prescribed rate. The nominal linear time-invariant (LTI) case uses the analogous set $\mathcal{C}(\bar{P}, \bar{\Lambda})$ obtained by removing the robust IQC block.

To extend our two-stage framework, Stage I replaces the observer error state η with the augmented closed-loop state ζ and uses sampled Lyapunov decrease as a pre-training signal. For sampled augmented states $\{\zeta_i\}_{i=1}^N$, define the one-step Lyapunov residual

$$F_{\mathrm{ctrl}}(\zeta, \tilde{\theta}) = V(\zeta^+) - \rho^2 V(\zeta), \qquad \zeta^+ = f_{\mathrm{cl}}(\zeta; \tilde{\theta}), \tag{39}$$

where $0 < \rho < 1$ is the desired exponential decay rate. The controller can then be trained with a task objective plus a point-guided Lyapunov penalty

$$\mathcal{L}^{\mathrm{ctrl}}_{\mathrm{point}}(\tilde{\theta}) = \mathcal{L}_{\mathrm{task}}(\tilde{\theta}) + \alpha \frac{1}{N} \sum_{i=1}^{N} \big[F_{\mathrm{ctrl}}(\zeta_i, \tilde{\theta}) + \delta_{\mathrm{ctrl}}\big]_+ + \gamma \|\tilde{\theta}\|^2, \tag{40}$$

where $\delta_{\mathrm{ctrl}} > 0$ is the controller Lyapunov margin, $\alpha, \gamma > 0$ are loss weights, and ζ_i may be sampled from rollouts, reset distributions, or a prescribed operating domain.

Stage II then starts from the pre-trained transformed controller $\tilde{\theta}_{\mathrm{pre}}$ and performs the LMI-certified refinement using the same projection structure as Gu et al. [2022], for example

$$\begin{aligned} \min_{Q_1, Q_2, \tilde{\theta}} \quad & \|\tilde{\theta} - \tilde{\theta}_{\mathrm{pre}}\|_F^2 + \beta_1 \|Q_1 - \bar{P}^{-1}\|_F^2 + \beta_2 \|Q_2 - \bar{\Lambda}^{-1}\|_F^2 \\ \text{s.t.} \quad & (36), \end{aligned} \tag{41}$$

where $\beta_1, \beta_2 > 0$ are regularization weights. Alternatively, one may use the eigenvalue-penalty variant from our observer experiments. After a feasible solution is found, one can update $\bar{P} = Q_1^{-1}$ and $\bar{\Lambda} = Q_2^{-1}$ and repeat the projection if desired. The final controller inherits the Lyapunov/LMI stability guarantee of Gu et al. [2022], while the point-guided stage reduces the amount of SDP-based projection or LMI fine-tuning needed for high-capacity RNN controllers.

Table 5: **Quantitative performance comparison under extreme perturbation ($\sigma = 0.5$).** Results are reported as mean $\pm$ standard error of the mean (SEM) across 100 independent trials.

Architecture	Tracking Error (m)	Success Rate	Architecture	Tracking Error (m)	Success Rate
Linear	0.2052 ± 0.0402	0.49	**Small NN**	0.0558 ± 0.0312	0.52
Tiny NN	0.2056 ± 0.0402	0.49	**Large NN**	$\mathbf{0.0118 \pm 0.0028}$	**0.56**

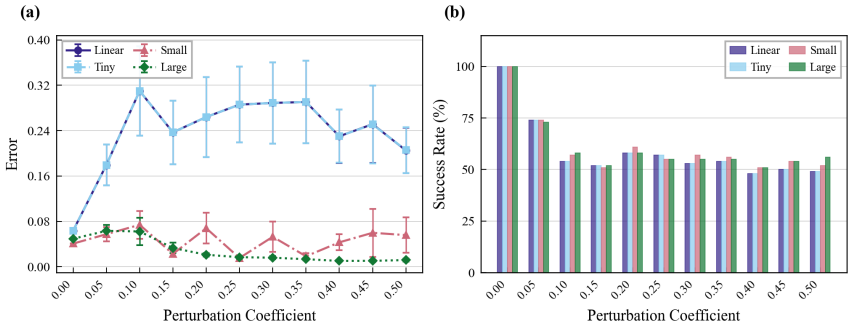


Figure 6: **Robustness of neural network observers under structural perturbations in the X-29 aircraft benchmark.** Tracking error (top) and success rate (bottom) across different network capacities. Error bars denote the mean $\pm$ SEM across 100 independent trials.

E Additional X-29 Capacity and Robustness Results

We use the X-29 aircraft state-space model to evaluate robustness under structural model mismatch and to study the effect of observer capacity. The nominal system is

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + d(t), \\ y(t) = Cx(t), \end{cases} \tag{42}$$

where $x \in \mathbb{R}^4$ and the nominal matrices A, B, and C are taken from Bosworth [1992]. Their numerical values are provided in Appendix G.3.

We compare four observers of increasing capacity: a Linear Baseline, Tiny neural network (Tiny NN), Small NN, and Large NN. A state-feedback controller $u(t) = -K\hat{x}(t)$ is used in all cases, and architectural details are given in Appendix G.3.

To test robustness to structural mismatch between the nominal model and the true system, we perturb the state transition matrix A and the observation matrix C as

$$X' = X + \sigma \frac{Y}{\|Y\|} \|X\|, \tag{43}$$

where $X \in \{A, C\}$, Y is a random matrix of the same dimension with entries sampled independently from the standard normal distribution, and $\sigma \in [0, 0.5]$ is the perturbation coefficient. For each σ, we simulate 20.0 seconds of closed-loop operation and average over 100 random seeds. We report two metrics: (i) *success rate*, the fraction of perturbed trials whose trajectories stay within the stability tube $\|x(t)\| \leq r_{\text{tube}}$ with $r_{\text{tube}} = 3.0$ m over the full horizon, and (ii) *mean tracking error*, the average Euclidean distance between the actual and desired trajectories over successful trials.

Figure 6 and Table 5 show that robustness improves markedly with network capacity. As σ increases, the linear observer and Tiny NN degrade quickly, whereas the Large NN maintains the lowest tracking error and the highest success rate. Under the hardest setting $\sigma = 0.5$, the Large NN reduces the tracking error to 0.0118 ± 0.0028, compared with 0.2052 ± 0.0402 for the linear baseline. Reporting tracking error over successful trials isolates control quality after stabilization, while the success-rate metric accounts for failures separately.

F Observer-Error Results

F.1 Quad-UAV

Table 6 compares steady-state ground-effect acceleration errors with Neural Lander Shi et al. [2019]. The neural observer has lower RMSE in both landing and take-off.

Table 6: **Quad-UAV steady-state ground-effect acceleration RMSE.** Mean $\pm$ sample SD over ten seeds per scenario, in $\mathrm{m/s^2}$. The marked entry reports the mean.

Scenario	Method	Steady-state RMSE
Landing	Neural Lander	$1.7295^{\dagger}$
Landing	Neural observer	0.02788 ± 0.01144
Take-off	Neural Lander	0.1815 ± 0.0257
Take-off	Neural observer	$(2.588 \pm 0.6381) \times 10^{-7}$

† Neural Lander landing: median 0.4507, range $[0.2534, 13.3322]$.

F.2 WaterLily

Table 7 reports steady-state errors in the shared velocity and disturbance channels. The neural observer has lower RMSE in v, ω, and d_v than ESO.

Table 7: **WaterLily steady-state observer-error RMSE.** Mean $\pm$ sample SD over five seeds for each observer, in native simulation units.

Channel	ESO	Neural observer
v	0.003759 ± 0.000318	$(3.540 \pm 1.055) \times 10^{-4}$
ω	$(6.039 \pm 0.136) \times 10^{-5}$	$(1.691 \pm 0.390) \times 10^{-5}$
d_v	0.12520 ± 0.01238	0.04993 ± 0.01238
d_ω	$(1.743 \pm 0.132) \times 10^{-6}$	$(2.363 \pm 0.621) \times 10^{-3}$

G Experimental Details

Unless otherwise stated, all neural network observers in our experiments use `tanh` as the activation function for hidden layers, and both Stage I pre-training and Stage II fine-tuning are optimized using the `Adam` optimizer Kingma and Ba [2015]. For point-guided Lyapunov pre-training, the sampled error states are drawn from a **truncated normal distribution** supported on the prescribed compact domain $\mathcal{X}$.

G.1 Experiment: Quad-UAV

The baseline control utilizes a PID controller with the following gains:

$$K_P = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.75 \\ 8.0 \\ 3.0 \\ 3.0 \end{bmatrix}, \quad K_I = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}, \quad K_D = \begin{bmatrix} 0.75 \\ 0.75 \\ 1.0 \\ 1.5 \\ 1.5 \\ 0.75 \end{bmatrix}.$$

To account for model uncertainties, the ground effect parameters are perturbed to $g_1 = 0.1 \pm 0.01$ and $g_2 = 1.0 \pm 0.25$.

For the Neural Lander baseline, we generated 40,000 data points to learn the dynamics, with training hyperparameters following those specified in Shi et al. [2019].

For the proposed Neural Network Observer, the pre-training phase is configured with $\mathrm{epoch} = 100000$ and $\mathrm{lr} = 0.01$, and the fine-tuning phase uses $\mathrm{epoch} = 15000$, $\mathrm{lr} = 0.01$, and $\epsilon = 0.01$. The

Lyapunov and LMI hyperparameters are

$$P = \mathrm{diag}(1,1,1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,\\ 1,1,1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1),\\ \rho = 0.1, \quad M_w = 0.5, \quad \xi = 10.0, \quad \beta = 0.01,$$

and the Stage II penalty function is

$$\phi(\lambda) = \begin{cases} 10^5\lambda, & \lambda > 0, \\ 0, & \lambda \le 0. \end{cases}$$

G.2 Experiment: AUV in WaterLily

The baseline control framework employs NMPC with a prediction horizon of $H = 20$. We evaluate three specific configurations:

- **Basic NMPC:** A baseline control implementation without disturbance compensation.
- **ESO + NMPC:** NMPC augmented with an ESO for disturbance compensation, using a gain bandwidth of $\omega_0 = 15$.
- **Neural Network Observer + NMPC (Ours):** Our proposed framework. The pre-training phase uses $\mathrm{epoch} = 100000$ and $\mathrm{lr} = 0.01$, while the fine-tuning phase uses $\mathrm{epoch} = 15000$, $\mathrm{lr} = 0.01$, and $\epsilon = 0.01$.

For the proposed Neural Network Observer in the AUV experiment, the Lyapunov and LMI hyperparameters are

$$P = \mathrm{diag}(0.1,0.1,0.5,0.5,1,1,1,1,1,1), \quad \rho = 0.1, \quad M_w = 0.5, \quad \xi = 10.0, \quad \beta = 0.01,$$

and the Stage II penalty function is

$$\phi(\lambda) = \begin{cases} 10^5\lambda, & \lambda > 0, \\ 0, & \lambda \le 0. \end{cases}$$

G.3 Experiment: X-29 Aircraft

The nominal matrices used in the X-29 aircraft benchmark are taken from Bosworth [1992]:

$$A = \begin{bmatrix} -0.4272\times10^{-1} & -0.8541\times10^{1} & -0.4451 & -0.3216\times10^{2} \\ -0.7881\times10^{-3} & -0.5291 & 0.9896 & 0.1439\times10^{-9} \\ 0.4010\times10^{-3} & 0.3542\times10^{1} & -0.2228 & 0.6150\times10^{-8} \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.3385\times10^{-1} & -0.9386\times10^{-1} & 0.4888\times10^{-2} \\ -0.1028\times10^{-2} & -0.1297\times10^{-2} & -0.4054\times10^{-3} \\ 0.2718\times10^{-1} & -0.5744\times10^{-2} & -0.1351\times10^{-1} \\ 0 & 0 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 57.3 & 0 & 0 \\ 0 & 0 & 57.3 & 0 \\ 0 & 0 & 0 & 57.3 \\ 0.007063 & 4.567 & 0.09867 & -0.3809\times10^{-4} \end{bmatrix}.$$

The baseline control employs a state-feedback control law $u(t) = -K\hat{x}(t)$, where the gain matrix K is given by:

$$K = \begin{bmatrix} -16.99 & 171.31 & 148.15 & 308.07 \\ -25.65 & -6.74 & -13.01 & 137.50 \\ 5.36 & -91.89 & -76.05 & -130.60 \end{bmatrix}.$$

The architectures of the four evaluated observers are detailed below:

- **Linear Baseline:** A conventional observer consisting solely of the linear shortcut $W^{[L+2]}$ (equivalent to zero hidden layers), representing a traditional LTI observer. Parameters are determined via pole-placement.
- **Tiny NN:** A minimal 3-layer architecture with a $[3, 3, 3]$ hidden unit configuration (9 total hidden neurons), solved using pole-placement combined with the *feasp* solver.
- **Small NN:** A medium-capacity 3-layer architecture with a $[32, 64, 32]$ hidden unit configuration (128 total hidden neurons), optimized through pole-placement, the *feasp* solver, and training.
- **Large NN:** An 8-layer deep ResNet with a $[32, 64, 64, 128, 128, 64, 64, 32]$ hidden unit configuration (576 total hidden neurons), optimized using pole-placement, the *feasp* solver, and the proposed training framework.

For the pole-placement method, the target poles are assigned at $[-0.5, -1 \pm 0.2i, -1.5 \pm 0.5i, -2 \pm 0.2i, -2.5]$.

In the *feasp* solver configuration, the gain parameter is set to $\epsilon = 0.01$ with a maximum of `max_iter` $= 10$ iterations.

For the neural network optimization, the pre-training phase uses $\mathrm{epoch} = 50000$ and $\mathrm{lr} = 0.01$, while the fine-tuning phase uses $\mathrm{epoch} = 10000$, $\mathrm{lr} = 0.01$, and a gain parameter of $\epsilon = 0.01$. For the Lyapunov and LMI objectives, we use

$$P = I_8, \quad \rho = 0.1, \quad M_w = 0.5, \quad \xi = 10.0, \quad \beta = 0.01,$$

and the Stage II penalty function is

$$\phi(\lambda) = \begin{cases} 10^5 \lambda, & \lambda > 0, \\ 0, & \lambda \leq 0. \end{cases}$$

G.4 Other Details

All experiments were conducted on a server equipped with a 12th Gen Intel(R) Core(TM) i5-12600KF CPU, an NVIDIA GeForce RTX 4060 GPU, and 32.0 GB of RAM. The software environment included Windows 10, MATLAB R2024a, Python 3.9 (for the solver), Python 3.8 (for training), and PyTorch 2.4.1.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction summarize the paper's main contributions, results, and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work are discussed in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The theoretical results are stated in Section 3, with complete proofs provided in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The implementation details needed to reproduce the main experimental results are provided in Section 4 and Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and reproduction instructions are available at `https://github.com/Berry-Myon/LearningNeuralNetworkObserver` and will also be included in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The training and test settings are detailed in Section 4 and Appendix G.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the mean and either the standard deviation (SD) or the standard error of the mean (SEM) for tracking errors over multiple trials with randomized perturbations. These quantities are explicitly indicated in Tables 5, 3, and 4 and Figures 6, 3, and 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computational resources used for the experiments are reported in Appendix G.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper focuses on foundational mathematical methods for observer stability in dynamical systems. We do not identify a direct path to negative societal applications or significant immediate societal impacts beyond general improvements in control engineering.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All code used for comparison methods is open source and is properly cited and credited in the experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as important, original, or non-standard components of the method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.