Learning Stability Attention in Vision-based End-to-end Driving Policies

Tsun-Hsuan Wang[*] TSUNW@MIT.EDU
Wei Xiao[*] WEIXY@MIT.EDU
Makram Chahine CHAHINE@MIT.EDU
Alexander Amini AMINI@MIT.EDU
Ramin Hasani RHASANI@MIT.EDU
Daniela Rus RUS@MIT.EDU
Computer Science and Artificial Intelligence Lab, MIT, Cambridge, MA, USA.

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

Today's end-to-end learning systems can learn to explicitly infer control from perception. However, it is difficult to guarantee stability and robustness for these systems since they are often exposed to unstructured, high-dimensional, and complex observation spaces (e.g., autonomous driving from a stream of pixel inputs). We propose to leverage control Lyapunov functions (CLFs) to equip end-to-end vision-based policies with stability properties and introduce stability attention in CLFs (att-CLFs) to tackle environmental changes and improve learning flexibility. We also present an uncertainty propagation technique that is tightly integrated into att-CLFs. We demonstrate the effectiveness of att-CLFs via comparison with classical CLFs, model predictive control, and vanilla end-to-end learning in a photo-realistic simulator and on a real full-scale autonomous vehicle.

Keywords: End-to-end Learning, Stability, Autonomous Driving

1. Introduction

End-to-end representation learning systems provide effective methods for autonomy in high dimensional observation spaces such as vision where ground-truth states are unavailable (e.g. situations that are difficult to solve by planning and control algorithms (Primbs et al., 1999).) A key challenge for end-to-end systems is guaranteeing their stability and robustness, which are necessary conditions for deployment in safety-critical applications. Stability and robustness are critical for complex systems such as autonomous driving. It is difficult to derive stability and robustness conditions for end-to-end systems because their observation space is large (e.g., streams of pixel inputs), resulting in increased uncertainty. For instance, in autonomous driving, even a slight change in the sunlight could drastically affect the output behavior of the autonomous control system. Many recent works marry the expressive power of neural networks with classical control modules that empower learning-based models with stability properties (Chang et al., 2019; Dai et al., 2021) and safety guarantee (Xiao et al., 2023). Advances in differentiable optimization further facilitate seamless integration of control theory to deep neural networks (Amos and Kolter, 2017; Amos et al., 2018). While learning-based approaches have demonstrated promising potential in state-space con-

[*] The authors are with equal contributions.

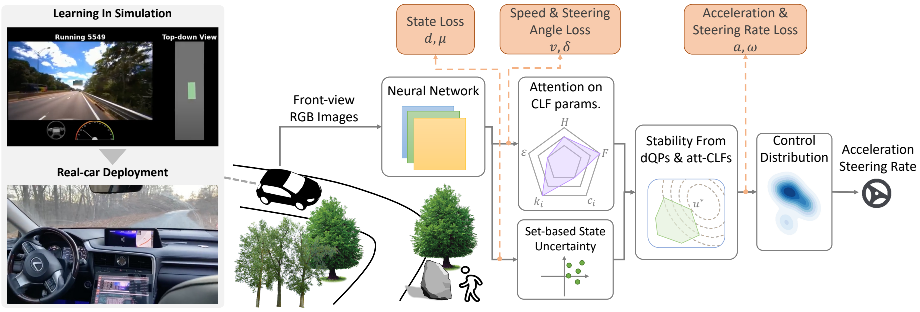


Figure 1: An overview of our end-to-end learning system with att-CLFs.

trol, there are limited attempts to extend this paradigm to end-to-end learning systems that infer control directly from perception.

In this paper, we propose to learn stability attention in control Lyapunov functions (CLFs) to equip end-to-end vision-based learning systems with stability properties. Lyapunov functions (Sontag, 1983; Artstein, 1983; Freeman and Kokotovic, 1996) are powerful tools for verifying the stability of a dynamical system. They have been applied to control systems to formulate stability conditions via CLFs (Ames et al., 2012). The time domain is discretized, and the state is assumed to be constant within each time step. The control problem is a sequence of quadratic programs (QPs). However, this simple and effective formulation is not easy to integrate with end-to-end learning models due to (i) the inability to respond to environmental changes based on perceptual inputs and (ii) limited flexibility from exponentially stabilizing behavior.

To this end, we make CLFs dependent on observations and incorporate them into end-to-end learning via differentiable QPs (Amos and Kolter, 2017). Then, we show how to construct Lyapunov Functions (LF) to achieve a stability attention mechanism (att-CLFs) that can handle environmental changes and higher relative degree.

Contributions: (i) we leverage CLFs to equip vision-based end-to-end driving policies with stability properties, and introduce stability attention in CLFs to tackle environmental changes and improve learning flexibility (ii) we present an uncertainty propagation method tailored to differentiable optimization (iii) we compare our method with classical CLFs, model predictive control and direct end-to-end learning methods, and verify our method in a photo-realistic simulator; we also deploy our model on a real full-scale autonomous vehicle.

2. Background and Preliminaries

Consider an affine control system of the form:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times q}$ are locally Lipschitz, and $\boldsymbol{u} \in U \subset \mathbb{R}^q$, where U denotes a control constraint set. The relative degree definition of the function is omitted, but can be found in (Xiao and Belta, 2019).

Definition 1 *(Control Lyapunov function (Ames et al., 2012)) A continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ is a globally and exponentially stabilizing control Lyapunov function (CLF) for*

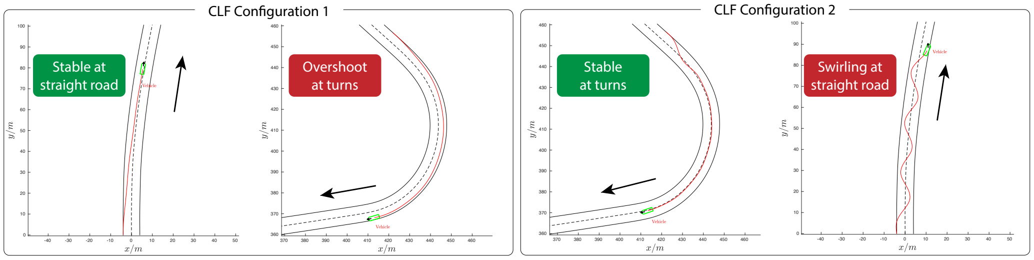


Figure 2: CLF-based QPs for lane tracking. A slow convergent CLF (left) may work well on a straight road, but fails to work for a sharp turn. A fast convergent CLF (right) may work well for a sharp turn, but fails to work on a straight road due to the myopic property of the QP method.

system (1) if there exist $c_1 > 0, c_2 > 0, c_3 > 0$ such that for all $\boldsymbol{x} \in \mathbb{R}^n$, $c_1||\boldsymbol{x}||^2 \le V(\boldsymbol{x}) \le c_2||\boldsymbol{x}||^2$ and

$$\inf_{u \in U}[L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} + c_3 V(\boldsymbol{x})] \le 0. \tag{2}$$

where L_f, L_g denote the Lie derivatives (Khalil, 2002) along f and g, respectively.

Consider an optimal control problem with the cost defined as $\frac{1}{2}\boldsymbol{u}(t)^T H \boldsymbol{u}(t) + F^T \boldsymbol{u}(t)$, where $H \in \mathbb{R}^{q \times q}$ is positive definite, $F \in \mathbb{R}^q$. Suppose the objective is to stabilize system (1) to the origin. We can formulate the following CLF-based optimization (Ames et al., 2012):

$$\begin{aligned} \boldsymbol{u}^*(t) = \arg \min_{\boldsymbol{u}(t) \in U} \frac{1}{2}\boldsymbol{u}(t)^T H \boldsymbol{u}(t) + F^T \boldsymbol{u}(t), \\ \text{s.t. } L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} + \varepsilon V(\boldsymbol{x}) \le 0, \end{aligned} \tag{3}$$

where $\varepsilon > 0$. The parameter ε in the CLF constraint and the definition of $V(\boldsymbol{x})$ significantly affect how the system trajectory converges to the origin. In order to solve (3), as in (Ames et al., 2012), one usually discretizes the time. The CLF constraint is linear in control since the state value is fixed at the beginning of the discrete interval. Therefore, each optimization is a quadratic program (QP). Note that the CLF constraint in (3) may conflict with the control bound. One usually relaxes the CLF constraint by replacing the bound 0 on the right hand side of the inequality with a slack variable, and minimizes it in the cost function.

Limitations of CLFs. The setup has limited flexibility for real world applications due to:

- The desired trajectories may not always exhibit exponentially stabilizing behavior.
- Under heavy environmental state changes, global stability via CLFs is almost impossible to obtain. For complex control synthesis such as vision-based driving, the definition of a CLF may depend on environmental variables.
- The CLF-based QP is solved point-wise yet can lead to aggressive performance and sub-optimalilty.

Example 1 *Suppose we wish to stabilize an autonomous vehicle to the lane center. The definition of a CLF involves road curvature. When the road has sharp turns that are unexpected or not predictable, it is almost impossible to find a globally stabilizing CLF, as shown in Fig. 2. The optimality of control synthesis is limited, and the system may overshoot around the equilibrium due to all possible perturbations (such as noisy dynamics, road curvature variation, etc.), as shown by the left case in Fig. 2. We may formulate a receding horizon optimization to address this issue. However, in this case the optimization is no longer convex.*

In this paper we aim to address these limitations by introducing attention CLFs (att-CLFs).

3. Problem Formulation and Approach

Problem setup. Given (i) a sensor measurement $\boldsymbol{z} \in \mathbb{R}^d$ (e.g., front-view RGB images of the vehicle in this work), where $d \in \mathbb{N}$ is its dimension, (ii) vehicle dynamics (1), (iii) a nominal controller $h^\star(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{u}^\star$ (such as a model predictive controller), and (iv) a learning-based end-to-end perception-to-control model $h(\boldsymbol{z}|\boldsymbol{\theta}) = \boldsymbol{u}$, parametrized by $\boldsymbol{\theta}$, our goal is to find optimal parameters:

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}}[l(h^\star(\boldsymbol{x}, \boldsymbol{z}), h(\boldsymbol{z}|\boldsymbol{\theta}))] \tag{4}$$

where l is a similarity measure (or a loss function). We also wish to equip the learning-based end-to-end controller h with stability properties which we will describe in the following.

Approach. Our solution uses an end-to-end trainable framework that includes a differentiable optimization layer (Amos and Kolter, 2017) formulated by att-CLF based QP. The proposed att-CLF is observation dependent, and thus, it is adaptive to changing environments. The model takes as input front-view RGB images $\boldsymbol{z}$ whose features are extracted by CNN. In order to enforce temporal coherency and prevent the model from being susceptible to sudden input change (e.g., affected by sunlight), we employ LSTM after the CNN. The outputs of the LSTM are used to construct the att-CLF, as well as to determine other optimization parameters in the QP. The outputs of the QP are the controls $\boldsymbol{u}$ of the vehicle. The model is trained using imitation learning where the ground truth control is obtained by a model predictive controller. We also propagate state uncertainty to control to improve performance in real-world deployment. The overview is shown in Fig. 1.

4. Method

In this section, we introduce CLFs with stability attention (att-CLFs) and show how we can propagate uncertainty to improve end-to-end control.

4.1. CLFs with Stability Attention

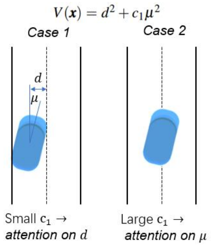


Figure 3: Lane tracking attention under different scenarios.

We focus on quadratic CLFs due to its effectiveness and ubiquity (Ames et al., 2012). Specifically, we construct dependencies of att-CLFs on the environment, i.e., conditioning on measurement $\boldsymbol{z}$:

$$V(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}^T P(\boldsymbol{z}) \boldsymbol{x} \tag{5}$$

where $P(\boldsymbol{z}) \in \mathbb{R}^{n \times n}$ is a positive definite matrix modelled by a neural network that takes $\boldsymbol{z}$ as input. However, learning to directly predict $P(\boldsymbol{z})$ for valid and effective att-CLF is challenging. Next, we describe how to properly construct $P(\boldsymbol{z})$ for better learning. First, we decompose $P(\boldsymbol{z}) = Q^T(\boldsymbol{z})\Lambda(\boldsymbol{z})Q(\boldsymbol{z})$, where $Q(\boldsymbol{z})$ is an orthogonal matrix and $\Lambda(\boldsymbol{z})$ is a diagonal matrix whose components are eigenvalues of $P(\boldsymbol{z})$. We define a notion of stability attention in an att-CLF as:

Definition 2 *(Stability Attention) The stability attention of an att-CLF $V(\boldsymbol{x}, \boldsymbol{z})$ in (5) under an observation $\boldsymbol{z}$ is defined by the state components of $Q(\boldsymbol{z})\boldsymbol{x}$ (with change of basis) corresponding to the top $K \in \mathbb{N}$ largest eigenvalues of $P(\boldsymbol{z})$ (i.e., the diagonal entries of $\Lambda(\boldsymbol{z})$).*

Example 2 *For the autonomous driving example in Fig. 3, we wish to stabilize the off-center distance $d \in \mathbb{R}$ and local heading error $\mu \in \mathbb{R}$ with respect to the lane center line to zero. An att-CLF can be defined as $V(\boldsymbol{x},\boldsymbol{z}) = [d,\mu]^T P(\boldsymbol{z})[d,\mu]$, where $P = [1,0;0,c_1(\boldsymbol{z})]$. Small $c_1(\boldsymbol{z})$ denotes that we pay more attention to the off-center distance d, thus a controller can quickly drive the vehicle position to the lane center if the vehicle is far from it. However, when the vehicle is near the lane center line, we wish to have a larger $c_1(\boldsymbol{z})$, i.e., pay more attention to the local heading error μ. This can make the vehicle orientation better aligned with the road direction.*

In a CLF as in Def. 1, the attention is fixed (constant P), leading to a fixed exponentially convergent behavior. In contrast, the attention of an att-CLF depends on $\boldsymbol{z}$, allowing flexibility to learn different types of behaviors and straightforward interpretation. The dependency on observation makes att-CLFs more versatile and able to imitate complex policies. When stabilizing full states, we can construct a diagonal $P(\boldsymbol{z})$ (an att-CLF has relative degree one); otherwise, we proceed as follows.

High relative degree. In an att-CLF, it is possible that $\frac{\partial V(\boldsymbol{x},\boldsymbol{z})}{\partial \boldsymbol{x}} g(\boldsymbol{x}) = 0, \forall \boldsymbol{x}, \forall \boldsymbol{z}$, which prevents us from obtaining a stabilizing controller as in (3). This may happen when we only wish to stabilize partial states (like the driving problem example), which introduces a high relative degree issue (Xiao and Belta, 2019). To address this, we propose state-feedback based att-CLFs, which can also be applied to non-input-to-state linearizable systems (Khalil, 2002). Suppose $\boldsymbol{x} := (x_1,\dots,x_n)$, and we only wish to stabilize $(x_1,\dots,x_m)$, where $m < n$. Suppose the relative degree of $x_i, i \in \{1,\dots,m\}$ is $r_i \in \mathbb{N}$, and define $n_0 = \sum_{i\in\{1,\dots,m\}} r_i$. We define the first derivative of x_i as a new state $\xi_{i,1} \in \mathbb{R}$, and then define the derivative of $\xi_{i,1}$ as another new state $\xi_{i,2} \in \mathbb{R}$. This can be done recursively until we define a new state $\xi_{i,r_i-1} \in \mathbb{R}$ whose first derivative will be a linear function of $\boldsymbol{u}$ (i.e., its relative degree is one) for system (1) under a certain state $\boldsymbol{x}$. Then, in order to drive x_i to 0, the desired state for ξ_{i,r_i-1} is $\hat{\xi}_{i,r_i-1} := -l_i(\boldsymbol{z})x_i - l_{i,1}(\boldsymbol{z})\xi_{i,1}\cdots - l_{i,r_i-2}(\boldsymbol{z})\xi_{i,r_i-2}$, where $l_i(\boldsymbol{z}) > 0, l_{i,1}(\boldsymbol{z}) > 0, \dots, l_{i,r_i-2}(\boldsymbol{z}) > 0$. Intuitively, making $r_i - 1$'th derivative being inversely proportional to lower degree states (0' to $r_i - 2$'th) allows to drive eventually lower degree states to 0.

Let $\boldsymbol{y} := (x_1,\dots,x_m,\xi_{1,1},\dots,\xi_{1,r_1-1},\dots,\xi_{m,1},\dots,\xi_{m,r_m-1})$ be the new state vector of the transformed dynamics $\dot{\boldsymbol{y}} = \hat{f}(\boldsymbol{y}) + \hat{g}(\boldsymbol{y})\boldsymbol{u}$, where $\hat{f}: \mathbb{R}^{n_0} \to \mathbb{R}^{n_0}, \hat{g}: \mathbb{R}^{n_0} \to \mathbb{R}^{n_0\times q}$ are determined by the above transformation process. We define a state-feedback based att-CLF with relative degree one, which stabilizes actual state ξ_{i,r_i-1} to the desired state $\hat{\xi}_{i,r_i-1}$:

$$V(\boldsymbol{y},\boldsymbol{z}) = \sum_{i=1}^{m} c_i(\boldsymbol{z})\Big(\frac{1}{l_i(\boldsymbol{z})}\xi_{i,r_i-1} - \frac{1}{l_i(\boldsymbol{z})}\hat{\xi}_{i,r_i-1}\Big)^2 = \sum_{i=1}^{m} c_i(\boldsymbol{z})(x_i + k_{i,1}(\boldsymbol{z})\xi_{i,1}\cdots + k_{i,r_i-1}(\boldsymbol{z})\xi_{i,r_i-1})^2 \tag{6}$$

where $c_i(\boldsymbol{z}) > 0, \dots, c_m(\boldsymbol{z}) > 0$, $k_{i,1}(\boldsymbol{z}) = \frac{l_{i,1}(\boldsymbol{z})}{l_i(\boldsymbol{z})}, \dots, k_{i,r_i-2}(\boldsymbol{z}) = \frac{l_{i,r_i-2}(\boldsymbol{z})}{l_i(\boldsymbol{z})}, k_{i,r_i-1}(\boldsymbol{z}) = \frac{1}{l_i(\boldsymbol{z})}$. We can then construct $P(\boldsymbol{z})$ with the new state vector $\boldsymbol{y}$ and rewrite the att-CLF in matrix form:

$$V(\boldsymbol{y},\boldsymbol{z}) = \boldsymbol{y}^T P(\boldsymbol{z})\boldsymbol{y} = (Q(\boldsymbol{z})\boldsymbol{y})^T \Lambda(\boldsymbol{z})(Q(\boldsymbol{z})\boldsymbol{y}), \quad \text{where} \tag{7}$$

$$Q(\boldsymbol{z}) = \begin{bmatrix} I_{m\times m} & k_{block}(\boldsymbol{z}) \\ 0_{(n_0-m)\times m} & 0_{(n_0-m)\times(n_0-m)} \end{bmatrix}, \Lambda(\boldsymbol{z}) = \begin{bmatrix} c_{m\times m} & 0_{m\times(n_0-m)} \\ 0_{(n_0-m)\times m} & 0_{(n_0-m)\times(n_0-m)} \end{bmatrix}.$$

where $k_{block}(\boldsymbol{z})$ is a block diagonal matrix composed by vectors $k_i(\boldsymbol{z}) = (k_{i,1}(\boldsymbol{z}),\dots,k_{i,r_i-1}(\boldsymbol{z})), i \in \{1\dots,m\}$, $c_{m\times m}(\boldsymbol{z})$ is a diagonal matrix composed by $c_i(\boldsymbol{z}), i \in \{1,\dots,m\}$, $0_{i\times j}$ denotes a zero matrix with dimension $i \times j$, and $I_{m\times m}$ denotes an identity matrix with dimension $m \times m$. Note that $k_i(\boldsymbol{z}), c_i(\boldsymbol{z})$ are the output of the previous layer (LSTM).

As per Def. 2, the stability attention in (8) is the top K largest diagonal entries of $\Lambda(\boldsymbol{z})$, and the corresponding state is $Q(\boldsymbol{z})\boldsymbol{y}$ after change of basis. We can get a similar att-CLF constraint as in (2):

$$\frac{\partial V(\boldsymbol{y},\boldsymbol{z})}{\partial \boldsymbol{y}}\hat{f}(\boldsymbol{y}) + \frac{\partial V(\boldsymbol{y},\boldsymbol{z})}{\partial \boldsymbol{y}}\hat{g}(\boldsymbol{y})\boldsymbol{u} + \frac{\partial V(\boldsymbol{y},\boldsymbol{z})}{\partial \boldsymbol{z}}\dot{\boldsymbol{z}} + \varepsilon V(\boldsymbol{y},\boldsymbol{z}) \leq 0, \tag{8}$$

In the above, the term $\frac{\partial V(\boldsymbol{y},\boldsymbol{z})}{\partial \boldsymbol{z}}\dot{\boldsymbol{z}}$ can be evaluated using the bound of $\dot{\boldsymbol{z}}$ if it is available to ensure robustness to the input $\boldsymbol{z}$. We have the following theorem to prove stability:

Theorem 3 *Given a state-feedback based att-CLF as in (6), any control $\boldsymbol{u}$ that satisfies the att-CLF constraint (8) stabilizes the partial state $(x_1,\dots,x_m)$ to the origin for system (1).*

Proof The att-CLF constraint (8) can be rewritten in the time derivative form $\frac{dV(\boldsymbol{y},\boldsymbol{z})}{dt} + \varepsilon V(\boldsymbol{y},\boldsymbol{z}) \leq 0$. Then, according to the Lyapunov stability theorem (Khalil, 2002), we have that $V(\boldsymbol{y},\boldsymbol{z})$ will be exponentially stabilized to zero. In other words, when $V(\boldsymbol{y},\boldsymbol{z}) = 0$, we have either case (i) $x_i = 0, \xi_{i,j} = 0, \forall j \in \{1,\dots,r_i-1\}, \forall i \in \{1,\dots,m\}$ or case (ii) $x_i = -k_{i,1}(\boldsymbol{z})\xi_{i,1}\cdots - k_{i,r_i-1}(\boldsymbol{z})\xi_{i,r_i-1}$. In case (i), the theorem holds. In case (ii), following (6), the condition is equivalent to $\xi_{i,r_i-1} = -l_i(\boldsymbol{z})x_i - l_{i,1}(\boldsymbol{z})\xi_{i,1}\cdots - l_{i,r_i-2}(\boldsymbol{z})\xi_{i,r_i-2}$. Since we have $\dot{x}_i = \xi_{i,1},\dots,\dot{\xi}_{i,r_i-2} = \xi_{i,r_i-1}$ following the above state transformation process, $\xi_{i,r_i-1} = -l_i(\boldsymbol{z})x_i - l_{i,1}(\boldsymbol{z})\xi_{i,1}\cdots - l_{i,r_i-2}(\boldsymbol{z})\xi_{i,r_i-2}$ is the state feedback control law that drives $x_i, \forall i \in \{1,\dots,m\}$ to the origin (Khalil, 2002). Thus, any control $\boldsymbol{u}$ that satisfies the constraint (8) stabilizes the partial state $(x_1,\dots,x_m)$ to the origin for system (1). ■

An optimization layer in NN. We can formulate an att-CLF based QP as in (3) by replacing its CLF constraint with (8). This optimization becomes a sequence of QPs using the time discretization from (Ames et al., 2012), and the QP can be incorporated into a neural network as a differentiable optimization layer (Amos and Kolter, 2017) (Fig. 1). This optimization layer allows error backpropagation through the solution of the QP and updates the trainable parameter utilizing gradient descent. The CLF constraint is with stability attention. Recall that we can get a relaxed form to handle the infeasibility of CLFs as mentioned in introduced in (Ames et al., 2012). We may also add a differentiable Control Barrier Functions (Xiao et al., 2023) for obstacle avoidance.

4.2. Uncertainty Propagation

The accuracy of state $\boldsymbol{x}$ plays a critical role in CLFs. An unbiased estimation but with high variance can lead to undesired activeness of CLF constraint in the QP, producing overall correct but jerky control. We resort to mitigating this issue with the introduction of state uncertainties. We use a set-based representation for state uncertainty since it is highly flexible to represent any kind of distribution, can be constructed based on parametric models, and is widely adopted in predictive uncertainty quantification (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017). Given a set of state samples $\{\boldsymbol{x}_i\}_{i=1}^N$, we solve for a set of controls $\{\boldsymbol{u}_i^*\}_{i=1}^N = \mathrm{QP}(\{\boldsymbol{x}_i\}_{i=1}^N)$, where QP is defined in (3) with (8) replacing the original CLF constraint and can be efficiently solved by batch operation. We then use kernel density estimation to convert the control sample set to a distribution,

$$p(\boldsymbol{u}|\boldsymbol{x},\boldsymbol{z}) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{\boldsymbol{u}-\boldsymbol{u}_i^*}{h}\right), \tag{9}$$

where K is a kernel function and $h > 0$ is the bandwidth. We use the Gaussian kernel with Scott's rule (Scott, 2015) for bandwidth selection. With control distribution at hand, we can easily integrate

Method	Obs.	Mean Dev.	Infer. Time (s)
NMPC	State	**0.011**	0.818
CLF (Ames et al., 2012)	State	0.150	0.005
att-CLF (ours)	State	0.085	0.028
V-E2E (Amini et al., 2022)	Image	0.573	0.002
E2E-CLF (Amos and Kolter, 2017)	Image	0.328	0.033
att-CLF (ours)	Image	**0.218**	0.032

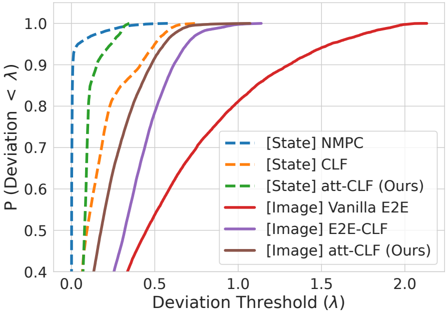


Table 1 & Figure 4: **Left:** Quantitative comparison. **Right:** Probability of deviation from lane center smaller than a threshold λ (the larger area under the curver, the better).

other (unnormalized) prior to obtaining the final control with a set of probing points,

$$\boldsymbol{u}^* = \arg\max_{\boldsymbol{u}\in\mathscr{U}} \; p(\boldsymbol{u})p(\boldsymbol{u}|\boldsymbol{x},\boldsymbol{z}), \tag{10}$$

where the probing point set $\mathscr{U}$ can be a grid that covers the control space or even simply the control samples obtained from the QP. The prior $p(\boldsymbol{u})$ can be any prior knowledge of the design or control distribution from other modules in the system. For simplicity, we use uniform distribution as prior. To obtain predictive uncertainty, we follow (Kendall and Gal, 2017) that predicts the mean and log variance of a normal distribution for state estimate, from which we draw samples and construct the set $\{\boldsymbol{x}_i\}_{i=1}^N$. Note that we can flexibly plug in other predictive uncertainty quantification techniques.

5. Experiments

5.1. Platform & Dataset

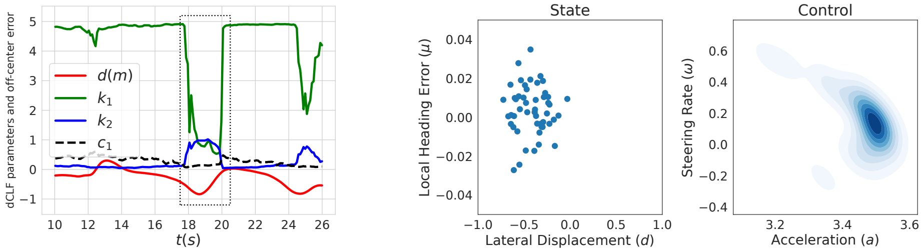
Hardware Setup. We collect data and deploy our models on a full-scale vehicle (2019 Lexus RX 450H) with autonomous driving capability. The onboard computation includes an NVIDIA 2080Ti GPU and an AMD Ryzen 7 3800X 8-Core Processor. The primary perception sensor is a 30Hz BFS-PGE-23S3C-CS RGB camera with resolution 960×600 and 130° horizontal field-of-view. The car is also equipped with IMUs and wheel encoders that provide steering feedback and odometry, as well as a centimeter-level accurate OxTS differential GPS for evaluation purposes only.

Simulation and Data Generation. We adopt a training strategy called guided policy learning that leverages a data-driven simulator *VISTA* (Amini et al., 2022) to augment a real-world dataset with diverse synthetic data for robust policy learning. The ground truth controls (labels) are generated using nonlinear model predictive control (NMPC). Although computationally expensive, NMPC is tractable offline. The simulation is built upon roughly 2-hour real-world driving data collected at a different time of day, weather conditions, and seasons of a year, in total roughly 200k images.

5.2. Implementation Details

Vehicle dynamics. We consider the vehicle dynamics defined with respect to a reference trajectory as shown in (Rucco et al., 2015):

$$\dot{s} = \frac{v\cos(\mu+\beta)}{1-d\kappa}, \quad \dot{d} = v\sin(\mu+\beta), \quad \dot{\mu} = \frac{v}{l_r}\sin\beta - \kappa\frac{v\cos(\mu+\beta)}{1-d\kappa}, \quad \dot{v} = u_a, \quad \dot{\delta} = u_\omega, \tag{11}$$



(a) (b)

Figure 5: **(a)** The profile of att-CLF attention k_1, k_2, c_1 under varying lateral off-center distance d. **(b)** Uncertainty propagation from sample set in state space to density in control space.

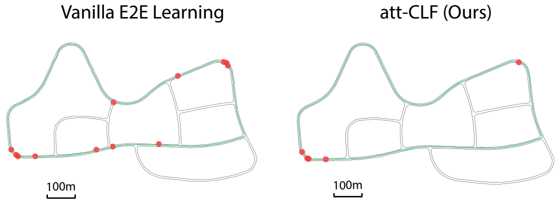
where $\boldsymbol{x} = (s, d, \mu, v, \delta)$, s and d are respectively the along-trajectory distance and lateral distance of the vehicle Center of Gravity (CoG) with respect to the closest point on the reference trajectory; μ is local heading error determined by the global vehicle heading ψ and the tangent angle ϕ of the closest point on the reference trajectory (i.e., $\psi = \phi + \mu$); v, u_a denote the vehicle linear speed and acceleration; δ, u_ω denote the steering angle and steering rate; κ is the curvature of the reference trajectory at the closest point; l_r is the length of the vehicle from the tail to the CoG and $\beta = \arctan\left(\frac{l_r}{l_r + l_f} \tan\delta\right)$, where l_f is the length of the vehicle from the head to the CoG.

Att-CLFs for end-to-end driving. In this problem, we only wish to stabilize d and μ to 0 for lane tracking. The relative degrees of both d and μ are two. Thus, we formulate a state-feedback based att-CLF following Sec. 4.1. The desired states for d and μ are $\hat{d} = -\xi_1, \hat{\mu} = -\xi_2$, where $\xi_1 = v\sin(\mu+\beta), \xi_2 = (\frac{v}{l_r}\sin\beta - \kappa\frac{v\cos(\mu+\beta)}{1-d\kappa})$. The transformed state vector is then $\boldsymbol{y} = (d, \mu, \xi_1, \xi_2)$. The att-CLF in this case is then $V(\boldsymbol{y}, \boldsymbol{z}) = (d + k_1(\boldsymbol{z})\xi_1)^2 + c_1(\boldsymbol{z})\left(\mu + k_2(\boldsymbol{z})\xi_2\right)^2$ (we set $c_2(\boldsymbol{z})$ to one). We assume the observation is piece-wise constant $\dot{\boldsymbol{z}} = 0$, similar to control approximation in (Ames et al., 2012). We leave the estimation of $\dot{\boldsymbol{z}}$ or its bound to future work.

Model & Learning. The input is an RGB image. We extract features with a 5-layer CNN with kernel size as 3, channels as (24, 36, 48, 64, 64) and group norm, followed by average pooling. The feature vector is then fed to a 64-dim LSTM for temporal reasoning. We then use independent MLP heads of hidden size as (32, 32) for different outputs u_a, u_ω and intermediate estimates $v, \delta, d, \mu, \kappa, k_1, k_2, c_1$. We use L2 loss for control u_a, u_ω with ground truth obtained from NMPC. For state estimation d, μ, κ, which is modeled as distribution as mentioned in Sec. 4.2, we follow predictive uncertainty loss in (Kendall and Gal, 2017). We also adopt auxiliary L2 loss on v, δ for more efficient and stable learning. We use Adam optimizer with a learning rate of 0.001.

5.3. Offline Closed-loop Evaluation

We evaluate our policies in a closed-loop setting in *VISTA* (Amini et al., 2022) based on a test set, where we collect 100 episodes (each of maximal 300 steps) with random scenes and initial poses. The baselines include i) *nonlinear MPC* (NMPC) ii) classical *CLF* (Ames et al., 2012) iii) *vanilla end-to-end learning* (V-E2E) (Amini et al., 2022) and iv) *end-to-end CLF* (E2E-CLF) where we use CNN+LSTM to extract state followed by differentiable QP (dQP) (Amos and Kolter, 2017) with fixed CLF parameters. The proposed att-CLF method (a) comparing to *V-E2E* – augment end-to-end learning with better lane tracking error (distribution) (b) comparing to *CLF* and *E2E-CLF* – outperform other CLF baselines with stability attention and (c) comparing to *NMPC* – runs much faster with simple dQP formulation, as shown in Fig. 4 and Tab. 1.



Method	Avg. # of Intervention	Mean Deviation (m)
Vanilla E2E	4.0 ± 1.6	0.779 ± 0.178
att-CLF (ours)	**1.6 ± 0.4**	**0.687 ± 0.083**

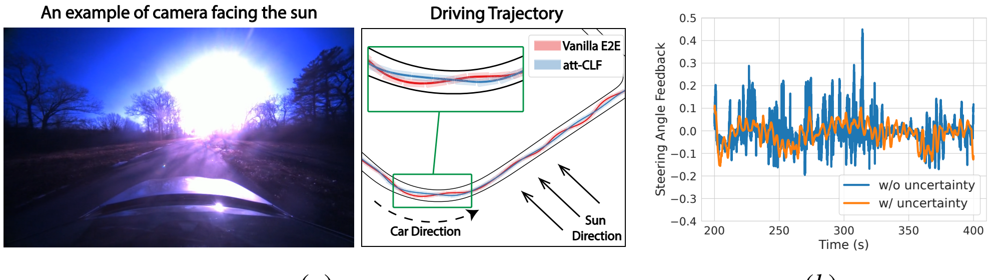
Figure 6 & Table 2: **Left:** Real-world deployment. Policy trajectories (3 trials at different times of day) and intervention locations (red dot). **Right:** Average performance of the real-car test.

Attention of stability. When the car is more off-lane ($|d|$ is large), the att-CLF pays more attention to the lateral distance d (larger k_1); conversely, when the car is closer to the lane center ($|d|$ is smaller), the att-CLF pays more attention to heading μ (larger k_2 and smaller c_1) to achieve stable lane following, as shown in Fig. 5a. Otherwise, the vehicle can overshoot around the lane center line due to the myopic solving method and state uncertainties. This result demonstrates that att-CLF can learn effective strategy by adapting its parameters in response to different situations.

Uncertainty propagation. A simple interpretation for Fig. 5b is to look at how the current state affects steering rate control ω. While all state samples indicate the car is on the right of the lane center ($d < 0$), there is some disagreement on whether the car heading is clockwise ($\mu < 0$) or counter-clockwise ($\mu > 0$) with respect to the road curvature. In the case where $d < 0, \mu < 0.01$ (car is at the right to the lane center and does not orient to the extreme left), the att-CLF will most likely suggest turning left (steering rate $\omega > 0$), which leads to the major mode in the control distribution. Also, the set-based uncertainty can be computed via batch operation in dQP and has acceptable computational overhead (28ms for 1 sample, 43ms for 10 samples, 50ms for 50 samples).

5.4. Deployment on a Physical Autonomous Vehicle

We deploy our end-to-end policy on a full-scale autonomous vehicle. We test the policies for 3 trials at different times of the day. Human make intervention when the vehicle is off the road. Tab. 2 shows that att-CLF achieves fewer human interventions and smaller deviation from the lane center. Fig. 6 overlays the driving trajectories from all trials. We observe that both vanilla E2E and att-CLF fail more frequently for lower-left sharp turn since direct sunlight to the camera along with the reflection from the snow at the side of the road severely corrupt the RGB image. Also, most interventions occur in the trials around 3-4 pm since the sun at that time most likely points directly to the camera, where lighting conditions and shadows in the real world pose great challenges.



(a) (b)

Figure 7: **(a)** Real car testing results when the visual input is corrupted by strong sunlight. **(b)** Steering angle feedback of att-CLF without and with uncertainty propagation on the real car.

Unstable maneuver in challenging scenarios. In Fig. 7a, we show an example [1] where the camera is pointed directly to the sun. Under the corruption of sunlight, policies tend to exhibit a chattering maneuver as opposed to running the test with mild lighting conditions (in the evening or at noon). The att-CLF augments end-to-end policies with stability and produces smoother trajectories, while vanilla E2E easily makes the vehicle overshoot around the lane center, shown as the red curves.

Uncertainty. Fig. 7b shows steering feedback from IMU. Introducing uncertainty greatly improve the smoothness of the control. Furthermore, we observe better stability with uncertainty at trials mostly corrupted by sunlight: mean deviation 1.29m (w/o uncertainty) vs 0.84m (w/ uncertainty).

6. Related Work

Lyapunov functions and learning. In spite of the huge success, the stability properties of deep learning models are less understood (Lechner et al., 2020b). One way to achieve this for neural controllers is Lyapunov stability (Slotine et al., 1991). The premise is to find a Lyapunov function that synthesizes a Lyapunov-stable controller. For example, (Jarvis-Wloszek et al., 2003; Majumdar et al., 2013) demonstrated that Lyapunov functions can be obtained by solving a sum-of-squares program for polynomial dynamics. (Ravanbakhsh and Sankaranarayanan, 2019) proposed a learner-falsifier framework to search for valid Lyapunov functions. Recent works further leveraged the expressive power of neural networks in a Lyapunov framework (Richards et al., 2018; Chang et al., 2019). Other works aim to verify the Lyapunov conditions in neural networks (Abate et al., 2020; Chang et al., 2019; Dai et al., 2020, 2021). In this paper, we build on control Lyapunov functions (CLFs) (Primbs et al., 1999) and leverage differentiable optimization to construct attention of CLFs.

Differentiable optimization for control. Recent advances in differentiable optimization enables seamless integration to neural network (Amos and Kolter, 2017; Agrawal et al., 2019). Such compatibility opens new ways of combining learning-based models with classical control, which can be mostly formulated as optimization problems, such as model predictive control (Amos et al., 2018), convex optimal control (Okada et al., 2017), path integral control (Agrawal et al., 2020), and control barrier function (Xiao et al., 2023). In this work, we leverage differentiable QP (Amos and Kolter, 2017) for CLFs to augment end-to-end driving policies with stability properties.

End-to-end driving policy learning. Apart from traditional autonomous driving pipelines with independent perception, planning, and control modules, many works (Pomerleau, 1989; Lechner et al., 2020a) have demonstrated the ability of end-to-end policy learning that directly infers control from perception. This framework has been shown to solve various tasks including lane keeping (Xu et al., 2017), obstacle avoidance (Wang et al., 2021; Bohez et al., 2017), and navigation (Amini et al., 2019; Codevilla et al., 2018). The major pitfall of end-to-end driving policy is the black-box model that hinders the possibility of introducing properties like stability as opposed to the classical control module. In this work, we leverage recent advances in data-driven simulation (Amini et al., 2022) for sim-to-real transfer and embed stability in end-to-end driving policies.

7. Conclusion

We propose to leverage CLFs to embed vision-based end-to-end driving policies with stability properties and introduce the novel notion of stability attention in CLFs that allows to handle environmental changes and brings flexible learning capability. We compare with other baselines and demonstrate the effectiveness of att-CLFs in a high-fidelity simulator and on a real autonomous car.

[1] Video: https://drive.google.com/file/d/12g0FctJkV7-F_YnCJD7Eurr8d4S7bhGm/view?usp=sharing

Acknowledgments

The research was supported in parts by Capgemini Engineering. It was also partially sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. This work was further supported by the Office of Naval Research (ONR) Grant N00014-18-1-2830.

References

Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, and Andrea Peruffo. Formal synthesis of lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, 2020.

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.

Akshay Agrawal, Shane Barratt, Stephen Boyd, and Bartolomeo Stellato. Learning convex optimization control policies. In *Learning for Dynamics and Control*, pages 361–373. PMLR, 2020.

A. D. Ames, K. Galloway, and J. W. Grizzle. Control lyapunov functions and hybrid zero dynamics. In *Proc. of 51rd IEEE Conference on Decision and Control*, pages 6837–6842, 2012.

Alexander Amini, Guy Rosman, Sertac Karaman, and Daniela Rus. Variational end-to-end navigation and localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8958–8964. IEEE, 2019.

Alexander Amini, Tsun-Hsuan Wang, Igor Gilitschenski, Wilko Schwarting, Zhijian Liu, Song Han, Sertac Karaman, and Daniela Rus. Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

Brandon Amos and J. Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 136–145, 2017.

Brandon Amos, Ivan Dario Jimenez Rodriguez, Jacob Sacks, Byron Boots, and J. Zico Kolter. Differentiable mpc for end-to-end planning and control. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 8299–8310. Curran Associates Inc., 2018.

Z. Artstein. Stabilization with relaxed controls. *Nonlinear Analysis: Theory, Methods & Applications*, 7(11):1163–1173, 1983.

Steven Bohez, Tim Verbelen, Elias De Coninck, Bert Vankeirsbilck, Pieter Simoens, and Bart Dhoedt. Sensor Fusion for Robot Control through Deep Reinforcement Learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural lyapunov control. *Advances in neural information processing systems*, 32, 2019.

Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-End Driving via Conditional Imitation Learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

Hongkai Dai, Benoit Landry, Marco Pavone, and Russ Tedrake. Counter-example guided synthesis of neural network lyapunov functions for piecewise linear systems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1274–1281. IEEE, 2020.

Hongkai Dai, Benoit Landry, Lujie Yang, Marco Pavone, and Russ Tedrake. Lyapunov-stable neural-network control. *arXiv preprint arXiv:2109.14152*, 2021.

R. A. Freeman and P. V. Kokotovic. *Robust Nonlinear Control Design*. Birkhauser, 1996.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Zachary Jarvis-Wloszek, Ryan Feeley, Weehong Tan, Kunpeng Sun, and Andrew Packard. Some controls applications of sum of squares programming. In *42nd IEEE international conference on decision and control (IEEE Cat. No. 03CH37475)*, volume 5, pages 4676–4681. IEEE, 2003.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, third edition, 2002.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Mathias Lechner, Ramin Hasani, Alexander Amini, Thomas A Henzinger, Daniela Rus, and Radu Grosu. Neural circuit policies enabling auditable autonomy. *Nature Machine Intelligence*, 2(10): 642–652, 2020a.

Mathias Lechner, Ramin Hasani, Daniela Rus, and Radu Grosu. Gershgorin loss stabilizes the recurrent neural network compartment of an end-to-end robot learning scheme. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5446–5452. IEEE, 2020b.

Anirudha Majumdar, Amir Ali Ahmadi, and Russ Tedrake. Control design along trajectories with sums of squares programming. In *2013 IEEE International Conference on Robotics and Automation*, pages 4054–4061. IEEE, 2013.

Masashi Okada, Luca Rigazio, and Takenobu Aoshima. Path integral networks: End-to-end differentiable optimal control. *arXiv preprint arXiv:1706.09597*, 2017.

Dean A Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1989.

James A Primbs, Vesna Nevistić, and John C Doyle. Nonlinear optimal control: A control lyapunov function and receding horizon perspective. *Asian Journal of Control*, 1(1):14–24, 1999.

Hadi Ravanbakhsh and Sriram Sankaranarayanan. Learning control lyapunov functions from counterexamples and demonstrations. *Autonomous Robots*, 43(2):275–307, 2019.

Spencer M Richards, Felix Berkenkamp, and Andreas Krause. The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In *Conference on Robot Learning*, pages 466–476. PMLR, 2018.

Alessandro Rucco, Giuseppe Notarstefano, and John Hauser. An efficient minimum-time trajectory generation strategy for two-track car vehicles. *IEEE Transactions on Control Systems Technology*, 23(4):1505–1519, 2015.

David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.

E. Sontag. A lyapunov-like stabilization of asymptotic controllability. *SIAM Journal of Control and Optimization*, 21(3):462–471, 1983.

Tsun-Hsuan Wang, Alexander Amini, Wilko Schwarting, Igor Gilitschenski, Sertac Karaman, and Daniela Rus. Learning interactive driving policies via data-driven simulation. *arXiv preprint arXiv:2111.12137*, 2021.

Wei Xiao and Calin Belta. Control barrier functions for systems with high relative degree. In *Proc. of 58th IEEE Conference on Decision and Control*, pages 474–479, Nice, France, 2019.

Wei Xiao, Tsun-Hsuan Wang, Ramin Hasani, Makram Chahine, Alexander Amini, Xiao Li, and Daniela Rus. Barriernet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics*, 2023.

Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-End Learning of Driving Models from Large-Scale Video Datasets. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.