ABNet: Adaptive explicit-Barrier Net for Safe and Scalable Robot Learning

Wei Xiao [1] **Tsun-Hsuan Wang** [1] **Chuang Gan** [2] **Daniela Rus** [1]

Abstract

Safe learning is central to AI-enabled robots where a single failure may lead to catastrophic results. Existing safe learning methods are not scalable, inefficient and hard to train, and tend to generate unstable signals under noisy inputs that are challenging to be deployed for robots. To address these challenges, we propose Adaptive explicit-Barrier Net (ABNet) in which barriers explicitly show up in the closed-form model that guarantees safety. The ABNet has the potential to incrementally scale toward larger safe foundation models. Each head of ABNet could learn safe control policies from different features and focuses on specific part of the observation. In this way, we do not need to directly construct a large model for complex tasks, which significantly facilitates the training of the model while ensuring its stable output. Most importantly, we can still formally prove the safety guarantees of the ABNet. We demonstrate the efficiency and strength of ABNet in 2D robot obstacle avoidance, safe robot manipulation, and vision-based end-to-end autonomous driving, with results showing much better robustness and guarantees over existing models[1].

1. Introduction

Robot learning usually requires to leverage scalable training and vast amount of data. There are many large models (Li et al., 2022) for complex robotic tasks including manipulation, locomotion, autonomous driving (Bommasani et al., 2021) (Singh et al., 2023) (Wang et al., 2023a). However, these models are not trustworthy and have no safety guarantees. Existing methods that incorporate guarantees or certificates into neural networks are not scalable and hard to train (Pereira et al., 2020) (Xiao et al., 2023) (Wang et al., 2023b). It is desirable to merge these models as we can get better performance controllers in general (Beygelzimer et al., 2015) (Agarwal et al., 2020). Traditional mixture of expert methods (Shazeer et al., 2017) (Riquelme et al., 2021) (Zhou et al., 2022) or other merging approaches (Huang et al., 2023) (Ramé et al., 2023) (Wang et al., 2024) are not designed to retain the safety of the models. In this work, we explore to leverage the collective power of many safety-critical models to handle complex tasks while preserving the safety of the models.

There are various definitions of safety for robotics and autonomy, and safety can be basically defined as something bad never happens. Mathematically, safety can be defined as a continuously differentiable constraint with respect to the system state and it can be further captured by the forward invariance of the safe set over such a constraint (Ames et al., 2017) (Xiao & Belta, 2021) (Glotfelter et al., 2017). In other words, we can use different constraints and approaches to enforce safety. The way we learn such safety enforcement methods may depend on the focused observation feature. For instance, some human drivers may focus on the left lane boundary in driving in order to achieve safe lane keeping, while others may focus on the right lane boundary, as shown in Fig. 1. Merging these models enables us to build robust and powerful learning models. However, the adaptivity of the merging method to different safe models is crucial, especially in retaining safety.

In the literature, differentiable Quadratic Programs (dQP) (Amos & Kolter, 2017) and differentiable Model Predictive Control (dMPC) (Amos et al., 2018) are widely used for safe robot learning. However, dMPC is restricted to linear systems with linear constraints. Barrier-based learning methods (Robey et al., 2020) (Pereira et al., 2020) (Srinivasan et al., 2020), such as the BarrierNet (Xiao et al., 2023) (Wang et al., 2023b) (Liu et al., 2023), are widely used to transform nonlinear problems into dQPs and can equip deep learning systems with safety guarantees. However, there are several limitations of these learning methods: (i) they are involved with solving batch QPs during training, which is inefficient, and dQPs tend to give awful solutions that significantly deteriorate the model; (ii) they can only implement a single safety enforcement method as the last layer of the neural network, which is not scalable to larger safe learning

[1]Computer Science and Artificial Intelligence Lab, MIT, USA [2]UMass Amherst and MIT-IBM Watson AI Lab, USA. Correspondence to: Wei Xiao <weixy@mit.edu>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

[1]Code is available at: https://github.com/Weixy21/ABNet

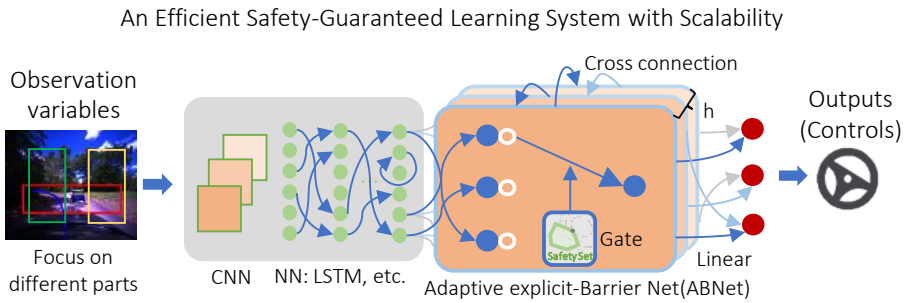

Figure 1: The proposed ABNet that is efficient, scalable and generates stable output while guaranteeing safety for robots. Each head of ABNet in the model could learn safe control policies with focus on different observation feature in a scalable or one-shot/direct manner. Barriers play the role of gates in determining the closed-form safe control and are more interpretable.

models; (*iii*) these methods tend to generate unstable output under noise, which cannot be deployed for robots.

In this paper, we propose the Adaptive explicit-Barrier Net (ABNet) to merge many safety-critical models while preserving the safety guarantees. The ABNet is efficient, scalable, robust to noise, and easy to be trained in an incremental manner. As shown in Fig. 1, we may build multi-head models within the ABNet. Each head of the ABNet may pay attention to different observation features to generate a safe control policy. *We combine the outputs of all the safe learning models in a way that is provably safe.* The weights of this combination quantify the importance of each head of the model, and they are trainable. The structure of the ABNet allows us to build larger safe foundation models for complicated robotic applications as we can incrementally train safe models corresponding to different robot skills and this will simply increase the head h of the ABNet.

In summary, we make the following **new contributions**:

- We propose a novel explicit-Barrier model that shows superior performance in stable training and computation than dQP (Amos & Kolter, 2017) and BarrierNet (Xiao et al., 2023) while guaranteeing safety of robot learning, and the explicit-Barrier model is crucial to build larger models via merging due to its high-efficiency in dealing with nonlinear systems and constraints.
- We propose a novel ABNet that merges many safety-critical learning models, and this new model is scalable, robust, and easy to be trained.
- We formally prove the safety guarantees of the proposed ABNet.
- We demonstrate the strength and effectiveness of our model on a variety of robot control tasks, including 2D robot obstacle avoidance, safe robot manipulation, and vision-based end-to-end autonomous driving in an open dataset. We also show that existing models/policies merging could make safety worse in complicated tasks (such as in vision-based driving).

2. Problem Formulation

We consider the following safe robot learning problem:

Problem. Given (a) a robotic system with dynamics; (b) a state-feedback nominal controller $\boldsymbol{\pi}^*(\boldsymbol{x}) = \boldsymbol{u}^*$ (such as a model predictive controller) that provides the training label; (c) a set of safety constraints $b_j(\boldsymbol{x}) \geq 0, j \in S$ (b_j is continuously differentiable, S is a constraint set); (d) a neural network controller $\boldsymbol{\pi}(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = \boldsymbol{u}$ parameterized by $\boldsymbol{\theta}$ (under observation $\boldsymbol{z}$);

Our goal is to find the optimal parameter

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}}[\ell(\boldsymbol{\pi}^*(\boldsymbol{x}), \boldsymbol{\pi}(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}))], \tag{1}$$

while satisfying all the safety constraints in (c) and the dynamics constraint (a). $\mathbb{E}$ is the expectation, and ℓ is a loss function.

3. Adaptive Explicit-Barrier Net

In this section, we present the architecture of the Adaptive explicit-Barrier Net (ABNet) and formally prove its safety guarantees in learning systems.

Our proposed method can fuse machine learning models that can strictly enforce system safety. In the literature, to make the safety model trainable without losing guarantees, we would usually require the model to be in the form of differentiable convex optimizations, such as differentiable QP (Amos & Kolter, 2017), differentiable MPC (Amos et al., 2018) or differentiable CBF (Xiao et al., 2023). In the former two cases, the considered robot learning problems are usually with linear dynamics and linear constraints. Otherwise, the optimization becomes nonlinear (i.e., not trainable in neural networks). Although one can transform constrained optimizations into unconstrained optimizations that are trainable using classical barrier functions (Boyd &

Vandenberghe, 2004), it may make the system lose safety guarantees.

3.1. Multi-head Explicit-Barrier

We focus on general safe robot learning problems with nonlinear dynamics and constraints. For such problems, it has been shown that we can use the CBF transformation to reduce nonlinear optimizations onto quadratic optimizations with safety guarantees (Ames et al., 2017) (Xiao & Belta, 2021), which gives rise to the so-called BarrierNet (Xiao et al., 2023).

Specifically, consider robot dynamics as: $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$, where $\boldsymbol{x} \in \mathbb{R}^n$ is the robot state, $f: \mathbb{R}^n \to \mathbb{R}^n$ and $g: \mathbb{R}^n \to \mathbb{R}^{n\times q}$ are locally Lipschitz, and $\boldsymbol{u} \in \mathbb{R}^q$ is the control. We can also consider non-affine control systems by defining auxiliary systems (Xiao et al., 2021).

Implicit-Barrier. The constrained optimal control in the considered problem in Sec. 2 is then transformed into the following differentiable CBF/BarrierNet (Xiao et al., 2023), which may form a head of the model:

$$\boldsymbol{u}_k = \arg\min_{\boldsymbol{u}(t)} \frac{1}{2}\boldsymbol{u}(t)^T H(\boldsymbol{z}_k|\theta_{h,k})\boldsymbol{u}(t) + F^T(\boldsymbol{z}_k|\theta_{f,k})\boldsymbol{u}(t) \tag{2}$$

s.t.

$$\begin{aligned} &L_f\psi_{j,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p) + [L_g\psi_{j,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)]\boldsymbol{u} \\ &\quad + p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{j,m}(\psi_{j,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)) \geq 0, j \in S, \\ &\quad \psi_{j,i}(\boldsymbol{x},\boldsymbol{z}|\theta_p) = \dot{\psi}_{j,i-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p) \\ &\quad + p_i(\boldsymbol{z}|\theta^i_p)\alpha_{j,i}(\psi_{j,i-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)), i \in \{1,\dots,m-1\}, j \in S, \\ &\quad \psi_{j,0}(\boldsymbol{x},\boldsymbol{z}|\theta_p) = b_j(\boldsymbol{x}), j \in S, \end{aligned} \tag{3}$$

where $H(\boldsymbol{z}_k|\theta_{h,k}) \in \mathbb{R}^{q\times q}$ is positive definite, and $-H^{-1}(\boldsymbol{z}_k|\theta_{h,k})F(\boldsymbol{z}_k|\theta_{f,k})$ can be interpreted as a reference control (the output of previous network layers). The constraints above are the High-Order CBFs (HOCBFs) constructed in enforcing the safety constraints $b_j(\boldsymbol{x}) \geq 0$ of relative degree m(Xiao & Belta, 2021). It can be shown that the satisfaction of the first HOCBF constraint in the above implies the satisfaction of $b_j(\boldsymbol{x}) \geq 0, \forall j \in S$, which proves the safety of differentiable CBF. In the above, $L_f\psi = \frac{d\psi}{dx}f(\boldsymbol{x}), L_g\psi = \frac{d\psi}{dx}g(\boldsymbol{x}), k \in \{1,\dots,h\}$, and h is the number of heads (as shown in Fig. 1). $p_i \geq 0, i \in \{1,\dots,m-1\}, p_{m,k} \geq 0$ are penalty functions (outputs of the previous network, as shown in Fig. 2) on the strictly increasing and zero-passing functions $\alpha_{j,i}, i \in \{1,\dots,m\}, j \in S$, and will determine the conservativeness of the robot. $\theta := (\theta_h, k, \theta_{f,k}, \theta^m_{p,k}, \theta_p), k \in \{1,\dots,h\}$, where $\theta_p := (\theta^1_p,\dots,\theta^{m-1}_p)$ are all trainable parameters. $\boldsymbol{z}_k$ is the observation of the head k, and it is possible that all heads share the same observation, i.e. $\boldsymbol{z}_k = \boldsymbol{z}, \forall k \in \{1,\dots,h\}$.

The training of the above differentiable CBF (3) involves solving batch QPs (Amos & Kolter, 2017), which is inefficient. Since CBFs do not explicitly show up in the solution, we call differentiable CBF (3) as *implicit-barrier*. In the following, we derive the trainable explicit solution of the differentiable CBF, which is our proposed *explicit-barrier*.

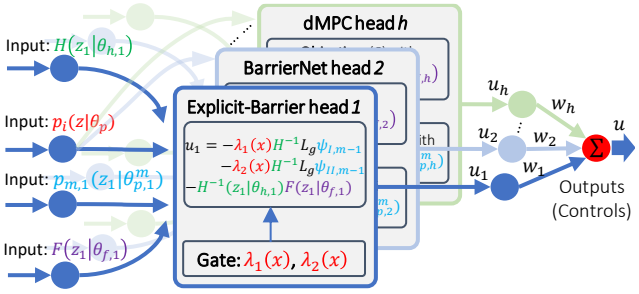

Figure 2: Architecture of multi-head explicit-barriers. ABNet is capable (adaptive) to fuse any safe learning models, such as the proposed explicit-barriers, BarrierNet, dMPC, etc. The ABNet is usually used in conjunction with any other neural networks and can be implemented in parallel. The parameters (inputs) of each head of ABNet are the outputs of previous layers (such as CNN or LSTM).

Explicit-Barrier. It has been shown in (Luenberger, 1997) that we can find the explicit solution of a QP if there are only two constraints. As the cardinality of the safety constraint set S may be greater than two, the number of HOCBFs in the differentiable CBF (3) will also be greater than two. In order to address this, first, we can define two safety functions $b_I(\boldsymbol{x}) = -ln(\sum_{j\in S_1} exp(-b_j(\boldsymbol{x})))$ and $b_{II}(\boldsymbol{x}) = -ln(\sum_{j\in S_2} exp(-b_j(\boldsymbol{x})))$, where $S = S_1 \cup S_2$. By (Boyd & Vandenberghe, 2004), we have that $\max_{j\in S} b_j(\boldsymbol{x}) \leq ln(\sum_{j\in S} exp(b_j(\boldsymbol{x})))$. It can be easily shown that $b_I(\boldsymbol{x}) \geq 0$ and $b_{II}(\boldsymbol{x}) \geq 0$ implies $b_j(\boldsymbol{x}) \geq 0, \forall j \in S$.

Alternatively, we can simply consider the two most risk safety specifications, i.e., $b_I(\boldsymbol{x}) = \min_{i\in S} b_j(\boldsymbol{x})$, $b_{II}(\boldsymbol{x}) = \min_{i\in S\setminus \arg\min_{i\in S} b_j(\boldsymbol{x})} b_j(\boldsymbol{x})$. This approach is simpler and it works well for most obstacle avoidance tasks.

Then, the explicit optimal solution of the differentiable CBF with the above two safety specifications can be given by

$$\begin{aligned} u_k = &-\lambda_1(\boldsymbol{x})H^{-1}L_g\psi_{I,m-1}(\boldsymbol{x}) \\ &-\lambda_2(\boldsymbol{x})H^{-1}L_g\psi_{II,m-1}(\boldsymbol{x}) - H^{-1}F, \end{aligned} \tag{4}$$

where H, F are given as in (2) with arguments omitted. $\lambda_1(\boldsymbol{x}) \leq 0, \lambda_2(\boldsymbol{x}) \leq 0$ are two gate functions ((14), (15) given in Appendix), and $\psi_{I,m-1}(\boldsymbol{x}), \psi_{II,m-1}(\boldsymbol{x})$ are the two HOCBFs corresponding to $b_I(\boldsymbol{x}), b_{II}(\boldsymbol{x})$ defined similarly as in (3). As we can see that $b_I(\boldsymbol{x}), b_{II}(\boldsymbol{x})$ explicitly show up in the above equation while it guarantees safety, we call it *explicit-Barrier*. This makes the explicit-Barrier more interpretable. The whole explicit solution derivation process is given in Appendix Sec. A.

Adaptive mechanism. Each head of ABNet may learn different safe control policies even if all the heads have the same observation $\boldsymbol{z}$. The benefit is that the final performance is achieved "collectively" by all heads and thus each head can just focus on the "subproblem" with safety. Alternatively, we may also make each head of ABNet focus on different observations $\boldsymbol{z}_k$. The observation $\boldsymbol{z}_k$ may come from different parts of the sensor observation (such as the left lane boundary and right lane boundary in driving shown in Fig. 1), or even different perceptions (such as vision, lidar, etc.)

Cross connection. It can be noted from (3) that each head of ABNet $k \in \{1,\dots,h\}$ has some cross connection with other heads, as also shown in Fig. 1. In other words, $\psi_{j,i}(\boldsymbol{x},\boldsymbol{z}|\theta_p), i \in \{1,\dots,m-1\}, j \in S$ are formulated in the same way through the shared parameter θ_p (independent from k). This is to ensure (i) the construction for provable safety (as shown later), and (ii) some shared information across different heads of ABNet as they all generate safe controls for the robot.

Fusion. Another important consideration is how should we fuse all these controls $\boldsymbol{u}_k, k \in \{1,\dots,h\}$ while preserving the safety property of each head of the ABNet. We propose the following form:

$$\boldsymbol{u} = \sum_{k=1}^{h} w_k \boldsymbol{u}_k, \quad \text{where } \sum_{k=1}^{h} w_k = 1. \tag{5}$$

In the above, $w_k \geq 0, k \in \{1,\dots,h\}$ are trainable parameters. The composition of explicit-Barrier (4), BarrierNet, and dMPCs, etc, in the form of (5) is our proposed *ABNet*, as shown in Fig. 2. The safety guarantees of the ABNet is shown in the following theorem:

Theorem 3.1. *(**Safety of ABNets**) Given the multi-head ABNet formulated as in (4) and all other safe learning models (BarrierNet, dMPC, etc.). If the system is initially safe (i.e., $b_j(\boldsymbol{x}(t_0)) \geq 0, \forall j \in S$), then a control policy $\boldsymbol{u}$ from the ABNet output (5) guarantees the safety of system, i.e., $b_j(\boldsymbol{x}(t)) \geq 0, \forall j \in S, \forall t \geq t_0$.*

All the proofs for theorems are given in Appendix B. If the system is not initially safe (i.e., $b_j(\boldsymbol{x}(t_0)) < 0, \exists j \in S$), then the system state $\boldsymbol{x}$ will be driven to the safe side of the state space due to the Lyapunov property of CBF/HOCBFs (Ames et al., 2017) (Xiao & Belta, 2021). This enables the possibility of utilizing data that violates safety to conduct adversary training of the ABNet.

Natural noise filter. The ABNet is a natural noise filter since $w_k \in [0,1], \forall k \in \{1,\dots,h\}$ in (5). This can ensure that the output $\boldsymbol{u}$ of the model is stable with a large enough head number h if all the heads have different observation $\boldsymbol{z}_k$ for the current environment. This feature makes ABNet a very robust and adaptive controller for robotic systems, and thus, ABNet can generate smooth signals.

Algorithm 1 Construction and training of ABNet

Input: the problem setup (a)-(d) given in the problem formulation (Sec. 2).
Output: a robust and safe controller $\boldsymbol{u}$ for the system.
(a) Formulate each head of explicit-Barriers as in (4).
(b) Build the cross connection among explicit-Barriers via $p_i(\boldsymbol{z}|\theta_p^i), i \in \{1,\dots,m-1\}$.
(c) Fuse all the heads of explicit-Barriers as in (5).
if *Incremental training* **then**
  Decouple $p_i(\boldsymbol{z}|\theta_p^i), i \in \{1,\dots,m-1\}$ and define them for each explicit-Barrier.
  Train each head of explicit-Barriers, respectively.
  Choose a $p_i(\boldsymbol{z}|\theta_p^i), i \in \{1,\dots,m-1\}$ from one of the explicit-Barriers to build cross connection.
  Fuse all the explicit-Barriers via (6).
else
  Directly train the ABNet via reverse mode error back propagation.
end if

Theorem 3.2. *(**Safety of merging of ABNets**) Given two ABNets, the merged model using the form as in (5) again guarantees the safety of system.*

3.2. Model Training

The ABNet can be trained incrementally or in one-shot. This is due to the fact that each head of ABNet can generate a control policy that is applicable to the system. The linear combination weights $w_k, k \in \{1,\dots,h\}$ in the ABNet denote the importance of the corresponding control policies.

Incremental training. In ABNet, we may train each head $k, k \in \{1,\dots,h\}$ of the model in a scalable way as we wish to minimize the loss between their output $\boldsymbol{u}_k$ and the label $\boldsymbol{u}^*$ as well. The training can be done by directly incorporating the explicit-Barrier (4) into the model. There are some cross connections via $p_i(\boldsymbol{z}|\theta_p)$ between explicit-Barriers in the ABNet that may prevent the implementation of the training. We may address this by training a $p_i(\boldsymbol{z}|\theta_p)$ for each head of the ABNet. After we train all heads of the model, we may fix the parameters of those models, choose a $p_i(\boldsymbol{z}|\theta_p)$ from one of the explicit-Barriers (or take an average of all $p_i(\boldsymbol{z}|\theta_p)$ among the models) to build the cross connection, and train the weights w_i for some more iterations. Another way is to fuse these explicit-Barriers by testing loss. In other words, the weight $w_k, k \in \{1,\dots,h\}$ can be determined by:

$$w_k = \frac{1/\ell_k(\boldsymbol{u}_k, \boldsymbol{u}^*)}{\sum_{k=1}^{h} 1/\ell_k(\boldsymbol{u}_k, \boldsymbol{u}^*)}, \tag{6}$$

where ℓ_k is a loss function.

If we already have some trained ABNet, and we wish to

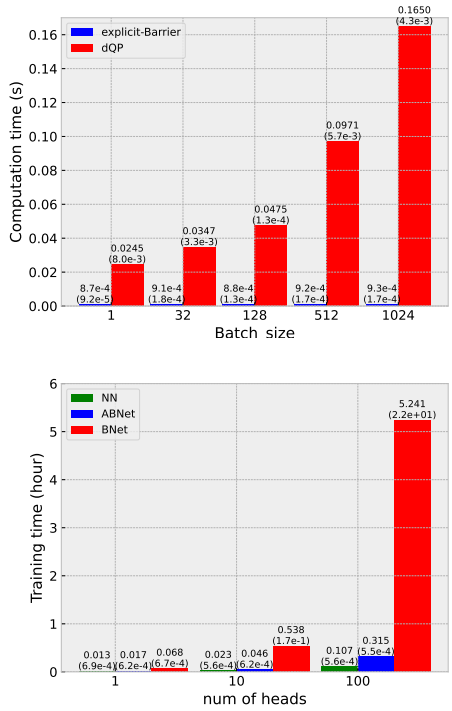

Figure 3: Computation (upper, numbers in the bracket denote variance) and training efficiency (lower, numbers in the bracket denote testing loss) comparison of our proposed explicit-Barrier (ABNet) with dQP and BarrierNet (BNet). The use of dQP in BarrierNet could give very bad solutions. NN is a normal neural network without safety guarantees.

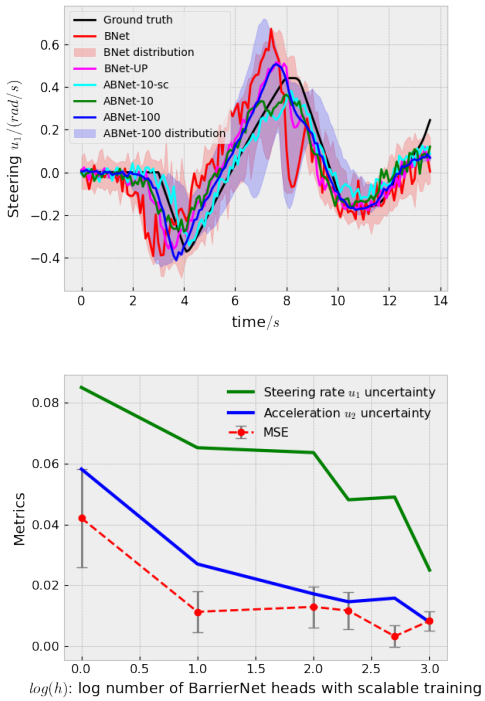

Figure 4: 2D robot obstacle avoidance closed-loop testing control profiles (upper) and ABNet performance with the increasing of ABNet heads using scalable training (lower). This scalable training for ABNet is with safety guarantees. The controls are subject to input noise, and thus are non-smooth.

add some new capabilities (such as safe driving by only focusing on the left lane boundary) to the model, then we can train some heads of ABNets based on the new data we have. Finally, we can fuse the models similarly with safety guarantees as shown in Thm. 3.2. This shows the scalability of the proposed ABNet that allows us to build larger foundational safe models in an incremental way.

One-shot/Direct training. The one-shot training of the ABNet can be directly done using the traditional reverse mode automatic differentiation. In addition to the loss between the eventual output $\boldsymbol{u}$ of the ABNet and the label $\boldsymbol{u}^*$, we may also consider the losses on $\boldsymbol{u}_k, k \in \{1,\dots,h\}$, as well as on the reference controls $H^{-1}(\boldsymbol{z}_k|\theta_{h,k})F(\boldsymbol{z}_k|\theta_{f,k})$, in order to improve the training performance.

The construction and training of the ABNet involve the formulation of each head of explicit-Barriers as in (4), the model fusion as in (5), and the scalable or direct training as shown above (Alg. 1).

4. Experiments

In this section, we conduct several experiments to answer the following questions:

- How does the proposed explicit-Barrier compare with dQP (Amos & Kolter, 2017) and BarrierNet (Xiao & Belta, 2021) in terms of computation and training efficiency?
- Does our method match the theoretic results in experiments and **is it scalable**?
- How does our method compare with state-of-the-art models in enforcing safety constraints?
- The benefit of models/policies merging and the robustness of our models in safety and smoothness?

Benchmark models: We compare with (i) *baseline*: **Tables 1, 2**–single end-to-end learning model (E2E) (Levine et al., 2016) and **Table 3**–single vanilla end-to-end (V-E2E) model (Amini et al., 2022), (ii) *safety guaranteed models*: (implicit-) BarrierNet (BNet) (Xiao et al., 2023), Deep forward and backward (DFB) model (Pereira et al., 2020), (iii) *policies merging*: BarrierNet policies merged with uncertainty propagation (BNet-UP) (Wang et al., 2023b) that employs Gaussian kernels with Scott's rule (Scott, 2015) to select the bandwidth, (iv) *models merging*: E2Es merged with Monte-Carlo Dropout (E2Es-MCD) (Gal & Ghahramani, 2016), E2Es merged with Deep Resembles (E2Es-DR) (Lakshminarayanan et al., 2017).

Table 1: 2D robot obstacle avoidance closed-loop testing under noisy input.

MODEL	SAFETY (≥ 0)	CONSER. (≥ 0& $\downarrow$)	MSE($\downarrow$)	u_1 UNCERTAINTY ($\downarrow$)	u_2 UNCERTAINTY ($\downarrow$)	THEORET. GUAR.
E2E (LEVINE ET AL., 2016)	-14.140	-2.976 ± 3.770	0.007 ± 0.004	0.063	0.049	$\times$
E2ES-MCD (GAL & GHAHRAMANI, 2016)	-2.087	-1.341 ± 0.824	0.004 ± 0.001	0.041	0.026	$\times$
E2ES-DR (LAKSHMINARAYANAN ET AL., 2017)	-35.130	-3.176 ± 4.299	0.080 ± 0.006	0.032	0.020	$\times$
DFB (PEREIRA ET AL., 2020)	36.659	47.810 ± 4.377	0.013 ± 0.003	0.062	0.052	$\surd$
BNET (XIAO ET AL., 2023)	5.045	7.966 ± 1.287	0.014 ± 0.006	0.074	0.047	$\surd$
BNET-UP (WANG ET AL., 2023B)	5.988	8.573 ± 1.738	0.008 ± 0.004	0.054	0.028	$\times$
ABNET-10-SC (OURS)	5.731	6.269 ± 0.319	0.011 ± 0.007	0.065	0.027	$\surd$
ABNET-10 (OURS)	12.639	13.887 ± 1.323	0.008 ± 0.005	0.049	0.030	$\surd$
ABNET-100 (OURS)	10.122	11.729 ± 0.816	0.012 ± 0.006	0.049	0.013	$\surd$

Our models: We consider the minimum function method in determining $b_I(\boldsymbol{x})$ and $b_{II}(\boldsymbol{x})$. *Sec. 4.2 and 4.3*: ABNet trained in a scalable way with 10 heads (ABNET-10-SC), ABNet trained in one shot with 10 heads (ABNET-10), ABNet trained in one shot with 100 heads (ABNET-100). *Sec. 4.4*: our ABNet trained in one shot with 10 heads using the same input images (ABNET), ABNet with attention images and 10 heads (ABNET-ATT), our ABNet first trained with ABNET scaled by ABNET-ATT (20 heads, ABNET-SC).

Evaluation metrics: The evaluation metrics are defined as follows: mean square error of the model testing (MSE), satisfaction of safety constraints where non-negative values mean safety guarantees (SAFETY), system conservativeness (CONSER.), steering control u_1 uncertainty (u_1 UNCERTAINTY), acceleration control u_2 uncertainty (u_2 UNCERTAINTY), and theoretical safety guarantees (THEORET. GUAR.) respectively. The metrics are explicitly defined in Appendx C.

4.1. Computation and Training Time

We first compare the training stability and efficiency of our proposed explicit-Barrier (or ABNet) with dQP (Amos & Kolter, 2017) and BarrierNet (Xiao et al., 2023). The dQP method is based on the "QPFunction" library from OptNet (Amos & Kolter, 2017), and BarrierNet is based on dQP. The computation times under different batch sizes are shown in Fig. 3 (upper). The computation time significantly increases as the increasing of the batch size, but the proposed explicit-Barrier remains to be efficient. Fig. 3 (lower) shows the training time (of the model based on the 2D robot case in Sec. 4.2) under different number of heads. The training time of our proposed ABNet is comparable to a normal NN (NN is without safety guarantees). While the BarrierNet (based on dQP) tends to give very bad training and testing solutions, as also shown in Fig. 7 in the Appendix, which significantly deteriorates the quality of model training.

4.2. 2D Robot Obstacle Avoidance

We aim to find a neural network controller for a 2D robot that can drive the robot from an initial location to an arbitrary destination while avoiding crash onto the obstacle. All the models (h copies/heads) have the same input (with uniformly distributed noise, 10% of the input magnitude in testing). The detailed problem setup and model introductions are given in Appendix C.2.

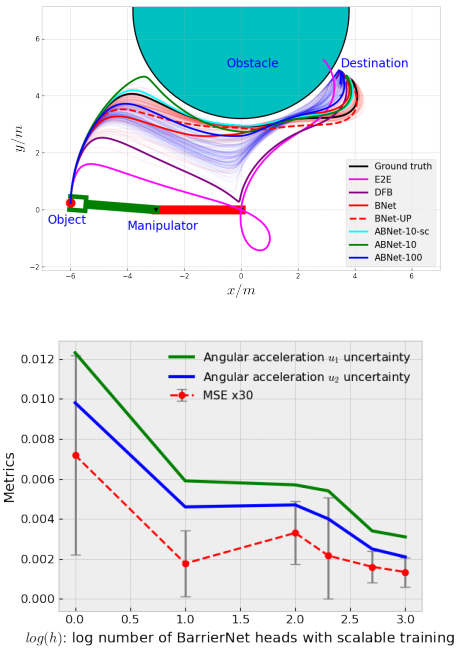

Figure 5: Robot manipulation closed-loop end-effector trajectories (upper) and ABNet performance with the increasing of model heads using scalable training (lower). The transparent trajectories in the upper figure are corresponding to results in all runs.

Models/policies merging can improve the performance as

Table 2: Robot manipulation closed-loop testing under noisy input and comparisons with benchmarks.

MODEL	SAFETY (≥ 0)	CONSER. (≥ 0& $\downarrow$)	MSE($\downarrow$)	u_1 UNCERTAINTY ($\downarrow$)	u_2 UNCERTAINTY ($\downarrow$)	THEORET. GUAR.
E2E (LEVINE ET AL., 2016)	-11.027	-1.082 ± 2.992	$3.6e\text{-}4\pm1.7e\text{-}4$	0.013	0.009	$\times$
E2ES-MCD (GAL & GHAHRAMANI, 2016)	-11.827	0.162 ± 2.085	$1.1e\text{-}4\pm7.3e\text{-}5$	0.008	0.005	$\times$
E2ES-DR (LAKSHMINARAYANAN ET AL., 2017)	-11.381	-0.958 ± 1.875	$1.3e\text{-}4\pm8.5e\text{-}5$	0.007	0.005	$\times$
DFB (PEREIRA ET AL., 2020)	2.905	6.023 ± 3.110	$8.7e\text{-}4\pm1.9e\text{-}4$	0.019	0.018	$\surd$
BNET (XIAO ET AL., 2023)	0.147	0.745 ± 0.505	$2.3e\text{-}4\pm1.2e\text{-}4$	0.010	0.009	$\surd$
BNET-UP (WANG ET AL., 2023B)	0.206	0.346 ± 0.098	$5.2e\text{-}5\pm3.2e\text{-}5$	0.005	0.005	$\times$
ABNET-10-SC (OURS)	0.233	0.570 ± 0.360	$5.9e\text{-}5\pm5.5e\text{-}5$	0.006	0.005	$\surd$
ABNET-10 (OURS)	0.039	0.272 ± 0.443	$1.2e\text{-}4\pm9.6e\text{-}5$	0.008	0.007	$\surd$
ABNET-100 (OURS)	0.053	0.123 ± 0.177	$1.1e\text{-}4\pm4.4e\text{-}5$	0.005	0.004	$\surd$

shown by the MSE metrics in Table 1 and the scalable training in Fig. 4. Note that our scalable training for ABNets has safety guarantees. The DFB tends to be very conservative as the CBFs within which are not differentiable, which presents a high conservative value shown in Table 1. Our proposed ABNets can significantly reduce the uncertainty of the outputs (controls) under noisy input while guaranteeing safety, and this uncertainty decreases as the increases of the heads in the ABNets, as shown by the last two and three columns in Table 1, as well as shown in Fig. 4 and 8 of Appendix C.2 where the control uncertainty of ABNet-100 is lower than the one of BNet. The smoothness of the controls also increases with the increase of model heads (e.g., blue from ABNet v.s. red from BNet in Fig. 8). In terms of performance, our proposed ABNets can also improve the testing errors compared to BNet and DFB, as shown by the MSE in Table 1. The E2Es-MCD model can achieve the best performance, but this is at the cost of safety (the SAFETY metric in Table 1 is negative, which implies violated safety).

4.3. Safe Robot Manipulation

In robot manipulation, we employ a two-link planar robot manipulator to grasp an object from an arbitrary point to an arbitrary destination while avoiding crashing onto obstacles. All the models (h copies/heads) have the same input (with uniformly distributed noise, 10% of the input magnitude in testing). We compare our proposed ABNets with the same benchmark models as in the last subsection. More detailed problem setup and model introductions are given in Appendix C.3.

Again, models/policies merging can improve the performance as shown by the MSE metrics in Table 2 and the scalable training in Fig. 5. All the E2E-related models are not robust to noise and violate safety constraints (i.e., crash onto obstacles) under noisy input since there are no formal guarantees, and such an example is shown by the magenta trajectory curve of the end-effector in Fig. 5. As shown in Table 2, the proposed ABNet-100 model is the least conservative one with the lowest control uncertainties as well under noisy inputs (significantly improved compared with BNet and DFB), which demonstrates its advantage over other models. This uncertainty improvement is also shown by the control distributions in Fig. 9 in Appendix C.3 (BNet: red area v.s. ABNet-100: blue area). The BNet-UP achieves the best performance without safety guarantees.

4.4. Vision-based End-to-End Autonomous Driving

We finally test our models in a more complicated and realistic task: vision-based driving, using an open dataset and benchmark from the VISTA (Amini et al., 2022). One of ABNets, named ABNet-att, is constructed such that different heads focus on different parts of the image (left lane boundary, right lane boundary, etc., the corresponding images are shown in Fig 10 of Appendix C.4). For more experiment and model details, please refer to Appendix C.4.

As shown in Table 3, the proposed ABNets can avoid crash onto obstacles with 100% obstacle passing rate, including the ABNet-sc that is trained in a scalable way with two ABNets (also shown by the scalable training in Fig. 6). This is because the ABNets can learn the correct steering control (the blue and green sine waves shown in Fig. 11 (right) in Appendix C.4) to avoid the obstacle without stopping in front of it. Compared to the baseline MPC, the proposed ABNet is much more efficient (0.004s v.s. 0.872s). Although linearization is possible in MPC to improve the efficiency, it may make the MPC lose safety guarantees. The DFB and BNet-related models learn a significant deceleration control (shown in Fig. 11) to avoid crashing onto obstacles, which explains why the corresponding obstacle passing rates are low compared to other models in Table 3 and why the blue trajectories (BNet) terminate near the obstacle in Fig. 6 (upper). Nonetheless, there are still some crash cases in DFB and BNet models due to badly learned CBF parameters that make the inter-sampling effect (i.e., safety violation

Table 3: Vision-based end-to-end autonomous driving closed-loop testing and comparisons with benchmarks. New items are short for obstacle crash rate (CRASH), obstacle passing rate (PASS).

MODEL	CRASH (↓)	PASS (↑)	SAFETY (≥ 0)	CONSER. (≥ 0& ↓)	u_1 UNCERTAINTY (↓)	u_2 UNCERTAINTY (↓)	THEORET. GUAR.
V-E2E (AMINI ET AL., 2022)	6%	94%	-60.297	−0.610±21.165	0.443	0.222	×
E2ES-MCD (GAL & GHAHRAMANI, 2016)	8%	92%	-60.566	−2.211±22.343	0.429	0.227	×
E2ES-DR (LAKSHMINARAYANAN ET AL., 2017)	9%	91%	-60.572	−1.499±21.500	0.431	0.224	×
DFB (PEREIRA ET AL., 2020)	4%	39%	-18.114	−0.828±5.444	0.513	0.125	√
BNET (XIAO ET AL., 2023)	3%	33%	-16.694	−4.882±4.817	0.724	0.385	√
BNET-UP (WANG ET AL., 2023B)	2%	35%	-23.252	−5.190±4.920	0.726	0.532	×
ABNET (OURS)	0%	100%	1.455	6.132±2.181	0.168	0.316	√
ABNET-ATT (OURS)	0%	100%	4.198	8.053±1.449	0.172	0.269	√
ABNET-SC (OURS)	0%	100%	2.221	7.224±1.667	0.130	0.256	√

between discretized times) serious. Most importantly, our proposed ABNet can learn less uncertain controls for this complicated task, as shown in Table 3, the scalable training in Fig. 6, and Fig. 11 (e.g., ABNet:blue or ABNet-att:green area v.s. BNet: red area).

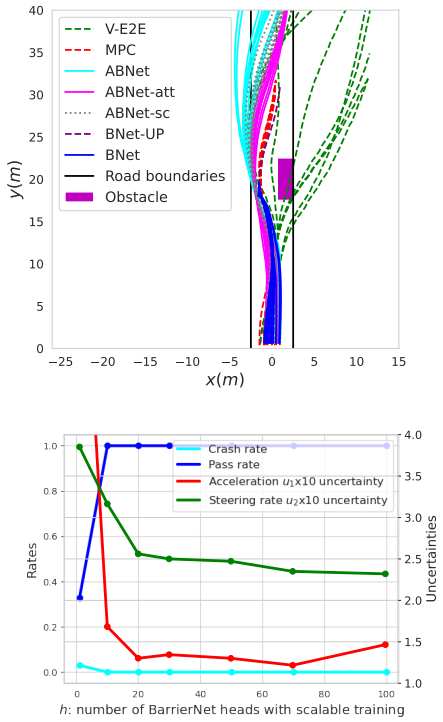

Figure 6: Vision-based end-to-end autonomous driving closed-loop testing trajectories in VISTA (upper) and ABNet performance with the increasing of model heads using scalable training (lower). This scalable training is done by both the ABNet and ABNet-att in Table 3 with safety guarantees.

The ABNet-att can learn more consistent autonomous driving behavior than the ABNet due to the image attention setting, as shown by the magenta (ABNet-att) and cyan (ABNet) trajectories in Fig. 6 (upper) and the green (ABNet-att) and blue (ABNet) areas in Fig. 11. **Ablation studies** on the robustness of our ABNets in terms of safety under high-noisy inputs (50% noise level) are given in Table 4 of Appendix C.4.

5. Related Works

Scalability, merging and uncertainty in safe robot learning. Machine learning has been widely used in robot control (Bommasani et al., 2021) (Singh et al., 2023) (Wang et al., 2023a). However, there is increasing concern for machine learning, especially large foundation models, being used in robotics (Bommasani et al., 2021). Mixture of expert methods (Shazeer et al., 2017) (Riquelme et al., 2021) (Zhou et al., 2022) are scalable but hard to retain the property (such as safety) of the models. The uncertainty resulting from noisy model input or dataset is preventing the deployment to real robots (Loquercio et al., 2020) (Kahn et al., 2017). To address this, predictive uncertainty quantification (Gal & Ghahramani, 2016) (Lakshminarayanan et al., 2017), also a model merging approach, has been widely adopted. It has been shown to work well in vision-based autonomous driving under noisy input (Wang et al., 2023b) using the Gaussian kernel with Scott's rule (Scott, 2015) to select bandwidth. The main challenge of this technique is that it may make the system lose performance guarantees, such as safety. Other model merging approaches (Huang et al., 2023) (Ramé et al., 2023) (Wang et al., 2024) do not preserve safety either. We address the uncertainty and scalablibity problem using the proposed ABNets with provable safety.

CBFs and set invariance. In control theory, the set invariance has been widely adopted to prove and enforce the safety of dynamical systems (Blanchini, 1999) (Rakovic et al., 2005) (Ames et al., 2017) (Xiao & Belta, 2021) (Xiao

et al., 2023). The Control Barrier Function (CBF) (Ames et al., 2017) (Xiao & Belta, 2021) is such a state of the art technique that can enforce set invariance (Aubin, 2009), (Prajna et al., 2007), (Wisniewski & Sloth, 2013), and transforms a nonlinear optimization problem to a quadratic problem that is very efficient to solve. CBFs originates from barrier functions that are originally used in optimization problems (Boyd & Vandenberghe, 2004). However, the CBF tends to make the system conservative (i.e., at the cost of performance) in order to enforce safety, and it is not scalable to build large models. Our proposed ABNet can address all these limitations.

Safety in neural networks. Safety is usually enforced using optimizations. Barrier functions have been widely used in safe Reinforcement Learning (RL) (Tessler et al., 2018; Achiam et al., 2017). However, safety cannot be guaranteed in safe RL as the barrier functions are used as part of the reward function (a soft constraint). Recently, differentiable optimizations show great potential for learning-based control with safety guarantees (Pereira et al., 2020; Amos et al., 2018; Xiao et al., 2023; Liu et al., 2023). The quadratic program (QP) can be employed as a layer in the neural network, i.e., the OptNet (Amos & Kolter, 2017). The OptNet has been used with CBFs in neural networks as a safe filter controls (Pereira et al., 2020), in which CBFs themselves are not trainable, which can significantly limit the learning capability. Neural network controllers with safety certificate have been learned through verification-in-the-loop training (Deshmukh et al., 2019; Zhao et al., 2021; Ferlez et al., 2020). However, the verification method cannot guarantee to cover the whole state space, and this method is also very computationally expensive. None of these methods are scalable to larger models, and are subject to uncertainty, which the proposed ABNet can address.

6. Conclusions, Limitations and Future Work

We propose Adaptive explicit-Barrier Net (ABNet) that merges many safety-critical learning models while preserving the safety in this paper. The proposed ABNet is efficient to train, scalable to build larger safe learning models, can achieve better performance, and is robust to input noise. We have demonstrated the effectiveness of the model on a series of robot control tasks. Nonetheless, our model (and all the other barrier-based learning models (Ferlez et al., 2020) (Xiao et al., 2023)) still have a few limitations motivating for further research.

Limitations. First, all the ABNets have the same safety constraints. We will explore how to merge ABNets with different safety constraints in the future. Second, the ABNet also requires safety specifications that may be unknown in some robot control tasks, we may learn the safety specifications from data (Robey et al., 2020), (Srinivasan et al., 2020), and this can also be done in conjunction with ABNet. Third, the model merging is done in the output space, future work will further focus on model merging with safety guarantees in the parameter space. Finally, we will apply the proposed model in environments that involve contact handling, such as grasping.

Acknowledgements

The research was supported in part 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 research was also supported in part by the AI2050 program at Schmidt Futures (Grant G-965 22-63172), and by the ONR Science of Autonomy program N00014-23-1-2354.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

Agarwal, N., Brukhim, N., Hazan, E., and Lu, Z. Boosting for control of dynamical systems. In *International Conference on Machine Learning*, pp. 96–103. PMLR, 2020.

Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

Amini, A., Wang, T.-H., Gilitschenski, I., Schwarting, W., Liu, Z., Han, S., Karaman, S., and Rus, D. 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)*, pp. 2419–2426. IEEE, 2022.

Amos, B. and Kolter, J. Z. Optnet: Differentiable optimiza-

tion as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pp. 136–145, 2017.

Amos, B., Rodriguez, I. D. J., Sacks, J., Boots, B., and Kolter, J. Z. Differentiable mpc for end-to-end planning and control. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8299–8310. Curran Associates Inc., 2018.

Aubin, J.-P. *Viability theory*. Springer, 2009.

Beygelzimer, A., Hazan, E., Kale, S., and Luo, H. Online gradient boosting. *Advances in neural information processing systems*, 28, 2015.

Blanchini, F. Set invariance in control. *Automatica*, 35(11): 1747–1767, 1999.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Boyd, S. P. and Vandenberghe, L. *Convex optimization*. Cambridge university press, New York, 2004.

Deshmukh, J. V., Kapinski, J. P., Yamaguchi, T., and Prokhorov, D. Learning deep neural network controllers for dynamical systems with safety guarantees: Invited paper. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–7, 2019.

Ferlez, J., Elnaggar, M., Shoukry, Y., and Fleming, C. Shieldnn: A provably safe nn filter for unsafe nn controllers. *preprint arXiv:2006.09564*, 2020.

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

Glotfelter, P., Cortes, J., and Egerstedt, M. Nonsmooth barrier functions with applications to multi-robot systems. *IEEE control systems letters*, 1(2):310–315, 2017.

Huang, C., Liu, Q., Lin, B. Y., Pang, T., Du, C., and Lin, M. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*, 2023.

Kahn, G., Villaflor, A., Pong, V., Abbeel, P., and Levine, S. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*, 2017.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Liu, W., Xiao, W., and Belta, C. Learning robust and correct controllers from signal temporal logic specifications using barriernet. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 7049–7054. IEEE, 2023.

Loquercio, A., Segu, M., and Scaramuzza, D. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, 2020.

Luenberger, D. G. *Optimization by vector space methods*. John Wiley & Sons, 1997.

Nagumo, M. Über die lage der integralkurven gewöhnlicher differentialgleichungen. In *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series. 24:551-559*, 1942.

Pereira, M. A., Wang, Z., Exarchos, I., and Theodorou, E. A. Safe optimal control using stochastic barrier functions and deep forward-backward sdes. In *Conference on Robot Learning*, 2020.

Prajna, S., Jadbabaie, A., and Pappas, G. J. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52 (8):1415–1428, 2007.

Rakovic, S. V., Kerrigan, E. C., Kouramas, K. I., and Mayne, D. Q. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on automatic control*, 50(3):406–410, 2005.

Ramé, A., Ahuja, K., Zhang, J., Cord, M., Bottou, L., and Lopez-Paz, D. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning*, pp. 28656–28679. PMLR, 2023.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34: 8583–8595, 2021.

Robey, A., Hu, H., Lindemann, L., Zhang, H., Dimarogonas, D. V., Tu, S., and Matni, N. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE*

Conference on Decision and Control (CDC), pp. 3717–3724, 2020.

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

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

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530. IEEE, 2023.

Srinivasan, M., Dabholkar, A., Coogan, S., and Vela, P. A. Synthesis of control barrier functions using a supervised machine learning approach. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7139–7145, 2020.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Wang, L., Zhao, J., Du, Y., Adelson, E. H., and Tedrake, R. Poco: Policy composition from and for heterogeneous robot learning. *arXiv preprint arXiv:2402.02511*, 2024.

Wang, T.-H., Maalouf, A., Xiao, W., Ban, Y., Amini, A., Rosman, G., Karaman, S., and Rus, D. Drive anywhere: Generalizable end-to-end autonomous driving with multi-modal foundation models. *arXiv preprint arXiv:2310.17642*, 2023a.

Wang, T.-H., Xiao, W., Chahine, M., Amini, A., Hasani, R., and Rus, D. Learning stability attention in vision-based end-to-end driving policies. In *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 1099–1111. PMLR, 15–16 Jun 2023b.

Wisniewski, R. and Sloth, C. Converse barrier certificate theorem. In *Proc. of 52nd IEEE Conference on Decision and Control*, pp. 4713–4718, Florence, Italy, 2013.

Xiao, W. and Belta, C. High-order control barrier functions. *IEEE Transactions on Automatic Control*, 67(7):3655–3662, 2021.

Xiao, W., Belta, C., and Cassandras, C. G. Adaptive control barrier functions. *IEEE Transactions on Automatic Control*, 67(5):2267–2281, 2021.

Xiao, W., Wang, T.-H., Hasani, R., Chahine, M., Amini, A., Li, X., and Rus, D. Barriernet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics*, 39(3):2289–2307, 2023.

Zhao, H., Zeng, X., Chen, T., Liu, Z., and Woodcock, J. Learning safe neural network controllers with barrier certificates. *Form Asp Comp*, 33:437–455, 2021.

Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V., Dai, A. M., Le, Q. V., Laudon, J., et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

A. Closed-form Solution of the Explicit-Barrier

Here, we show the process of deriving the closed-form solution of the explicit-Barrier following (Luenberger, 1997) (Ames et al., 2017).

Similarly as in (3), we consider the following optimization (with the two specifications $b_I(x), b_{II}(x)$ shown in the main text) corresponding to the explicit-Barrier:

$$\boldsymbol{u}_k = \arg\min_{\boldsymbol{u}(t)} \frac{1}{2}\boldsymbol{u}(t)^T H(\boldsymbol{z}_k|\theta_{h,k})\boldsymbol{u}(t) + F^T(\boldsymbol{z}_k|\theta_{f,k})\boldsymbol{u}(t) \tag{7}$$

s.t.

$$\begin{aligned} &L_f\psi_{I,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p) + [L_g\psi_{I,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)]\boldsymbol{u} + p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{I,m}(\psi_{I,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)) \geq 0,\\ &L_f\psi_{II,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p) + [L_g\psi_{II,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)]\boldsymbol{u} + p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{II,m}(\psi_{II,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)) \geq 0, \end{aligned}$$

We first define

$$\begin{aligned} g_1(\boldsymbol{x}) &= [-L_g\psi_{I,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)], \quad h_1(\boldsymbol{x}) = L_f\psi_{I,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p) + p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{I,m}(\psi_{I,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)),\\ g_2(\boldsymbol{x}) &= [-L_g\psi_{II,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)], \quad h_2(\boldsymbol{x}) = L_f\psi_{II,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p) + p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{II,m}(\psi_{II,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)). \end{aligned} \tag{8}$$

The matrix H is positive definite in the above optimization (7), we then define

$$\begin{aligned} [\hat{g}_1(\boldsymbol{x}), \hat{g}_2(\boldsymbol{x})] &= H(\boldsymbol{z}_k|\theta_{h,k})^{-1}[g_1(\boldsymbol{x}), g_2(\boldsymbol{x})],\\ \begin{bmatrix} \hat{h}_1(\boldsymbol{x}) \\ \hat{h}_2(\boldsymbol{x}) \end{bmatrix} &= \begin{bmatrix} h_1(\boldsymbol{x}) \\ h_2(\boldsymbol{x}) \end{bmatrix} - \begin{bmatrix} g_1(\boldsymbol{x})^T \\ g_2(\boldsymbol{x})^T \end{bmatrix} \hat{\boldsymbol{u}}_k \end{aligned} \tag{9}$$

where

$$\hat{\boldsymbol{u}}_k = -H(\boldsymbol{z}_k|\theta_{h,k})^{-1}F(\boldsymbol{z}_k|\theta_{f,k}). \tag{10}$$

Next, let $\boldsymbol{v}_k := \boldsymbol{u}_k - \hat{\boldsymbol{u}}_k$ and $\langle\cdot,\cdot\rangle$ define an inner product with weight matrix $H(\boldsymbol{z}_k|\theta_{h,k})$ so that $\langle \boldsymbol{v}_k, \boldsymbol{v}_k\rangle = (\boldsymbol{v}_k)^T H(\boldsymbol{z}_k|\theta_{h,k})\boldsymbol{v}_k$. The optimization problem (7) is equivalent to:

$$\begin{aligned} \boldsymbol{v}^*_k = \arg\min_{\boldsymbol{v}_k} &\langle \boldsymbol{v}_k, \boldsymbol{v}_k\rangle,\\ \text{s.t., } &\langle \hat{g}_1(\boldsymbol{x}), \boldsymbol{v}_k\rangle \leq \hat{h}_1(\boldsymbol{x}),\\ &\langle \hat{g}_2(\boldsymbol{x}), \boldsymbol{v}_k\rangle \leq \hat{h}_2(\boldsymbol{x}). \end{aligned} \tag{11}$$

Finally, we have that the optimal solution of (7) is given by

$$\boldsymbol{u}_k = \boldsymbol{v}^*_k + \hat{\boldsymbol{u}}_k. \tag{12}$$

Let $G(\boldsymbol{x}) = [G_{ij}(\boldsymbol{x})] = [\langle \hat{g}_i(\boldsymbol{x}), \hat{g}_j(\boldsymbol{x})\rangle], i,j = 1,2$ is the Gram matrix. Following (Luenberger, 1997) [Ch. 3], the unique solution $\boldsymbol{v}^*_k$ to (11) is given by

$$\boldsymbol{v}^*_k = \lambda_1(\boldsymbol{x})\hat{g}_1(\boldsymbol{x}) + \lambda_2(\boldsymbol{x})\hat{g}_2(\boldsymbol{x}) \tag{13}$$

where the two gate functions $\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x})$ are given by:

$$\lambda_1(\boldsymbol{x}) = \begin{cases} 0 & \text{if } G_{21}(\boldsymbol{x})\max(\hat{h}_2(\boldsymbol{x}),0) - G_{22}(\boldsymbol{x})\hat{h}_1(\boldsymbol{x}) < 0\\ \frac{\max(\hat{h}_1(\boldsymbol{x}),0)}{G_{11}(\boldsymbol{x})} & \text{if } G_{12}(\boldsymbol{x})\max(\hat{h}_1(\boldsymbol{x}),0) - G_{11}(\boldsymbol{x})\hat{h}_2(\boldsymbol{x}) < 0\\ \frac{\max(G_{22}(\boldsymbol{x})\hat{h}_1(\boldsymbol{x}) - G_{21}(\boldsymbol{x})\hat{h}_2(\boldsymbol{x}),0)}{G_{11}(\boldsymbol{x})G_{22}(\boldsymbol{x}) - G_{12}(\boldsymbol{x})G_{21}(\boldsymbol{x})} & \text{otherwise .} \end{cases} \tag{14}$$

$$\lambda_2(\boldsymbol{x}) = \begin{cases} \frac{\max(\hat{h}_2(\boldsymbol{x}),0)}{G_{22}(\boldsymbol{x})} & \text{if } G_{21}(\boldsymbol{x})\max(\hat{h}_2(\boldsymbol{x}),0) - G_{22}(\boldsymbol{x})\hat{h}_1(\boldsymbol{x}) < 0\\ 0 & \text{if } G_{12}(\boldsymbol{x})\max(\hat{h}_1(\boldsymbol{x}),0) - G_{11}(\boldsymbol{x})\hat{h}_2(\boldsymbol{x}) < 0\\ \frac{\max(G_{11}(\boldsymbol{x})\hat{h}_2(\boldsymbol{x}) - G_{12}(\boldsymbol{x})\hat{h}_1(\boldsymbol{x}),0)}{G_{11}(\boldsymbol{x})G_{22}(\boldsymbol{x}) - G_{12}(\boldsymbol{x})G_{21}(\boldsymbol{x})} & \text{otherwise .} \end{cases} \tag{15}$$

B. Proof of Theorems

Theorem 3.1. (**Safety of ABNets**) Given the multi-head ABNet formulated as in (4) and all other safe learning models (BarrierNet, dMPC, etc.). If the system is initially safe (i.e., $b_j(\boldsymbol{x}(t_0)) \geq 0, \forall j \in S$), then a control policy $\boldsymbol{u}$ from the ABNet output (5) guarantees the safety of system, i.e., $b_j(\boldsymbol{x}(t)) \geq 0, \forall j \in S, \forall t \geq t_0$.

Proof: The proof outline is to first show the existence of new HOCBF constraints (corresponding to all the safety specifications) that are defined over the output of the ABNet. Then, we can use Nagumo's theorem (Nagumo, 1942) to recursively show the forward invariance of each safety set in the HOCBFs, and this can eventually imply the satisfaction of the safety specifications $b_j(\boldsymbol{x}) \geq 0, \forall j \in S$.

First, we show how we may ensure the safety of ABNet when there are other safe learning models, such as BarrierNet, dMPC, etc. Given a safe learning model, we have that $b_j(\boldsymbol{x}) \geq 0, \forall j \in S$. By the adaptive CBF theorem (Xiao et al., 2021), we have that the satisfaction of the adaptive CBF constraint is a necessary and sufficient condition for the safety of the system. In other words, $b_j(\boldsymbol{x}) \geq 0, \forall j \in S$ implies that there exists an adaptive CBF:

$$L_f \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p) + [L_g \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)]\boldsymbol{u}_k + p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{j,m}(\psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)) \geq 0, j \in S, \tag{16}$$

where $p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k}) > 0$ is the penalty (adaptive) function, and $\psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)$ is defined as in (4).

Next, we consider the explicit-Barrier model. As shown in Appendix sec. A, the explicit-barrier (4) is the exact solution of the QP (7). The solution of the QP (7) further implies the satisfaction of $b_I(\boldsymbol{x}) \geq 0, b_{II}(\boldsymbol{x}) \geq 0$ by the HOCBF theory (Xiao & Belta, 2021), which is equivalent to have that $b_j(\boldsymbol{x}) \geq 0, \forall j \in S$ (shown right before (4)). Again, by the adaptive CBF theorem, we have that there exist adaptive CBFs as in the form of (16).

Finally, we only need to consider the case of fusing controllers $\boldsymbol{u}_k, k \in \{1, \ldots, h\}$ that satisfy the following:

$$L_f \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p) + [L_g \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)]\boldsymbol{u}_k + p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{j,m}(\psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)) \geq 0, j \in S, \tag{17}$$

Multiplying the weight $w_k \geq 0$ to the last equation, we have

$$w_k L_f \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p) + w_k [L_g \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)]\boldsymbol{u}_k + w_k p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{j,m}(\psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)) \geq 0, j \in S, \tag{18}$$

Taking a summation of the last equation over all $k \in \{1, \ldots, h\}$, the following equation establishes:

$$\begin{aligned} &\sum_{k=1}^{h} w_k L_f \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p) + \sum_{k=1}^{h} w_k [L_g \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)]\boldsymbol{u}_k \\ &+ \sum_{k=1}^{h} w_k p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{j,m}(\psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)) \geq 0, j \in S, \end{aligned} \tag{19}$$

Since $L_g \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p$ is a vector that is independent of k and $\sum_{k=1}^{h} w_k = 1$, the last equation can be rewritten as:

$$\begin{aligned} &L_f \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p) + L_g \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p) \left(\sum_{k=1}^{h} w_k \boldsymbol{u}_k \right) \\ &+ \sum_{k=1}^{h} w_k p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{j,m}(\psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)) \geq 0, j \in S, \end{aligned} \tag{20}$$

The summation of class $\mathcal{K}$ functions is also a class $\mathcal{K}$ function. Since $\alpha_{j,m}$ are class $\mathcal{K}$ functions, the $\sum_{k=1}^{h} w_k p_{m,k}(\boldsymbol{z}_k|\theta^m_{p,k})\alpha_{j,m}(\psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p))$ is also a class $\mathcal{K}$ function over $\psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p)$. Therefore, equations (20) are the **new HOCBF constraints** defined over the output of the ABNet, i.e., $\sum_{k=1}^{h} w_k \boldsymbol{u}_k$. In other words, whenever $\psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p) = 0$, we have

$$L_f \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p) + L_g \psi_{j,m-1}(\boldsymbol{x}, \boldsymbol{z}|\theta_p) \left(\sum_{k=1}^{h} w_k \boldsymbol{u}_k \right) \geq 0, j \in S, \tag{21}$$

The controls (outputs of the ABNet) $\sum_{k=1}^{h} w_k \boldsymbol{u}_k \equiv \boldsymbol{u}$ are directly used to drive the system, and $\boldsymbol{z}$ is taken as a piece-wise constant within discretized time intervals (Xiao et al., 2023). Therefore, the last equation can be rewritten as

$$\frac{\partial \psi_{j,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)}{\partial \boldsymbol{x}}(f(\boldsymbol{x})+g(\boldsymbol{x})\boldsymbol{u}) = \frac{\partial \psi_{j,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p)}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \dot{\psi}_{j,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p) \geq 0, j \in S, \tag{22}$$

Since $b_j(\boldsymbol{x}(t_0)) \geq 0$, we can always initialize the HOCBF definition such that $\dot{\psi}_{j,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p) \geq 0$ is satisfied at t_0 (Xiao & Belta, 2021). By Nagumo's theorem (Nagumo, 1942) and (20)-(22), we have that $\psi_{j,m-1}(\boldsymbol{x},\boldsymbol{z}|\theta_p) \geq 0, \forall t \geq t_0$.

Recursively, we can show that $\psi_{j,i}(\boldsymbol{x},\boldsymbol{z}|\theta_p) \geq 0, \forall t \geq t_0, \forall i \in \{0,\ldots,m-1\}$ from $i = m-1$ to $i = 0$. Since $b_j(\boldsymbol{x}) = \psi_{j,0}(\boldsymbol{x},\boldsymbol{z}|\theta_p)$ by (3), we have that $b_j(\boldsymbol{x}(t)) \geq 0, \forall t \geq t_0, \forall j \in S$, which the safety guarantees of the ABNet for the system. ■

Theorem 3.2. (**Safety of merging of ABNets**) Given two ABNets, the merged model using the form as in (5) again guarantees the safety of system.

Proof: The proof outline is similar to that of Theorem 3.1. From each ABNet, we can show the existence of new HOCBF constraints (corresponding to all the safety specifications) that are defined over the output of each ABNet. Then we can again show the existence of another set of new HOCBF constraints (corresponding to all the safety specifications) that are defined over the output of the merged ABNet. Finally, we can also use Nagumo's theorem (Nagumo, 1942) to recursively show the forward invariance of each safety set in the HOCBFs, and this can eventually imply the satisfaction of the safety specifications $b_j(\boldsymbol{x}) \geq 0, \forall j \in S$.

The mathematical proof is similar to that of Theorem 3.1, and thus is omitted. ■

C. Experiment Details

Metrics used in all the tables. The SAFETY metric is defined as:

$$\text{SAFETY} = \min_k \{ \min_{t\in[t_0,T]} b(\boldsymbol{x}(t))\}_k, k \in \{1,\ldots,N\}, \tag{23}$$

where N is the number of testing runs ($N = 100$ in this case). T is the final time of each run. $b(\boldsymbol{x}) \geq 0$ is the safety constraint that is given explicitly in each experiment below.

The CONSER. metric is defined as

$$\begin{aligned} \text{CONSER. mean} &= \operatorname{mean}_k \{ \min_{t\in[t_0,T]} b(\boldsymbol{x}(t))\}_k, k \in \{1,\ldots,N\}, \\ \text{CONSER. std} &= \operatorname{std}_k \{ \min_{t\in[t_0,T]} b(\boldsymbol{x}(t))\}_k, k \in \{1,\ldots,N\}. \end{aligned} \tag{24}$$

The UNCERTAINTY metric for both controls are calculated by:

$$u_i \text{ UNCERTAINTY} = \operatorname*{mean}_{t\in[t_0,T]} \{ \operatorname*{std}_k \{u_i(t)\}_k, k \in \{1,\ldots,N\}\}, i \in \{1,2\}. \tag{25}$$

All the class $\mathscr{K}$ functions in the BarrierNets/ABNets are implemented as linear functions with trainable slopes.

C.1. Training Stability and Efficiency

The dQP (Amos & Kolter, 2017) could give very bad solutions (although it still satisfies the safety constraints), as shown in Fig. 7 (right), this could significantly deteriorate the training quality of the model.

C.2. 2D Robot Obstacle Avoidance

Models. All the models include fully connected layers of shape [5, 128, 32, 32, 2] with RELU as activation functions. There are some additional layers of differentiable QPs in other models (other than E2E-related models). The model input is the system state and the goal.

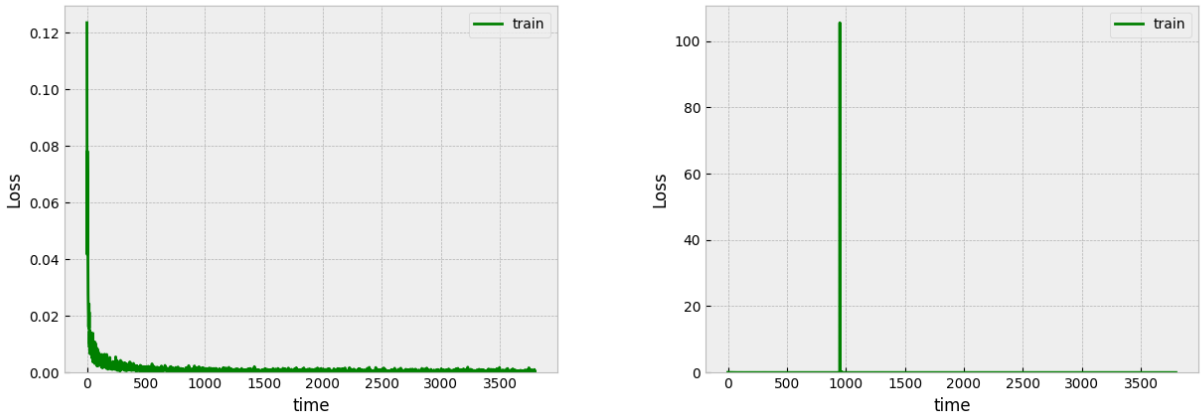

Figure 7: Comparison of ABNet (left) and BarrierNet (right) (based on dQP) in training stability. BarrierNet tends to give very bad solutions. "time" in the x-axis denotes training iterations.

Training and Dataset. The dataset includes 100 trajectories, and each trajectory has 137 trajectory points. The ground truth controls (i.e., training labels) are obtained via solving HOCBF-based QPs (Xiao & Belta, 2021). We use *Adam* as the optimizer to train the model with a MSE loss function and a learning rate 0.001. We use the *QPFunction* from the OptNet (Amos & Kolter, 2017) to solve the dQPs. The training time of the ABNet is about 1 hour for 20 epochs on a RTX-3090 computer.

Robot dynamics and safety constraints. We employ the bicycle model as the robot dynamics:

$$\underbrace{\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \\ \dot{v}(t) \end{bmatrix}}_{\dot{\boldsymbol{x}}(t)} = \underbrace{\begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \\ 0 \\ 0 \end{bmatrix}}_{f(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{g(\boldsymbol{x})} \underbrace{\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}}_{\boldsymbol{u}} \tag{26}$$

where $(x, y) \in \mathbb{R}^2$ denotes the 2D location of the robot, $\theta \in \mathbb{R}$ is the heading angle of the robot, $v \in \mathbb{R}$ is the linear speed of the robot. u_1, u_2 are the angular speed and acceleration controls, respectively.

The safety constraint of the robot is defined as:

$$b(\boldsymbol{x}) = (x - x_0)^2 + (y - y_0)^2 - R^2 \geq 0, \tag{27}$$

where $(x_0, y_0) \in \mathbb{R}^2$ is the 2D location of the obstacle, and $R > 0$ is its size.

Acceleration control profiles. We show the acceleration control profiles in Fig. 8. The corresponding uncertainty is also significantly decreased with the proposed ABNet.

C.3. Safe Robot Manipulation

Models. All the models include fully connected layers of shape [6, 128, 256, 128, 128, 32, 32, 2] with RELU as activation functions. There are some additional layers of differentiable QPs in other models (other than E2E-related models). The model input is the system state and the goal.

Training and Dataset. The dataset includes 1000 trajectories, and each trajectory has about 350 trajectory points. The ground truth controls (i.e., training labels) are obtained via solving HOCBF-based QPs (Xiao & Belta, 2021). We use *Adam* as the optimizer to train the model with a MSE loss function and a learning rate 0.001. We use the *QPFunction* from the OptNet (Amos & Kolter, 2017) to solve the dQPs. The training time of the ABNet is about 2 hours for 10 epochs on a RTX-3090 computer.

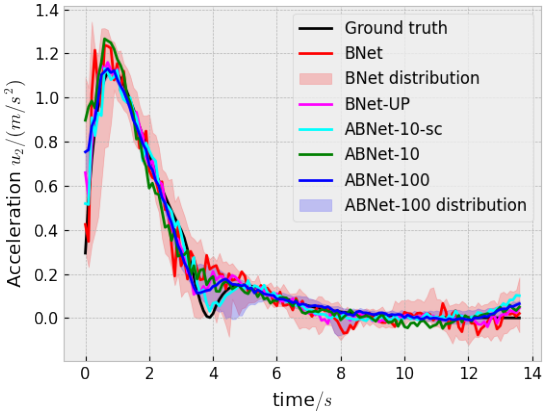

Figure 8: 2D robot obstacle avoidance acceleration control profiles and their distributions. The controls are subject to input noise, and thus are non-smooth. All the testings are done in a closed-loop fashion, i.e., the model outputs are directly used to control the robot.

Robot dynamics and safety constraints. We employ the following model as the manipulator dynamics:

$$\underbrace{\begin{bmatrix} \dot{\theta}_1 \\ \dot{\omega}_1 \\ \dot{\theta}_2 \\ \dot{\omega}_2 \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \omega_1 \\ 0 \\ \omega_2 \\ 0 \end{bmatrix}}_{f(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{g(\boldsymbol{x})} \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\boldsymbol{u}} \tag{28}$$

where $(\theta_1, \theta_2) \in \mathbb{R}^2$ denotes the angles of the two-link manipulator joints (defined in the Cartesian space, we may get the joint space angles $q_1 = \theta_1, q_2 = \theta_2 - \theta_1$), $(\omega_1, \omega_2) \in \mathbb{R}^2$ is the angular speed of the two-link manipulator joints, u_1, u_2 are the angular acceleration controls corresponding to the two joints, respectively.

The safety constraint of the robot is defined as:

$$b(\boldsymbol{x}) = (l_1 \cos\theta_1 + l_2 \cos\theta_2 - x_0)^2 + (l_1 \sin\theta_1 + l_2 \sin\theta_2 - y_0)^2 - R^2 \geq 0, \tag{29}$$

where $(x_0, y_0) \in \mathbb{R}^2$ is the location of the obstacle, and $R > 0$ is its size. $l_1 > 0, l_2 > 0$ are the length of the two links of the manipulator, respectively. In the current setting, the non-collision of the end-effector implies the non-collision of the link. Therefore, we only need to consider the safety of the end-effector. We show both the u_1, u_2 control profiles in Fig. 9 to demonstrate the advantage of the proposed ABNet. The metric definitions are the same as in the 2D robot obstacle avoidance, and the number of testing runs is $N = 100$.

C.4. Vision-based End-to-End Autonomous Driving

Models. All the models include CNN ([[3, 24, 5, 2, 2], [24, 36, 5, 2, 2], [36, 48, 3, 2, 1], [48, 64, 3, 1, 1], [64, 64, 3, 1, 1]]) and LSTM layers (size: 64) and some fully connected layers of shape [32, 32, 2] $\times 2$ with RELU as activation functions. The dropout rates for both CNN and fully connected layers are 0.3. There are some additional layers of differentiable QPs in other models (other than E2E-related models). The model input is the front-view RGB images (shape: $3 \times 45 \times 155$) of the ego vehicle, and the outputs are the steering rate and acceleration controls of the vehicle.

Training and Dataset. The dataset is open-sourced including 0.4 million image-control pairs from a closed-road sim-to-real driving field. Static and parked cars of different types and colors are used as obstacles in the dataset. The dataset is collected from the VISTA simulator (Amini et al., 2022). The ground truth controls (i.e., training labels) are obtained via solving a nonlinear model predictive control (NMPC). We use *Adam* as the optimizer to train the model with a MSE loss function and a learning rate 0.001. We use the *QPFunction* from the OptNet (Amos & Kolter, 2017) to solve the dQPs. The training time of the ABNet is about 15 hours for 5 epochs on a RTX-3090 computer.

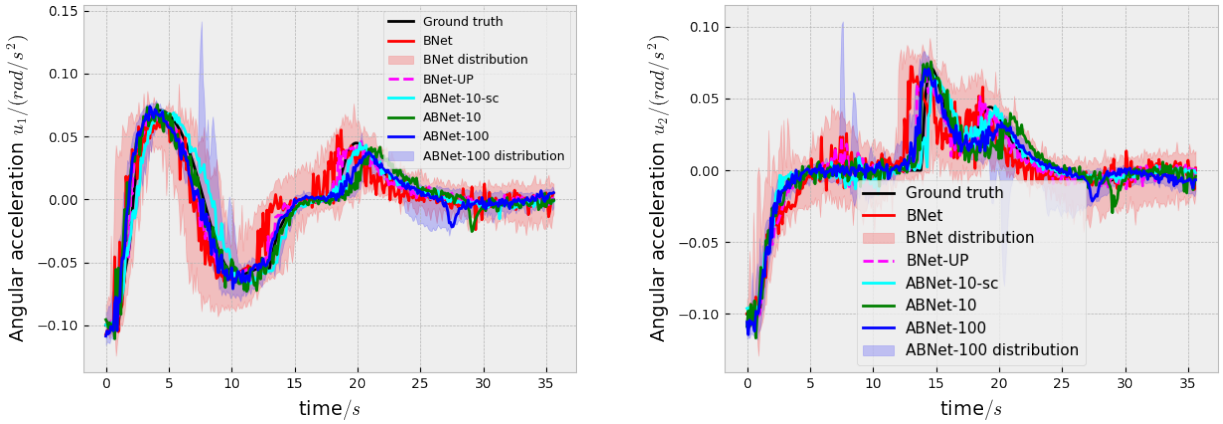

Figure 9: Robot manipulation joint control profiles and their distributions. The controls are subject to input noise, and thus are non-smooth. All the testings are done in a closed-loop fashion, i.e., the model outputs are directly used to control the manipulator.

Brief introduction to VISTA. VISTA is a sim-to-real driving simulator that can generate driving scenarios from real driving data (Amini et al., 2022). The VISTA allows us to train our model with guided policy learning. This learning method has been shown to work for model transfer to a full-scale real autonomous vehicle. There three steps to generate the data: (i) In VISTA, we randomly initialize the locations and poses of ego- and ado-cars that are associated with the real driving data; (ii) we use NMPC to collect ground-truth controls (training labels) with corresponding states, and (iii) we collect front-view RGB images along the trajectories generated from NMPC.

Vehicle dynamics and safety constraints. The vehicle dynamics are specified with respect to a reference trajectory (Rucco et al., 2015), such as the lane center line. The two most important states are the along-trajectory progress $s \in \mathbb{R}$ and the lateral offset distance $d \in \mathbb{R}$ of the vehicle center with respect to the trajectory. The dynamics are defined as:

$$\underbrace{\begin{bmatrix} \dot{s} \\ \dot{d} \\ \dot{\mu} \\ \dot{v} \\ \dot{\delta} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \frac{v\cos(\mu+\beta)}{1-d\kappa} \\ v\sin(\mu+\beta) \\ \frac{v}{l_r}\sin\beta - \kappa\frac{v\cos(\mu+\beta)}{1-d\kappa} \\ 0 \\ 0 \end{bmatrix}}_{f(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{g(\boldsymbol{x})} \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\boldsymbol{u}}, \tag{30}$$

where μ is the local heading error of the vehicle with respect to the reference trajectory, v is the linear speed of the vehicle, κ is the curvature of the trajectory at the progess s. l_r is the length of the vehicle from the tail to the center, $\beta = \arctan\left(\frac{l_r}{l_r+l_f}\tan\delta\right)$, where l_f is the length of the vehicle from the head to the center. u_1, u_2 are the steering rate and acceleration controls of the vehicle, respectively.

The safety constraint of the vehicle is defined as:

$$b(\boldsymbol{x}) = (s-s_0)^2 + (d-d_0)^2 - R^2 \geq 0, \tag{31}$$

where $(s_0, d_0) \in \mathbb{R}^2$ is the location of the obstacle in the curvi-linear frame (i.e., defined with respect to the reference trajectory), and $R > 0$ defines its size that is chosen such that the satisfaction of the above constraint can make the ego vehicle avoid crashing onto the obstacle.

Closed-loop testing. We test all of our models in a closed-loop manner in VISTA. In other words, at each time step, we get the front-view RGB image observation from VISTA. Then, the model generates a control based on the image. Finally, the control is used to drive the "virtual" vehicle in VISTA. This process is done recursively until the final time. The total number of testing runs is $N = 100$ for all the tables. The obstacles are randomly initialized (in uniform probability distribution) with lateral distance d_0 ranges from $\pm 0.1m$ to $\pm 1.5m$. In Figs. 6 and 11, the ego vehicle is randomly initialized with $d \in [-0.5, 0.5]m$ (in uniform probability distribution).

Figure 10: Attention-based image observations for the ABNet-att model. From left to right and top to down: attentions on full image, left-most part, left lane boundary, lane center, right lane boundary, and right-most part.

Image observations for the ABNet-att model. We generate the attention-based observations as shown in Fig. 10. Each of the attention images may play an important role in a specific driving scenario (e.g., attention on the left-most part may be crucial for sharp-left turn).

Acceleration control profiles. We present both the acceleration control and steering rate control profiles in Fig. 11. Both the BNet and BNet-UP models have forced the ego vehicle to have a large deceleration instead of making it to pass the obstacle using the steering control when the vehicle approaches the obstacle. This can make the ego vehicle get stuck at the obstacles, and thus, the obstacle passing rate (as shown in Table 3) is low in these two models.

Ablation studies on the model robustness in terms of safety under noisy input. To further test the model safety robustness, we add random noise (50% magnitude of the image values) to all the image observations. The results are presented in Table 4. Our proposed ABNets can still guarantee the safety of the vehicle under noisy input (0% crash rate), while the crash rates using other models significantly increase except the DFB model. This is because the HOCBFs in the DFB model are not trainable, and the corresponding parameters are fixed. Badly trained HOCBFs could make the method fail to guarantee safety due to the inter-sampling effect.

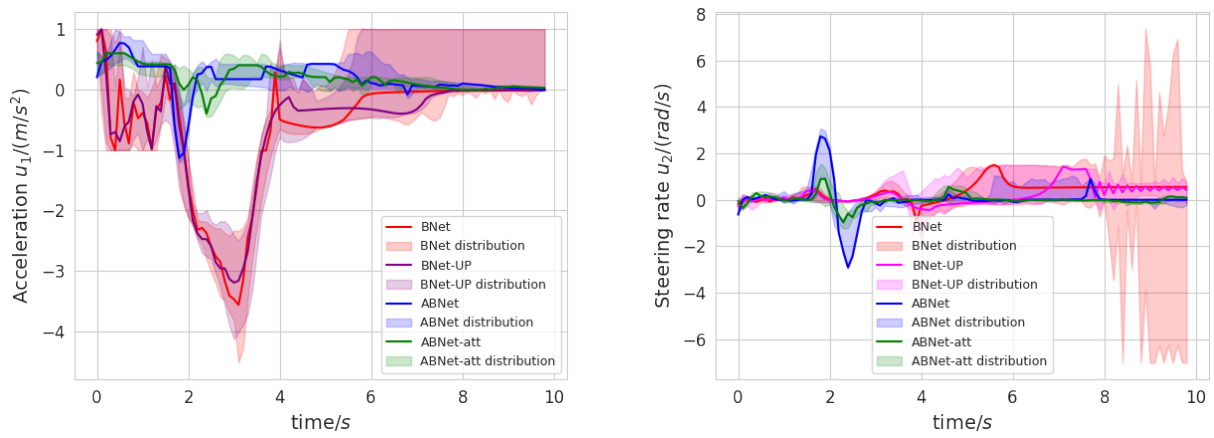

Figure 11: Vision-based end-to-end autonomous driving closed-loop testing control profiles. The models directly take images as inputs, and output controls for the vehicle. All the testings are done in closed-loop in VISTA.

Table 4: Ablation study: vision-based end-to-end autonomous driving closed-loop testing **under noise** and comparisons with benchmarks. Items in the first row are short for obstacle crash rate (CRASH), Obstacle passing rate (PASS), satisfaction of safety constraints where non-negative values mean safety guarantees (SAFETY), system conservativeness (CONSER.), acceleration control u_1 uncertainty (u_1 UNCERTAINTY), steering rate control u_2 uncertainty (u_2 UNCERTAINTY), and theoretical safety guarantees (THEORET. GUAR.) respectively. In the model column, items are short for single vanilla end-to-end driving model (V-E2E), E2Es merged with Monte-Carlo Dropout (E2Es-MCD), E2Es merged with deep resembles (E2Es-MERG), deep forward and backward model (DFB), single BarrierNet (BNET), BarrierNet policies with uncertainty propagation (BNET-UP), ABNet with 10 heads (ABNET), ABNet with attention images and 10 heads (ABNET-ATT), ABNET-SC denotes our ABNet first trained with ABNET-ATT scaled by ABNET (20 heads)respectively. The safety metric is defined as the **minimum** value of the safety specification $b_j(\boldsymbol{x}), j \in S$ among all runs. The conservativeness metric is defined as the **mean** (with std) of the minimum value (in each run) of the safety specification $b_j(\boldsymbol{x}), j \in S$ among all runs. The uncertainty metrics for both u_1 and u_2 are measured by the standard deviations of the model outputs (two controls) among all runs.

MODEL	CRASH (↓)	PASS (↑)	SAFETY (≥ 0)	CONSER. (≥ 0& ↓)	u_1 UNCERTAINTY (↓)	u_2 UNCERTAINTY (↓)	THEORET. GUAR.
V-E2E (AMINI ET AL., 2022)	31%	69%	-59.455	−8.932±19.741	0.529	0.239	×
E2Es-MCD (GAL & GHAHRAMANI, 2016)	28%	72%	-58.405	−8.116±20.802	0.524	0.232	×
E2Es-DR (LAKSHMINARAYANAN ET AL., 2017)	27%	73%	-60.267	−8.781±20.910	0.512	0.225	×
DFB (PEREIRA ET AL., 2020)	1%	37%	-13.281	−0.256±4.348	0.482	0.127	√
BNET (XIAO ET AL., 2023)	23%	37%	-45.415	−9.114±13.382	0.730	0.316	√
BNET-UP (WANG ET AL., 2023B)	24%	39%	-44.634	−8.866±13.167	0.747	0.278	×
ABNET (OURS)	0%	100%	4.268	8.315±2.147	0.151	0.326	√
ABNET-ATT (OURS)	0%	100%	5.986	7.032±0.405	0.118	0.213	√
ABNET-SC (OURS)	0%	100%	4.118	7.515±1.120	0.128	0.255	√