Fail-Safe Adversarial Generative Imitation Learning

Anonymous authors
Paper under double-blind review

Abstract

For flexible yet safe imitation learning (IL), we propose theory and a modular method, with a safety layer that enables a closed-form density/gradient of the overall safe generative policy, end-to-end training using generative adversarial IL (GAIL), and worst-case safety guarantees. The safety layer maps unsafe actions into a set of safe actions, and uses the change-of-variables formula plus additivity of measures for the density. The set of safe actions is inferred by first checking safety of a finite sample of actions via adversarial reachability analysis of fallback maneuvers, and then concluding on the safety of these actions' infinite "neighborhoods" using, e.g., Lipschitz continuity. We provide theoretical analysis showing the robustness advantage of using the safety layer already during training (imitation error linear in the horizon) compared to only using it at test time (quadratic error). In an experiment on real-world driver interaction data, we empirically demonstrate tractability, safety and imitation performance of our approach.

1 Introduction & Related Work

For several problems at the current forefront of agent learning, such as decision making of self-driving vehicles or robots, or modeling/simulation of realistic agents, imitation learning (IL) is gaining momentum as a method, or as one part in methods (Suo et al., 2021; Igl et al., 2022; Bhattacharyya et al., 2019; Bansal et al., 2018; Xu et al., 2020; Zeng et al., 2019; Cao et al., 2020). But *safety* and *robustness* of IL for such tasks remain a challenge that is surprisingly little explored.

(Generative) IL The basic idea of imitation learning is as follows: we are given recordings of the sequential behavior of some *demonstrator* agent, and then we train the *imitator* agent on this data to make it behave similarly to the demonstrator (Osa et al., 2018). One powerful recent IL method is *generative adversarial imitation learning (GAIL)* (Ho and Ermon, 2016; Song et al., 2018). GAIL's idea is, on the one hand, to use policy gradient methods borrowed from reinforcement learning (RL; including, implicitly, a form of *planning*) to generate an imitator behavior that matches the demonstrator's over whole trajectories. And on the other hand, to measure the "matching", GAIL uses a *discriminator* as in *generative adversarial networks (GANs)* that seeks to distinguish between demonstrator and imitator distributions. While the neural net and GAN aspects help the flexibility of the method, the RL aspects stir against compounding of errors over rollout horizons, which other IL methods like *behavior cloning (BC)*, which do not have such planning aspects, suffer from. One recent line of research generalizes classic Gaussian policies to more flexible (incl. multi-modal) conditional *normalizing flows* as imitator policies (Ward et al., 2019; Ma et al., 2020) which nonetheless give exact densities/gradients for training via the *change-of-variables formula*.

IL's safety/robustness challenges While these are substantial advances in terms of *learning flexibility/capacity*, it remains a challenge to make the IL methods (1) guaranteeably *safe*, especially for multi-agent interactions with humans, and (2) *robust*, in the sense of certain generalizations "outside the training data", e.g., longer simulations than training trajectories. Both is non-trivial, since safety and long-term behavior can heavily be affected already by small learning errors (similar as the *compounding error* of BC).

Safe control/RL frameworks we build on Several ideas from safe control/RL are relevant for us: (1) In reachability analysis/games (Fridovich-Keil and Tomlin, 2020; Bansal et al., 2017), the safety prob-

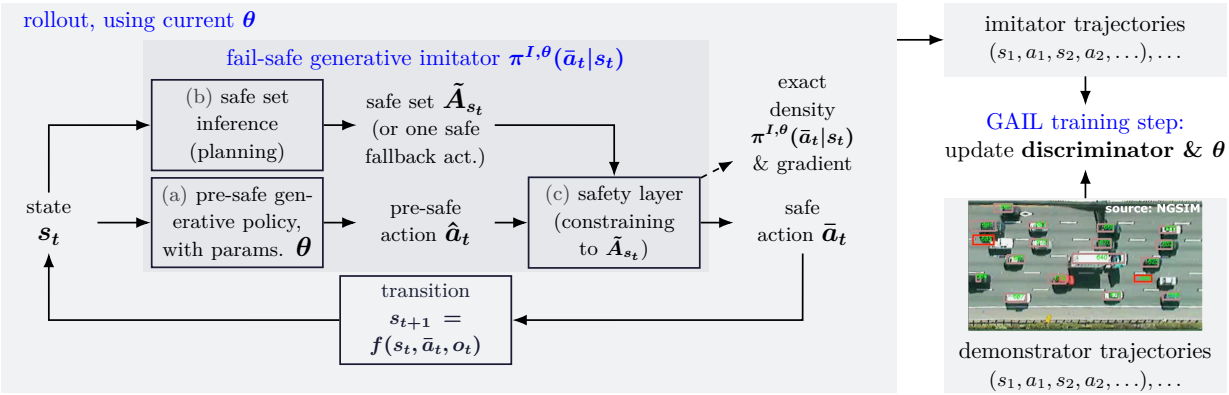


Figure 1: Our theoretically grounded method *FAGIL*: It consists of the *fail-safe generative imitator policy* (inner box on l.h.s.), that takes current state s_t as input, and then it (a) lets a standard pre-safe generative policy (e.g., Gauss or normalizing flow) generate a pre-safe action $\hat{a}_t$, (b) *infers a provably safe set* $\tilde{A}_{s_t}$ of actions, and then (c) uses our *safety layer* to constrain the pre-safe action $\hat{a}_t$ into the safe set $\tilde{A}_{s_t}$, yielding safe action $\bar{a}_t$. It has *closed-form density and gradient*, so we train it end-to-end via GAIL (r.h.s.). While our method is for general safe IL tasks, we take driver imitation as running example/experiments.

lem is formulated as finding controllers that provably keep a dynamical system within a pre-defined set of safe/collision-free states over some finite or infinite horizon, possibly under adversarial perturbations/other agents. (2) Any learning-based policy can be made safe by, at each stage, if it *fails* to produce a safe action, *enforcing* a safe action given by some sub-optimal but safe fallback controller (given there is such a controller) (Wabersich and Zeilinger, 2018). (3) Often it is enough to search over a reasonably small set of candidates of fallback controllers, such as emergency brakes and simple evasive maneuvers in autonomous driving (Pek and Althoff, 2020). And (4) a simple way to enforce a policy's output to be safe is by composing it with a *safety layer* that maps into the safe set, i.e., constrains the action to the safe set, and *differentiable* such safety layers can even be used during *training* (Donti et al., 2021a; Dalal et al., 2018).

Safe IL, theoretical guarantees, and their limitations While incorporating *safety into RL* has received significant attention in recent years (Wabersich and Zeilinger, 2018; Berkenkamp et al., 2017; Alshiekh et al., 2018; Chow et al., 2019; Thananjeyan et al., 2021), *safety in IL*, and its *theoretical analysis*, have received comparably little attention. To have a rough classification of the existing safe IL work, a first distinction can be made in terms of safety via *reward augmentation* versus *safety layers/constraints within the policy*. *In reward augmentation*, loss terms are added to the imitation loss that penalizes undesired state-action pairs (e.g., collisions) (Bhattacharyya et al., 2019; 2020; Bansal et al., 2018; Suo et al., 2021; Zeng et al., 2020). Then there are *hard-constrained policies* (Tu et al., 2022; Havens and Hu, 2021; Cheng et al., 2022) , and, in particular, one line of research that is deep IL with safety layers, but *only during test time* (Chen et al., 2019). There is little work that uses safety layers during *training and test time*, with the exception of (Yin et al., 2021) which gives guarantees but is not generative. More broadly related are methods that control the probabilistic risk of high imitation costs (Javed et al., 2021; Brown et al., 2020; Lacotte et al., 2019); but they do not consider safety constraints *separate* from the imitation costs, as we do.

Main contributions and paper structure In this sense we are not aware of general approaches for the problem of safe generative IL that are (1) end-to-end trainable *and* (2) safe/robust with theoretical guarantees. In this paper, we contribute theory (Sec. 3), method (Fig. 1, Sec. 4) and experiments (Sec. 5) towards understanding and addressing these crucial gaps:[1][2]

[1]Note that our approach works with both, demonstrators that are already fully safe, and those that may be unsafe.

[2]Regarding the task of realistic modeling/simulation: Adding safety constraints can, in principle, lead to unrealistic biases in case of unsafe demonstrations. But the other extreme is pure IL where, e.g., collisions rates can be unrealistically high Bansal et al. (2018). So this is always a trade-off.

• We propose a simple yet flexible type of differentiable *safety layer* that maps a given "pre-safe" action into a given safe action set (i.e., constrains the action). It allows to have a *closed-form density/gradient* (Prop. 3; by a non-injective "piecewise" change of variables), in contrast to common existing differentiable safety layers (Donti et al., 2021a; Dalal et al., 2018).

• We contribute two types of *sample-based safe set inference approaches*, which, for a given state, output a provably safe set of actions – in spite of just checking safety of a *finite* sample of actions, using Lipschitz continuity/convexity (Prop. 1 and 2). This advances sample-based approaches Gillula et al. (2014) to overcome limitations of exact but restrictive, e.g., linear dynamics, approaches (Rungger and Tabuada, 2017).

• A general question is to what extent it helps to use the *safety layer already during training (as we do)*, compared to just concatenating it, at *test* time, to an unsafely trained policy, given the latter may computationally be much easier. We theoretically quantify the imitation performance advantage of the former over the latter: essentially, the former imitation error scales *linearly* in the rollout horizon, the latter *quadratically* (Sec. 3.3, rem. 1, and thm. 1) – reminiscent of BC. The intuition: only the former method learns how to properly deal (plan) with the safety layer, while in the latter case, the safety layer may *lead to unvisited states at test time from which we did not learn to recover*. (Generally, proofs are in Appendix A.)

• Based on these results, we provide the method *fail-safe generative adversarial imitation learner (FAGIL;* Sec. 4). – It is sketched in Fig. 1.

• In experiments on real-world highway data with multiple interacting driver agents (which also serves as running example throughout the paper), we empirically show tractability and safety of our method, and show that its imitation/prediction performance comes close to unconstrained GAIL baselines (Sec. 5).

2 Setting & Problem Formulation

Setting and definitions

• We consider a dynamical system consisting of: *states* $s_t \in S$, at *time stages* $t \in 1{:}T := \{1, \dots, T\}$;

• an *ego agent* that takes *action* $a_t \in A \subset \mathbb{R}^n, n \in \mathbb{N}$, according to its *ego policy* $\pi_t \in \Pi_t$ (for convenience we may drop subscript t); we allow π_t to be either a conditional (Lebesgue) density, writing $a_t \sim \pi_t(a|s_t)$, or deterministic, writing $a_t = \pi_t(s_t)$;

• the ego agent can be either the *demonstrator*, denoted by π^D (a priori unknown); or the *imitator*, denoted by $\pi^{I,\theta}$, with θ its parameters;

• *other agents*, also interpretable as *noise/perturbations* (a priori unknown), with their *(joint) action* o_t according to *others' policy* $\sigma_t \in \Phi_t$ (density/deterministic, analogous to ego);

• the system's *transition function* (environment) f, which we assume to be *known*, s.t.,

$$s_{t+1} = f(s_t, a_t, o_t). \tag{1}$$

• We are given a finite set of *demonstrator trajectories* of the form $(s_1, a_1), (s_2, a_2), \dots, (s_T, a_T)$ of π^D's episodic rollouts in an instance of this dynamical system including other agents/noise, *as training data*.

• We consider a *(momentary) imitation cost function* $c(s, a)$ which formalizes the similarity between imitator and demonstrator trajectory data (details follow in Sec. 4.2 on the GAIL-based c), and *(momentary) safety cost function* $d(s)$; states s for which $d(s) \leq 0$ are referred to as *momentarily safe* or *collision-free*.

• Notation: $P(\cdot)$ denotes probability, and $p_x(\cdot)$ the density of a variable x, if it exists (we may drop subscript x).

Goal formulation The goal is for the imitator to generate trajectories that minimize the expected imitation cost, while satisfying a safety cost constraint, formally:

$$\pi^I = \arg\min_{\pi} \underbrace{\mathbb{E}_{\pi}\left(\sum_{t=1}^{T} c(s_t, a_t)\right) - \psi(\pi)}_{=:\, v^{\pi,c,\psi}}, \qquad d(s_t) \leq 0, \text{ for all } t, \tag{2}$$

where $v^{\pi,c,\psi}$ is called *total imitation cost function* (we drop superscript ψ if it is zero), $\psi(\pi)$ is a *policy regularizer*, and $\mathbb{E}_\pi(\cdot)$ denotes the expectation over roll-outs of π.[3] We purportedly stay fairly abstract here. We will later instantiate c, ψ (Sec. 4.2) and d (Sec. 5), and specify our safety-relevant modeling/reasoning about the other agents/perturbations (Sec. 3.1).

3 General Theoretical Tools for Policy and Training of Safe Generative IL

We start by providing, in this section, several general-purpose theoretical foundations for safe generative IL, as tools to help the design of the safe imitator *policy* (Sec. 3.1 and 3.2), as well as to understand the advantages of end-to-end *training* (Sec. 3.3). We will later, in Sec. 4, build our method on these foundations.

3.1 Sample-based Inference of Safe Action Sets

For the design of safe imitator policies, it is helpful to know, at each stage, a *set of safe actions*. Roughly speaking, we consider an action as safe, if at least *one safe* "best-case" future ego policy π exists, i.e., keeping safety cost $d(s_t)$ below 0 for all t, under *worst-case/adversarial* other agents/perturbations σ. To make this formal, we define the *safe (action) set* in state s at time t as

$$\bar{A}_t^s := \{a \in A : \text{ it exists } \pi_{t+1:T}, \text{ s.t. for all } \sigma_{t:T} \text{ and for all } t{<}t'{\leq}T,\ \ d(s_{t'}){\leq}0 \text{ holds, given } s_t{=}s, a_t{=}a\} \quad (3)$$

(based on our general setting of Sec. 2). We refer to actions in $\bar{A}_t^s$ as *safe actions*. Such a worst-case uncertainty reasoning resembles common formulations from reachability game analysis (Fridovich-Keil and Tomlin, 2020; Bansal et al., 2017). Furthermore, define the *total safety cost (to go)* as[4]

$$w_t(s,a) := \min_{\pi_{t+1:T}} \max_{\sigma_{t:T}} \max_{t' \in t+1:T} d(s_{t'}), \text{ for all } t. \quad (4)$$

Observe the following:

- The total safety cost allows us to quantitatively characterize the safe action set as *sub-zero set*:

$$\bar{A}_t^s = \{a : w_t(s,a) \leq 0\}.$$

Note that requiring momentary safety cost $d(s_{t'}) \leq 0$ *for each* future t' in Eq. (3) translates to requiring that $d(s_{t'}) \leq 0$ for the *maximum* over future t' in Eq. (4).

- Under the worst-case assumption, for each t, s: an ego policy π_t can be part of a candidate solution of our basic problem, i.e., satisfy safety constraint in Eq. (2), *only if π_t's outputted actions are all in $\bar{A}_t^s$* (because otherwise there exists no feasible continuation $\pi_{t+1:T}$, by definition of $\bar{A}_t^s$). This fact will be crucial to define our safety layer (Sec. 3.2).

Now, regarding *inference* of the safe action set $\bar{A}_t^s$: while in certain limited scenarios, such as linear dynamics, it may be possible to efficiently calculate it exactly, this is not the case in general (Rungger and Tabuada, 2017). We take an approach to circumvent this by checking $w_t(s,a)$ for just a *finite sample* of a's. And then concluding on the value of $w_t(s,\cdot)$ on these a's "neighborhoods". This at least gives a so-called *inner approximation $\tilde{A}_t^s$ of the safe set $\bar{A}_t^s$*, meaning that, while $\tilde{A}_t^s$ and $\bar{A}_t^s$ may not coincide, we know for sure that $\tilde{A}_t^s \subset \bar{A}_t^s$, i.e, $\tilde{A}_t^s$ is safe. The following result gives us Lipschitz continuity and constant to draw such conclusions.[5] It builds on the fact that maximization/minimization preserves Lipschitz continuity (Appendix A.1.1). We assume all spaces are implicitly equipped with norms.

[3] As usual, the arg min is well defined under appropriate compactness and continuity assumptions. The "=" may strictly be a "∈". We assume an initial (or intermediate) state as given.

[4] Note: In the scope of safety definitions, we generally let ego/other policies π, σ range over compact sets Π, Φ of deterministic policies. Then the minima/maxima are well-defined, once we make appropriate continuity assumptions. The finite horizon can be extended to the infinite case using terminal safety terms Pek and Althoff (2020).

[5] Once we know that $w_t(s,\cdot)$ is γ-Lipschitz, and that $w_t(s,a) < 0$, then we also know that $w_t(s,\cdot)$ is negative on a ball of radius $\frac{w_t(s,a)}{\gamma}$ around a.

Proposition 1 (Lipschitz constants for Lipschitz-based safe set)**.** *Assume the momentary safety cost d is α-Lipschitz continuous. Assume that for all (deterministic) ego/other policies $\pi_t \in \Pi_t, \sigma_t \in \Phi_t$, $t \in 1{:}T$, the dynamics $s \mapsto f(s, \pi_t(s), \sigma_t(s))$ as well as $a \mapsto f(s, a, \sigma_t(s))$ for fixed s are β-Lipschitz. Then $a \mapsto w_t(s, a)$ is $\alpha \max\{1, \beta^T\}$-Lipschitz.*

We also give an another approach to inner-approximate the safe set. Note that the following is just an alternative to Prop. 1, and will not be needed for our method FAGIL-L later on. Sometimes, safety of a finite set of corners/extremalities that span a set already implies safety of the full set, e.g., using convexity:[6]

Proposition 2 (Extremality-based safe set)**.** *Assume the dynamics f is linear, and the ego/other policy classes $\Pi_{1:T}, \Phi_{1:T}$ consist of open-loop policies, i.e., action trajectories, and that safety cost d is convex. Then $a \mapsto w_t(s_t, a)$ is convex. In particular, for any convex polytope $B \subset A$, $w_t(s, \cdot)$ takes its maximum at one of the (finitely many) corners of B.*

For several broader remarks on the elements of this section, which are not necessary to understand this main part though, see also Appendix D.1.

3.2 Piecewise Diffeomorphisms for Flexible Safety Layers with Differentiable Closed-Form Density

An important tool to *enforce safety* of a policy's actions are safety layers. Assume we are given $\tilde{A} \subseteq \bar{A}$ as an (inner approximation of the) safe action set (e.g., from Sec. 3.1). A *safety layer* is a neural net layer that maps $A \to \tilde{A}$, i.e., it constrains a *"pre-safe"*, i.e., potentially unsafe, action $\hat{a} \in A$ into $\tilde{A}$. In contrast to previous safety layers (Donti et al., 2021a; Dalal et al., 2018), for our generative approach, we want a safety layer that gives us an overall policy's *closed-form differentiable density* (when plugging it on top of a closed-form-density policy like a Gaussian or normalizing flow), for end-to-end training etc. In this section we introduce a new function class and density formula that helps to construct such safety layers.

Remember the classic *change-of-variables formula* (Rudin et al., 1964): If $y = e(x)$ for some diffeomorphism[7], and x has density $p_x(x)$, then the implied density of y is $|\det(J_{e^{-1}}(y))| p_x(e^{-1}(y))$.[8] The construction of safety layers with exact density/gradient *only* based on change of variables can be difficult or even impossible. One intuitive reason for this is that diffeomorphisms require *injectivity*, and therefore one cannot simply map unsafe actions to safe ones and simultaneously leave safe actions where they are.

These limitations motivate our simple yet flexible approach of using the following type of function as safety layers, which relaxes the rigid injectivity requirement that pure diffeomorphisms suffer from:

Definition 1 (Piecewise diffeomorphism safety layers)**.** *We call a function $g : A \to \tilde{A}$ a* piecewise diffeomorphism *if there exists a countable partition $(A_k)_k$ of A, and diffeomorphisms (on the interiors) $g_k : A_k \to \bar{A}_k \subset \tilde{A}$, such that $g|_{A_k} = g_k$.*

Now we can combine the diffeomorphisms' change-of-variables formulas with additivity of measures:

Proposition 3 (Closed-form density for piecewise diffeomorphism)**.** *If g is such a piecewise diffeomorphism, $\bar{a} = g(\hat{a})$ and $\hat{a}$'s density is $p_{\hat{a}}(\hat{a})$, then $\bar{a}$'s density is*

$$p_{\bar{a}}(\bar{a}) = \sum_{k : \bar{a} \in g_k(A_k)} |\det(J_{g_k^{-1}}(\hat{a}))| p_{\hat{a}}(g_k^{-1}(\bar{a})). \tag{5}$$

Specific instances of such piecewise diffeomorphism safety layer will be given in Sec. 4. For further remarks on motivation, details and illustration of piecewise diffeomorphism layers, which are not necessary to understand this main part though, we refer the interested reader to Appendix D.2.

[6] Note that Gillula et al. (2014) also inner-approximate safe sets from finite samples of the current action space, by using the *convex hull* argument. But, among other differences, they do not allow for other agents, as we do in Prop. 2.

[7] A *diffeomorphism* e is a continuously differentiable bijection whose inverse is also cont. differentiable.

[8] J_e denotes the Jacobian matrix of e.

3.3 Analysis of End-to-End Training with Safety Layers vs. Test-Time-Only Safety Layers

After providing tools for the design of safe *policies*, now we focus on the following general question regarding *training* of such policies: *What is the difference, in terms of imitation performance, between using a safety layer only during test time, compared to using it during training and test time?* (Note: this question/section is a *justification*, but is not strictly necessary for understanding the *definition* of our method in Sec. 4 below. So this section may be skipped if mainly interested in the specific method and experiments.)

The question is important for the following reason: While we in this paper advocate end-to-end training including safety layer, in principle one could also use the safety layer *only* during test/production time, by composing it with an "unsafely trained" policy. This is nonetheless safe, but much easier computationally in training. The latter is an approach that other safe IL work takes (Chen et al., 2019), but it means that the test-time policy differs from the trained one.

The analysis we give in this section is inspired by analysis of the *compounding errors* incurred by behavior cloning (BC) (Ross and Bagnell, 2010; Syed and Schapire, 2010; Xu et al., 2020) For this section, let us make the following assumptions and definitions (some familiar from the mentioned BC/IL work):[9]

- the state space S and action space A are discrete; the *time-averaged state-action distribution* is defined – as usual – as $\rho(s,a) := \left(\frac{1}{T}\sum_{t=1}^{T} p_{s_t}(s)\right)\pi(a|s)$, for all s, a; and $\rho(s)$ denotes the marginal.
- We assume some safety layer (i.e., mapping from action set into safe action set) to be given and fixed. As before, π^D is the demonstrator, π^I the *imitator trained with train-and-test-time safety layer*. Additionally define π^U as a classic, *unsafe/unconstrained trained imitator* policy[10], and π^O as the *test-time-only safety policy* obtained by concatenating π^U with the safety layer at test time. Let $\rho^D, \rho^I, \rho^U, \rho^O$ denote the corresponding state-action distributions.
- As is common (Ross and Bagnell, 2010; Xu et al., 2020), we measure performance deviations in terms of the (unknown) demonstrator cost function, which we denote by c^* (i.e., in the fashion of inverse reinforcement learning (IRL), where the demonstrator is assumed to be an optimizer of Eq. (2) with c^* as c); and assume c^* only depends on s, and $\|c^*\|_\infty$ is its maximum value. Let $v^D := v^{\pi^D, c^*}$, $v^I := v^{\pi^I, c^*}$, $v^O := v^{\pi^O, c^*}$ be the corresponding total imitation cost functions (Sec. 2).
- We assume that (1) the demonstrator always acts safely, and (2) the safety layer is the non-identity only on actions that are never taken by the demonstrator (mapping them to safe ones). Importantly, note that *if we relax these assumptions, in particular, allow the demonstrator to be unsafe, the results in this section would get even stronger* (the difference of adding a safety layer would be even bigger).

Keep in mind that what we can expect to achieve by GAIL training is a population-level closeness of imitator to demonstrator state-action distribution of the form $D(\rho^{\mathrm{I}}, \rho^{\mathrm{D}}) \leq \varepsilon$, for some ε decreasing in the sample size (Xu et al., 2020), and D, e.g., the Jensen-Shannon distance (due to the usual bias/generalization error). So we make this an assumption in the following results. Here we use $D(\rho^{\mathrm{I}}, \rho^{\mathrm{D}}) = D_{\mathrm{TV}}(\rho^{\mathrm{I}}, \rho^{\mathrm{D}}) = \frac{1}{2}\sum_{s\in S, a\in A} |\rho^{\mathrm{I}}(s,a) - \rho^{\mathrm{D}}(s,a)|$, the *total variation distance* (Klenke, 2013), which simplifies parts of the derivation, though we believe that versions for other D such as Jensen-Shannon are possible.

Now, first, observe that for our "train *and* test safety layer" approach, we can have a *linear, in the horizon* T, imitation performance guarantee. This is because the *imitator* π^I *is explicitly trained to deal with the safety layer* (i.e., to plan to avoid safe but poor states, or to at least recover from them; see Appendix A.3.1 for the derivation).

Remark 1 (Linear error in T of end-to-end train-and-test-time safety layer)**.** *Assume* $D_{TV}(\rho^I, \rho^D) \leq \varepsilon$. *Then we get*

$$|v^I - v^D| \leq 2\varepsilon T \|c^*\|_\infty.$$

[9] We believe that most of the simplifying assumptions we make can be relaxed, but the analysis will be much more involved.
[10] In our method's case this means: the pre-safe policy's, e.g., Gaussian, output $\hat{a}$ is directly taken as action a.

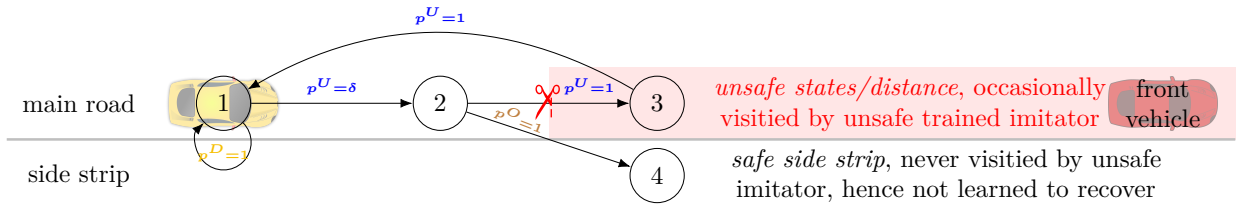


Figure 2: Staying with our running example of driver imitation, this figure illustrates the following: if we use safety layers only at test time and not already during training, the test-time imitator has *not learned to deal with the states that the safety layer may lead to* (i.e. to either plan to avoid, or recover from them), yielding *poor imitation performance*. In a nutshell: The demonstrator is always ($p^D = 1$) in state 1, meaning a constant velocity and distance to the front vehicle. The unsafely trained imitator π^U, due to learning error, with low probability $p^U{=}\delta$, deviates from state 1 and reaches first state 2 and then (due to, say, inertia) unsafe state 3, from which it has learned to always recover though. The test-safe-only imitator π^O, like the unsafely trained imitator it builds upon, with low probability $p^O{=}\delta$ reaches state 2. But from there, the *safety constraints (red scissors symbol) kick in* and the only remaining action is to go to the side strip 4 with $p^O = 1$, where there was *no data on how to recover (never visited/learned by unsafe* π^U*)*, getting trapped there forever. Details are in Appendix A.3.

In contrast, using test-time-only safety, the test-time imitator π^O *has not learned to plan with, and recover from, the states that the safety layer may lead to*, leading to an (essentially tight) *quadratic* error bound (proved in Appendix A.3):

Theorem 1 (Quadratic error in T of test-time-only safety layer)**.** ***Lower bound*** *(an "existence" statement): We can construct an environment*[11] *with variable horizon T and with a demonstrator, sketched in Fig. 2 and additional details in Appendix A.3.2, a universal constant ι, and, for every $\varepsilon > 0$, an unconstrained imitator π^U with $D_{TV}(\rho^D, \rho^U) \leq \varepsilon$, such that for the induced test-time constrained imitator π^O we have, for all $T \geq 2$*[12]*,*

$$|v^O - v^D| \geq \iota \min\{\varepsilon T^2, T\} \|c^*\|_\infty. \tag{6}$$

Upper bound *(a "for all" statement): Assume $D_{TV}(\rho^D, \rho^U) \leq \varepsilon$ and assume $\rho^U(s)$ has support wherever $\rho^D(s)$ has. Then*

$$|v^O - v^D| \leq \frac{4\varepsilon}{\nu} T^2 \|c^*\|_\infty, \tag{7}$$

where ν is the minimum mass of $\rho^D(s)$ within the support of $\rho^D(s)$.

4 Method Fail-Safe Adversarial Generative Imitation Learner

Building on the above theoretical tools and arguments for end-to-end training, we now describe our modular methodology FAGIL for safe generative IL. Before going into detail, note that, for our *general continuous* setting, tractably inferring safe sets and tractably enforcing safety constrains are both commonly known to be challenging and usually require non-trivial modeling trade-offs (Rungger and Tabuada, 2017; Gillula et al., 2014; Achiam et al., 2017; Donti et al., 2021b). Therefore, here we choose to provide (a) one tractable *specific method instance* for the *low-dimensional* case (where we can efficiently partition the action space via a grid), which we also use in the experiments, alongside (b) an abstract template for a *general approach*. We discuss policy, safety guarantees (Sec. 4.1), and training (Sec. 4.2).

[11] In this discrete settings, the environment amounts to a Markov decision process (MDP) (Sutton and Barto, 2018).

[12] The quadratic bound in Eq. (6) is substantial once we look at small ε, because then the quadratic bound holds even for large T, while the train-and-test-time safety layer in Rem. 1 keeps scaling linearly in T. See also Rem. 4 in Appendix A.3.2.

4.1 Policy with Invariant Safety Guarantee

First, our method consists of the *fail-safe (generative) imitator* $\pi^{I,\theta}(a|s)$. Its structure is depicted on the l.h.s. of Fig. 1, and contains the following modules: at each roll-out stage t, with state s_t as input,

(a) the **pre-safe generative policy module** outputs a pre-safe (i.e., potentially unsafe) action sample $\hat{a}_t \in A$ as well as explicit probability density $p_{\hat{a}|s}(\hat{a}|s_t)$ and its gradients. **Specific instantiations:** This can be, e.g., a usual Gauss policy or conditional normalizing flow.[13]

(b) The **safe set inference module** outputs an inner-approximated safe action set $\tilde{A}_t^{s_t}$, building on our results form Sec. 3.1. The rough idea is as follows (recall that an action a is safe if $w_t(s_t, a) \leq 0$): We check[14] the total safety cost $w_t(s_t, a)$ for a *finite sample* of a's, and then conclude on the value of $w_t(s_t, \cdot)$ on these a's *"neighborhoods"*, via Prop. 1 and 2.[15]

Specific instantiations: We partition the action set A into boxes (hyper-rectangles) based on a regular grid (assuming A is a box). Then we go over each box A_k and determine its safety by evaluating $w_t(s_t, a)$, with either a the center of A_k, and then using the Lipschitz continuity argument (Prop. 1) to check if $w_t(s_t, \cdot) \leq 0$ on the full A_k (*FAGIL-L*), or a ranging over all corners of box A_k, yielding safety of this box iff $w_t(s_t, \cdot) \leq 0$ on all of them (*FAGIL-E*; Prop. 2). Then, as inner-approximated safe set $\tilde{A}_t^{s_t}$, return the union of all safe boxes A_k.

(c) Then, pre-safe action $\hat{a}_t$ plus the safe set inner approximation $\tilde{A}_t^{s_t}$ are fed into the **safety layer**. The idea is to use a piecewise diffeomorphism g from Sec. 3.2 for this layer, because the "countable non-injectivity" gives us quite some flexibility for designing this layer, while nonetheless we get a closed-form density by summing over the change-of-variable formulas for all diffeomorphisms that map to the respective action (Prop. 3). As output, we get a sample of the safe action $\bar{a}_t$, as well as density $\pi^{I,\theta}(\bar{a}_t|s_t)$ (by plugging the pre-safe policy's density $p_{\hat{a}|s}(\hat{a}|s_t)$ into Prop. 3) and gradient w.r.t. θ.

Specific instantiations: We use the partition of A into boxes $(A_k)_k$ from above again. The piecewise diffemorphism g is defined as follows, which we believe introduces a helpful inductive bias: (a) On all safe boxes k, i.e., $A_k \subset \bar{A}$, g_k is the identity, i.e., we just keep the pre-safe proposal $\hat{a}$. (b) On all unsafe boxes k, g_k is the translation/scaling to the box where the pre-safe density $p_{\hat{a}}$ has the most mass (approximately). Alternatively, in (b) a probabilistic version (b') can be used where g_k translates to safe part A_ℓ with (approximate) probability proportional to $p_{\hat{a}}$ on A_ℓ (intuitively, this amounts to a conditioning on the safe set, i.e., a renormalization, when the box size goes to zero). Further safety layer instances and illustrations are in Appendix D.3.

Adding a safe fallback memory To cope with the fact that it may happen that at some stage an *individual* safe action exists, but no *fully* safe part A_k exists (since the A_k are usually sets of more than one action and not all have to be safe even though one or some are safe), we propose the following, inspired by (Pek and Althoff, 2020): starting at a presumably safe stage 1, at each stage t, the fail-safe imitator remembers *one* fail-safe fallback future ego policy $\pi^f_{t+1:T}$. Then, if no fully safe part A_k (including a new fail-safe fallback) is found anymore at $t+1$, execute the fail-safe fallback π^f_{t+1}, and keep $\pi^f_{t+2:T}$ as new fail-safe fallback (see also Appendices B and D.3.5 on potential issues with this).

Remark 2 (Invariant safety guarantee)**.** *Based on the above, if we know at least one safe action at stage* 1*, then we invariably over time have at least one safe action plus subsequent fallback policy. So we are* guaranteed *that the ego will be safe throughout the horizon* $1{:}T$*.*

4.2 Imitation Cost and Training Based on GAIL

In principle, various frameworks can be applied to specify imitation loss $c(s, a)$ and training (which we left abstract in Sec. 2) for our approach. Here, we use (Wasserstein) GAIL (Ho and Ermon, 2016; Xiao et al.,

[13] If A is box-shaped, then we add a squashing layer to get actions into A using component-wise sigmoids.

[14] Clearly, already the "checker" task of evaluating a single action's safety, Eq. (4), can be non-trivial. But often, roll-outs, analytic steps, convex optimizations or further inner approximations can be used, or combinations of them, see also Sec. 5.

[15] Due to modularity, other safety frameworks like Responsibility-Sensitive Safety (RSS) Shalev-Shwartz et al. (2017) can be plugged into our approach as well. The fundamental safety trade-off is between being safe (conservativity) versus allowing for enough freedom to be able to "move at all" (actionability).

2019) for these parts. In this section, we give a rough idea of how this works, while some more detailed background and comments for the interested reader are in Appendix B. In GAIL, roughly speaking, $c(s, a)$ is given by a GAN's *discriminator* that measures the similarity between imitator and demonstrator (average state-action distribution). As policy regularizer $\psi(\pi)$ in Eq. (2), we take π's differential entropy.

Training As illustrated on the r.h.s. of Fig. 1, analogous to usual GAIL, essentially training consists of steps alternating between: (a) doing roll-outs of our fail-safe imitator $\pi^{I,\theta}(a|s)$ under current parameter θ to sample imitator state-action trajectories (the *generator*), (b) obtain new imitator parameter θ' by optimizing an estimated Eq. (2) from these trajectory samples, using policy gradient approaches (Haarnoja et al., 2018; Sutton and Barto, 2018) for which the exact policy density formula (Prop. 3) and its gradient is crucial, (c) obtain new discriminator parameters based on optimizing an estimate of the discriminator loss, based on trajectory samples of both, imitator π^I and demonstrator π^D.

5 Experiments

The goal of the following experiments is to demonstrate tractability of our method, and empirically compare safety as well as imitation/prediction performance to baselines. *(Code will be made available upon publication.)* We pick driver imitation from real-world data as task because it is relevant for both, *control* (of vehicles), as well as *modeling* (e.g., of real "other" traffic participants, for validation of self-driving "ego" algorithms Suo et al. (2021); Igl et al. (2022)). Furthermore, its characteristics are quite representative of various challenging safe control and robust modeling tasks, in particular including fixed (road boundary) as well as many moving obstacles (other drivers), and multi-modal uncertainties in driver behavior.

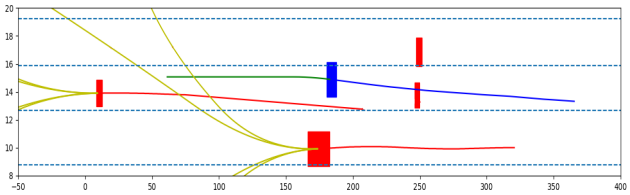


Figure 3: Driving from right to left, our fail-safe imitator (blue) with past trajectory (blue) and future fail-safe fallback emergency brake (green); other vehicles (red), with worst-case reachable boundaries indicated in yellow (rear vehicles are ignored as described).

Data set and preprocessing We use the open *highD data set* (Krajewski et al., 2018), which consists of 2-D car trajectories (each ∼20s) recorded by drones flying over highway sections (we select a straight section with ~ 1500 trajectories). It is increasingly used for benchmarking (Rudenko et al., 2019; Zhang et al., 2020). We filter out other vehicles in a roll-out once they are behind the ego vehicle. This is substantial filtering, but is, in one way or another, commonly done Pek and Althoff (2020); Shalev-Shwartz et al. (2017).[16] Additionally, we filter out initially unsafe states (using the same filter for all methods).

Setting and simulation: We do an open-loop simulation where we only replace the ego vehicle by our method/the baselines, while keeping the others' trajectories from the original data. The action is 2-D (longitudinal and lateral acceleration) and as the simulation environment's dynamics f we take a component-wise double integrator. As collision/safety cost $d(s)$, we take minus the minimum l_∞ distance of ego to other vehicles (axis-aligned bounding boxes) and road boundary (it is particularly easy to compute), which is 2-Lipschitz (from norm $\|\cdot\|_\infty$ to $|\cdot|$).

Policy fail-safe imitator (FAGIL) We use our FAGIL imitator policy from Sec. 4.1. As learnable *state feature*, the state is first rendered into an abstract abstract birds-eye-view image, depicting other agents and road boundaries, which is then fed into a convolutional nets (CNNs) for dimensionality reduction, similar as described in (Chai et al., 2019). *Regarding safe set inference:* In this setting, where the two action/state dimensions and their dynamics are separable, the others' reachable rectangles can simply be computed

[16]This filter is based on the idea that the ego is not blamable for other cars crashing into it from behind (and otherwise safety may become too conservative; most initial states could already be unsafe if we consider adversarial rear vehicles that deliberately crash from behind).

Table 1: Outcome on real-world driver data, in terms of imitation and safety (i.e., collision) performance.

	Method	*Imitation performance*		*Safety performance*
Pre-safe	Overall	ADE	FDE	Probability crash & off-road
Gauss	FAGIL-E (ours)	0.99	2.68	0.00
	FAGIL-L (ours)	0.96	2.59	0.00
	GAIL Ho and Ermon (2016)	0.96	2.52	0.13
	RAIL Bhattacharyya et al. (2020)	0.96	2.52	0.13
	TTOS (Sec. 3.3)	1.09	2.94	0.00
Flow	FAGIL-E (ours)	1.56	4.20	0.00
	FAGIL-L (ours)	1.16	3.12	0.00
	GAIL Ho and Ermon (2016)	0.85	2.25	0.15
	RAIL Bhattacharyya et al. (2020)	0.83	2.20	0.14
	TTOS (Sec. 3.3)	1.60	4.53	0.00

exactly based on the others' maximum longitudinal/lateral acceleration/velocity (this can be extended to approximate separability (Pek and Althoff, 2020)). As ego's fallback maneuver candidates, we use non-linear shortest-time controllers to roll out emergency brake and evasive maneuver trajectories (Pek and Althoff, 2020). Then the total safety cost w is calculated by taking the minimum momentary safety cost d (as defined above, i.e., minimum distance) between ego's maneuvers and others' reachable rectangles over time. Note that the safety cost d and dynamics f with these fallbacks are each Lipschitz continuous as required by Prop. 1, implying the *safety guarantee* for FAGIL-L. (FAGIL-E roughly but not strictly satisfies the conditions of Prop. 2, but we conjecture it can be extended appropriately.) *As safety layer*, we use the probabilistic one ((a) plus (b')) from Sec. 4.1. **Training:** Training happens according to Sec. 4.2, using Wasserstein GAIL, with soft actor-critic (SAC) (Kostrikov et al., 2018; Haarnoja et al., 2018) as policy-gradient-based training (part (b) in Sec. 4.2), based on our closed-form density/gradient from Sec. 3.2. Further details on setup, methods and outcome are in Appendix C.

Baselines, pre-safe policies, ablations, metrics and evaluation Baselines are: *reward-augmented generative adversarial imitation learning (RAIL)* (closest comparison from our method class of safe generative IL, discussed in Sec. 1) Bhattacharyya et al. (2020; 2019), *test-time-only safety (TTOS)* from Sec. 3.3, and classic *GAIL* Ho and Ermon (2016). Note that TTOS can be seen as an *ablation study* of our method where we drop the safety layer during train-time, and GAIL an ablation where we drop it during train and test time. For all methods including ours, as *(pre-safe) policy* we evaluate both, *Gaussian policy* and *conditional normalizing flow*. We evaluate, on the test set, overall probability (frequency) of collisions over complete roll-outs (including crashes with other vehicles and going off-road) as *safety performance* measure; and *average displacement error (ADE*; i.e., average distance between trajectories) and *final displacement error (FDE)* over trajectories of 4s length as *imitation/prediction performance* measure (Suo et al., 2021).

Outcome and discussion The outcome is in Table 1, rounded to two decimal points. It empirically validates our theoretical *safety* claim by reaching 0% collisions of our methods (with fixed and moving obstacles). At the same time we come close in *imitation* performance to the baselines RAIL/GAIL, which, even if using reward augmentation to penalize collisions (RAIL), have significant collision rates. (The rather high collision rate of RAIL may also be due to the comparably small data set we use.) The poorer imitation performance of test-time-only safety layers (TTOS) compared to ours can be read as an empirical average-case confirmation of our theoretical worst-case analysis (Thm. 1). Fig. 3 gives one rollout sample, incl. ego's fail-safe and others' reachable futures considered by our safe set inference.

6 Conclusion

In this paper, we considered the problem of safe and robust generative imitation learning, which is of relevance for robust realistic simulations of agents, as well as when bringing learned agents into many real-world applications from robotics to autonomous driving. We contributed to filling several gaps that existed so far towards solving this problem: inference of guaranteed adversarially safe action sets, safety layers with closed-form density/gradient, and the theoretical understanding of end-to-end generative training with safety layers. Combining these results, we described a general abstract method for the problem, as well as one specific tractable instantiation for a restricted low-dimensional setting. The latter we experimentally evaluated on the task of driver imitation from real-world data, which includes fixed and moving, uncertain obstacles (the other drivers).

Limitations of our work come from the core challenges that safety goals often bring with them, in terms of tractability and cautiousness: First, the grid-based instantiation we give of our general theory is tractable mainly in the low-dimensional case, and in the experimental setting we harnessed tractability via separability of longitudinal and lateral dynamics of the other agents. Second, worst-case safety can lead to overly cautious actions while human "demonstrators" often achieve surprisingly good trade-offs in this regard; incorporating more restrictions about other agents, such as the responsibility-sensitive safety (RSS) framework, would still be covered by our general theory but help to be less conservative. Additionally, in the experimental evaluation, we concentrated on one challenging but restricted setting, leaving an extensive purely empirical study to future work.

In this sense, our work constitutes one step on the path of combining *flexibility and scalability* (i.e., general capacity for learning with little hand-crafting) of neural net-based probabilistic policies, with theoretic worst-case *safety guarantees*, while maintaining *end-to-end* generative trainability.

Broader Impact Statement

In the domain of control of physical systems that involve humans, such as robots that interact with humans ("cobots") or autonomous driving, where even small and/or rare mistakes of methods can injur people, "pure" deep imitation learning can be problematic, since often they are "black boxes" for which we do not understand precisely enough how they behave "outside the training data". Our approach can ideally have a postive impact on making imitation learning-based methods more safe and robust for such domains (or making them more applicable for such domains in the first place). One limitation of our work (beyond the technical limitations discussed above) is that in the end, society a has to debate and decide on the difficult trade-offs in terms of cautiousness versus actionability that safe control often involves (clearly safety is of highest value, but the question is how much residual risk society wants to take, given there rarely exists perfect safety). Science can just acknowledge the problem and provide understanding and tools. And overall, automation via machine learning approaches, including our approach, can have positive and negative impacts on jobs and the economoy overall, which society needs to consider and make decisions about.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

Lukáš Adam, Rostislav Horčík, Tomáš Kasl, and Tomáš Kroupa. Double oracle algorithm for computing equilibria in continuous games. *Proc. of AAAI-21*, 2021.

Anayo K Akametalu, Jaime F Fisac, Jeremy H Gillula, Shahab Kaynama, Melanie N Zeilinger, and Claire J Tomlin. Reachability-based safe learning with gaussian processes. In *53rd IEEE Conference on Decision and Control*, pages 1424–1431. IEEE, 2014.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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. JMLR. org, 2017.

Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J Zico Kolter. Differentiable mpc for end-to-end planning and control. In *Advances in Neural Information Processing Systems*, pages 8289–8300, 2018.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, pages 688–699, 2019.

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253. IEEE, 2017.

Nir Baram, Oron Anschel, Itai Caspi, and Shie Mannor. End-to-end differentiable adversarial imitation learning. In *International Conference on Machine Learning*, pages 390–399. PMLR, 2017.

Felix Berkenkamp, Matteo Turchetta, Angela P Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *arXiv preprint arXiv:1705.08551*, 2017.

Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

Michel Besserve and Bernhard Schölkopf. Learning soft interventions in complex equilibrium systems. *arXiv preprint arXiv:2112.05729*, 2021.

Raunak Bhattacharyya, Blake Wulfe, Derek Phillips, Alex Kuefler, Jeremy Morton, Ransalu Senanayake, and Mykel Kochenderfer. Modeling human driving behavior through generative adversarial imitation learning. *arXiv preprint arXiv:2006.06412*, 2020.

Raunak P Bhattacharyya, Derek J Phillips, Changliu Liu, Jayesh K Gupta, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 789–795. IEEE, 2019.

Damian Boborzi, Florian Kleinicke, Jens Buchner, and Lars Mikelsons. Learning to drive from observations while staying safe. 2021. upcoming.

Daniel Brown, Russell Coleman, Ravi Srinivasan, and Scott Niekum. Safe imitation learning via fast bayesian reward inference from preferences. In *International Conference on Machine Learning*, pages 1165–1177. PMLR, 2020.

Zhangjie Cao, Erdem Bıyık, Woodrow Z Wang, Allan Raventos, Adrien Gaidon, Guy Rosman, and Dorsa Sadigh. Reinforcement learning based control of imitative policies for near-accident driving. *arXiv preprint arXiv:2007.00178*, 2020.

Michael R Caputo. The envelope theorem and comparative statics of nash equilibria. *Games and Economic Behavior*, 13(2):201–224, 1996.

Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.

Jianyu Chen, Bodi Yuan, and Masayoshi Tomizuka. Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety. *arXiv preprint arXiv:1903.00640*, 2019.

Zhihao Cheng, Li Shen, Meng Fang, Liu Liu, and Dacheng Tao. Lagrangian generative adversarial imitation learning with safety, 2022. URL `https://openreview.net/forum?id=11PMuvv3tEO`.

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

Priya L Donti, Melrose Roderick, Mahyar Fazlyab, and J Zico Kolter. Enforcing robust control guarantees within neural network policies. In *International Conference on Learning Representations*, 2021a.

Priya L Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. *arXiv preprint arXiv:2104.12225*, 2021b.

Laurent El Ghaoui, Fangda Gu, Bertrand Travacca, and Armin Askari. Implicit deep learning. *arXiv preprint arXiv:1908.06315*, 2019.

David Fridovich-Keil and Claire J Tomlin. Approximate solutions to a class of reachability games. *arXiv preprint arXiv:2011.00601*, 2020.

Philipp Geiger and Christoph-Nikolas Straehle. Learning game-theoretic models of multiagent trajectories using implicit layers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4950–4958, 2021.

Jeremy H Gillula, Shahab Kaynama, and Claire J Tomlin. Sampling-based approximation of the viability kernel for high-dimensional linear sampled-data systems. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 173–182, 2014.

Colas Le Guernic and Antoine Girard. Reachability analysis of hybrid systems using support functions. In *International Conference on Computer Aided Verification*, pages 540–554. Springer, 2009.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

Aaron Havens and Bin Hu. On imitation learning of linear control policies: Enforcing stability and robustness constraints via lmi conditions. In *2021 American Control Conference (ACC)*, pages 882–887. IEEE, 2021.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.

Junning Huang, Sirui Xie, Jiankai Sun, Qiurui Ma, Chunxiao Liu, Jianping Shi, Dahua Lin, and Bolei Zhou. Learning a decision module by imitating driver's control behaviors. *arXiv preprint arXiv:1912.00191*, 2019.

Maximilian Igl, Daewoo Kim, Alex Kuefler, Paul Mougin, Punit Shah, Kyriacos Shiarlis, Dragomir Anguelov, Mark Palatucci, Brandyn White, and Shimon Whiteson. Symphony: Learning realistic and diverse agents for autonomous driving simulation. *arXiv preprint arXiv:2205.03195*, 2022.

Zaynah Javed, Daniel S Brown, Satvik Sharma, Jerry Zhu, Ashwin Balakrishna, Marek Petrik, Anca Dragan, and Ken Goldberg. Policy gradient bayesian robust optimization for imitation learning. In *International Conference on Machine Learning*, pages 4785–4796. PMLR, 2021.

Achim Klenke. *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013.

Vilmos Komornik. *Topology, calculus and approximation*. Springer, 2017.

Philip Koopman, Aaron Kane, and Jen Black. Credible autonomy safety argumentation. In *27th Safety-Critical Systems Symposium*, 2019.

Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *arXiv preprint arXiv:1809.02925*, 2018.

Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

Jonathan Lacotte, Mohammad Ghavamzadeh, Yinlam Chow, and Marco Pavone. Risk-sensitive generative adversarial imitation learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2154–2163. PMLR, 2019.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. *arXiv preprint arXiv:1711.00832*, 2017.

Chun Kai Ling, Fei Fang, and J Zico Kolter. What game are we playing? End-to-end learning in normal and extensive form games. In *IJCAI-ECAI-18: The 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence*, 2018.

Chun Kai Ling, Fei Fang, and J Zico Kolter. Large scale learning of agent rationality in two-player zero-sum games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6104–6111, 2019.

Bjørnar Steinnes Luteberget. Numerical approximation of conformal mappings. Master's thesis, Institutt for matematiske fag, 2010.

Xiaobai Ma, Jayesh K Gupta, and Mykel J Kochenderfer. Normalizing flow policies for multi-agent systems. In *International Conference on Decision and Game Theory for Security*, pages 277–296. Springer, 2020.

Silvia Magdici and Matthias Althoff. Fail-safe motion planning of autonomous vehicles. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 452–458. IEEE, 2016.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *arXiv preprint arXiv:1811.06711*, 2018.

Fabian Otto, Philipp Becker, Ngo Anh Vien, Hanna Carolin Ziesche, and Gerhard Neumann. Differentiable trust region layers for deep reinforcement learning. *arXiv preprint arXiv:2101.09207*, 2021.

Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

Christian Pek and Matthias Althoff. Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1447–1454. IEEE, 2018.

Christian Pek and Matthias Althoff. Fail-safe motion planning for online verification of autonomous vehicles using convex optimization. *IEEE Transactions on Robotics*, 37(3):798–814, 2020.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *arXiv preprint arXiv:1905.06113*, 2019.

Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

Matthias Rungger and Paulo Tabuada. Computing robust controlled invariant sets of linear systems. *IEEE Transactions on Automatic Control*, 62(7):3665–3670, 2017.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*, 2017.

Chelsea Sidrane, Amir Maleki, Ahmed Irfan, and Mykel J Kochenderfer. Overt: An algorithm for safety verification of neural network control policies for nonlinear systems. *Journal of Machine Learning Research*, 23(117):1–45, 2022.

Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. *arXiv preprint arXiv:1807.09936*, 2018.

Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. *arXiv preprint arXiv:2101.06557*, 2021.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Umar Syed and Robert E Schapire. A reduction from apprenticeship learning to classification. *Advances in neural information processing systems*, 23:2253–2261, 2010.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922, 2021.

Stephen Tu, Alexander Robey, Tingnan Zhang, and Nikolai Matni. On the sample complexity of stability constrained imitation learning. In *Learning for Dynamics and Control Conference*, pages 180–191. PMLR, 2022.

Renée Vidal, Shawn Schaffert, John Lygeros, and Shankar Sastry. Controlled invariance of discrete time systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 437–451. Springer, 2000.

Kim P Wabersich and Melanie N Zeilinger. Linear model predictive safety certification for learning-based control. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 7130–7135. IEEE, 2018.

Patrick Nadeem Ward, Ariella Smofsky, and Avishek Joey Bose. Improving exploration in soft-actor-critic with normalizing flows policies. *arXiv preprint arXiv:1906.02771*, 2019.

Huang Xiao, Michael Herman, Joerg Wagner, Sebastian Ziesche, Jalal Etesami, and Thai Hong Linh. Wasserstein adversarial imitation learning. *arXiv preprint arXiv:1906.08113*, 2019.

Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. *arXiv preprint arXiv:2010.11876*, 2020.

He Yin, Peter Seiler, Ming Jin, and Murat Arcak. Imitation learning with stability and safety guarantees. *IEEE Control Systems Letters*, 2021.

Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8660–8669, 2019.

Wenyuan Zeng, Shenlong Wang, Renjie Liao, Yun Chen, Bin Yang, and Raquel Urtasun. Dsdnet: Deep structured self-driving network. In *European Conference on Computer Vision*, pages 156–172. Springer, 2020.

Chengyuan Zhang, Jiacheng Zhu, Wenshuo Wang, and Junqiang Xi. Spatiotemporal learning of multivehicle interaction patterns in lane-change scenarios. *arXiv preprint arXiv:2003.00759*, 2020.

Appendix

A Proofs with remarks

This section contains all proofs for the mathematical statements of the main text, as well as some additional remarks.

A.1 Proofs for Sec. 3.1

A.1.1 Proof of Prop. 1

We will need the following known result (a fact that lies somewhere between *envelope* and *maximum theorem*):

Fact 1. *Let h be a function from $X \times Y$ to $\mathbb{R}$, and assume X and $\mathbb{R}$ to be equipped with some given norms (to which Lipschitz continuity refers in the following). Assume $h(x, y)$ is Lipschitz in x with constant L uniformly in y and assume that maxima exist. Then the function $x \mapsto \max_y h(x, y)$ is also L-Lipschitz.*

For the sake of completeness, we provide a proof of Fact 1. Note: We show the statement for the case that a supremum is always taken, i.e., is a maximum. For the supremum version we refer the reader to related work.[17]

Proof of Fact 1. Let $\|\cdot\|$ and $|\cdot|$ denote the given norms on $X, \mathbb{R}$, respectively. Let $e(x) := \arg\max_y h(x, y)$ (or pick one argmax if there are several).

Consider an arbitrary but fixed pair x, x' and let, w.l.o.g., $\max_y h(x, y) \geq \max_y h(x', y)$. Then:

$$
\begin{aligned}
&|\max_y h(x, y) - \max_y h(x', y)| && (8)\\
&= \max_y h(x, y) - \max_y h(x', y) && (9)\\
&= h(x, e(x)) - h(x', e(x')) && (10)\\
&\leq h(x, e(x)) - h(x', e(x)) \qquad (\text{since } h(x', e(x')) \geq h(x', e(x))) && (11)\\
&\leq L\|x - x'\|. && (12)
\end{aligned}
$$

□

Now to the proof itself:

Proof for Prop. 1. The idea is to first propagate the Lipschitz continuity through the dynamics, and then iteratively apply Fact 1 to the min/max operations.

Here, for any Lipschitz continuous function e, let L_e denote its constant. Furthermore, regarding policies, we may write π, σ as shorthand for $\pi_t, \sigma_t, t \in 1 : T$, respectively.

First observe the following, given an arbitrary but fixed t and π, σ:

Let

$$g(s, \pi, \sigma) := f(s, \pi(s), \sigma(s)) \tag{13}$$

and $h(s, \pi, \sigma, k)$ be defined as the k-times concatenation of $g(\cdot, \pi, \sigma)$, i.e., a roll out; and then define function i by also including the initial action, i.e.,

$$i(s, a, \pi, \sigma, k) := h(f(s, a, \sigma(s)), \pi, \sigma, k). \tag{14}$$

[17] For a proof, see e.g., `https://math.stackexchange.com/q/2532116/1060605`.

Then, by our assumptions, for π, σ, k arbitrary but fixed,

$$s \mapsto d(h(s, \pi, \sigma, k)) \tag{15}$$

is Lipschitz continuous in s with constant (uniformly in π, σ)

$$L_d L^k_{f(\cdot,\pi(\cdot),\sigma(\cdot))} = \alpha\beta^k, \tag{16}$$

since it is the composition of d, which is α-Lipschitz, with the k-times concatenation of g, which is β-Lipschitz by assumption. This together with our assumption implies that, for any arbitrary but fixed s, also

$$a \mapsto d(i(s, a, \pi, \sigma, k))(= w_t(s, a)) \tag{17}$$

is Lipschitz uniformly in π, σ, with constant (for any k)

$$\alpha\beta^k\beta = \alpha\beta^{k+1} \leq \alpha \max\{1, \beta^T\}. \tag{18}$$

Now let state s be arbitrary but fixed. Also let π be arbitrary but fixed. Then, based on the above together with Fact 1,

$$a \mapsto \max_{\sigma_{t:T}} \max_{t \in t+1:T} d(i(s, a, \pi, \sigma, t))(=: e(a, \pi)) \tag{19}$$

is Lipschitz with constant $\alpha \max\{1, \beta^T\}$. Again applying Fact 1, we see that also

$$a \mapsto \min_{\pi_{t+1:T}} \max_{\sigma_{t:T}} \max_{t \in t+1:T} d(i(s, a, \pi, \sigma, t))(= \min_{\pi} e(a, \pi)) \tag{20}$$

is Lipschitz with constant $\alpha \max\{1, \beta^T\}$.

□

Remark 3. *Alternative versions of this result could be based, e.g., on the "Envelope Theorem for Nash Equilibria" (Caputo, 1996).*

A.1.2 Proof of Prop. 2

Proof for Prop. 2. Let the mapping i be defined as in Eq. (14). Fix s.

Observe that, based on our assumption that d is convex and the dynamics f linear, for any σ, t fixed,

$$(a, \pi) \mapsto d(i(s, a, \pi, \sigma, t)) \tag{21}$$

is convex. But then, also the maximum over σ, t, i.e.,

$$(a, \pi) \mapsto \max_{\sigma_{t:T}} \max_{t \in t+1:T} d(i(s, a, \pi, \sigma, t)) \tag{22}$$

is convex, because the element-wise maximum over a family of convex functions – in our case the family indexed by σ, t – is again convex.

One step further, this means (since convexity is preserved by "minimizing out" one variable over a convex domain) that also taking the minimum over π preserves convexity, i.e.,

$$w_t(s, a) = \min_{\pi_{t+1:T}} \max_{\sigma_{t:T}} \max_{t \in t+1:T} d(i(s, a, \pi, \sigma, t)) \tag{23}$$

is convex as a function of a.

But a convex function over a (compact) convex set that is spanned by a set of corners, takes its maximum at (at least one) of the corners.

□

A.2 Proofs for Sec. 3.2

A.2.1 Proof of Prop. 3

Proof for Prop. 3. Intuitively it is clear that the results follows from combining σ-additivity of measures with the change-of-variables formula. Nonetheless we give a rigorous derivation here.

Keep in mind the following explicit specifications that were left implicit in the main part: In the main part we just stated to assume Lebesgue densities on $A \subset \mathbb{R}^n$, implicitly we assume $\mathbb{R}^n$ with the usual Lebesgue σ-algebra as underlying measurable space, and that all parts A_k are measurable. Furthermore, to be precise, we assume differentiability of the g_k on the *interior* of their respective domain.

Recall that $(A_k)_k$ is a countable partition of A, and we assumed diffeomorphisms $g_k : A_k \to \bar{A}_k \subset \tilde{A}$ (diffeomorphic on the interior of A_k) and that $g|_{A_k} = g_k$.

Note that based on our assumptions, either g is already measurable, or, otherwise, we can turn it into a measurable function by just modifying it on a Lebesgue null set (the boundaries of the A_k), not affecting the below argument.

Let P_A denote the original measure on outcome space A (restriction from $\mathbb{R}^n$), and $P_{\tilde{A}}$ its push-forward measure on $\tilde{A}$ induced by g.

Then, up to null sets, for any measurable set $M \subset \tilde{A}$,

$$
\begin{aligned}
& P_{\tilde{A}}(M) && (24)\\
&= P_A(g^{-1}(M)) && (25)\\
&= P_A(g^{-1}(M) \cap \bigcup_k A_k) && (26)\\
&= P_A(\bigcup_k g^{-1}(M) \cap A_k) && (27)\\
&= P_A(\bigcup_k g_k^{-1}(M)) && (28)\\
&= \sum_k P_A(g_k^{-1}(M)) && (29)\\
&= \sum_k \int_M |\det(J_{g_k^{-1}}(\bar{a}))| p_{\hat{a}}(g_k^{-1}(\bar{a}))[\bar{a} \in g_k(A_k)] d\bar{a} && (30)\\
&= \int_M \sum_{k:\bar{a}\in g_k(A_k)} |\det(J_{g_k^{-1}}(\bar{a}))| p_{\hat{a}}(g_k^{-1}(\bar{a})) d\bar{a} && (31)
\end{aligned}
$$

where Eq. (30) is the classic change-of-variables (also called transformation formula/theorem) (Klenke, 2013) applied to the push-forward measure of P_A under g_k. Therefore, the[18] density of $P_{\tilde{A}}$, i.e., $p_{\bar{a}}(\bar{a})$, is given by $\sum_{k:\bar{a}\in g_k(A_k)} |\det(J_{g_k^{-1}}(\hat{a}))| p_{\hat{a}}(g_k^{-1}(\bar{a}))$.

□

A.3 Proofs and precise counterexample figure/MDP/agents for Sec. 3.3

In this section we give all elaborations and proofs for the end-to-end versus test-time-only safety analysis of Sec. 3.3.

For this section, keep in mind:

- In the discrete setting of this section, the imitation cost function c etc. are vectors in the Euklidean space here; and, for instance, $cP^D_{S_t}$ stands for the inner product, the expectation of the cost variable c.

[18] Rigorously speaking, it is *a* density.

- If not stated otherwise, $\|\cdot\|$ refers to the l_1-norm $\|\cdot\|_1$.
- Keep in mind that for the total variation distance D_{TV} we have the relation to the l_1 norm $D_{\mathrm{TV}}(p,q)=\frac{1}{2}\|p-q\|_1$.
- Keep in mind that, as in the main text, letters D, I, U, O stand for demonstrator, imitator (our fail-safe imitator with safety layer already during training time), unsafe/unconstrained trained imitator, and test-time-only-safety imitator; and P^D, P^I, P^U, P^O etc. denote the probabilities under the respective policies, and same for ρ^D etc.

A.3.1 Derviation of Rem. 1

The following derivation works along the line of related established derivations, e.g., the one for infinite-horizon GAIL guarantees (Xu et al., 2020).

Derviation of Rem. 1. Keep in mind that for any discrete distributions $p_{s,a}(s,a), q_{s,a}(s,a)$, when marginalizig out a, we get for p_s, q_s,

$$\begin{aligned}
&\|p_s - q_s\| && (32)\\
=&\sum_s |p(s)-q(s)| && (33)\\
=&\sum_s |\sum_a p(s,a)-q(s,a)| && (34)\\
\leq&\sum_s\sum_a |p(s,a)-q(s,a)| \qquad \text{(triangle inequality)} && (35)\\
=&\|p_{s,a}-q_{s,a}\|. && (36)
\end{aligned}$$

Now we have

$$\begin{aligned}
&|v^I - v^D| && (37)\\
=&|\sum_t cP^I_{s_t} - \sum_t cP^D_{s_t}| && (38)\\
=&|cT(\sum_t \frac{1}{T}P^I_{s_t} - \sum_t \frac{1}{T}P^D_{s_t})| && (39)\\
=&T|c(\rho^{\mathrm{I}}_s - \rho^{\mathrm{D}}_s)| \qquad \text{(Definition } \rho) && (40)\\
\leq&T\|c\|_\infty\|\rho^{\mathrm{I}}_s - \rho^{\mathrm{D}}_s\| && (41)\\
\leq&T\|c\|_\infty\|\rho^{\mathrm{I}}_{s,a} - \rho^{\mathrm{D}}_{s,a}\| \qquad \text{(Eq. (36))} && (42)\\
=&T\|c\|_\infty 2D_{\mathrm{TV}}(\rho^{\mathrm{I}}_{s,a}, \rho^{\mathrm{D}}_{s,a}). && (43)
\end{aligned}$$

□

A.3.2 Proof of Thm. 1, including elaboration of example (figure)

Elaboration of the example for the lower bound (Fig. 2) Here, let us first give a precise version – Fig. 4 – of the example in Fig. 2, that is used to get the lower bound in Thm. 1. Instead of the exact Fig. 2, we give an "isomorphic" version with more detailed/realistic states and actions in terms of relative position/velocity/accleration/lane.

Specify the Markov decision process (MDP) and agents (policies) as follows, with imitation error $p^U{=}\delta$ as paramteter:

MDP:

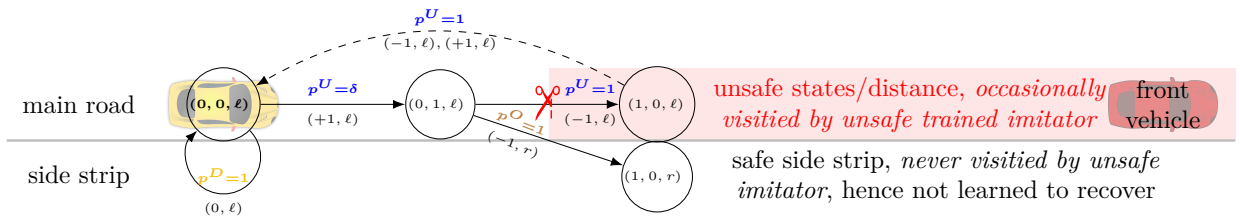


Figure 4: More detailed (but essentially isomorphic) version of the example MDP/agents from Fig. 2.

- *States:* The states (of the ego) are triplets (Δx, Δv, lane), where Δx is deviation from demonstrator position, Δv is deviation from demonstrator velocity, lane is lane (ℓ for left, or r for right). (The demonstrator is in a safe state.) There are five possible states, all depicted in the figure, except for $(1, -1, \ell)$, to keep the figure simple.

- *Actions:* The possible actions are depicted as arrows, with the action (acc, lane) written below the arrow, where acc means the longitudinal acceleration, and lane the target lane.

- *Transition:* The longitudinal transition is the usual double integrator of adding acc to Δv, while, lateral-wise, action lane instantaneously sets the new lane (more realistic but complex examples are possible of course).

- *Costs:* The true demonstrator's cost c^* is 0 on the demonstrator state $(0, 0, \ell)$, and 1 on every other state.

- *Horizon:* form $t = 1$ to T, where we consider T a parameter (so it is strictly speaking a family of MDPs).

Agents:

- *Demonstrator:* Yellow car, always ($p^D = 1$) staying at the depicted state, referred to as s^D ($= (0, 0, \ell)$), at a more than safe distance to front car (red).

- *Unconstrained trained imitator (blue):* due to a slight imitation error, it is sometimes (with probability $p^U{=}\delta$) accelerating, and if it accelerates, it necessarily reaches an unsafe state (after one more step, due to the "inertia" of the double integrator), but, as implied by the GAIL loss, always recovers (dashed arrows) back to true demonstrator state.

- *Safety-constrained test-time imitator (same, except for last action in brown):* as the unconstrained imitator, it occasionally ($p^O{=}\delta$) ends up at rightmost safe state, *and then the only remaining safe action is to change lane to the side strip, where there was no data on how to recover, so getting trapped there forever.*

Initialization – agent-dependent: While the overall agrument works as well for an agent-independent initialization, we do an agent-specific initialization, since it simplifies calculations substantially. At $t = 1$, for the demonstartor, we let $p(s_1)$ have full mass on the demonstrator state s^D. For the unconstrained imitator, as $p(s_1)$, we take the stationary distribution under the induced Markov process, which is $\frac{1}{1+3\delta}(1, \delta, \delta, \delta)$, where the first component means the demonstartor state s^D.

Proof of Thm. 1 Keep in mind that we will occasionally use upper case letters like S_t, A_t to refer to the actual random variables of state and action, distinguishing them from the values s_t, a_t that those random variables can take.

Proof for Thm. 1. **Lower bound part:**

Keep in mind the counterexample construction above, in particular its deviation parameter δ, to which the following arguments refer. Let s^D, a^D denote the state and action that the demonstrator is constantly in.

Construction of δ; implication $D_{TV}(\rho^D, \rho^U) \leq \varepsilon$:

By construction, we have $\rho^D(s^D, a^D) = 1$, while for the unconstrained imitator, $\rho^U(s^D, a^D) = \frac{1}{1+3\delta} \cdot (1-\delta)$ (recall that $\rho^U(s)$ is the time-average over the state distribution, which we took as the stationary one for the unconstrained imitator, giving mass $\frac{1}{1+3\delta}$ on s^D, and $1-\delta$ is the action probability).

So

$$|\rho^D(s^D, a^D) - \rho^U(s^D, a^D)| = |1 - \frac{1-\delta}{1+3\delta}| = \frac{4\delta}{1+3\delta}.$$

So, by symmetry, $D_{\mathrm{TV}}(\rho^D, \rho^U)$ is one half times twice this,

$$D_{\mathrm{TV}}(\rho^D, \rho^U) \leq \frac{4\delta}{1+3\delta}.$$

Additionally, it is easy to see that for $1 \geq \delta \geq 0$, $\frac{4\delta}{1+3\delta} \leq 4\delta$, and so

$$D_{\mathrm{TV}}(\rho^D, \rho^U) \leq 4\delta. \tag{44}$$

Therefore choose

$$\delta := \frac{1}{4}\varepsilon, \tag{45}$$

based on the ε given by assumption. This implies $D_{\mathrm{TV}}(\rho^D, \rho^U) \leq \varepsilon$.

Preparation of a bound statement we need later:

As preparation which we repeatedly need later, observe that there is some constant $\kappa > 0$, such that for all $T \geq 2$, we have

$$\kappa \leq (1 + \frac{-1}{T})^T \leq e^{-1}, \tag{46}$$

where e here is the Euler number (exponential) as usual, and for κ we can take, e.g., $\frac{1}{4}$. To have a first hint why this holds, see that $(1+\frac{-1}{T})^T \xrightarrow{T\to\infty} e^{-1}$. For the full argument, observe the following: First, observe that at $T=2$, we have $(1+\frac{-1}{T})^T = \frac{1}{4}$. Second, it is sufficient to show that the expression is monotonically growing, i.e., the derivative $\frac{\mathrm{d}}{\mathrm{d}T}(1+\frac{-1}{T})^T$ is positive for $T \geq 2$. But this is equivalent to $\frac{\mathrm{d}}{\mathrm{d}T}\log((1+\frac{-1}{T})^T) = \frac{1}{T-1} + \log(\frac{T-1}{T})$ being positive, since log is a monotonic function. But the latter follows from the fact that always $x \geq \log(1+x)$ and then plugging in $\frac{1}{T-1}$ for x, yieldig $\frac{1}{T-1} \geq \log(1+\frac{1}{T-1}) = \log(\frac{T}{T-1}) = -\log(\frac{T-1}{T})$.

Now, before going into the main proof parts, recall that we want to show

$$|v^O - v^D| \geq \iota \min\{\varepsilon T^2, T\}\|c^*\|_\infty \tag{47}$$

for ι some constant indepentent of ε, T.

We do so by considering the case where it is quadratic separately from where it is linear.[19]

Proof for the lower bound, case $\delta T \leq 1$:

Consider the case $\delta T \leq 1$.

By construction, at each fixed stage t, the probability that the test-time constrained imitator π^O is not in the demonstrator state is the summed probability of not deviating from s^D for k times, $k \leq t$, and then deviating once, i.e.,

$$P^O(S_t \neq s^D) = \sum_{k=1}^{t} \delta(1-\delta)^{k-1}. \tag{48}$$

[19]Note that our construction is different from behavior cloning error analysis (Ross and Bagnell, 2010; Syed and Schapire, 2010) especially in that we do not need one separate starting state.

We want to bound the term $(1-\delta)^{k-1}$ from below, to show that the quadratic bound of Eq. (47) holds under the current case $\delta T \leq 1$.

We have, for any $k \leq T$,

$$(1-\delta)^k \geq (1-\frac{1}{T})^k \geq (1-\frac{1}{T})^T = (1+\frac{-1}{T})^T \geq \kappa \tag{49}$$

where the first inequality holds because we assumed $\delta \leq \frac{1}{T}$ and so $1-\delta \geq 1-\frac{1}{T}$, and the second one because always $k \leq T$ and third is Eq. (46).

Therefore Eq. (48) can be bounded from below by

$$\sum_{k=1}^{t} \delta\kappa = t\delta\kappa. \tag{50}$$

Now we get, with the cost vector $\vec{c}$ (i.e., the safety cost function c, but making explicit it is a vector since we are in the discrete case) being 0 in the demonstrator state, and $\|\vec{c}\|_\infty$ elsewhere,

$$\begin{aligned}
&|v^O - v^D| && (51)\\
&= |\sum_t (P^O_{s_t} - P^D_{s_t}) \cdot \vec{c}| && (52)\\
&= |\sum_t (P^O(S_t \neq s^D) - 0)\|\vec{c}\|_\infty| && (53)\\
&\geq \sum_t (t\delta\kappa) \cdot \|\vec{c}\|_\infty \qquad \text{(Eq. (50) etc.)} && (54)\\
&\geq \delta\kappa\|\vec{c}\|_\infty \sum_t t && (55)\\
&\geq \delta\kappa'\|\vec{c}\|_\infty T^2 && (56)
\end{aligned}$$

for some κ' (that absorbs both, κ and the deviation between T^2 and $\sum_t t$.

Proof for the lower bound, case $\delta T > 1$:

First note that, alternatively to the previous deviation, we can also write $P^O(S_t \neq s^D)$ as the complementary probability of test-time constrained imitator *not* having deviated from the demonstrator state s^D so far, i.e.,

$$P^O(S_t \neq s^D) = 1-(1-\delta)^t \geq 1-(1-\frac{1}{T})^t, \tag{57}$$

since in the current case, $\delta T > 1$ and so $\delta > \frac{1}{T}$ and so $1-\delta < 1-\frac{1}{T}$ and so $(1-\delta)^t < (1-\frac{1}{T})^t$ and so $1-(1-\delta)^t > 1-(1-\frac{1}{T})^t$.

Now, for $t \geq \frac{T}{2}$, we get,

$$(1-\frac{1}{T})^t \leq (1-\frac{1}{T})^{\frac{T}{2}} = ((1+\frac{-1}{T})^T)^{\frac{1}{2}} \leq (e^{-1})^{\frac{1}{2}} < 1, \tag{58}$$

where the second last inequality is Eq. (46). So $(1-\frac{1}{T})^t$ is strictly below and bounded away from 1, and therefore Eq. (57) is uniformly strictly above and bounded away from 0, i.e., $P^C(S_t \neq s^D)$ is greater $\kappa'' > 0$ for $t \geq \frac{T}{2}$.

Then

$$
\begin{aligned}
&|v^O - v^D| && (59)\\
&= |\sum_{t=1}^{T}(P^O_{s_t} - P^D_{s_t}) \cdot \vec{c}| && (60)\\
&= |\sum_{t=1}^{T}(P^O(S_t \neq s^D) - 0)\|\vec{c}\|_\infty| && (61)\\
&\geq \sum_{t=\frac{T}{2}}^{T} P^O(S_t \neq s^D)\|\vec{c}\|_\infty && (62)\\
&\geq \sum_{t=\frac{T}{2}}^{T} \kappa''\|\vec{c}\|_\infty && (63)\\
&= \frac{T}{2}\kappa''\|\vec{c}\|_\infty && (64)\\
&= \kappa'''T\|\vec{c}\|_\infty. && (65)
\end{aligned}
$$

Proof for the lower bound, final remarks:

ι is given by taking the minima of the above κ's and conbining it with the relation Eq. (45).

Upper bound part:

In what follows, P stand for probability vectors/matrices, slightly overriding notation, and $P_{S'|S}$ for the transition probability (corresponding to f plus potential noise).

First observe that generally (here P stand for probability vectors/matrices, slightly overriding notation),

$$
\begin{aligned}
\Delta_{t+1} &:= \|P^O_{S_{t+1}} - P^D_{S_{t+1}}\|_1 && (66)\\
&= \|P^O_{S'|S}P^O_{S_t} - P^D_{S'|S}P^D_{S_t}\|_1 && (67)\\
&= \|P^O_{S'|S}P^O_{S_t} - P^O_{S'|S}P^D_{S_t} + P^O_{S'|S}P^D_{S_t} - P^D_{S'|S}P^D_{S_t}\|_1 && (68)\\
&= \|P^O_{S'|S}(P^O_{S_t} - P^D_{S_t}) + (P^O_{S'|S} - P^D_{S'|S})P^D_{S_t}\|_1 && (69)\\
&\leq \|P^O_{S'|S}(P^O_{S_t} - P^D_{S_t})\|_1 + \|(P^O_{S'|S} - P^D_{S'|S})P^D_{S_t}\|_1 && (70)
\end{aligned}
$$

First consider the first term in Eq. (70)*.* We can generally, just since the 1-norm of any stochastic matrix is 1, bound it as follows:

$$
\begin{aligned}
&\|P^O_{S'|S}(P^O_{S_t} - P^D_{S_t})\|_1 && (71)\\
&\leq \|P^O_{S'|S}\|_1\Delta_t = 1\Delta_t && (72)
\end{aligned}
$$

Regarding the second term in Eq. (70)*,* we have the following way to bound it.

Let $\bar{S}$ denote the subset of the set S of states where $\rho^D(s)$ has support, and, as in the main text, $\nu > 0$ is the minimum value of $\rho^D(s)$ on $\bar{S}$. Then

$$\nu \sum_{s\in\bar{S}} \|\pi^D(\cdot|s) - \pi^U(\cdot|s)\|_1 \tag{73}$$

$$= \nu \sum_{s\in\bar{S},a} |\pi^D(a|s) - \pi^U(a|s)| \tag{74}$$

$$\leq \sum_{s\in\bar{S},a} \rho^D(s)|\pi^D(a|s) - \pi^U(a|s)| \tag{75}$$

$$= \sum_{s,a} \rho^D(s)|\pi^D(a|s) - \pi^U(a|s)| \tag{76}$$

$$= \sum_{s,a} |\pi^D(a|s)\rho^D(s) - \pi^U(a|s)\rho^D(s)| \tag{77}$$

$$\tag{78}$$

$$= \sum_{s,a} |\pi^D(a|s)\rho^D(s) + \pi^U(a|s)\rho^U(s) - \pi^U(a|s)\rho^U(s) - \pi^U(a|s)\rho^D(s)| \tag{79}$$

$$\leq \sum_{s,a} |\pi^D(a|s)\rho^D(s) - \pi^U(a|s)\rho^U(s)| + |\pi^U(a|s)\rho^U(s) - \pi^U(a|s)\rho^D(s)| \qquad \text{(triangle inequality)} \tag{80}$$

$$= \sum_{s,a} |\pi^D(a|s)\rho^D(s) - \pi^U(a|s)\rho^U(s)| + \pi^U(a|s)|\rho^U(s) - \rho^D(s)|? \tag{81}$$

$$\leq \sum_{s,a} |\pi^D(a|s)\rho^D(s) - \pi^U(a|s)\rho^U(s)| + |\rho^U(s) - \rho^D(s)| \tag{82}$$

$$= \|\rho^D_{s,a} - \rho^U_{s,a}\|_1 + \|\rho^D_s - \rho^U_s\|_1 \qquad \text{(recall definition } \rho\text{)} \tag{83}$$

$$\leq 2\varepsilon + 2\varepsilon \qquad \text{(by assumption and Eq. (36))}, \tag{84}$$

so in particular

$$\max_{s\in\bar{S}} \|\pi^D(\cdot|s) - \pi^U(\cdot|s)\|_1 \leq 4\varepsilon/\nu. \tag{85}$$

Therefore

$$\|P^D_{a|s|_{\bar{S}}} - P^U_{a|s|_{\bar{S}}}\|_1 = \|\pi^D_{a|s|_{\bar{S}}} - \pi^U_{a|s|_{\bar{S}}}\|_1 = \max_{s\in\bar{S}} \|\pi^D(\cdot|s) - \pi^U(\cdot|s)\|_1 \leq \frac{4\varepsilon}{\nu}, \tag{86}$$

where the first equation is just a rewriting (note that here, $\pi^D_{a|s|_{\bar{S}}}$ etc. denote matrices (transition matrix), while $\pi^D(\cdot|s)$ etc. denotes a vector (probability vector)). Note that this also implies the same bound for the test-time-constrained imitator, i.e.,

$$\|P^D_{a|s|_{\bar{S}}} - P^O_{a|s|_{\bar{S}}}\|_1 \leq \frac{4\varepsilon}{\nu}, \tag{87}$$

for the following reason: only the deviating mass between D and U that lies on actions where D does not have support can be redistributed by the safety layer (because we assumed that D is safe, and hence those actions are safe and are not affected by adding a safety layer, based on our assumption about the safety layer). But this deviation mass already lies where it has the maximum effect on the norm (namely: where D does not have any mass at all). So redistributing it will make the deviation not bigger.

Furthermore we have, just by the definition of the conditional probability matrix $P_{S,A|S}$

$$\|(P^O_{S,A|S} - P^D_{S,A|S})\|_1 = \|(P^O_{A|S} - P^D_{A|S})\|_1, \tag{88}$$

and the same holds when restricting the state to any subset, in particular $\bar{S}$.

Now, to finalize our argument to also bound the second term in Eq. (70), observe

$$\|(P^O_{S'|S} - P^D_{S'|S})P^D_{S_t}\|_1 \tag{89}$$
$$= \|(P^O_{S'|S|_{\bar{S}}} - P^D_{S'|S|_{\bar{S}}})P^D_{S_t|_{\bar{S}}}\|_1 \qquad \text{(since } \bar{S} \text{ is the demonstrator support)} \tag{90}$$
$$\leq \|(P^O_{S'|S|_{\bar{S}}} - P^D_{S'|S|_{\bar{S}}})\|_1 \|P^D_{S_t|_{\bar{S}}}\|_1 \tag{91}$$
$$= \|P_{S'|S,A}(P^O_{S,A|S|_{\bar{S}}} - P^D_{S,A|S|_{\bar{S}}})\|_1 \|P^D_{S_t|_{\bar{S}}}\|_1 \tag{92}$$
$$\leq \|P_{S'|S,A}\|_1 \|P^O_{S,A|S|_{\bar{S}}} - P^D_{S,A|S|_{\bar{S}}}\|_1 \|P^D_{S_t|_{\bar{S}}}\|_1 \tag{93}$$
$$= \|P_{S'|S,A}\|_1 \|P^O_{A|S|_{\bar{S}}} - P^D_{A|S|_{\bar{S}}}\|_1 \|P^D_{S_t|_{\bar{S}}}\|_1 \qquad \text{(Eq. (88))} \tag{94}$$
$$\leq 1 \cdot \frac{4\varepsilon}{\nu} \cdot 1 \qquad \text{(Eq. (87), 1-norm of stochastic matrix/vector is 1).} \tag{95}$$

Coming towards the end, observe that plugging Eq. (70), (72) and (95) together gives

$$\Delta_{t+1} - \Delta_t \leq \frac{4\varepsilon}{\nu}.$$

Therefore

$$\Delta_t = \sum_{t' \leq t} \Delta_{t'+1} - \Delta_{t'} \leq t\frac{4\varepsilon}{\nu}.$$

Then (keep in mind that we assumed that the reward depends only on the state)

$$|v^O - v^D| = |\sum_t (P^O_{s_t} - P^D_{s_t}) \cdot \vec{c}| \tag{96}$$
$$\leq \sum_t \|(P^O_{S_t} - P^D_{S_t})\|_1 \|\vec{c}\|_\infty \tag{97}$$
$$= \|\vec{c}\|_\infty \sum_t \Delta_t \tag{98}$$
$$\leq \|\vec{c}\|_\infty \sum_t t\frac{4\varepsilon}{\nu} \tag{99}$$
$$\leq \|\vec{c}\|_\infty \frac{4\varepsilon}{\nu} T^2. \tag{100}$$

□

Remark 4. *Why do we have this "min" lower bound in Eq.* (6) *of Thm. 1, and not just a purely quadratic one? Note that in the above proof, if the case $\delta > \frac{1}{T}$ holds, then $\delta T > 1$, but the maximum deviation between imitator and demonstator cannot be larger than the imitator constantly being at a state different from the demonstartor with probability 1, i.e., can asymptotically not be larger than $T\|c\|_\infty$ (which is then smaller than $\delta T T\|c\|_\infty$ in this case).*

B Additional background on GAIL cost and training

We use (Wasserstein) GAIL, an established methodology, for imitation loss and training in our approach. The rough idea of this was already given in Sec. 4.2.

For the full account on details of (Wasserstein) GAIL, we refer the reader to the original work (Ho and Ermon, 2016; Xiao et al., 2019). Here let us nonetheless summarize some background on GAIL cost function and training, and comment on the parts relevant to our approach.

Imitation loss Let us elaborate on the GAIL-based (Ho and Ermon, 2016) cost function. We do it more generally here to also include Wasserstein-GAIL/GAN. The reason why we consider using the Wasserstein-version Xiao et al. (2019) is the following: Due to safety constraints, if the demonstrator is unsafe, we may deal with distributions of disjoint support. Then we want to encourage "geometric closeness" of distributions. But this is not possible with geometry-agnostic information-theoretic distance measures.

To specify the imitation cost c that we left abstract in Sec. 2 and 4.2, let

$$c(s,a) = e_2(s,a), \tag{101}$$

$$(e_1, e_2) = \arg\max \mathbb{E}_{\pi^D}(\sum_t e_1(s_t,a_t)) + \mathbb{E}_{\pi^I}(\sum_t e_2(s_t,a_t)) \tag{102}$$

where the arg max ranges over pairs (e_1, e_2) of bounded functions from some underlying space of bounded functions (Xiao et al., 2019), for which, besides Eq. (102), the following is required (Xiao et al., 2019):

$$e_1(x) + e_2(y) \leq \delta(x,y), \tag{103}$$

for some underlying distance δ. This condition is referred to as *Lipschitz regularity.* e_1, e_2 are called *Kantorovich potentials.*

As classic example, when taking $e_1(s,a) := \log(1 - D(s,a))$ and $e_2(s,a) := \log(D(s,a))$, for some function D, then this exactly amounts to the classic GAN/GAIL formulation with discriminator D.

Training Regarding the policy optimization step (b) sketched in Sec. 4.2, note that: For each sampled state s_t, $\pi^{I,\theta}(a|s_t)$ implicitly remembers the safe set $\tilde{A}_t^s$, such that also after any policy gradient step its mass will always remain within the safe set. Note that for this to work, it can be a problem if at some point we took a safe fallback action (Sec. 3.1) and thus only know a *singleton* safe set (and it is not a function of current state only). Since this is rare, it should pose little problems to training. Rigorously addressing this is left to future work.

C Additional details of experiments and implementation

Here we give some additional details for our experiments and implementation. These details are not necessary for the rough idea, but to get the detailed picture of these parts.

Remark: Code will be made available upon publication.

C.1 Additional architecture, computation and training information

Details of policy architectures:

- For the Gaussian policy (i.e., parameterizing mean and covariance matrix), we take two hidden layers each 1024 units.
- As conditional flow policy, we take a conditional flow with two hidden layers a 128 units.
- As discriminator, we take a NN with two hidden layers and 32 units each.
- As state feature CNN, we take a ResNet18 applied to the abstract birds-eye-view image representation of the state (Chai et al., 2019).

For the custom NNs, we use leaky ReLUs as non-linearities.

The data set (scene) consists of $\sim$ 1500 tracks (trajectories), of which we use 300 as test set and 100 as validation, and the rest as training set.

One full safe set computation with some limited vectorized parts takes around 1.5s on a standard CPU. We believe vectorizing/parallelizing more of the code can further reduce this substantially.

C.2 Regarding safety guarantees and momentary safety cost

Regarding the fact that Prop. 2 does not strictly cover FAGIL-E yet: We conjecture that, extending Prop. 2, a guarantee can be given for the l_∞ setting even without convexity over all dimensions, simply by the l_∞ norm being a minimum over dimensions and then in some way considering each dimension individually.

Regarding Lipschitz continuity of momentary safety cost: Note that, as briefly stated in the main text, the momentary safety cost d in our experiments is Lipschitz. Intuitively, it is some form of negative (minimum) *distance function* between ego and other vehicles. Distance/cost $d < 0$ is safe, while $d > 0$ is a collision. That is, d is a Lipschitz continuous function, but $d = 0$ is the boundary between the set of safe states and unsafe states. (Similar to the safe action set being the sub-zero (i.e., $w < 0$) set of the total safety cost w.) We believe that Lipschitz continuity is a fairly general assumption for safety, since safety is often considered in the physical world. And in the physical world, many of the underlying dependencies are in fact sufficiently continuous in a sense.

D Broader supplementary remarks

In this section, we gather several further discussions that are more broadly related to the results of the main part.

This section is not necessary to describe or understand the main part.

However, given that safe generative IL is a field that is rather little explored, we feel it can be particularly useful to provide some further comments here that may help understanding and may serve as a basis for next steps.

D.1 Further details on Sec. 3.1

Several broader remarks regarding safe set inference, which are not necessary to understand the core definitions and results:

- A-temporally, Eq. (4) can be seen as a Stackelberg game.
- In Eq. (4), the others' maxima may often collapse to open-loop trajectories.
- We use the general formulation where policies π_t can depend on t. This allows us to easily treat the case where we directly plan an *open*-loop action trajectory $a_{1:T}$ simultaneously, by simply $\pi_t(s_t) := a_t$, for all t, s_t.
- The state set corresponding to our action set would be an "invariant safe set" (Bansal et al., 2017).
- Lipschitz/continuity understanding is helpful also for other things: e.g., understanding the safe set's topology for Sec. 3.2.
- Our sample-based approach can also interpreted as follows: we can use *single-action safety checkers* (in the sense of the *"doer-checker"* approach (Koopman et al., 2019)) as oracles for action-set safety inference.
- On Prop. 2: Open-loop is a restriction but note that we do search over *all* trajectories in Π, Φ, so it is similar to trajectory-based planning.
- *Regarding inner approximations of the safe set:* For as little as possible bias and conservativity, we need the inner approximation of the safe set to be as large as possible. In principle, one could also just give a finite set of points as inner approximation, but this could be very unflexible and biased, and injectivity for change-of-variables Prop. 3 would not be possible.
- *Regarding safety between discrete time steps:* Note that our setup is discrete time, but it can directly also imply safety guarantees between time stages: e.g., for our experiments, we know maximum

velocity of the agents, so we know how much they can move at most between time stages, and we can add this as a margin to the distance-based safety cost.

- *Small set of candidate ego policies, to simplify calculations:* As an important simplification which preserves guarantees, in the definition/evaluation of safe set and total safety cost $w_t(s, a)$, we let the ego policies range over some small, sometimes finite set $\bar{\Pi}_{t+1:T} \subset \Pi_{t+1:T}$ of reasonable *fallback* future continuations of ego policies. In the case of autonomous driving (see experiments), $\bar{\Pi}_{t+1:T}$ consists of *emergency brake* and *evasive maneuvers* (additional details are in Appendix D.1).[20]

- A popular approach for trajectory optimization is *model predictive control (MPC)* Bertsekas (2012), so let us briefly comment on the relation between our trajectory calculations for safety and MPC: In a sense, MPC could solve the trajectory optimization, i.e., the "min part" of Eq. (4). So, to some extent, it is related. However, MPC would need one specific policy/dynamics for the other agents, while we just have a set of possible other agents policies (and take the worst-case/adversarial case over this set). And MPC alone would also not give us a set of safe actions (which we need since we want to allow as much support for the generative policy density as possible), but only one action/trajectory.

D.2 Further details on Sec. 3.2

Here we go more into detail on the construction of safety layer that are differentiable and give us an overall density, in particular (1) the challenges and (2) potential existing tools to build on. (See also Appendix D.3.2.)

One purpose of this section is to provide further relevant background for future work on safety layers that allow to have closed-form density/gradient – given this is a rather new yet highly non-trivial topic (from the mathematics as well as tractability side).

D.2.1 Definition and remark

Let us introduce the topological notion of "connected" we use below.

Roughly, an open set in $\mathbb{R}^n$ is called *(path-) connected*, if any two of its elements can be connected by a continuous path within the set; and *simply connected* if these connecting paths are, up to continuous deformations within the set, unique.

For formal definitions we refer to textbooks, such as (Komornik, 2017).

Remark on Def. 1: Usually, parts A_k will be topologically simply connected sets.

D.2.2 Basic remarks on simplest safety layer approaches

Let us elaborate on the issue of designing safety layers, mappings $A \to \bar{A}$, in particular ones that are differentiable and give us an overall density.

Besides the more advanced approaches we discuss in the main text and below, let us also mention the two simplest ones and their limitations.

First, maybe the simplest safetly layer is to map all unsafe actions, i.e., $A \setminus \bar{A}$, to *one* safe action, i.e., *one point*. But this would lead to stronger biases and we would lose the property of having an overall density (at least the change of variables would not be applicable due to non-injectivity; and the policy gradient theorem would also be problematic).

[20] Regarding the ego, note that the search over a small set of future trajectories in the form of the *"fallback maneuvers"* (instead of an exhaustive search/optimization over the full planning/trajecotry space) can be seen as a simple form of motion primitives or graph search methods and related to "sampling-based" planning approaches (Paden et al., 2016). Regarding the other agents/perturbations, one could also just consider a small subset of possible behaviors, though we do not do this in the current work since it weakens guarantees of course. This would be somewhat related to sampling-based methods in continuous game theory (Adam et al., 2021) as well as empirical/oracle-based game theory (Lanctot et al., 2017).

Second, another simple solutions would be move/scale the whole A into *one safe open set*, given the safe set contains at least one open set (which should generally be the case, e.g., using Prop. 1). This in fact would be a diffeomorphism so we could apply the change-of-variables formula. However, obviously this would be very rigid: the possible safe actions to take would then usually be a very small part of the full safe set – a strong and usually poor bias. This is because, be aware that, also the actually safe parts of the action space would be re-mapped to the tiny safe open set. That is, it would force us to also move actions that are actually safe to new positions. And this would leave little space of designing more sensible inductive biases, as in the two instances we propose.

D.2.3 Simple versatility of piecewise diffeomorphism safety layers

In contrast to these overly biased simplest solutions from Appendix D.2.2, let us give an example of how piecewise diffeomorphisms give us more room for design. Note that this is not meant as a safetly layer that should actually be used, but rather as a simple argument to show the universality:

Remark 5. *Piecewise diffeomorphic safety layers are versatile: If a given safe set $\bar{A}$ contains at least some area (open set), then we can usually map A into $\bar{A}$, simply by moving and scaling the unsafe areas into safe ones, while leaving the safe actions where they are.*

Elaboration of Rem. 5. Assume the safe set $\bar{A}$ is given as the subzero set of some continuous function $h : A \to \mathbb{R}$, i.e., $\bar{A} = h^{-1}((-\infty, 0])$, which is the case, e.g., under Prop. 1, and that it contains at least some open set. Let B be some (any) ball in this open set.

Here is the simplest construction: Just take $A_1 := h^{-1}((-\infty, 0])$ and $A_2 := h^{-1}((0, \infty))$. Let g_1 (on A_1) be the identity. And let g_2 (on A_2) be a shrinking, such that A_2 has less diameter than B, and a subsequent translation of A_2 into B.

Another construction, which has a better "topological" bias in that it tries to map unsafe parts into *nearby* safe parts works as follows:

If $\bar{A}$ is the subzero set of a continuous function, then the unsafe set is open. So it can be written as the union of its connected components, and each connected component is also open. Therefore, in each connected component k, there is at least one rational number (or, in higher dimensions, the analogous element of $\mathbb{Q}^n$), so there are at most countably many of them.

So define g_k with domain the component k, and simply being a translation and scaling that moves this component k into a *nearby* open part of the safe set (this is where it becomes more appealing in terms "topological" bias), or, as a default, into B.

Let g be defined by the "union" of these g_k, and on the rest of A, i.e., on the safe set $\bar{A}$, let it simply be the identity.

□

D.2.4 On general existence, construction/biases and tractability of closed-form-density-preserving differentiable safety layer

Here we add some notes on the general discussion of safety layers with closed-form densities, beyond the particular approach we take in this paper.

There could be many ways to construct safety layers with closed-form density, and it is hard, especially within the scope of just this paper, to get general answers on questions of (1) possibility and (2) tractable construction of such safety layers. Especially since there may be trivial solutions (see Appendix D.2.2) which are not satisfactory in terms of rigidity/bias though. So let us narrow down the problem to try to make at least some basic assertions.

First, a *simple example of impossibility when using pure diffeomorphisms*: already for topological reasons, if A is connected but $\tilde{A}$ not, then no such safety layer can exist that is bijective onto $\tilde{A}$.

Nonetheless, overall the starting point in this paper is the change of variables, or the piecewise change of variables we use. This implies that we extensively need diffeomorphisms to build our safety layer on. And this implies that we need homeomorphisms, when relaxing the differentiability to continuity.

Regarding the shapes of (un-)safe sets, or their (inner/outer) approximations, general answers would be desirable too, but particularly here we focus on polytopes and hyperrectangles, since for those are also typical results of reachability analysis etc. (Sidrane et al., 2022).

Here are some basic concepts and results from topology and related areas. We only give an idea, while detailed formalities are beyond the scope of this paper.

- In $\mathbb{R}^2$ there is a lot of relevant theory from complex analysis, which studies holomorphic, i.e., complex differentiable functions (another way to state it is that they are conformal, i.e., locally angle-preserving mappings; for detailed formalities see, e.g., (Luteberget, 2010)). Biholomorphic (i.e., a holomorphic bijection whose inverse is also holomorphic) implies diffeomorphic, so this is relevant for us.

  For instance, the *Riemann mapping theorem* (Luteberget, 2010) tells us that such a biholomorphic mapping exists from $A \subset \mathbb{R}^2$ to $\bar{A}$, *if* $\bar{A}$ *is a simply connected open set in* $\mathbb{R}^2$ (assuming per se that A is a box, i.e., simply connected as well). While this could be a powerful result for safety layers, the *issue comes with tractability.* Often, Riemann mappings are defined just implicitly, based on some variational problem, or some differential equation problem, whose solution can be arbitrarily complex to calculate. More tractable (though it still remains unclear how to make them fully tractable) concepts and results (Luteberget, 2010) include:

  - Schwarz-Christoffel mappings – a special case for polytope domains;
  - sphere packing-based algorithms for approximating Riemann mappings via combinatorical solutions on a finite graph of the problem, and then extrapolation to the continuous original problem.
  - Note that it is easy to see (since holomorphic functions are globally uniquely determined from any local neighborhood, based on them being fully described by their Taylor expansion around any single point) that holomorphic functions are a quite rigid function class for safety layer though: for instance, there cannot be a holomorphic function which would be the identity on the safe set, but the non-indentity outside the safe set.

- Adding on the negative, i.e., non-existence side, often the non-existence of safety layers or parts of them can be proved using topological arguments (often invariants), e.g., that the domain and co-domain do have to have the same connectivity property (connected, simply conntected, etc.), or other based on homology.

- There are also positive, i.e., existence results, from topology, and very general ones. For instance, if two open subsets of $\mathbb{R}^n$ are contractible and simply connected at infinity, then they are homeomorphic (i.e., continuous bijections whose inverse is also continuous) to each other.[21] This, even though not sufficient, is a strong necessary requirement for being diffeomorphic of course. Also convexity can be helpful.[22]

D.3 Details on FAGIL method Sec. 4, in particular its safety layers

D.3.1 Regarding safe set inference module

Note that, in principle, due to modularity, many safety approaches can be plugged as safety modules into our overall method (Fig. 1). This includes frameworks based on set fixed points (Vidal et al., 2000), Lyapunov functions (Donti et al., 2021a), Hamilton-Jacobi type equations (continuous time) (Bansal et al., 2017), or Responsibility-Sensitive Safety (RSS) Shalev-Shwartz et al. (2017).

[21] See `https://math.stackexchange.com/q/55114/1060605`.

[22] See `https://math.stackexchange.com/q/165629/1060605` and `http://relaunch.hcm.uni-bonn.de/fileadmin/geschke/papers/ConvexOpen.pdf`.

A fundamental problem in safe control is the trade-off between being safe (conservative) versus allowing for enough freedom to be able to "move at all" (actionability). Ours and many safe control frameworks build on more conservative adversarial/worst-case reasoning. RSS's idea is to prove safety under the less conservative (i.e., "less worst-case") assumption of others sticking to (traffic) *conventions*, and if collisions happen nonetheless, then at least the ego is "not to blame".

D.3.2 An alternative, distance-based grid-based safety layer

Here is another instance of a piecewise diffeomorphism safety layer g, based on the inductive bias of mapping unsafe actions to *nearby* safe ones, which we refer to as our *distance-based safety layer instance*:

(a) On all safe parts k, i.e., $A_k \subset \bar{A}$, g_k is the identity, i.e., we just keep the pre-safe proposal $\hat{a}$. (b) On all unsafe parts k, g_k is the translation and scaling to the most nearby rectangle in Euclidean distance.

D.3.3 Illustrative examples of piecewise diffeomorphisms for safety layers

Let us give some illustrative examples of piecewise diffeomorphism safety layers, see Fig. 5.

This includes an example of our *distance-based, grid-based safety layer* described in Appendix D.3.2. (The probability-based, grid-based version from Sec. 4 works in a related way, but choosing the targets A_ℓ based on their pre-safe probability, instead of their distance to the sources A_k.)

Note that in the grid-based example, here we just need translations, because all parts (rectangles) of the partition are of the same size, while in other cases, some parts (e.g., if there is a boundary not in line with the grid) may have different sizes, and then we also need scaling.

D.3.4 Computational complexity of the safety layer

We here give a rough analysis of the computation complexity of piecewise diffeomorphisms (Sec. 3.2) and thus the safety layer, as part of the fail-safe imitator policy $\pi^{I,\theta}(\bar{a}|s)$ (Sec. 4), and their gradients. Let us look at Eq. (5) and recall the structure of the fail-safe imitator with safe action and pre-safe action from Fig. 1.

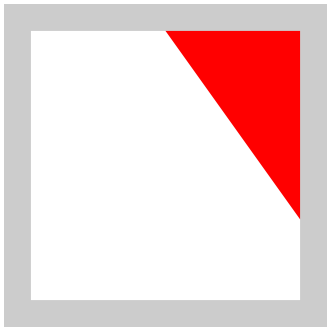
To calculate safe density $p_{\bar{a}}(\bar{a})$ of a safe action $\bar{a}$, given the pre-safe density $p_{\hat{a}}$ of the pre-safe policy, this is a sum (Eq. (5)) over the relevant parts A_k of the partition $(A_k)_k$; and then each k-th summand is the change-of-variables formula applied to the diffeomorphism g_k and pre-safe density $p_{\hat{a}}$, where the determination of g_k may itself require some calculation (based, e.g., on distances or pre-safe densities of the parts). So, essentially, the complexity is at most the number of parts in the partition $(A_k)_k$ times the complexity of calculating the change-of-variables formula (and determination of g_k if necessary, see instances below).

Let us give some instances of this:

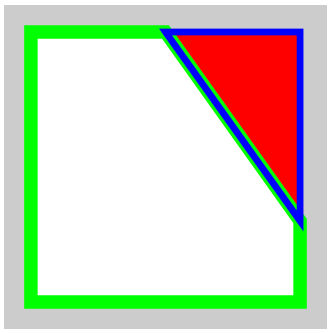
- When using any sort of general invertible (normalizing flow) neural net (Papamakarios et al., 2019) as transformations g_k, then we inherit their complexity for each summand of Eq. (5).
- For our proposed safety layers, e.g., the grid- and distance-based one we describe in Appendix D.3 the number of parts in the partition is around 100 (based on a 10 times 10 grid). And each g_k is a translation/scaling to a target part, yielding a simple scaling factor as change-of-variables formula (the determination of the target part, i.e., of g_k, requires, for each state, one single run over all pairs of parts in the partition to get the distances; on this side, the probability-based safety layer version we describe in Sec. 4 is faster, since we do not have to go over all pairs, just all single ones).

To calculate the *gradient* of $\pi^{I,\theta}(\bar{a}|s)$ w.r.t. θ, this is analogous, where we simply can drag the gradient into the sum of Eq. (5), and the gradients of the summands are given by the gradients of the pre-safe policy that is being used (i.e., for instance conditional Gaussian or normalizing flow), times the factor $|\det(J_{g_k^{-1}}(\hat{a}))|$ which is constant w.r.t. θ (in the our distance-based safety layer). So here the complexity is essentially number of (at most) all parts in the partition $(A_k)_k$, times complexity of gradient calculation for pre-safe policy.

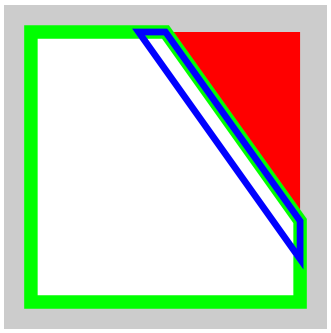
(a) **Setup:** Assume the full action set A is given by the set enclosed by the gray frame. ...



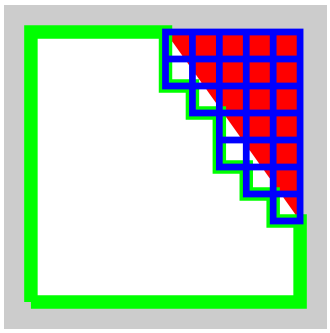
(b) ... And the unsafe actions $A \setminus \bar{A}$ are given by the red subset of the full action set.



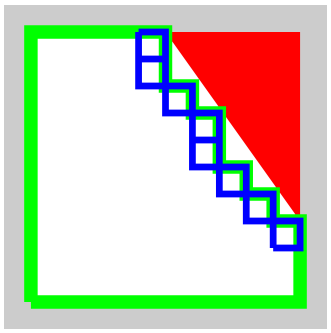
(c) **First example piecewise diffeomorphism:** The partition $(A_k)_k$ has just two elements: A_1 is the safe set enclosed by the green line, and A_2 the unsafe set, encircled by the blue line. ...



(d) ... The diffeomorphism g_1 on A_1 is just the identity. And the diffeomorphism g_2 maps the unsafe A_2 to the blue polygon at the boundary within the safe set. Note that such a diffeomorphism exists by the Riemann (or Schwarz-Christoffel) mapping theorem, or, in this simple case, also by convexity (see our discussion of all these concepts in Appendix D.2.4). Together, g_1 and g_2 form our piecewise diffeomorphism g on $A = A_1 \cup A_2$.



(e) **Second example piecewise diffeomorphism:** Here, we consider our grid-based partition of the full action set, i.e., a complete regular tiling of the action set with rectangles A_k, as partition $(A_k)_k$. We only illustrated the individual rectangles A_k for the unsafe area, in blue, while for the safe area we only drew the outer boundary, in green.



(f) ... On all safe A_k, the diffeomorphism g_k is simply the identity. And on all unsafe A_k (meaning fully or partially unsafe), g_k is the shifting to the most nearby safe rectangle A_ℓ. Those safe target rectangles A_ℓ are drawn in blue. Together, these g_k form our piecewise diffeomorphism g on the full A. Note that this is an example of our *distance-based safety layer* described in Appendix D.3.2.

Figure 5: Examples of piecewise diffeomorphism safety layers. Top row: setup. Middle and bottom row: two examples.

D.3.5 Invariant safety on full sets, not just singletons

Recall that for the general method we used singleton safe sets as backup to guarantee invariant safety.

It is important to emphasize though that in stable linear systems it can in fact often be guaranteed that always a *full safe box* $\bar{A}_k$ *is found*. And then the extension of memorizing a safe fallback singleton is not necessary.

The idea towards proving this is as follows: The hard part is that we have to check safe future trajecotries not just for *individual* trajectories, but where at each stage, any action out of thes safe *set* can be taken. But these sets can be modelled as *bounded perturbations* of individual actions. Then stability arguments can show us that the error (deviation from a single safe trajectory over time) can be bounded, and then we can recur to planning individual trajectories plus simple margins.

D.4 Further related work and remarks

More broadly related is also the following work:

Boborzi et al. (2021) similar to us propose a safe version of GAIL, but there are several differences: they build on a different safety reasoning (RSS), which is specific for the scope of driving tasks, while our full method with its worst-case safety reasoning is applicable to general IL domains, and they do not provide exact densities/gradients of the full safe policy as we do.

Certain safety layers (Donti et al., 2021a; Dalal et al., 2018) are a special case of so-called *implicit layers* (Amos and Kolter, 2017; Amos et al., 2018; El Ghaoui et al., 2019; Bai et al., 2019; Ling et al., 2018; 2019), which have recently been applied also for game-theoretic multi-agent trajectory model learning (Geiger and Straehle, 2021), including highway driver modeling; robust learning updates in deep RL (Otto et al., 2021); and learning interventions that optimize certain equilibria in complex systems Besserve and Schölkopf (2021).

Worth mentioning is also work on safe IL that builds on hierarchical/hybrid approaches that combine imitation learning with classic safe planning/trajectory tracking (Chen et al., 2019; Huang et al., 2019).

Besides GAIL, which only requires samples of rollouts, one could also think about using fully known and differentiable dynamics (Baram et al., 2017) for the training of our method.

Also note reachability-based safe learning with Gaussian processes (Akametalu et al., 2014).

At the intersection of safety/reachability and learning, also work on outer approximations based on verified neural net function class properties is connected Sidrane et al. (2022).

Beyond the paper mentioned in the main text, there are several papers on fail-safe planning for autonomous driving (Magdici and Althoff, 2016; Pek and Althoff, 2018; 2020).

Convex sets, various kinds of them, have proved a helpful concept in reachability analysis and calculation, see., e.g, (Guernic and Girard, 2009) (their Section 1 gives an overview).

On the meaning of the term "fail-safe" The term "fail-safe" seems to be used in varying ways in the literature. In this work, in line with common uses, we have the following intuitive meaning in mind: We have an imitation learning component and a worst-case planning component in our system (the fail-safe imitator). The *pure IL component* – the pre-safe policy – can in principle always *fail* in the sense of leading to "failure", i.e., momentarily unsafe/collision, states, since it does not have strong guarantees.[23] Specifically, this would happen when it would put all mass on an action that leads to a collision.[24] But having our *worst-case fallback planner plus the safety layer* as additional component in our system, we make sure that, even if the IL component fails, nonetheless there exists at least one safe action plus future trajectory. This makes the system *safe against a failure of the pure IL component*, and this is what we mean by *fail-safe*. This is similar

[23]There can also be some confusion because "fail" can be used in two senses: in the sense of a restricted failure, that can be mitigated and still be safe though; or in the sense that "failure" means "collision" or "momentarily unsafe" state.

[24]Note that for Gauss as well as flow policies, we have the conceptual issue here that they have mass everywhere; and in this probabilistic setting it is hard to say what precisely "fail" would be; however, one can as well plug in pre-safe policies into our approach that do not have support everywhere, and then the aforementioned holds rigorously.

to (Magdici and Althoff, 2016) who use "fail-safe" in the sense that if other agents behave other than their "normal" prediction module predicts (i.e., the prediction module fails in this sense), then there is nonetheless a safe fallback plan. All this also relates to notions of *redundancy* etc.