TITRATED: Learned Human Driving Behavior without Infractions via Amortized Inference

Vasileios Lioutas[1,2] **Adam Ścibior**[1,2] **Frank Wood**[1,2,3]
[1]Inverted AI [2]University of British Columbia [3]Mila
{firstname.lastname}@inverted.ai

The ability to accurately predict the motion of nearby agents such as vehicles and pedestrians is of paramount importance for achieving fully autonomous driving [6, 10, 7, 5, 13, 12]. Apart from the use in on-board prediction systems facilitating planning in self-driving cars, learned models of human driving behaviors are now also being used to create realistic non-playable characters (NPCs) for simulators in which self-driving cars can be tested and trained [9, 1]. In this setting, it is particularly problematic if the models predict behaviors that result in serious infractions, such as colliding with other agents or going off-road too often, as the resulting simulations would be drastically unrealistic. The problem of excess infractions can be to some extent mitigated by introducing additional penalties for infractions at behavior model training time, which are referred to as “common sense losses” in [9].

Like all machine learning models, the ones predicting human driving behavior suffer from degraded performance under domain shift, which in this case, in particular, occurs when the models are deployed in locations not covered in the training dataset. This includes increased infraction rates, which is a major obstacle when creating simulations with NPCs learned from data collected on different roads. In this paper, we present a general algorithm called TITRATED (Training ITRA to Emulate Desiderate) for fine-tuning behavior prediction models to novel settings where human demonstrations are not available.

We use ITRA [8] as the model of choice for predicting driving behaviors, but any other probabilistic behavior prediction model can be used with the methodology presented in this paper. While the high level algorithm for performing rejection sampling and amortized inference is model-agnostic, we focus the presentation on ITRA for concreteness.

Each agent in ITRA is modeled as a conditional variational recurrent neural network (VRNN) [2], with latent variables z_t^i. The joint predictive distribution of ITRA factorizes as

$$p(s_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{N_t} \int \int p(z_t^i) p(a_t^i | \mathcal{I}_t^i, z_t^i, h_{t-1}^i) p(s_t^i | s_{t-1}^i, a_t^i) \; dz_{1:T}^{1:N_t} \; da_{1:T}^{1:N_t}$$

where s_t^i and a_t^i are the state and action of the individual agent i at time t, s_t is the full state of the world at time t, $\mathcal{I}_t^i$ is the observation of s_t available to agent i in the form of a birdview representation, and h_t^i is the recurrent state. The transition to the next state $s_t^i = kin(s_{t-1}^i, a_t^i)$ is produced by a kinematic bicycle model. The model is trained as usual, jointly with a separate inference network, by maximizing the evidence lower bound (ELBO).

We focus on addressing infractions that we never want to see in predictions, specifically, collisions and off-road invasions. Figure 1 illustrates the aim and results of our work. For each agent i and time t, the current state s_t^i and the fixed dimensions (l^i, w^i) define a bounding box $C_t^i \in \mathcal{R}^{4\times 2}$, represented as four corners of the vehicle. We define our collision metric as the sum of individual Intersection-over-Union (IoU) values across time

$$\mathcal{L}_{\text{C}}(C_{1:T}) = \sum_{t=1}^{T} \sum_{j \neq i}^{N_t} \text{IoU}(C_t^i, C_t^j).$$

Machine Learning for Autonomous Driving Workshop at the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

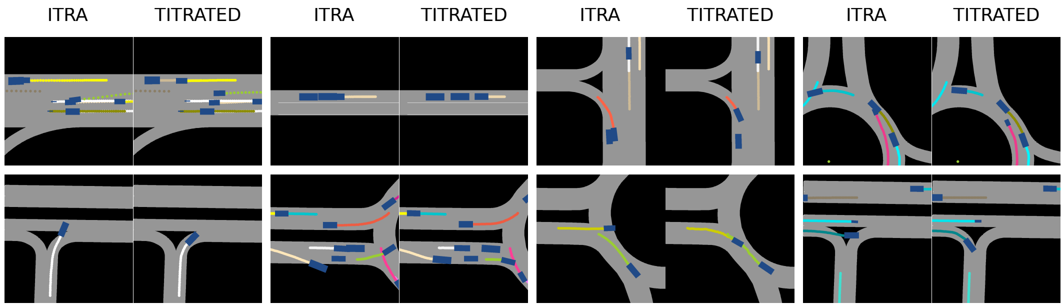

Figure 1: Examples from finetuning ITRA to the novel setting of CARLA. On its own ITRA produces unacceptable levels of infractions on CARLA maps, both in terms of collisions (top row) and off-road-driving (bottom row). Our method, TITRATED, substantially reduces both forms of infractions. Colored lines indicate the path moving vehicles took to get where they are.

using a differentiable version of the IoU metric as described in [14].

For off-road infractions, we assume that we have access to a triangle mesh V defining the driveable area (road surfaces, etc.). Given a function Φ that computes the distance from a point to a triangle, we define the off-road infraction metric as the sum of distances of the corners of all vehicles from the road mesh across time

$$\mathcal{L}_{\text{OR}}(C_{1:T}, V) = \sum_{t=1}^{T} \sum_{i=1}^{N_t} \sum_{c_j \in C_t^i} \min_{v \in V} \Phi(c_j, v).$$

This metric can be computed efficiently in a differentiable manner [4]. Note that it is zero only when all four corners of all vehicles are contained within the driveable area.

We define A to be the event that no vehicle performed an infraction within the specified time window

$$A := (\mathcal{L}_{\text{C}} = 0) \wedge (\mathcal{L}_{\text{OR}} = 0).$$

and obtain a better behavior model by further conditioning on not performing infractions, denoted here as the conditional density $p(s_{1:T}|A)$.

Unfortunately, sampling from the conditional distribution $p(s_{1:T}|A)$ can be computationally expensive. For this reason, we learn a model $p_\theta(s_{1:T}) \approx p(s_{1:T}|A)$, which can be sampled from sequentially in fixed time. To do so, we generate examples of infraction-free trajectories in an off-line setting as shown in Algorithm 1 given a collection D of initial conditions on target maps.

Algorithm 1 Infraction-Free Dataset Generation

Input: Initial conditions dataset D
Driving behaviour model $p(s_{1:T})$
Maximum number of trials max_trials
Output: Infraction-Free dataset $\tilde{D}$

```
D̃ ← ∅
for (s_0, V) ∈ D do
    found ← false
    for n ← 1 to max_trials do
        Sample rollout s_{1:T} from p(s_{1:T}|s_0)
        Convert states s_{1:T} to bounding boxes C_{1:T}
        if L_C(C_{1:T}) = 0 and L_OR(C_{1:T}, V) = 0 then
            found ← true
            break
    if found then
        D̃ ← D̃ ∪ {(s_{1:T}, V)}
return D̃
```

This procedure is simple but can be computationally expensive, since rejection sampling can take arbitrarily long time to produce acceptable samples. In order to limit computational cost, we introduce

Scene	Collision Rate $\times 10^{-4}$			Off-road Rate $\times 10^{-4}$		
	ITRA	TITRATED	Reduction	ITRA	TITRATED	Reduction
Town01_Straight	26.0	2.0	91.97%	42.7	0	100%
Town01_3way	19.4	4.8	75.06%	11.3	0	100%
Town02_Straight	11.2	2.8	80.53%	51.5	1.6	96.74%
Town02_3way	10.0	3.3	66.20%	33.9	0	100%
Town03_Roundabout	5.0	1.7	65.60%	134.0	47.1	64.83%
Town03_5way	4.6	1.6	63.26%	95.8	37.5	60.78%
Town03_4way	6.0	2.5	58.33%	67.2	34.2	49.07%
Town03_3way_Unprotected	6.3	2.7	56.82%	74.8	42.3	43.42%
Town03_3way_Protected	10.4	4.9	52.50%	84.9	41.5	51.14%
Town03_GasStation	6.6	0.6	90.54%	32.8	13.2	59.69%
Town04_Merging	0.2	0.1	13.63%	47.8	0	100%
Town04_3way_Large	8.0	2.3	71.25%	11.2	0	100%
Town04_3way_Small	16.6	6.4	61.20%	29.6	0.8	97.14%
Town04_4way_Stop	10.9	4.7	56.60%	32.4	11.6	64.12%
Town04_Parking	6.5	4.7	26.76%	44.7	5.3	87.98%
Town06_Merge_Single	1.7	1.1	32.94%	109.6	27.6	74.83%
Town06_4way_large	1.7	0.3	79.70%	7.3	1.5	79.21%
Town06_Merge_Double	1.8	0.5	71.72%	2.3	0	100%
Town07_3way	6.9	2.2	66.95%	45.3	3.1	93.14%
Town07_4way	7.8	2.3	70.51%	60.6	3.1	94.80%
Town10HD_4way	3.6	1.7	51.11%	51.1	21.1	58.76%
Town10HD_3way_Protected	4.5	1.6	63.33%	76.2	26.3	65.52%
Town10HD_3way_Stop	3.6	1.9	45.00%	62.5	17.2	72.43%
Average	7.8	2.4	68.20%	52.6	14.5	72.26%

Table 1: Rates of driving infractions in various CARLA scenes, with (TITRATED) and without (ITRA) the fine-tuning proposed in this paper.

a max_trials parameter, which indicates the maximum number of sampling attempts performed, after which the item is excluded from the dataset if an acceptable sample was not found.

Using the synthetic dataset of infraction-free trajectories, we amortize the inference process $\mathbb{E}_{s_{1:T} \sim p(s_{1:T}|A)} \left[\log p_\theta(s_{1:T}) \right]$ by optimizing the ELBO objective, following the standard training process of ITRA

$$\mathcal{L}_{\text{ELBO}} = \mathop{\mathbb{E}}_{s_{1:T} \sim p(s_{1:T}|A)} \Bigg[\sum_{t=1}^{T-1} \sum_{i=1}^{N_t} \Bigg(\mathop{\mathbb{E}}_{q_\phi(z_t^i|a_t^i, \mathcal{I}_t^i, h_t^i)} \left[\log p_\theta(s_{t+1}^i | \mathcal{I}_t^i, z_t^i, h_t^i) \right] \\ - KL \left[q_\phi(z_t^i | a_t^i, \mathcal{I}_t^i, h_t^i) || p_\theta(z_t^i) \right] \Bigg) \Bigg]$$

where q_ϕ is a separate inference network trained jointly with the model p_θ. Learned models of human driving behavior, such as ITRA, are prone to generate excessive infractions. We ameliorate this problem by introducing explicit infraction penalties obtained when sampling from the amortized model p_θ, as additional loss terms

$$\mathcal{L} = -\mathcal{L}_{\text{ELBO}} + \lambda_{\text{C}} \mathcal{L}_{\text{C}} + \lambda_{\text{OR}} \mathcal{L}_{\text{OR}}.$$

This is the final training objective for TITRATED, although some care needs to be taken when minimizing it, since $\mathcal{L}_{\text{ELBO}}$ involves sampling from the inference network, while $\mathcal{L}_{\text{C}}$ and $\mathcal{L}_{\text{OR}}$ involve sampling from the prior. We address this by performing two separate rollouts for each training example.

In our experiments, we, at a high level, learn human driving behaviors from the INTERACTION dataset [11] and use them to create NPCs in CARLA [3]. We take a trained ITRA model obtained exactly as described in [8] and fine tune it to obtain TITRATED models for a collection of selected locations in CARLA. Our goal is to maintain maximum similarity to ITRA predictions while reducing the incidence of driving infractions, consisting of collisions and off-road invasions, in these novel

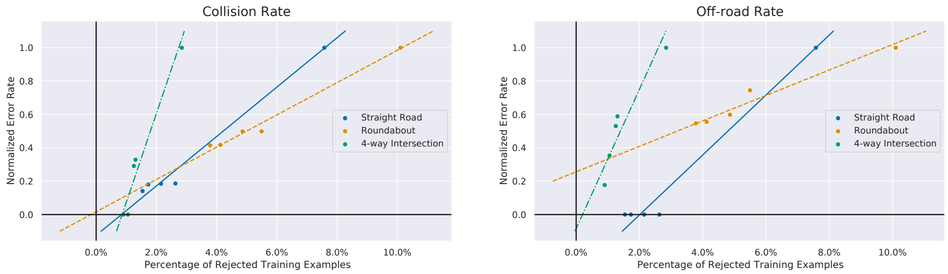

Figure 2: Scatter plots of infraction rates achieved by TITRATED versus the number of items rejected from the synthetic dataset. Different points correspond to max_trials values $\in \{1, 10, 20, 50, 100\}$ used in rejection sampling, increasing to the left. Straight lines are fit with least squares to extrapolate performance to a setting with perfect inference algorithms where no items are rejected.

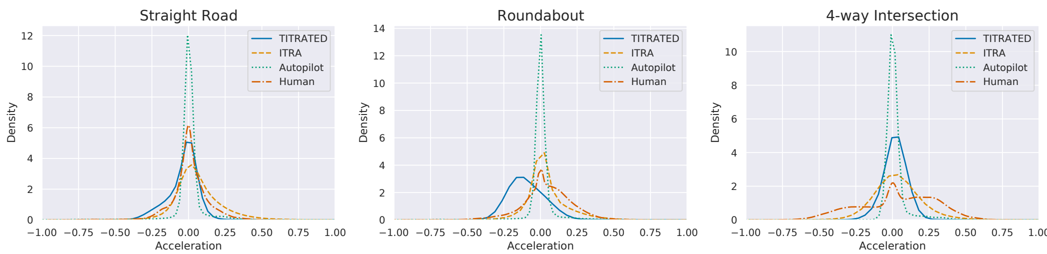

Figure 3: Kernel density estimates of acceleration distributions for TITRATED, ITRA, CARLA autopilot, and real human data. Acceleration values are normalized using highest values present in human data.

simulator contexts. We created a custom synthetic dataset of initial conditions using the built-in autopilot in CARLA. Table 1 shows that TITRATED is able to significantly reduce the collision and off-road rates of ITRA. Specifically, we first generate an ITRA rollout for each initial condition in the corresponding CARLA dataset and compute infraction rates. We do the same for TITRATED rollouts. We see that, averaged across all CARLA locations, TITRATED is able to reduce collisions by 68% and off-road invasions by 72%.

The key component of TITRATED is the Bayesian inference algorithm for computing and sampling from the conditional distribution $p(s_{1:T}|A)$. We investigated how using increasingly more powerful inference algorithms could impact the performance of TITRATED, trying to extrapolate to the setting of a perfect inference oracle that is able to produce samples with no infractions in all cases. We selected three representative locations from the CARLA dataset, covering a straight road, a roundabout, and a 4-way intersection. For each of those locations, we ran the full TITRATED training procedure with varying values of $max\_trials \in \{1, 10, 20, 50, 100\}$. To analyze the scaling behavior as the inference algorithm improves, we plotted the infraction rate of TITRATED against the fraction of rejected examples in Figure 2. Somewhat surprisingly, this scaling tends to be close to linear, allowing us to extrapolate to a setting with a perfect inference algorithm that finds acceptable samples for all training examples. In most, but not all cases, the extrapolated line achieves zero infraction rate before achieving full dataset coverage, suggesting that the use of better inference algorithms could remove infractions entirely.

In Figure 3, we present the probability densities of vehicle acceleration. The primary motivation for this work was to create NPCs for CARLA exhibiting human-like driving behavior. While this is largely subjective and no definitive metrics are agreed-upon by the community, in this experiment we seek to demonstrate that in some sense, TITRATED indeed generates more human-like driving behavior than the CARLA autopilot. In all three cases, CARLA autopilot has a highly peaked acceleration distribution around zero, while the distribution of human acceleration values is more spread, corresponding to a higher diversity of controls applied by humans. We see that both ITRA and TITRATED are more realistic than the CARLA autopilot, in the sense of the distribution of accelerations being more spread.

References

[1] Luca Bergamini, Yawei Ye, Oliver Scheel, Long Chen, Chih Hu, Luca Del Pero, Błażej Osiński, Hugo Grimmet, and Peter Ondruska. SimNet: Learning Reactive Self-driving Simulations from Real-world Observations. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages –. IEEE, 2021.

[2] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A Recurrent Latent Variable Model for Sequential Data. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2980–2988, Cambridge, MA, USA, 2015. MIT Press.

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[4] D. Eberly. Distance Between Point and Triangle in 3D. In *Geometric Tools*, 1999.

[5] Joey Hong, Benjamin Sapp, and James Philbin. Rules of the Road: Predicting Driving Behavior with a Convolutional Model of Semantic Interactions. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. doi: 10.1109/cvpr.2019.00865.

[6] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, and Manmohan Chandraker. DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. doi: 10.1109/cvpr.2017.233.

[7] Nicholas Rhinehart, Rowan Mcallister, Kris Kitani, and Sergey Levine. PRECOG: PREdiction Conditioned On Goals in Visual Multi-Agent Settings. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. doi: 10.1109/iccv.2019.00291.

[8] Adam Ścibior, Vasileios Lioutas, Daniele Reda, Peyman Bateni, and Frank Wood. Imagining The Road Ahead: Multi-Agent Trajectory Prediction via Differentiable Simulation. In *2021 IEEE 24rd International Conference on Intelligent Transportation Systems (ITSC)*, 2021.

[9] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. TrafficSim: Learning To Simulate Realistic Multi-Agent Behaviors. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 10400–10409. Computer Vision Foundation / IEEE, 2021.

[10] Charlie Tang and Russ R Salakhutdinov. Multiple Futures Prediction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[11] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kümmerle, Hendrik Königshof, Christoph Stiller, Arnaud de La Fortelle, and Masayoshi Tomizuka. INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps. *arXiv:1910.03088 [cs, eess]*, 2019.

[12] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, Congcong Li, and Dragomir Anguelov. TNT: Target-driveN Trajectory Prediction. *arXiv:2008.08294 [cs]*, page 12, August 2020. URL `http://arxiv.org/abs/2008.08294`. arXiv: 2008.08294.

[13] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-Agent Tensor Fusion for Contextual Trajectory Prediction. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. doi: 10.1109/cvpr.2019.01240.

[14] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. *2019 International Conference on 3D Vision (3DV)*, Sep 2019. doi: 10.1109/3dv.2019.00019. URL `http://dx.doi.org/10.1109/3DV.2019.00019`.