Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients

Oliver Scheel, Luca Bergamini, Maciej Wołczyk, Błażej Osiński, Peter Ondruska
Woven Planet, Level 5
{firstname.lastname}@woven-planet.global

Appendix A: Qualitative results

Figure 1 shows our method handling diverse, complex traffic situations well - it is identical to Figure 4 of the paper, but enlarged. For more qualitative results we refer to the supplementary video.

In-car testing

In this section we report additional results of deploying our trained policy to SDVs. Figure 2 shows our planner navigating through a multitude of challenging scenarios. For more results we refer to the supplementary video - where she show additional results in the form of videos, which also contain more information, namely different camera angles, the resulting scene understanding and planned trajectory of the SDV.

Appendix B: Additional Quantitative Results

Results for Optimizing Auxiliary Costs

In this section we investigate the ability to not only imitate expert behavior, but also to directly optimize metrics of interest. This mode blends pure imitation learning with reinforcement learning and allows tailoring certain aspects of the behavior, i.e. to optimize comfort or safety. To illustrate this, we consider optimising a mixed cost function that optimizes both L1 imitation loss and auxiliary losses:

$$L_{\bar{\tau}}(s_t, a_t) = \|\bar{p}_t - p_t\|_1 + \alpha|\texttt{acc}(a_t)| + \beta \sum_{e_i \in V} \texttt{coll}(e_i, p_t) \tag{1}$$

Here $\texttt{acc}(a_t)$ is the magnitude of the acceleration at time t and $\texttt{coll}(e_i, p_t)$ is a differentiable collision indicator, with V denoting the set of other vehicles. This loss is based on [1], more details can be found in Appendix D. α, β allow to weigh the influence of these different losses.

The ability to succeed on this task requires optimally trading-off short- and long-term performance between pure imitation and other goals. Tables 1 and 2 summarize performance when including acceleration and collision loss, respectively. When including the acceleration term, we note our method is the only one to successfully trade-off performance between imitation and comfort cost, thanks to its capability to directly optimize over the full distribution: while I1K slightly increases with growing α – which is expected – we can push comfort failures down to arbitrary levels. All other models fail for at least one of these metrics, and / or are insensitive to α. When including the collision loss, results are closer together. We hypothesize this is due to $\alpha = 0$, allowing one-step corrections and thus requiring less reasoning over the full time horizon.

Ablation Studies

Figure 3 shows the impact of training dataset size on performance. We see the performance of the method improving with more data. Figure 4 demonstrates the effect of different K on the performance of closed-loop training and thus demonstrates the importance of proper sampling.

5th Conference on Robot Learning (CoRL 2021), London, UK.

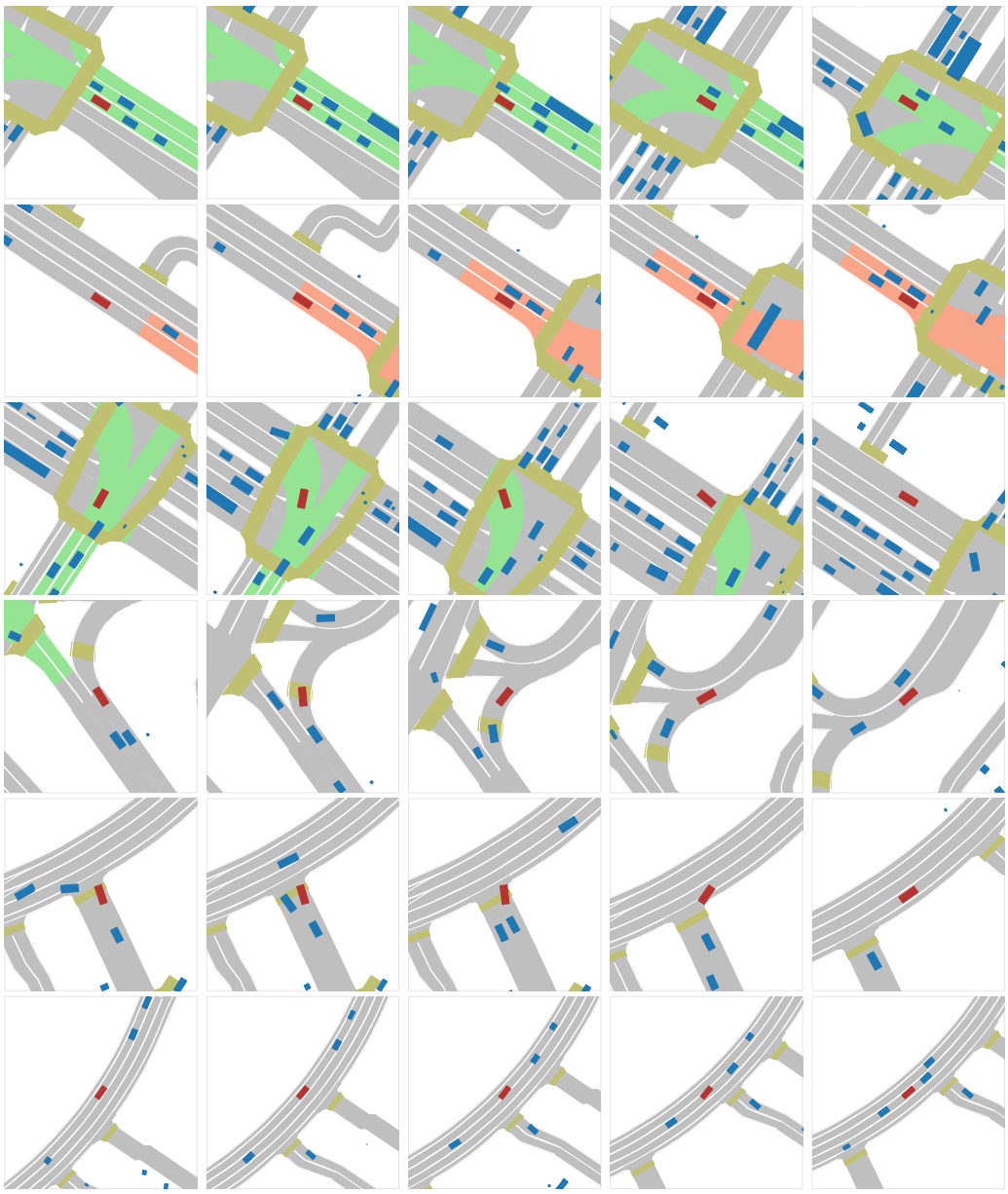

Figure 1: Qualitative results of our method controlling the SDV. Every row depicts one scene, images are 2s apart. The SDV is drawn in red, other agents in blue and crosswalks in yellow. Traffic lights colors are projected onto the affected lanes. Best view on a screen.

Discussion on Used Metrics

Metrics and their definition are naturally crucial for evaluating experiments - thus in the remainder of this section we list additional results using different thresholds and metrics. As reported in the paper, our default threshold for capturing deviations from the expert trajectory is 2m - which is based on average lane widths. Still, one can image wider lanes and less regulated traffic scenarios. Due to this Table 3 shows results of all examined methods using a threshold of 4m. Naturally, off-road failures increase, while other metrics improve due to our process of resetting after interventions. Still, one can observe that the reported results are relatively robust against such changes, i.e. the differences are small and relative trends still hold.

In the paper, for simplicity we measure comfort with one value, namely acceleration - which itself is based on differentiating speed, i.e. the travelled lateral and longitudinal distance divided by time.

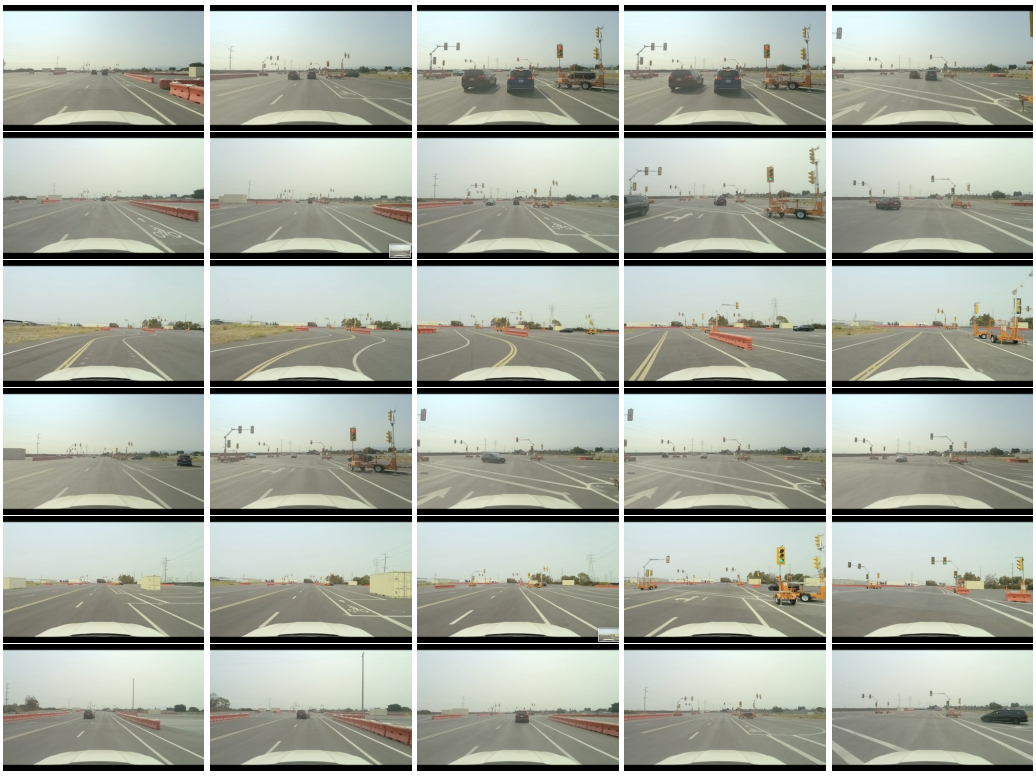

Figure 2: Qualitative results of our method controlling an SDV in the real. Every row depicts one scene, read left to right.

Configuration			Metrics	
α	β	Model	I1K	Comfort
0.01	0	BC-perturb	10,553	23,526
		MS Prediction	**1,428**	20,980
		Ours	2,512	**10,168**
0.03	0	BC-perturb	11,026	9,815
		MS Prediction	2,205	15,546
		Ours	**2,147**	**4,670**
0.1	0	BC-perturb	11,068	7,679
		MS Prediction	**2,316**	28,780
		Ours	2,737	**3,307**

Configuration			Metrics	
α	β	Model	Collisions	Comfort
0	0	BC-perturb	772	600,778
		MS Prediction	1,654	188,189
		Ours	2,055	205,131
0	1000	BC-perturb	264	858,546
		MS Prediction	612	388,632
		Ours	765	258,114
0	10000	BC-perturb	568	943,144
		MS Prediction	380	599,985
		Ours	669	508,679

Table 1: Left: influence of the acceleration term weight α. Only ours manages to find trade-offs and yields reasonable I1K and Comfort values. Right: influence of the collision term weight β. For simplicity both experiments were run with $K = 5$, note that larger K further improves performance of ours (compare Table 1 of the paper and Appendix B 2.2).

However, to reflect actual felt driving comfort, (longitudinal) jerk and lateral acceleration are better suited and more common in the industry. Therefore, Table 4 contains these additional values, and otherwise is identical to Table 1 of the original paper. These values yield more interesting insights into obtained driving behaviour, for example indicating that most discomfort is caused by longitudinal acceleration and jerk, while the lateral movement for all methods is much smoother. We further observe a similar theme as reported in the paper - namely that our method is the only one to be able to jointly optimize for performance and comfort, and that larger α yield smoother driving. Still, we note that the number of jerk failures is higher than the number of acceleration failures - which leads to promising future experiments in the form of explicitly penalizing jerk instead, or in addition to, acceleration.

To complete this excursion on metrics, we briefly discuss rear collisions. Often, they can be attributed to mistakes of other traffic participants, or non-reactive simulation (consider choosing a

Configuration			Metrics		
α	β	Model	I1K	Jerk	Lat. Acc.
0.01	0	BC-perturb	10,553	47,579	921
		MS Prediction	**1,428**	632,608	118
		Ours	2,512	621,180	128
0.03	0	BC-perturb	11,026	19,033	931
		MS Prediction	2,205	578,189	133
		Ours	**2,147**	354,341	138
0.1	0	BC-perturb	11,068	18,599	2062
		MS Prediction	**2,316**	691,540	133
		Ours	2,737	131,589	128

Configuration			Metrics		
α	β	Model	Collisions	Jerk	Lat. Acc.
0	0	BC-perturb	772	1,914,230	8,052
		MS Prediction	1,654	1,315,563	133
		Ours	2,055	1,311,728	607
0	1000	BC-perturb	264	1,922,438	29,234
		MS Prediction	612	1,455,627	1879
		Ours	765	1,935,049	261
0	10000	BC-perturb	568	1,764,526	155,852
		MS Prediction	380	1,580,696	3,131
		Ours	669	1,498,776	1,158

Table 2: Repeating Table 1, but listing (longitudinal) jerk and lateral acceleration for comfort.

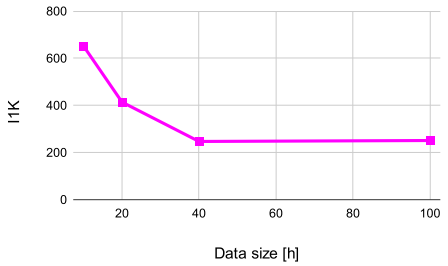

Figure 3: Influence of training data on our planner's performance: more data helps, but we seem to be reaching a plateau in performance.

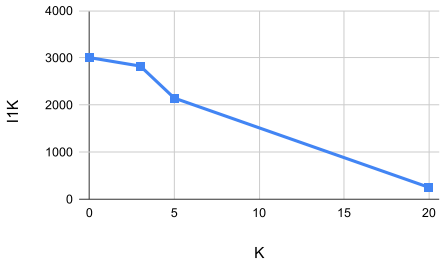

Figure 4: Importance of proper sampling: performance increases with growing K.

slightly different velocity profile, resulting in a rear collision over time). Still, rear collisions can indicate severe misbehavior, such as not starting at green traffic lights, or sudden, unsafe braking maneuvers. See Figure 5 for an example. Due to this, we do include rear collisions in our aggregation metric I1K - however note that we report all metrics separately, as well, to allow a detailed, customized performance analysis.

Appendix C: Policy architecture and state representation

In this section we disclose full details of the proposed network architecture, shown in Figure 6: Each high level object (such as an agent, lane, cross walk) is comprised of a certain number of points of feature dimension F. All points are individually embedded into a 128-dimensional space. We then add a sinusoidal embedding to points of each object to introduce an understanding of order to the model, and feed this to our PointNet implementation. This consists of 3 PointNet layers, in the end producing a descriptor of size 128 for each object. We follow this up with one layer of scaled dot-product attention: for this, the feature descriptor corresponding to ego is used as key, while all feature descriptors are taken as keys / value. We add an additional type embedding to the keys, s.t. the model can attend the values using also the object types – inspired by [2]. Via a final MLP the output is projected to the desired shape, i.e. $T \times 3$ for a trajectory of length T, in which each step is described via xy position and a yaw angle.

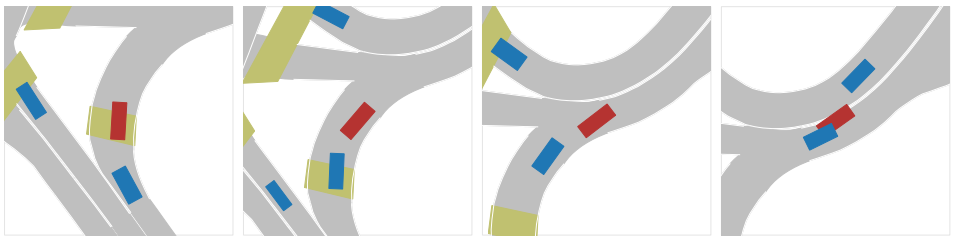

Figure 5: Showing one example of a critical rear-collision: in this case, the planner controlling the SDV (BC-perturb without ego history) decides to abruptly stop after short turn, causing the trailing car to crash into it.

Configuration		Collisions			Imitation			
Model	SDV history	Front	Side	Rear	Off-road	L2	Comfort	**I1K**
BC		153 ± 42	482 ± 203	1,043 ± 67	974 ± 298	8.27 ± 1.75	**102K** ± 1K	2,653 ± 483
BC-perturb		22 ± 4	57 ± 8	414 ± 142	**27** ± 5	3.06 ± 0.06	204K ± 6K	512 ± 127
BC-perturb	✓	**14** ± 6	74 ± 10	680 ± 12	**27** ± 6	3.18 ± 0.02	629K ± 23K	796 ± 12
MS Prediction	✓	22 ± 3	55 ± 3	125 ± 12	60 ± 13	2.07 ± 0.14	598K ± 49K	265 ± 17
Ours	✓	17 ± 7	**51** ± 5	**102** ± 12	40 ± 6	**1.83** ± 0.04	638K ± 41K	**210** ± 9

Table 3: Repeating Table 1 of the paper, but with a threshold of 4m for off-road failures.

Configuration		Collisions			Imitation		Comfort		
Model	SDV history	Front	Side	Rear	Off-road	L2	Jerk	Lat. Acc.	**I1K**
BC		79 ± 23	395 ± 170	997 ± 74	1618 ± 459	1.57 ± 0.27	958K ± 46K	71 ± 23	3,091 ± 601
BC-perturb		16 ± 2	56 ± 6	411 ± 146	82 ± 11	0.74 1,15± 0.01	1,156K ± 672K	1,115 ± 278	567 ± 128
BC-perturb	✓	**14** ± 4	73 ± 7	678 ± 11	**77** ± 6	0.77 ± 0.01	1,862K ± 46 K	7,285 ± 593	843 ± 6
MS Prediction	✓	18 ± 6	55 ± 4	125 ± 14	141 ± 31	0.46 ± 0.02	1,600K ± 14K	211 ± 21	341 ± 39
Ours	✓	15 ± 7	**46** ± 5	**101** ± 13	97 ± 6	**0.42** ± 0.00	1,750K ± 196K	507 ± 321	**260** ± 9

Table 4: Repeating Table 1 of the paper, but listing more fine-grained comfort metrics, namely (longitudinal) jerk and lateral acceleration.

A full description of our model input state is included in Table 5. We define the state as the whole set of static and dynamic elements the model receive as input. Each element is composed of a variable number of points, which can represent both time (e.g. for agents) and space (e.g. for lanes). The number of features per point depends on the element type. We pad all features to a fixed size F to ensure they can share the first fully connected layer. We include all elements up to the listed maximal number in a circular FOV of radius 35m around the SDV. Note that for performance and simplicity we only execute this query once, and then unroll within this world state.

Appendix D: Differentiable Collision Loss

We use a similar differentiable collision loss as proposed in [1]: idea is approximating each vehicle via $N = 3$ circles, and checking these for intersections. Assume loss calculation for timesteps $T-K$ to T, we then define our collision loss as:

$$\sum_{e_i \in V} \texttt{coll}(e_i, p_t) = \sum_{e_i \in V} \sum_{t=K}^{T} \texttt{pair}(e_i, p_t) \tag{2}$$

Here, $\texttt{pair}(e_i, p_t)$ describes a pair-wise collision term between our SDV and vehicle e_i at timestep t. Assume, e_i and SDV (given by pose p_t) are represented via circles C_i and C_{SDV}, then $\texttt{pair}(e_i, p_t)$

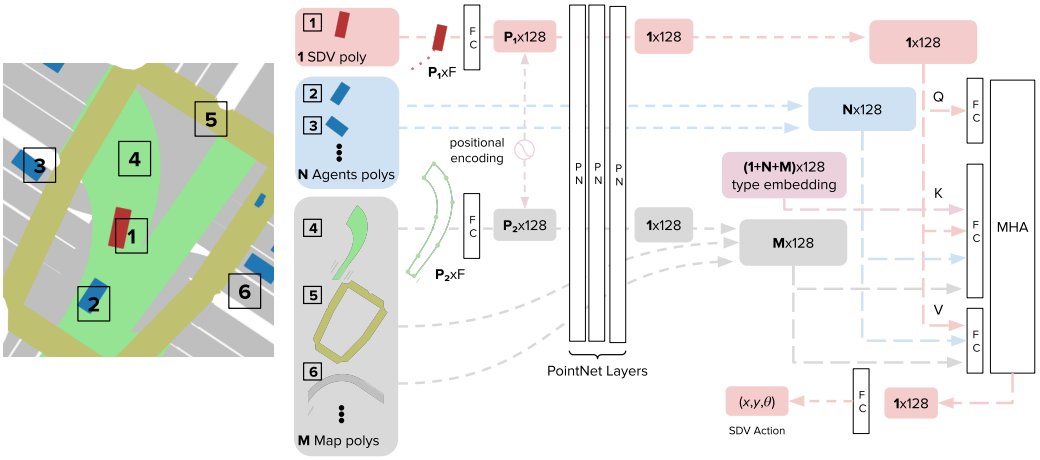

Figure 6: Overview of our policy model. Each element in the state is independently forwarded to a set of PointNet layers. The resulting features go through a Multi-Head Attention layer which takes into account their relations to output the final action for the SDV. The bird's-eye-view image on the left is only for illustrative purposes; we do not employ any rasterizations in our pipeline.

State element(s)	Elements per state	Points per element	Point features description
SDV	1	4	SDV X, Y, yaw pose of the current time step and in recent past
Agents	up to 30	4	other agents X, Y, yaw poses of the current time step and in recent past
Lanes mid	up to 30	20	interpolated X,Y points of the mid lanes with optional traffic light signal
Lanes left	up to 30	20	interpolated X,Y points of the left lanes boundaries
Lanes right	up to 30	20	interpolated X,Y points of the right lanes boundaries
Crosswalks	up to 20	up to 20	crosswalks polygons boundaries X,Y points

Table 5: Model input state description. The state is composed of multiple elements (e.g. agents and lanes) and each of them has multiple points according to its type. Each point is a multi-dimensional feature vector.

is calculated as the maximum intersection of any two such circles:

$$\texttt{pair}(e_i, p_t) = \texttt{max}_{c_i \in C_i, c_s \in C_{SDV}} \texttt{overlap}(c_i, c_s) \tag{3}$$

with

$$\texttt{overlap}(c_i, c_s) = \begin{cases} 1 - \frac{d}{r_{c_i} + r_{c_s}}, \text{ if } d \leq r_{c_i} + r_{c_s} \\ 0, \text{ otherwise} \end{cases} \tag{4}$$

in which d denotes the distance between the respective circles' centers and r their radius. Thus, this term is 0 when the circles do not intersect, and otherwise grows linearly to a maximum value of 1.

References

[1] S. Suo, S. Regalado, S. Casas, and R. Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. 2021.

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. *End-to-End Object Detection with Transformers*. 2020.