The supplementary materials:
Vehicle trajectory prediction works, but not everywhere

Mohammadhossein Bahari[*,1] Saeed Saadatnejad[*,1] Ahmad Rahimi[2] Mohammad Shaverdikondori[2] Amir Hossein Shahidzadeh[2] Seyed-Mohsen Moosavi-Dezfooli[3] Alexandre Alahi[1]

[1]EPFL Switzerland [2]Sharif university of technology [3] Imperial College London

{mohammadhossein.bahari, saeed.saadatnejad}@epfl.ch

In this document, we present the supplementary materials to the main paper including the overal algorithm, more qualitative results, more quantitative results and how the model works in rasterized space.

1. Overall algorithm

In this section, we demonstrate the overall algorithm for the chosen search method. The pseudo-code of the algorithm for generating a scene is shown in Algorithm 1. The goal is to generate the scene S^* for a given scenario x, a, S and predictor g. The process is called for k_{max} iterations. In each iteration, we start with selecting a transformation function (L. 3). Then, the transformation function generates the corresponding scene (L. 4). After that, the observation trajectory is scaled to ensure feasibility of the scenario (L. 5). Next, the prediction of the model in the new scenario is computed and used to calculate the loss (L. 6, L. 7). The best-achieved loss determines the final generated scene.

Algorithm 1: Scene search method

Input: Sequence h, Scene S, Predictor g, Surrounding vehicles a, Transformation set f, Number of iterations k_{max}

Output: Generated scene S^*

1 Initialize $l^* \leftarrow 1$
2 **for** $k = 1$ ***to*** k_{max} **do**
3   Choose a transformation function
4     $\tilde{S}$ = $[\tilde{s}]$ where $\tilde{s} \leftarrow$ Eq. 2
5   Obtain $\tilde{h}$, $\tilde{a}$ from phys constraints Sec. 3.3
6     $\tilde{z} = g(\tilde{h}, \tilde{S}, \tilde{a})$
7   Calculate l using Eq. 7
8   **if** $l < l^*$ **then**
9         $S^* = \tilde{S}$
10   **end**
11 **end**

2. Additional qualitative results

1. **Real-world retrieval images.** We show more real-world examples for both cases where the trajectory prediction model fails and succeeds in Figure 1.

2. **More generated scenes.** Figure 2 provides more visualizations for the performance of the baselines in our generated scenes.

[*] Equal contribution as the first authors.

(a) Paris location (b) New York location (c) Hong Kong location

(d) New Mexico location (e) Hong Kong location (f) Paris location

(g) New York location (h) New Mexico location (i) New York location

Figure 1. **Retrieving real-world places using our real-world retrieval algorithm.** We observe that the model fails in Paris (a), New York (b), Hong Kong (c) and New Mexico (d). The model also successfully predicts in the drivable area in the remaining figures.

3. **Noise in the drivable area map.** The models predict near perfect in the original dataset with HOR of less than 1%. Our exploration shows that most of the 1% failed cases are due to the annotation noise in the drivable area maps of the dataset and the models are almost error-free with respect to the scene. Some figures are provided in Figure 3.

4. **Gifs.** We provide gifs on the perfromance of model when smoothly transforming the scene in Figure 4. We observe

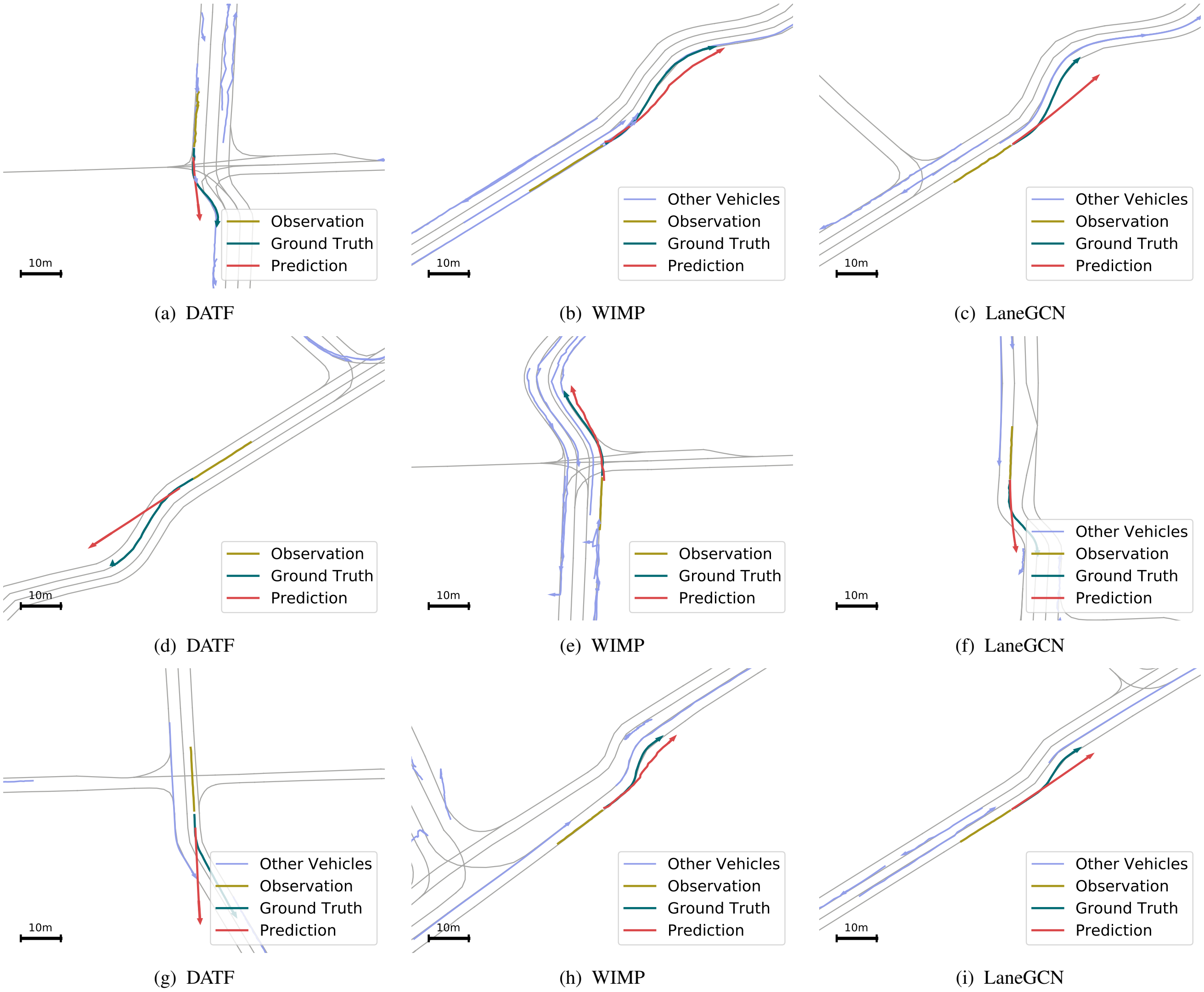

Figure 2. **The predictions of different models in some generated scenes.** All models are challenged by the generated scenes and failed in predicting in the drivable area.

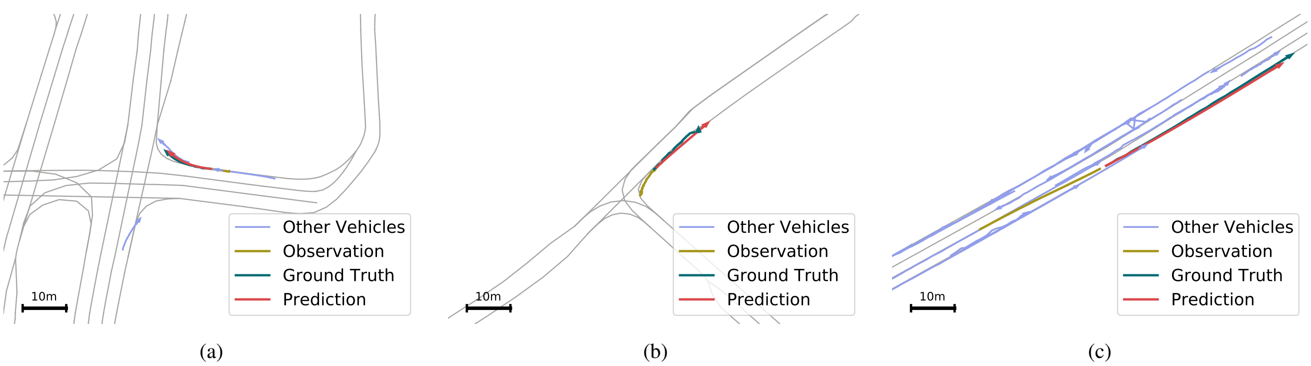

Figure 3. **Some examples showing the noise in the drivable area map.** All these predictions were considered as off-road because of an inaccurate drivable area map.

Model	Original	Generated (**Ours**)			
		Smooth-turn	Double-turn	Ripple-road	All
	SOR / HOR	SOR / HOR	SOR / HOR	SOR / HOR	SOR / HOR
DATF [5]	1 / 2	44 / 92	43 / 91	50 / 95	51 / 99
WIMP [2]	0 / 1	30 / 80	23 / 71	29 / 77	31 / 82
LaneGCN [3]	0 / 1	23 / 65	32 / 75	34 / 77	37 / 81
MPC [8]	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0

Table 1. **Comparing the performance of different baselines in the original dataset scenes and our generated scenes after removing trivial scenarios.** SOR and HOR are reported in percent and the lower represent a better reasoning on the scenes by the model. Numbers are rounded to the nearest integer.

Optimization algorithm	on LaneGCN [3] SOR / HOR	GPU Hours
Baysian [6,7]	13 / 40	17.5
GA [4]	14 / 45	25.0
TPE [1]	14 / 45	12.1
Brute force	23 / 66	4.2

Table 2. **Comparing the performance and computation time of different optimization algorithms in the generated scenes.**

that in some cases the model fails and in some succeeds.

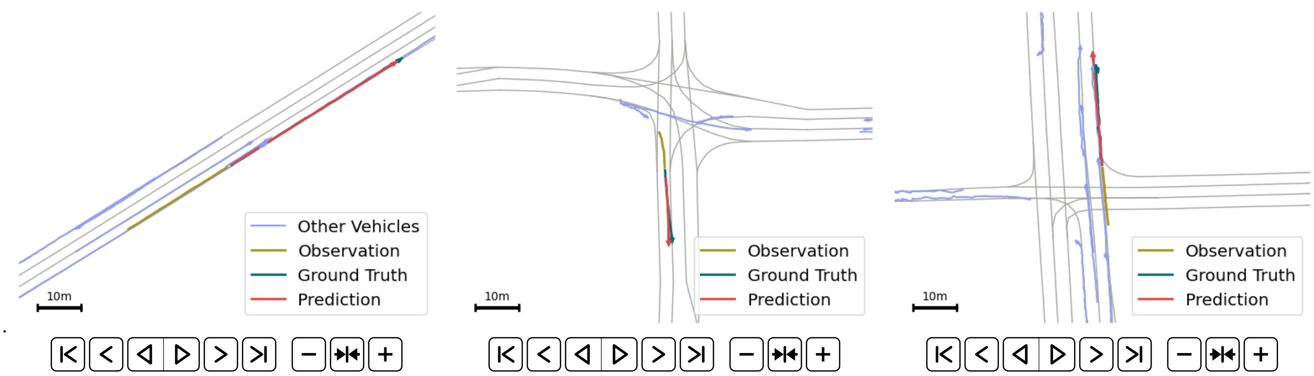

Figure 4. **The animations show the changes of the models predictions in different scenes.** It is best viewed using Adobe Acrobat Reader.

3. Additional quantitative results

3.1. Excluding trivial scenes.

In this section, we remove some trivial scenes, i.e., the scenes that fooling is near impossible, e.g., the scenes with zero velocity. Excluding them, we report in Table 1 and compared to table 1 of the paper, the off-road numbers substantially increase.

3.2. Exploring black box algorithms

In the paper, we mentioned that we used a brute-force approach for finding the optimal values as the search space is not huge. Here, we investigate different block box algorithms for the search. The results of applying different search algorithms are provided in Table 2. They cannot overcome the brute-force approach because of their bigger search spaces (the continuous space instead of the discrete space) and the large required computation time.

4. Generalization to rasterized scene

In the paper, we assumed S is in the vector representation, i.e., it includes x-y coordinates of road lanes points. In the case of a rasterized scene, an RGB value is provided for each pixel of the image. Therefore, it is the same as the vector representation unless here we have information (RGB value) about other parts of the scene in addition to the lanes. Hence, the transformation function can be applied directly on all pixels of the image. In other words, in image representation, s is the coordinate of each pixel which has an RGB value and $\hat{s}$ represents the new coordinate with the same RGB value as s.

References

[1] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems(NeurIPS)*, 2011. 4

[2] Siddhesh Khandelwal, William Qi, Jagjeet Singh, Andrew Hartnett, and Deva Ramanan. What-if motion prediction for autonomous driving. *arXiv preprint arXiv:2008.10587*, 2020. 4

[3] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision (ECCV)*. Springer, 2020. 4

[4] Kim-Fung Man, Wallace Kit-Sang Tang, and Sam Kwong. Genetic algorithms: concepts and applications [in engineering design]. *IEEE Trans. Ind. Electron.*, 1996. 4

[5] Seong Hyeon Park, Gyubok Lee, Jimin Seo, Manoj Bhat, Minseok Kang, Jonathan Francis, Ashwin Jadhav, Paul Pu Liang, and Louis-Philippe Morency. Diverse and admissible trajectory forecasting through multimodal context understanding. In *European Conference on Computer Vision (ECCV)*. Springer, 2020. 4

[6] Binxin Ru, Adam D. Cobb, Arno Blaas, and Yarin Gal. Bayesopt adversarial attack. *International Conference on Learning Representations (ICLR)*, 2020. 4

[7] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *International Conference on Machine Learning (ICML)*, 2010. 4

[8] Julius Ziegler, Philipp Bender, Markus Schreiber, Henning Lategahn, Tobias Strauß, Christoph Stiller, Thao Dang, Uwe Franke, Nils Appenrodt, Christoph Gustav Keller, Eberhard Kaus, Ralf G. Herrtwich, Clemens Rabe, David Pfeiffer, Frank Lindner, Fridtjof Stein, Friedrich Erbs, Markus Enzweiler, Carsten Knöppel, Jochen Hipp, Martin Haueis, Maximilian Trepte, Carsten Brenk, Andreas Tamke, Mohammad Ghanaat, Markus Braun, Armin Joos, Hans Fritz, Horst Mock, Martin Hein, and Eberhard Zeeb. Making bertha drive—an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6, 2014. 4