Risk-Aware Self-Consistent Imitation Learning for Trajectory Planning in Autonomous Driving

Supplementary Materials

Yixuan Fan, Yali Li, and Shengjin Wang*

[1] Department of Electronic Engineering, Tsinghua University
[2] Beijing National Research Center for Information Science and Technology
fan-yx21@mails.tsinghua.edu.cn, {liyali13, wgsgj}@tsinghua.edu.cn

1 Overview

In Sec. 2, we summarize and provide more implementation details of the RaSc model. In Sec. 3, we compare and discuss RaSc with two state-of-the-art methods. In Sec. 4, we describe how our research responsibly uses data from human subjects.

2 The RaSc Model

In Algorithm 1, we delineate the specific training procedure of the RaSc model. By pre-computing the pairwise future TTCs of the recorded trajectories and parallel computing the TTCs of trajectories planned and predicted by the model on GPUs, we constrain the training time overhead incurred by loss computation to within 20% of the total training time. The symbols used in Algorithm 1 have the same meanings as the corresponding symbols in the main paper.

Tab. 1 lists the hyperparameters used by RaSc. The resulting model contains 7.3 million trainable parameters. The data augmentation method we employed only includes randomly replacing values in the current state of the ego vehicle and the historical states of context agents with trainable mask embeddings.

3 More Detail Comparison with State-of-the-Arts

PlanTF [2], the previously best purely learning-based planner, demonstrated the importance of feature modeling and data augmentation for input features related to the ego vehicle state in enhancing the performance of learning-based planning methods. PDM [3], the best rule-based planner equipped with post-processing, showed the effectiveness of using the metrics of interest directly as scorers to evaluate trajectory proposals.

* Corresponding Author.

Our ideas are independent of theirs. We improve the model's understanding of the driving environment and decision-making motivations by adding modules that bridge the fundamental gap between open-loop training and closed-loop deployment on a baseline that is as simple as possible.

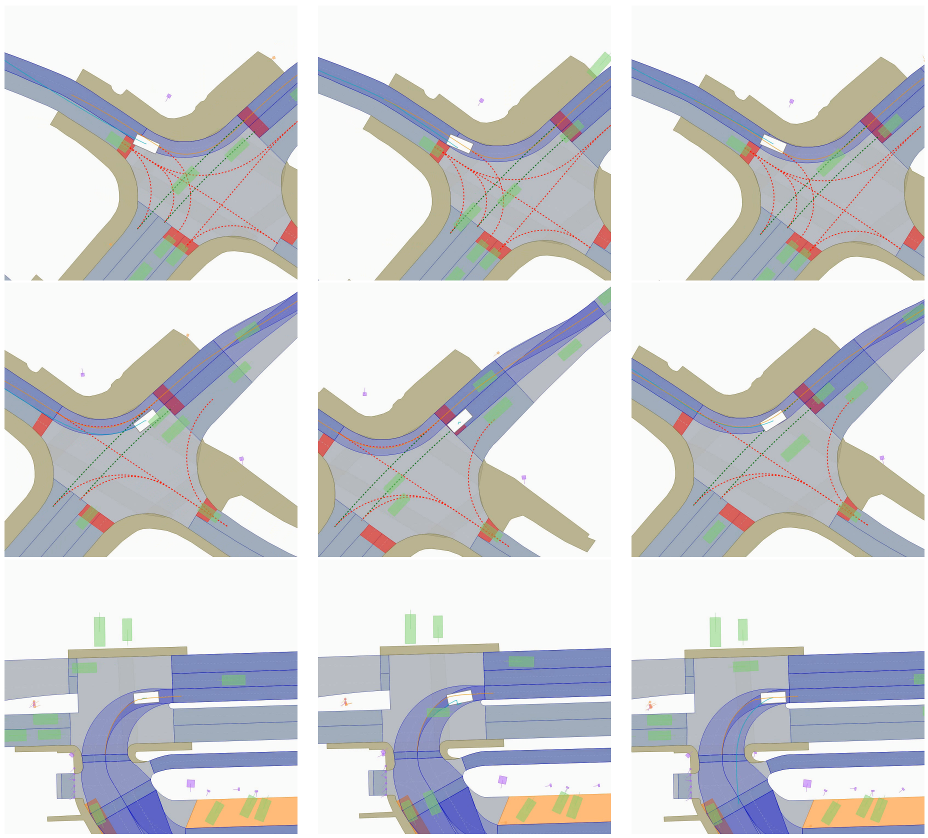
In simulations, we observed that (also reported in [3]), limited by the capability of the annotation system, the ego vehicle may collide with objects that suddenly appear on the road. Additionally, because human drivers sometimes violate traffic rules under specific circumstances (including brief instances of wrong-way driving and surpassing the speed limit), adhering to traffic rules and achieving the progress level of human drivers can be conflicting metrics in some situations. Due to these phenomena, combining or extending existing methods to achieve significantly higher closed-loop performance in the current testing platform may be challenging. In the future, more challenging test scenarios and more comprehensive evaluation criteria could positively impact the development of this field.

4 Responsibility to Human Subjects

We utilize the nuPlan dataset [1] in our experiments. This dataset, collected and made open-source by Motional, Inc., encompasses approximately 1,200 hours of human driving data across four cities: Boston, Pittsburgh, Las Vegas, and Singapore. The data was gathered by professional vehicle operators specifically for data collection purposes, ensuring no risk of privacy breaches. The human traffic participants in the vehicle's environment were processed by the annotation system into categories and bounding boxes, further mitigating any concerns regarding privacy leakage.

References

1. Caesar, H., Kabzan, J., Tan, K.S., Fong, W.K., Wolff, E., Lang, A., Fletcher, L., Beijbom, O., Omari, S.: nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. arXiv preprint arXiv:2106.11810 (2021)
2. Cheng, J., Chen, Y., Mei, X., Yang, B., Li, B., Liu, M.: Rethinking imitation-based planner for autonomous driving. arXiv preprint arXiv:2309.10443 (2023)
3. Dauner, D., Hallgarten, M., Geiger, A., Chitta, K.: Parting with misconceptions about learning-based vehicle motion planning. In: CoRL (2023)
4. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)



(a) PlanTF [2] **(b)** PDM-Hybrid [3] **(c) RaSc (ours)**

Fig. 1: Comparative visualized results. Each row showcases key frames of the results of three state-of-the-art planners (from left to right: PlanTF [2], PDM-Hybrid [3], and our proposed RaSc) in one scenario, in the non-reactive closed-loop simulation. The bounding box in white indicates the ego vehicle, the orange line depicts a 15-second human driving trajectory from the dataset, and the blue line represents the trajectory planned by our model for the next 8 seconds. From top to bottom, the scenarios illustrate: (1) RaSc maintained a safer distance from oncoming traffic and demonstrated better long-term planning; (2) RaSc avoided potential collisions and smoothly completed the turn; (3) RaSc accurately identified the appropriate timing to initiate an unprotected left turn and ensured safety.

Algorithm 1 RaSc Training Procedure.

Input: Historical states of all agents: $s_{-H:0}^{0:N}$;
High-definition map elements: $M_{0:M}$;
Future states of all agents: $s_{1:F}^{0:N}$;
Pre-computed pair-wise future TTCs of the recorded trajectories: $\left\{ttc_{t,i,j}^{*}\right\}_{(t,i,j)\in\mathbb{Q}}$.
Output: Loss for backpropagation: $\mathcal{L}$

\# Scene elements encoding in local coordinates.
$A_0 \leftarrow \mathrm{MLP}\left(s_0^0\right), A_{1:N} \leftarrow \mathrm{LSTM}\left(s_{-H:0}^{1:N}\right), M_{0:M} \leftarrow \mathrm{PointNets}\left(M_{0:M}\right)$
\# Global position encoding.
$p_{0:N+M+1} \leftarrow \mathrm{MLP}([\Delta x_{0:N+M+1}, \Delta y_{0:N+M+1}, \sin \Delta\theta_{0:N+M+1}, \cos \Delta\theta_{0:N+M+1}])$
\# Global interaction modeling. (cat for concatenate)
$A_{0:N}, M_{0:M} \leftarrow \mathrm{Transformer}\left(\mathrm{cat}\left[A_{0:N}, M_{0:M}\right] + \mathrm{cat}\left[p_{0:N+M+1}\right] + \mathrm{cat}\left[attributes\right]\right)$
\# Trajectory decoding.
$s_{1:L,1:F}^{0}, confidence_{1:L}^{0} \leftarrow \mathrm{MLP}\left(A_0\right)$
$s_{1:K,1:F}^{1:N}, confidence_{1:K}^{1:N} \leftarrow \mathrm{MLP}\left(A_{1:N}\right)$
\# Calculate $\mathcal{L}_{traj}$.
$\mathcal{L}_{traj} \leftarrow \mathcal{L}_{plan}\left(s_{1:L,1:F}^{0}, confidence_{1:L}^{0}; s_{1:F}^{0}\right) + \mathcal{L}_{pred}\left(s_{1:K,1:F}^{1:N}, confidence_{1:K}^{1:N}; s_{1:F}^{1:N}\right)$
\# Get the highest confidence planned/predicted trajectories.
$s_{traj,1:F}^{0} \leftarrow s^{0}_{\mathrm{argmax}\left(confidence_{1:L}^{0}\right),1:F}$
$s_{traj,1:F}^{1:N} \leftarrow s^{1:N}_{\mathrm{argmax}\left(confidence_{1:L}^{1:N}\right),1:F}$
\# TTC decoding. (p in inputs means global position embeddings)
$\left\{p_{t,i,j}\right\}_{(t,i,j)\in\mathbb{Q}}, \left\{ttc_{t,i,j}\right\}_{(t,i,j)\in\mathbb{Q}} \leftarrow \mathrm{TTC\_Decoder}\left(A_{0:N}, p_{0:N}\right)$
\# Calculate TTCs from trajectories planned/predicted by the model.
$\left\{ttc_{t,i,j}^{traj}\right\}_{(t,i,j)\in\mathbb{Q}} \leftarrow \mathrm{Calculate\_TTC}\left(s_{traj,1:F}^{0:N}\right)$
\# Calculate $\mathcal{L}_{ra}$.
$\mathcal{L}_{ra} \leftarrow \mathcal{L}_{ttc}\left(\left\{p_{t,i,j}\right\}, \left\{ttc_{t,i,j}\right\}; \left\{ttc_{t,i,j}^{*}\right\}\right)$
\# Calculate $\mathcal{L}_{sc}$.
$\mathcal{L}_{sc} \leftarrow \mathcal{L}_{ttc}\left(\left\{p_{t,i,j}\right\}, \left\{ttc_{t,i,j}\right\}; \left\{ttc_{t,i,j}^{traj}\right\}\right)$
\# Calculate $\mathcal{L}$.
$\mathcal{L} \leftarrow \lambda_{traj}\mathcal{L}_{traj} + \lambda_{ra}\mathcal{L}_{ra} + \lambda_{sc}\mathcal{L}_{sc}$
\# Apply the self-consistency-induced OHEM.
if $step > total\_steps * ohem\_start\_progress$ **then**
    do OHEM, set $\mathcal{L}$ to 0 for some samples.
end if
return $\mathcal{L}$

Table 1: Hyperparameters of RaSc

training procedure	optimizer	AdamW [4]
	base learning rate	0.0006
	weight decay	0.05
	optimizer momentum	β_1: 0.9, β_2: 0.999
	batch size	256
	training epochs	50
	learning rate schedule	cosine decay
	warmup epochs	1.0
loss weight	classification loss weight for trajectory imitation	0.1
	regression loss weight for trajectory imitation	1.0
	λ, classification loss weight for $\mathcal{L}_{ttc}$	1.0
	λ_{traj}, weight for the trajectory imitation loss	1.0
	λ_{ra}, weight for the risk awareness imitation loss	3.0
	λ_{sc}^0, maximum weight for the self-consistency constraint	9.0
model	hidden dimension	256
	H, observable history time steps	20
	F, future time steps for planning	80
	time step size	0.1s
	L and K, planning and prediction modes	6
	Transformer layers	5
	agent encoders layers (LSTM)	1
	map encoders layers (PointNet)	2
	map encoders hidden dimension	128
	attention heads for Transformer layers	4
	FFN dimension for Transformer layers	1024
	attention heads for the cross attention-based TTC decoder	4
data augmentation	ego state mask ratio	0.75
	context agents state mask ratio	0.2
data	total scenarios	1.0M (122GB storage)
	perception radius	80m