HPTP: Including Human Preference in Trajectory Prediction Models

Ahmad Rahimi

Alexandre Alahi

STRC conference paper 2024 **May 5, 2024**

STRC | **24th Swiss Transport Research Conference**
Monte Verità / Ascona, May 15-17, 2024

HPTP: Including Human Preference in Trajectory Prediction Models

Ahmad Rahimi
VITA
EPFL
ahmad.rahimi@epfl.ch

Alexandre Alahi
VITA
EPFL
alexandre.alahi@epfl.ch

May 5, 2024

Abstract

Recent advancements in vehicle trajectory prediction have notably improved data-driven models, yet they struggle with complex scenarios, showing limited prediction diversity and sometimes failing to comply with road constraints. These critical yet hard-to-evaluate issues hinder the development of safer, more robust models. Reflecting on similar challenges in natural language processing (NLP), where Reinforcement Learning from Human Feedback (RLHF) has effectively enhanced model quality, our study investigates integrating similar human feedback into vehicle trajectory models. We propose a novel approach using a learned reward model to infuse human judgment, aiming to improve prediction accuracy and reliability. This research marks a pivotal step in combining artificial intelligence with human expertise for more precise and secure vehicle trajectory forecasting.

Keywords

Vehicle trajectory prediction, Human feedback

Suggested Citation

Contents

List of Tables

List of Figures

1 Introduction

Vehicle trajectory prediction is the problem of predicting possible future positions of agents in a driving scenario, given the past trajectory of the traffic participants, as well as the map of surrounding environment. It plays a crucial role for the comfortable and safe planning of autonomous vehicles.

The problem has attracted researchers and companies, shown by the many emerged benchmarks and papers in the field. Researchers try to solve this problem from different aspects; some encode the road map and past trajectories as images which are then fed to convolutional neural networks, while others use the continuous positions of the vehicles and map centerlines and leverage transformers to solve the forecasting. However, most of these models have failure cases in predicting all possible future possibilities, or having predictions going off-road or the opposite direction of the road. Reducing these failure cases is crucial considering the task is safety critical.

Designing hand-crafted rules and loss functions to guide models correct themselves in these failure cases is difficult and costly. Therefore we propose a human feedback framework, where a human provides preference annotation on the predictions of multiple models. Then these preferences are used to train a reward model, which outputs a higher reward value for the predictions that are preferred by humans. Finally, the knowledge of human preference available in this reward model is used to fine-tune the original model with the objective to increase the reward.

This framework would pave the way to more easily insert human knowledge in vehicle trajectory prediction models, through simple and cheap preference labels.

2 Related works

Trajectory prediction. Trajectory prediction has been a hot topic in the past few years, due to the increasing need and importance of the problem for autonomous vehicles. The pioneer of the field was Alahi *et al.* (2016) where they used LSTMs to model the temporal and social interactions between human trajectories. Another challenge which is more pronounced for vehicle trajectory prediction is the consistency to map of the scenario, *i.e.*, the predicted trajectories should not go off-road and should be consistent with the road direction. One line of work Liang *et al.* (2020); Deo *et al.* (2022) constructs a graph based on the map of the scene and observes the problem from a graph learning perspective. Another line of work Gu *et al.* (2021); Gilles *et al.* (2021); Mangalam *et al.* (2020) decouples the trajectories from their end-goal and first predicts a heatmap of the end position of the vehicle in the scene, and then rolls out the full trajectory conditioned on different final goal points. After the very successful spread of transformer architectures Vaswani *et al.* (2023) from natural language processing, more recently, many researcher Shi *et al.* (2023); Nayakanti *et al.* (2022); Girgis *et al.* (2022) are using these architectures to solve trajectory prediction.

Reinforcement Learning from Human Feedback. A framework called Reinforcement Learning from Human Feedback (RLHF) was first introduced in the reinforcement learning field Christiano *et al.* (2017). They used human feedback to learn the reward model in an efficient way for tasks where defining a good reward model is difficult for human researchers. More recently, usage of this framework has extended to fine-tuning large language models (LLMs) to make their text generations more safe and usefull Ziegler *et al.* (2020); Stiennon *et al.* (2022). It is the de-facto alignment framework for fine-tuning LLMs nowadays, with all the major chat bots using it, often iteratively Touvron *et al.* (2023). Given the similarities between NLP domain and ours, we explore the use of RLHF in vehicle trajectory prediction. For collecting human preference data and training a reward model, we use the predictions coming from Girgis *et al.* (2022) and Deo *et al.* (2022) which are transformer and graph based vehicle trajectory prediction models, respectively.

3 Methodology

In this section we will formally describe our methodology. We will first define the trajectory prediction problem. Then, we introduce three steps to incorporate human feedback into trajectory prediction models: (*i*) human preference data collection, (*ii*) reward model training, (*iii*) model finetuning.

Trajectory prediction. Consider a trajectory forecasting problem where an ego agent $\mathbf{e}$ is surrounded by a set of neighboring agents $\mathcal{N}$ in a scene with surrounding area map $\mathcal{M}$. Let $s_t^i = (x_t^i, y_t^i)$ denote the state of agent i at time t and $s_t = \{s_t^1, \cdots, s_t^{|\mathcal{N}|}\}$ denote the joint state of all the agents in the scene. Given a sequence of history observations

$\boldsymbol{x} = (s_1, \cdots, s_t)$, the task is to predict future trajectories of all agents $\boldsymbol{y} = (s_{t+1}, \cdots, s_T)$ until time T. However, as the trajectory prediction task is multimodal, and given different intentions of agents in the scene, multiple futures are possible, we allow the model to predict K possible future trajectories, *i.e.*, $\boldsymbol{\hat{y}^1}, \cdots, \boldsymbol{\hat{y}^K}$. Modern forecasting models are largely composed of encoder-decoder neural networks, where the encoder $f(.)$ first extracts a compact representation h_t^i with respect to agent i and the decoder $g(.)$ subsequently rolls out its predicted future trajectory $\boldsymbol{\hat{y}_i} = \hat{s}^i_{t+1:T}$:

$$\begin{aligned} h_t^i &= f(s_{1:t}, i, \mathcal{M}), \\ \hat{s}^i_{t+1:T} &= g(h_t^i, \mathcal{M}). \end{aligned}$$

Human preference data collection. Given a pool of trajectory prediction models $\mathbf{M} = \{M_1, \ldots, M_m\}$, we collect a dataset $\mathcal{D}$ of the failure cases of models in $\mathbf{M}$. Then, we sample $d \in \mathcal{D}$ from the dataset, and $M_a, M_b \in \mathbf{M}$ from the prediction models. We generate the predictions $\boldsymbol{\hat{y}_a}, \boldsymbol{\hat{y}_b}$ of models M_a, M_b, given the failure case d. These predictions are shown to a human annotator and one of them is chosen to be preferred over the other one. These predictions together with the preferred index are gathered to form the human preference dataset $\mathcal{D}_{HF}$ containing tuples of form $(d, \boldsymbol{\hat{y}_a}, \boldsymbol{\hat{y}_b}, w \in \{L, R\})$.

Reward model training. Having the human feedback dataset $\mathcal{D}_{HF}$, we train a reward model $\mathcal{R}$ which returns a reward r, given the scenario d and predicted trajectory $\boldsymbol{\hat{y}}$:

$$r = \mathcal{R}(d, \boldsymbol{\hat{y}}).$$

Training procedure on the $\mathcal{D}_{HF}$ dataset follows:
For each record $(d, \boldsymbol{\hat{y}_a}, \boldsymbol{\hat{y}_b}, w \in \{L, R\})$ in the dataset, r_a, r_b are calculated using the reward model:

$$\begin{aligned} r_a &= \mathcal{R}(d, \boldsymbol{\hat{y}}_a), \\ r_b &= \mathcal{R}(d, \boldsymbol{\hat{y}}_a). \end{aligned}$$

If $w = L$, then the first prediction is preferred by the human annotator, therefore r_a should be higher than r_b, which is enforced by the loss function

$$\mathcal{L}_{RM} = -\log\big(\sigma(r_a - r_b)\big).$$

If $w = R$, then the second prediction is preferred, hence the loss function has the form

$$\mathcal{L}_{RM} = -\log\big(\sigma(r_b - r_a)\big).$$

Trajectory prediction model finetuning. Having a reward model which encodes the human preference knowledge, we optimize the trajectory prediction model to increase the reward obtaining by the reward model. The procedure is as follows:
Given a sample from the trajectory prediction dataset d, the trajectory prediction model M is used to generate predictions $\hat{\boldsymbol{y}} = M(d)$. Finally, the prediction is fed to the reward model to get reward $r = \mathcal{R}(d, \hat{\boldsymbol{y}})$. To maximize the reward, the final loss function is negative expectation of reward:

$$\mathcal{L}_{FT} = -\mathbb{E}[r].$$

However, if we just aim to maximize the reward, a phenomenon called reward hacking might happen. Since our reward model is trained on the predictions produced by the raw models, fine-tuning the model would change the input distribution of the reward model, making its output rewards not reliable. In some cases, the model might be able to find some shortcuts in the reward model to artificially increase the reward without learning something meaningful. In order to prevent reward hacking, we add a KL divergence penalty to our loss function to keep models predictions from changing too much from the original model.

4 Experiments

In this section we explain our experimental setup and results. We first describe the baseline trajectory prediction models we use and the metrics we report, then we present some statistics of the human preference data we have collected. Reward model training and prediction model fine-tuning finalize this section.

4.1 Experimental setup

Dataset. We use nuScenes dataset Caesar *et al.* (2020) which is a widely used vehicle trajectory prediction dataset used in the literature. It has around 40,000 clean and challenging driving scenarios, with the surrounding area map. nuScenes allows for 10 predicted trajectories per driving scenario, *i.e.*, $K = 10$.

Baselines. We incorporate two diverse baselines with state of the art performance on nuScenes, namely AutoBots Girgis *et al.* (2022) and PGP Deo *et al.* (2022). The former is a lightweight transformer based model consisting of social and temporal transformers for modeling interactions among agents and map. The latter first constructs a directed graph $\mathcal{G}$ based on the map of the surrounding area, where a directed path in $\mathcal{G}$ corresponds to a feasible route of the vehicle in the driving scenario. It then learns weights for each edge in the graph, representing the probability of using that edge. Given this probabilistic graph, PGP then generates its final predictions conditioned on random traversals of the graph.

Metrics. There are a few commonly used metrics in the field, which we briefly describe here:

- **minADE** stands for minimum average displacement error. It calculates the average distance between each prediction and the ground truth future trajectory and takes the minimum ADE among all predictions. Given predictions $\boldsymbol{\hat{y}^1}, \cdots, \boldsymbol{\hat{y}^K}$ and the ground truth future trajectory $\boldsymbol{y}$, the minADE metric could be calculated as:

$$\text{minADE} = \min_{1 \leq i \leq K} \frac{1}{T-t} \sum_{\tau=t+1}^{T} ||\boldsymbol{\hat{y}^i_\tau} - \boldsymbol{y_\tau}||_2.$$

- **minFDE** stands for minimum final displacement error. It is very simliar to minADE, with the difference that the displacement error is only calculated for the final time step. More precisely, it is defined as:

$$\text{minFDE} = \min_{1 \leq i \leq K} ||\boldsymbol{\hat{y}^i_T} - \boldsymbol{y_T}||_2.$$

- **Offroad** measures the proportion of predicted trajectories that went off-road. It could be formally defined as:

$$\text{Offroad} = \frac{\sum_{i=1}^{K} \mathbb{1}\{\boldsymbol{\hat{y}^i} \text{ goes off road}\}}{K}$$

In this work, we have designed some metrics to better asses the quality of our improved models predictions. They are defined in the following:

- **DDA** stands for Distance to Drivable Area. It is a differentiable metric, replacing the non-differentiable Offroad metric. For each point in a given trajectory $\hat{\boldsymbol{y}}$, it calculates the distance of that point to the closes drivable area, and averages for all the points. A DDA of 0 means the trajectory is inside drivable area, otherwise it shows by how much the trajectory has gone off road.
- **RDC** measures the road direction consistency of the trajectory. For each point in the predicted trajectory, it measures the distance of this point to the closest point on the map centerlines, and the angle difference between the trajectory at that point and the centerline. It penalises trajectories that their direction is not consistent with that of the road.
- **Diversity** measures how diverse the set of predictions $\hat{\boldsymbol{y}}^{\boldsymbol{1}}, \cdots, \hat{\boldsymbol{y}}^{\boldsymbol{K}}$ are. It calculates the sum of pairwise distance between the end points of the predicted trajectories, more precisely:
$$\text{Diversity} = \sum_{i \neq j} ||\hat{\boldsymbol{y}}^{\boldsymbol{i}}_{\boldsymbol{T}} - \hat{\boldsymbol{y}}^{\boldsymbol{j}}_{\boldsymbol{T}}||_2$$

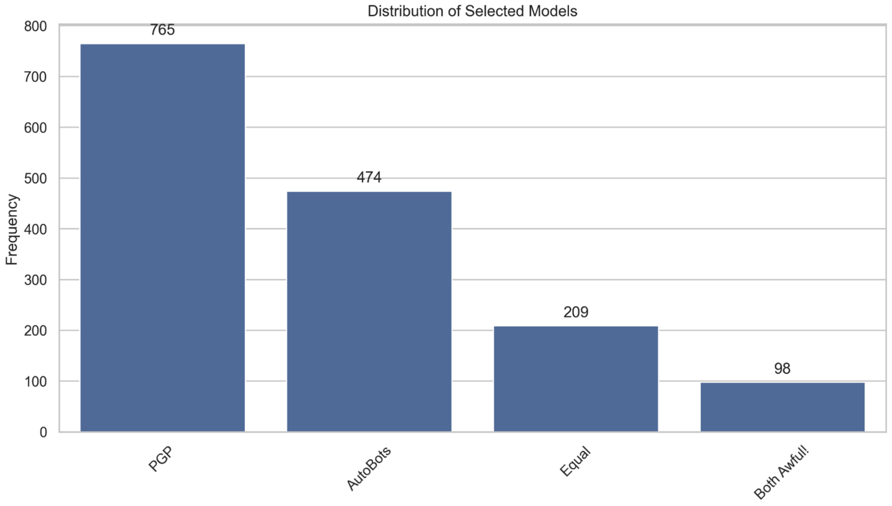
4.2 Human feedback data

We defined the dataset of failure cases $\mathcal{D}$ for AutoBots and PGP models to be all the driving scenarios that either of the two models has a minFDE higher than 5 meters. It consists of about 1500 driving scenarios. Each of the two models are used to predict trajectories for the scenarios in $\mathcal{D}$, which are then shown to human annotators. Our labelers selected the set of 10 trajectories that they thought is more diverse, while also adhering to driving rules. Each labeling on average took 20 seconds, and you can see the statistics of chosen models in figure 1.

4.3 Reward model

We design three types of reward models:

Figure 1: The distribution of selected options by human labelers.



- **Feature Engineering** reward model, for which we manually design some differentiable features to be given as input. For each of the 10 trajectories we calculate the ADE, FDE, DDA, and RDC. We also calculate the minADE, minFDE, Diversity, and number of offroad trajectories as global features. In total, our feature engineering reward model receives 44 features and uses a multi layer perceptron (MLP) to calculate the reward.
- **Data Driven** reward model uses only the human preference data to learn the reward, and does not use any domain knowledge. For this model, took the pre-trained encoder of AutoBots and added several layers to incorporate the predictions and return the reward.
- **Mixture** of both, uses the same architecture of the Data Driven model, but in the final layers before outputting the reward, we include the features from the Feature Engineering model.

Table 1 shows their performance on human preference prediction accuracy. As expected, the mixture model performs better than both, having the best of both worlds. Notably, the Data Driven model alone is not performing well. We believe this is due to the limited data we have, which makes it hard to learn human preference from data alone.

Table 1: Human preference prediction accuracy of the different reward models. The mixture model performs best.

Reward Model	Feature Engineering	Data Driven	Mixture
Accuracy (%)	78.5	69	80

Table 2: Effect of fine-tuning AutoBots using the reward model coming from human feedback. The Full Val is the full validation set of nuScenes, while the Hard Val is the subset of the validation set on which AutoBots has a minFDE higher than 5 meters.

Model	Full Val				Hard Val			
	minADE	minFDE	DDA	RDC	minADE	minFDE	DDA	RDC
baseline	1.029	1.68	0.26	0.36	3.92	9.28	2.82	2.12
fine-tuned (ours)	1.025	1.68	0.25	0.31	3.80	8.63	2.68	2.00

4.4 Model Fine-tuning

Using the reward model introduced in the previous section, in this section we fine-tune AutoBots model to increase the reward given by the reward model. We directly use our fully differentiable reward model on top of the predictions coming from AutoBots model, and optimise the loss function

$$\mathcal{L} = -\mathbb{E}[r]$$

The comparison of our fine-tuned model with the original AutoBots model could be seen in table 2. As one could see, all the metrics have been improved compared to the baseline model (AutoBots), and the improvement is even more pronounced on the hard validation set on which the model usually failed to predict the ground truth trajectory.

5 References

Alahi, A., K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei and S. Savarese (2016) Social LSTM: Human Trajectory Prediction in Crowded Spaces, 961–971.

Caesar, H., V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan,

G. Baldan and O. Beijbom (2020) nuScenes: A multimodal dataset for autonomous driving, http://arxiv.org/abs/1903.11027. ArXiv:1903.11027 [cs, stat].

Christiano, P. F., J. Leike, T. Brown, M. Martic, S. Legg and D. Amodei (2017) Deep Reinforcement Learning from Human Preferences, paper presented at the *Advances in Neural Information Processing Systems*, vol. 30.

Deo, N., E. Wolff and O. Beijbom (2022) Multimodal Trajectory Prediction Conditioned on Lane-Graph Traversals, paper presented at the *Proceedings of the 5th Conference on Robot Learning*, 203–212. ISSN: 2640-3498.

Gilles, T., S. Sabatini, D. Tsishkou, B. Stanciulescu and F. Moutarde (2021) GOHOME: Graph-Oriented Heatmap Output for future Motion Estimation, http://arxiv.org/abs/2109.01827. ArXiv:2109.01827 [cs].

Girgis, R., F. Golemo, F. Codevilla, M. Weiss, J. A. D'Souza, S. E. Kahou, F. Heide and C. Pal (2022) Latent Variable Sequential Set Transformers For Joint Multi-Agent Motion Prediction, http://arxiv.org/abs/2104.00563. ArXiv:2104.00563 [cs].

Gu, J., C. Sun and H. Zhao (2021) DenseTNT: End-to-end Trajectory Prediction from Dense Goal Sets, http://arxiv.org/abs/2108.09640. ArXiv:2108.09640 [cs].

Liang, M., B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng and R. Urtasun (2020) Learning Lane Graph Representations for Motion Forecasting, http://arxiv.org/abs/2007.13732. ArXiv:2007.13732 [cs].

Mangalam, K., H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik and A. Gaidon (2020) It Is Not the Journey but the Destination: Endpoint Conditioned Trajectory Prediction, http://arxiv.org/abs/2004.02025. ArXiv:2004.02025 [cs].

Nayakanti, N., R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat and B. Sapp (2022) Wayformer: Motion Forecasting via Simple & Efficient Attention Networks, http://arxiv.org/abs/2207.05844. ArXiv:2207.05844 [cs].

Shi, S., L. Jiang, D. Dai and B. Schiele (2023) Motion Transformer with Global Intention Localization and Local Movement Refinement, http://arxiv.org/abs/2209.13508. ArXiv:2209.13508 [cs].

Stiennon, N., L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei and P. Christiano (2022) Learning to summarize from human feedback, http://arxiv.org/abs/2009.01325. ArXiv:2009.01325 [cs].

Touvron, H., L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov and T. Scialom (2023) Llama 2: Open foundation and fine-tuned chat models.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin (2023) Attention Is All You Need, http://arxiv.org/abs/1706.03762. ArXiv:1706.03762 [cs].

Ziegler, D. M., N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano and G. Irving (2020) Fine-Tuning Language Models from Human Preferences, http://arxiv.org/abs/1909.08593. ArXiv:1909.08593 [cs, stat].