Exploring Different Loss Functions for Trajectory Prediction

1st Ashton Frias
University of Central Florida
Orlando, Florida
as700045@ucf.edu

I. Introduction

In this project, I experimented with testing different loss functions to see if they would improve the trajectory prediction in autonomous vehicles. The loss function used is crucial to the model's performance, as it instructs the model on what constitutes a good prediction from a poor one. This was evident when comparing the results in Part 1 of the final project. As shown in Table II, the minimum average displacement error (minADE) significantly outperformed Huber's loss function. This is because Huber's loss function attempts to minimize the error uniformly across the entire time series, resulting in subpar scores. It became apparent after testing and analysis that using it was not the optimal approach to solving multi-trajectory predictions. This highlights the importance of selecting an efficient loss function. For this project, I developed two loss functions. The first was the minimum final displacement error (minFDE), which computes a loss based on the final time step. The second loss function was a combination of minFDE and minADE, with the goal of providing two perspectives that would enhance trajectory predictions. While analyzing the results, minFDE did not improve the model's performance because it focused solely on enhancing the results for the last timestep, at the expense of all previous timesteps. However, for the combined loss function (minADE + minFDE), the model was able to maintain and even improve the trajectory predictions when compared to the original performance of minADE [1].

A. Motivation

Multi-agent trajectory prediction intends to provide autonomous vehicles with more information about the future trajectories of all their neighboring agents or obstacles in their environment. An example of this is tracking the positions of multiple pedestrians on a city sidewalk. By generating their future movements based on their current trajectory, this enables the autonomous vehicle to make safer decisions. These decisions may include stopping itself because the pedestrian is predicted to cross the road, or continuing on its current path to its destination if no collision is detected. As a result, this would lead to fewer accidents and improve the overall performance of autonomous vehicles [1].

II. Methods

A. Loss Function

As stated, the loss functions used for this project were minimum final displacement error (minFDE) and a combination loss function that consisted of minimum final displacement error (minFDE) and minimum average displacement error (minADE).

When developing the first loss function, minimum final displacement error (minFDE), I chose this loss function to determine if the model could make accurate decisions based solely on a single timeframe within the entire sequence. To calculate the minFDE, I first extracted the final predicted trajectory from each mode in the batch. This was the last predicted position of the target agent in the final timestep. Next, the Euclidean distance was calculated between each final predicted trajectory and the ground truth for each mode in the batch. After the minimum error across all modes for each sample was stored, the mean across the entire batch was returned. This loss function is equivalent to the equations seen in 2 and 3.

When developing the combined loss function, which consisted of minimum final displacement error (minFDE) and minimum average displacement error (minADE), I utilized the same minADE loss function from the previous assignment.

MinADE loss is computed by taking the Euclidean distance between all the predicted trajectories and the ground truth for each element in the batch. Next, the average displacement error was calculated for each mode, and then the minimum error across all modes was taken. Finally, the mean of all minimum errors in the batch was returned. This loss function is equivalent to the equations seen in 4 and 5.

After receiving the output from both minADE and minFDE, both outputs were then multiplied by a weight ranging from 0 to 1, resulting in the amount of influence each loss function had, as shown in Equation 1.

$$\text{Combined Loss} = \text{MinADE} \cdot w_1 + \text{MinFDE} \cdot w_2 \tag{1}$$

The combined loss function minFDE focused on the final position prediction, while minADE evaluates the overall trajectory. By combining these two loss functions, the model gained two unique perspectives on trajectory prediction, encouraging it to make accurate predictions over the entire timeseries while also producing accurate final predictions.

B. Dataset

The dataset used in the project is a preprocessed version of the NuScenes prediction challenge dataset, which is utilized for multi-agent trajectory prediction. The objective of this dataset is to predict future trajectories of the target agent and its neighboring agents as seen in Table I. Our model receives a sample that includes the trajectory data from the target and up to 10 neighboring agents over a 6 second time period, which results in 13 time indices. Our model is tasked with predicting the future trajectory of each agent for 4 seconds, based on 2 seconds of previously observed trajectories [2].

Time Step	x	y	vx	vy	v
1	-0.2881	-8.1348	1.5715	4.9208	0.0036
2	-0.2301	-5.8934	1.5668	4.4804	-0.8799
3	-0.1719	-3.8474	1.5621	4.0858	-0.7878
...	...	...	...	...	...
13	0.5079	15.2900	1.5233	0.9409	2.9357

TABLE I: Agent trajectory example (13 Time Steps, 5 Features).

C. Metrics

To evaluate the performance of each loss function, the minimum final displacement error (minFDE) and minimum average displacement error (minADE) were calculated for each sample over the entire validation set.

When calculating MinADE, we first compute the Euclidean distance between the predicted trajectory $(\hat{x}_{s,t}, \hat{y}_{s,t})$ and the ground-truth trajectory $(x_{s,t}, y_{s,t})$ for each time step t in each sample s, and then we take the average over all T time steps in that sample. After computing average displacement error (ADE) for each mode m, we return the mode that achieved the lowest error in that sample.

$$\mathrm{MinADE}_s = \min_{m\in\{1,\ldots,M\}} \frac{1}{T}\sum_{t=1}^{T}\sqrt{(\hat{x}_{s,m,t}-x_{s,t})^2+(\hat{y}_{s,m,t}-y_{s,t})^2} \tag{2}$$

Finally, we return the average minimum error across all samples from the entire batch.

$$\mathrm{MinADE} = \frac{1}{S}\sum_{s=1}^{S}\mathrm{MinADE}_s \tag{3}$$

When calculating MinFDE, we compute the Euclidean distance on the last time step T between the predicted trajectory $(\hat{x}_{s,t}, \hat{y}_{s,t})$ and the ground-truth trajectory $(x_{s,t}, y_{s,t})$ for each mode m across all samples s. After computing final displacement error (FDE) for each mode m, we return the mode that achieved the lowest error in that sample.

$$\mathrm{MinFDE}_s = \min_{m\in\{1,\ldots,M\}} \sqrt{(\hat{x}_{s,m,T}-x_{s,T})^2+(\hat{y}_{s,m,T}-y_{s,T})^2} \tag{4}$$

After, we return the average minimum final displacement error across all samples from the entire batch.

$$\mathrm{MinFDE} = \frac{1}{S}\sum_{s=1}^{S}\mathrm{MinFDE}_s \tag{5}$$

Lastly, I computed the average and median scores across all epochs for both *minFDE* and *minADE* using the values obtained from the validation set. This provides a summary of the model's overall performance during training.

$$\overline{\text{Metric}} = \frac{1}{E}\sum_{e=1}^{E}\text{Metric}^{(e)}$$

$$\widetilde{\text{Metric}} = \text{Median}\left(\left\{\text{Metric}^{(e)}\right\}_{e=1}^{E}\right)$$

III. Analysis and Results

A. Training

All models were trained using the default setup from Part 1 and one NVIDIA 3070. Each model was trained on a batch size of 32 for 50 epochs, and had a learning rate of 5e-5 using the Adam optimizer. When training the combination loss function, I experimented with different weights, ranging from 0 to 1. After testing various combinations, I found that assigning equal weights to both models, with a weighted value of 1, produced the best results.

B. Results

When analyzing the results from each of the loss functions, it is clear that the original loss function, minimum average displacement error (minADE), still provides the best overall performance. This is because it analyzes the model's predictions over the entire timeseries, rather than limiting itself to a single point in time. As a result, minADE guides the model to produce a more accurate overall trajectory prediction.

In contrast, the minimum final displacement error (minFDE) encourages the model to focus on making accurate predictions on the final trajectory. As a result, all prior predictions suffer in accuracy, as seen in Table II, where Huber's Loss trajectory prediction over the entire time series exceeds the results of minFDE. Although Huber's loss is not the ideal loss function for trajectory prediction, this emphasizes the importance of making predictions based on the entire timeseries, rather than limiting results to a single point in time.

Furthermore, as shown in Table II, minFDE loss performs the worst amongst all models in both average and median minADE. In addition, this is further illustrated in the cumulative distribution curve in Figure 1, where the tracking error across the 10 modes significantly favors FDE. Specifically, FDE exhibits a 10-meter error difference in tracking error when compared to ADE. This proves that minFDE prioritizes the final trajectory prediction at the expense of the model's overall performance. Although this loss function achieves an optimal average and median minFDE score, it would not be suitable for real-world applications alone, as it forces the

Loss Function	Average minADE	Average minFDE	Median minADE	Median minFDE
Huber's Loss	1.84	4.11	1.43	3.2
minFDE	8.9	1.76	8.4	1.01
minADE	**0.80**	**1.55**	**0.59**	0.994
Combined	0.84	1.63	0.61	**0.990**

TABLE II: Comparison of three loss functions across different prediction accuracy metrics. Lower values indicate better performance.

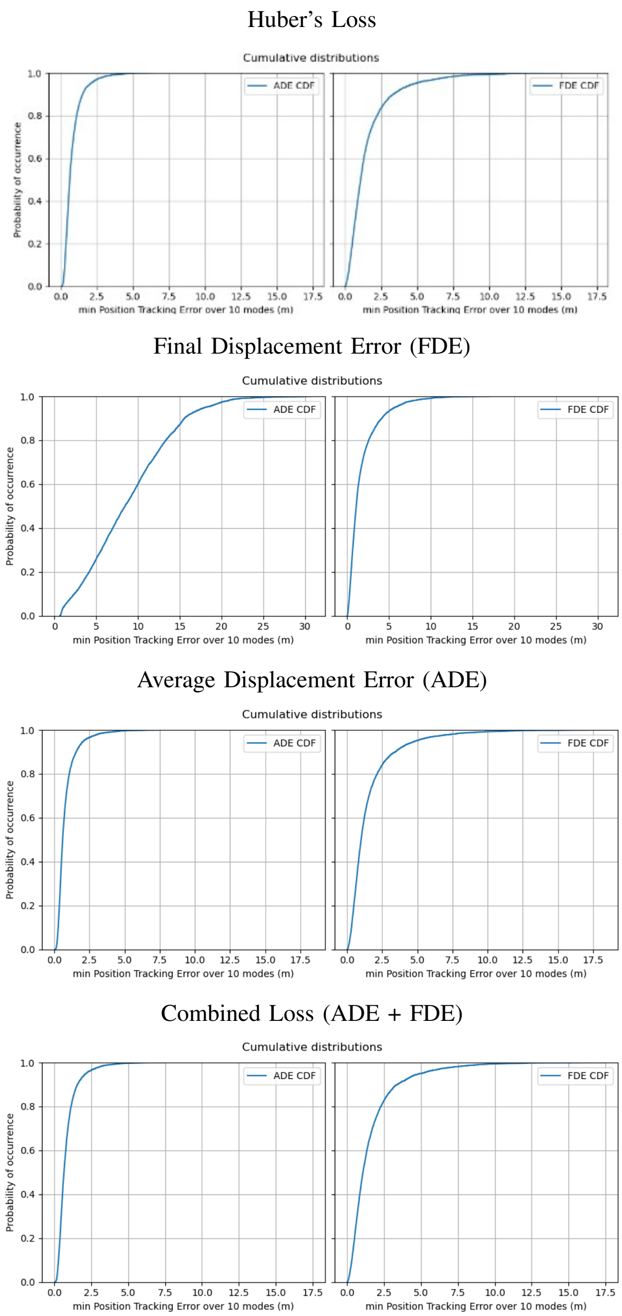

Fig. 1: Cumulative distribution function of minimum Final Displacement Error (minFDE) and minimum Average Displacement Error (minADE) for each loss function.

model to prioritize the final trajectory of the agent, which can lead to unsafe predictions.

This trend did not apply when testing the combined loss function, which consisted of both minimum final displacement error (minFDE) and minimum average displacement error (minADE). When analyzing the model's performance, all of the problems with the previous loss function were not a factor, as the model made accurate trajectory predictions on all time steps. Interestingly, the combined loss function produced similar results to minADE, but also outperformed it when measuring median minFDE, as shown in Table II. This suggests that the combined loss function of MinADE and MinFDE introduces a trade-off. Even though it may moderately improve the final position prediction while also maintaining an accurate overall trajectory prediction, using two loss functions increases the computational complexity, which may affect scalability and runtime. In Figure 1, the cumulative distribution curve further supports this observation, as the error is up to 3 meters for ADE and 7.5 for FDE, similar to the performance set initially by ADE alone. More fine-tuning of each weight could improve the trajectory prediction.

IV. Conclusion

In conclusion, when comparing the results of the minFDE and the combined loss function, it is evident that the combined loss function provides a more balanced representation of each trajectory for every instance in the time series. Additionally, it does not compromise or prioritize the accuracy of one timestep in the entire series, as seen in minFDE. For the combined loss function, there are improvements in median minFDE scores compared to the minADE loss function, while also maintaining similar results across the other metrics. Although the performance of the model doesn't drastically improve, it does showcase the potential benefit of using multiple loss functions.

A. Future Work

My work highlights the potential to improve trajectory prediction by using different loss functions. Future work could explore testing different combinations of loss functions to see if alternative perspectives can also enhance the model's performance. Additionally, using different weights for each component in the combined loss function could be beneficial to performance.

REFERENCES

[1] L. Shi, L. Wang, S. Zhou and G. Hua, ”Trajectory Unified Transformer for Pedestrian Trajectory Prediction,” 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2023, pp. 9641-9650, doi: 10.1109/ICCV51070.2023.00887.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan and O. Beijbom, ”nuScenes: A multimodal dataset for autonomous driving,” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 11621–11631.