A Vehicle Motion Prediction Approach for the 2021 Shifts Challenge

Abstract

This paper details the solution developed for the 2021 Shifts Challenge, which focused on robustness and uncertainty in real-world distributional shifts. The competition sought methods for addressing motion prediction in cross-domain scenarios. A key issue is the variance between input and ground truth data distributions, known as the domain shift problem. The method proposed features a novel architecture utilizing a self-attention mechanism and a specifically designed loss function. Ultimately, this approach achieved 3rd place in the competition.

1 Introduction

This paper examines the crucial issue of prediction in autonomous driving. Predicting vehicle trajectories to generate control commands is essential for avoiding collisions. While deep learning has shown promise in specific domains, real-world conditions, such as varying environments, weather, and driver behaviors, create challenges for models trained on single datasets. These models may not perform well across diverse datasets.

The 2021 Shifts Challenge concentrated on prediction tasks across different domains. The goal was to predict 25 timestamps of trajectories from given raster images. To address this, a new architecture was developed using insights from current research. The feature extractor was modified using NFNet for stability, and a self-attention layer was included to enhance time-related predictions. The loss function was also adjusted for improved robustness, leading to a 3rd place ranking with 8.637 R-AUC CNLL in the competition.

2 Our Solution

This section explains the solution for the domain-shift problem through the design of new model architectures. The domain-shift problem arises when training and validation datasets come from different distributions. Given input raster images X that contain the first 5 seconds of vehicle data, the objective is to predict the last 5 seconds of trajectories Y for the objects. These images include details about the positions, orientations, accelerations, and velocities of dynamic objects. The proposed model has two main parts: (1) a new backbone model and feature extractor, and (2) a revised loss function for better performance.

[width=0.8]./Recurrent$_m odel.png$

Figure 1: Base Model Architecture: The baseline model uses the backbone model to extract features and utilizes recurrent model to generate prediction according to latent vectors.

2.1 Baseline Model

The competition provided two baseline models and used an ensemble method to improve robustness. Both Behavior Cloning (BC) and Deep Imitation Model (DIM) use convolutional backbones to

.

convert raster image data into a latent vector, and then apply an autoregressive model to predict vehicle paths based on the latent vector. BC models the autoregressive likelihood as a single-variate Gaussian, while DIM uses a multivariate normal distribution. After assessing the performance of BC and DIM, BC was selected as the baseline due to its better performance. The BC method is broken down into two components: the feature extraction backbone and the recurrent model.

Feature Extraction Backbone Using the input raster image X, a feature extraction backbone and a self-attention layer (described below) are used to encode both spatial and temporal information about dynamic objects into a latent embedding.

$$Z = f(X) \tag{1}$$

The baseline applies MobileNetV1 as its backbone. MobileNetV2 and MobileNetV3 were also considered but produced worse results, likely due to the simplicity of input data and model complexity. Ultimately, the NFNet was chosen as the backbone (feature extractor) because of its training stability.

Self-Attention Layer To further refine the raster image features, a self-attention layer was incorporated. Self-attention, a key part of the Transformer model, allows for the consideration of long-range dependencies and global information. The feature map was divided into pixel groups, and self-attention was used to aggregate pixel-wise information.

Recurrent Model The GRU model was selected for the recurrent component due to superior performance compared to other models. Using the embedding from feature extraction as hidden states, the recurrent model makes predictions recursively. Given the embedding Zt at time t, with the output vector Y0 as zero vector, the recurrent model g is used to generate predictions:

$$Z_t = g_{encoder}(Y_{t-1}, Z_{t-1}) \tag{2}$$

$$Y_t = g_{decoder}(Y_{t-1}, Z_t) \tag{3}$$

Where $Y_t \in R^{B\times T\times 2}$ represents the vehicle's position on a 2D bird's-eye-view map, and $Z_t \in R^{B\times K}$ represents the hidden vector. B and T refer to the batch and time dimensions, respectively.

2.2 Loss Function

The model was initially trained using negative log-likelihood (NLL) loss. However, because of the inadequate performance of the model on Average Distance Error (ADE) and Final Distance Error (FDE), these metrics were added to minimize the distance between predicted and actual positions.

$$NLL(Y) = -log(p(Y)) \tag{4}$$

$$Loss = -\log(p(Y;\theta)) + \gamma_1||Y - \hat{Y}|| + \gamma_2||Y_f - \hat{Y}_f|| \tag{5}$$

Here, $p(Y;\theta)$ indicates the probability of a predicted trajectory Y based on model parameters θ. Y_f represents the trajectory's final location. In the equation, the first component is the original loss, the second is the ADE loss, and the last is the FDE loss.

2.3 Ensemble Method

To improve performance, the Robust Imitative Planning (RIP) method was employed to combine several models.

3 Experiments

3.1 Dataset and Evaluation

Dataset The dataset provided by Yandex Self-Driving Group was utilized for motion prediction. The training set contains 27036 scenes, and the testing set contains 9569 scenes. The dataset for the Shifts

Vehicle Motion Prediction includes 600000 scenes that vary in season, weather, location and time of day.

Evaluations metrics The evaluation used three metrics: Average Distance Error (ADE), Final Distance Error (FDE), and Negative log-likelihood (NLL). ADE measures the sum of squared errors between predicted and actual positions at each time step. FDE calculates the sum of squared errors of the final positions. NLL measures the unlikelihood of predicted trajectories matching the actual ones.

3.2 Implementation Details

Models were trained on a single V100 machine for one day, with a batch size of 512 and a learning rate of 1e-4. Input feature maps were resized to 128 x 128. The AdamW optimizer and gradient clipping with a value of 1.0 was used.

3.3 Ablation Study and Comparison Results

Ablation Study Table 1 displays the results of the ablation study. The baselines selected were DIM and BC. Various backbones, including EfficientNet, NFNet, and MobileNet, were compared, but models with more parameters performed worse. This result suggests that simpler models are sufficient for extracting raster image information. Adding a self-attention mechanism improved the results. Finally, incorporating ADE and FDE loss further improved performance, as shown in Table 1. Although the DIM method resulted in the lowest Negative Log Likelihood(NLL), it was not as competitive as other models. Therefore, the DIM model was not chosen to pursue performance.

Table 1: Ablation Study on Shift Vehicle Motion Prediction Dataset

Method	ADE↓ In Domain	FDE↓	NLL↓	ADE↓ Out of Domain	FDE↓	NLL
DIM + MobileNetV2(baseline)	2.450	5.592	-84.724	2.421	5.639	-85.13
BC + MobileNetV2(baseline)	1.632	3.379	-42.980	1.519	3.230	-46.88
BC + NFNet18	1.225	2.670	-53.149	1.300	2.893	-53.13
BC + NFNet50	1.360	2.963	-50.605	1.392	3.066	-51.31
BC + NFNet18 + Attention	1.174	2.549	-56.199	1.325	2.852	-54.47
BC + NFNet50 + Attention	1.155	2.504	-56.291	1.265	2.770	-54.73
BC + NFNet18 + ADE Loss	1.197	2.55	-54.047	1.299	2.821	-53.05
BC + NFNet18 + Attention + ADE Loss	1.139	2.488	-55.208	1.227	2.714	-54.28

Comparison Results After verifying the base model's effectiveness, the aggregation model, RIP, was used along with the Worst Case Method (WCM). The WCM method samples multiple predictions per model and picks the one with the lowest confidence for more reliable results. Table 2 shows the competition results, where our model outperformed baselines in weighted sums of ADE and FDE. However, the MINADE and MINFDE results were not as strong. Overall, this approach secured 3rd place.

Table 2: Quantitative Result of Top3 Final Submission: CNLL represents the weighted sum of NLL; WADE represents the weighted sum of ADE; WFDE represents the weighted sum of FDE;

Rank	Method	Score (R-AUC CNLL)	CNLL↓	WADE↓	WFDE↓	MINADE↓	MINFDE↓
-	baseline	10.572	65.147	1.082	2.382	0.824	1.764
1	SBteam	2.571	15.676	1.850	4.433	0.526	1.016
2	Alexey & Dmitry	2.619	15.599	1.326	3.158	0.495	0.936
3	Ours	8.637	61.864	1.017	2.264	0.799	1.719

4 Conclusion

In this challenge focused on distributional shifts, we introduced a novel base model architecture, which combined with an ensemble method, yielded competitive results. Other state-of-the-art methods were implemented, and results were compared with analysis. The robustness of the provided ensemble method was verified. This methodology resulted in the third prize in the competition.