Motion Forecasting via Coordinate Transformations and Object Trajectory Modifications

Jungwan Woo*, Jaeyeul Kim*, Sunghoon Im
Department of Electrical Engineering & Computer Science, DGIST, Daegu, Korea
{friendship1, jykim94, sunghoonim}@dgist.ac.kr

1. Introduction

For autonomous driving, it is essential to proficiently detect and track multiple objects on the road, including but not limited to vehicles, motorcycles, and pedestrians. Furthermore, the prediction of the future movement of objects is indispensable in realizing an impeccable and safe autonomous driving system. While the domains of detection and tracking have been extensively studied, research on predicting objects' future motion is still nascent. Recent work [4] introduces a new metric, denoted as $\mathrm{mAP_f}$, which incorporates a penalty for false positives for prediction tasks, which has not been considered in existing ADE or FDE. Based on the evaluation criteria, we propose an LSTM-based predictive framework tailored for forecasting the objects' future motion. The framework employs a three-tiered approach: commencing with object detection, followed by tracking, and culminating in predictions from the tracking results. For the path prediction task, a coordinate transformation is executed, transforming from the world coordinate system to the individual object coordinate system. In addition, we propose effective data augmentation strategies, conceived specifically to enhance the accuracy of forecasting tasks. We achieve 42.91 $\mathrm{mAP_f}$ in the Argoverse 2 Forecasting Challenge held at The CVPR 2023 Workshop on Autonomous Driving (WAD), which demonstrates the effectiveness of the proposed method.

2. Method

Our approach is delineated into three phases: detection, tracking, and forecasting. The detection phase yields output that is instrumental for tracking object trajectories, which is subsequently employed to generate the final forecasting results. This tripartite structure facilitates the integration of techniques from existing literature, enabling an assessment of the performance contribution form each part of the process.

*Denotes equal contribution

2.1. Detection and Tracking

The Argoverse 2 [6] sensor dataset encompasses 26 classes with a long-tail data distribution that poses inherent challenges for perception tasks. To address this, we employ the LT3D [3] detector, which leverages nested class labels to counterbalance the uneven distributions. We adopt the Centerpoint [8] variant of the LT3D with a voxel size of $0.075m$. The detection model is trained with five stacked LiDAR frames, employing data augmentation methods as introduced in previous work [8]. Notably, RGB filtering, which is a post-processing method used in original work [3], is not adopted in inference time. Subsequent to the detection output, we leverage a non-learning-based tracking method, AB3DMOT [5], to meticulously trace the trajectories of the objects. This model utilizes the capabilities of a 3D Kalman filter coupled with the Hungarian algorithm for multi-object tracking.

2.2. Forecasting

2.2.1 Base architecture and loss

Following the baseline framework, we opt for an LSTM [1] based model to forecast future object positions, capitalizing on the adeptness of LSTMs in processing time-series data. Our model ingests the features from preceding timesteps and predicts all six future frames with intervals of 0.5 second. As illustrated in Fig. 1, we designed the model to predict all six future frames concurrently at the current LSTM node, rather than invoking the LSTM iteratively for each frame. The input features fed to the LSTM encompass information regarding the object's position, velocity, and angle. The output label, used for training, is the incremental change in the movement distance between each timestep. The model is trained using ground-truth data, which consists of the positional differences between subsequent timesteps. The training data is generated by direct extraction from the ground-truth dataset, and not derived from predictions.

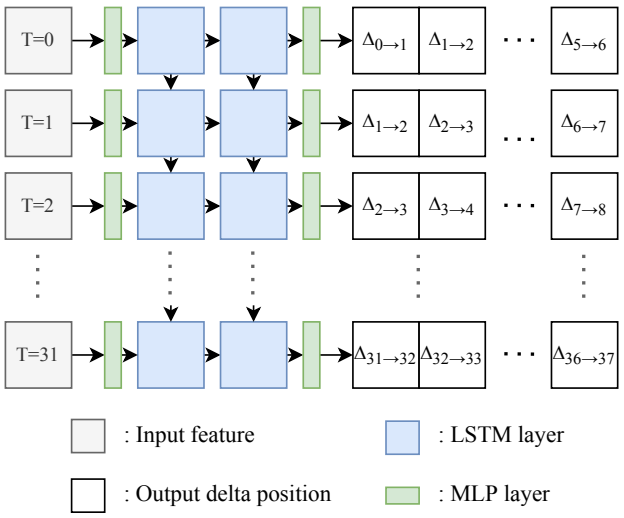

Figure 1. Schematic diagram of the proposed forecasting model. It consists of two MLP and LSTM layers of input and output. It ingests data pertaining to the angle, position, and velocity for each timestep, and yields the positional differences for the subsequent six frames relative to the preceding frame. Numerical annotations within the diagram serve illustrative purposes to facilitate comprehension. The number of input frames, depicted as 32 in the example, may vary during the training and inference phases.

2.2.2 Warping from world to object-wise coordinate

When predicting a path based on the world coordinate system for road-bound objects, there is a tendency for an escalation in the variance of the predicted value. This escalation in the output variance compromises the generalization capabilities of the forecasting framework. To alleviate this issue, we transform each object's coordinates to an object-wise centric perspective, which serves to enhance the forecasting results. As depicted in Fig. 2, this transformation entails centralizing the translational component and rotating the heading direction in accordance with the initially predicted yaw angle ascertained from the detection model. Concurrently, the discrepancy in the target position relative to the ground-truth is rotated by the corresponding angle.

2.2.3 Class conditioned Gaussian noise

Throughout the testing phases, the methodology commences afresh with detection and tracking. This intrinsic sequence engenders a domain gap between the training and testing phases. owing to this, the testing phase is subject to imprecise input features, which potentially leads to significant degradation in the forecasting performance. To tackle this issue, we emulate the compromised input features by injecting random Gaussian noise. Given the dataset's heterogeneity in terms of object classes and a broad spectrum of speed variances, we employ disparate noise intensities to each class (See Fig. 3(a)). We abstain from implementing class-wise noise intensity modulation for the heading angle feature, given its independence from the average velocity per class.

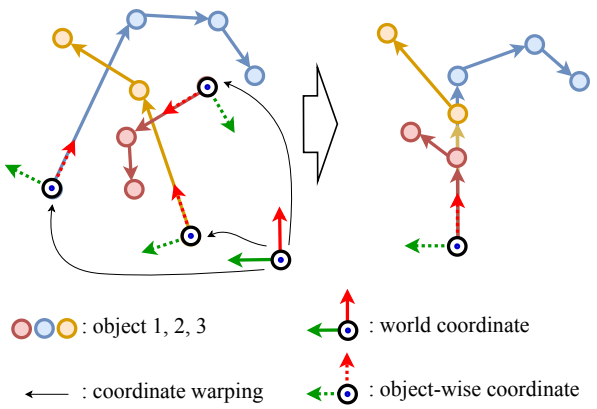

Figure 2. Trajectories of multiple objects in time series are expressed as nodes and edges. Through the transformation method from world coordinate system to each object coordinate system, the paths of objects that exist in the world coordinate system are warped based on the heading direction of their first frame. In addition to position, the same transformation applies to velocity and angle input features.

2.2.4 Missing timestep simulation

For the same reasons as mentioned in Sec. 2.2.3, we employ random masking of timesteps within each ground-truth trajectory to simulate the prediction domain. Within the training domain, which is grounded in the ground-truth, most objects are tracked with high fidelity, barring instances of occlusion. Conversely, within the prediction domain, trajectory discontinuities are commonplace. To emulate this prediction degradation in prediction, we randomly select timesteps within the ground-truth trajectory and mask a subset of them (See Fig. 3(b)). The data pertaining to the masked timesteps is synthesized through linear interpolation between the adjacent timesteps.

2.3. Implementation Details

The implementation details of the proposed method are as follows. We employ the Kaiming uniform initialization method [2] on the initial input projection MLP layer to maintain the magnitude of the input vector. Our forecasting model is optimized with the minimum loss among five predictions in every iteration of the training phase. It is essential to enforce that each head predicts different outcomes. Rather than enforcing additional regularization losses or constraints for this issue, we ascertain the optimal initial network weight via iterative random initialization and evaluation on the validation set. The model exhibiting the high-

		Average	static			linear			non-linear		
Detector	Tracker	mAP_f	mAP_f	ADE	FDE	mAP_f	ADE	FDE	mAP_f	ADE	FDE
Centerpiont [8]	Greedy	43.93	69.14	0.33	0.39	40.05	3.92	4.67	4.43	1.99	3.39
+ LT3D [3]	AB3DMOT	45.57	69.80	0.33	0.40	45.37	3.89	4.58	2.59	2.07	3.52
Transfusion-L [7]	Greedy	38.44	53.40	0.29	0.36	44.26	3.45	4.00	3.41	5.16	6.31
+ LT3D [3]	AB3DMOT	39.37	53.62	0.29	0.36	46.10	3.45	4.01	4.33	5.17	6.38
GT	GT	72.34	99.62	0.12	0.20	83.82	0.43	0.83	7.22	0.95	1.97

Table 1. Comparison between different detection and tracking model choices. Combined with our proposed forecasting model, we measure the forecasting performance on the Argoverse 2 validation set. The evaluation metric is measured using the code provided by the challenge organizer.

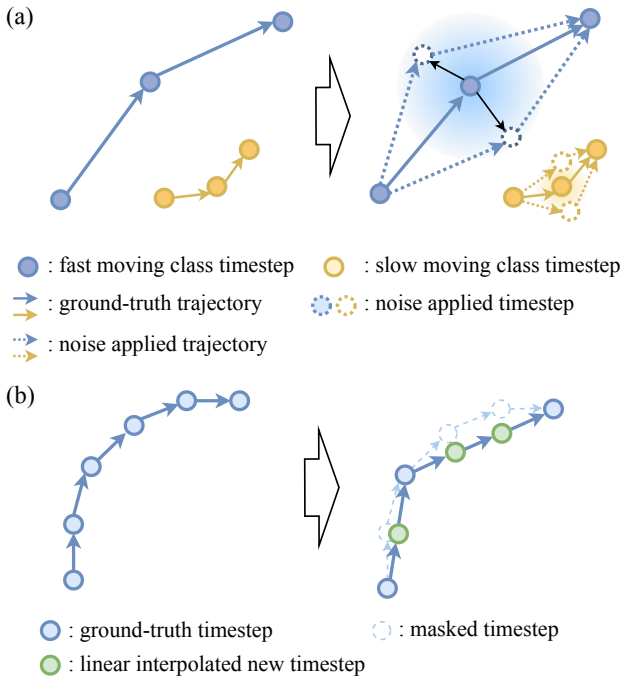

Figure 3. Trajectories of multiple objects in time series are expressed as nodes and edges. (a) From the per-class velocity statistics, we add class conditional Gaussian noise to the input feature. The faster the average velocity of an object class, the stronger the applied Gaussian noise. (b) The random timestep simulation is designed to make the input ground-truth similar to predicted input values.

est $\mathrm{mAP_f}$ on the validation set, among several models initialized and trained on ten disparate random seeds, is selected. In addition, we augment the training data using the trajectory flipping scheme. We flip the input translation and velocity feature coordinates along both the x-axis and y-axis and adjust the angle accordingly.

3. Experiments and Analysis

In Tab. 1, we compare several detection and tracking methods with our forecasting model on the validation set. Our experimentation incorporated Centerpointnet [8] and Transfusion-L [7] as candidates for detector models. For tracking, the greedy method and the AB3DMOT [5] method were evaluated as viable candidates. As shown in the second line of Table 1, the optimal $\mathrm{mAP_f}$ performance is attained through the amalgamation of Centerpoint for detection and AB3DMOT for tracking predictions. While the detection performance of the Transfusion-L detector outperforms that of Centerpoint, the discrepancy in the Non-Maximum-Suppression (NMS) post-processing rendered Centerpoint more efficacious. The transition from the Greedy method to AB3DMOT for tacking substantiated and enhancement in performance of the future motion prediction framework. In addition, an $\mathrm{mAP_f}$ performance of 72.34 was achieved when the ground-truth data served as the input for tracking prediction. Even if the ground-truth value is used, it can be seen that our model exhibits limitations in making accurate predictions in non-linear cases. For the competition, we integrated a Centerpoint-based detector with AB3DMOT tracking method and our forecasting model, attaining an $\mathrm{mAP_f}$ of 42.91 across 26 classes on the AV2 test set.

4. Conclusion

In this challenge, we found the most effective detection and tracking framework for future motion prediction. We incorporated object-wise coordinate transformation and infused noise into the tracking information serving as input data during the training process to mitigate the domain gap between training and testing. Additionally, by employing missing timestep simulation and flip augmentation, we bolstered performance. These strategies effectively enhance forecasting performance by bridging the domain gap between training and testing datasets. Our experiments demonstrated that superior performance is achieved when ground-truth trajectories are served in the test environment. This underscores the challenges inherent in the forecating task, namely the domain gap between training and testing, as well as limitations within the forecasting network. In conclusion, to ensure higher prediction performance, it is

essential to either enhance the performance of the tracking algorithm or incorporate the intrinsic errors of the tracking algorithm into the training phase. Therefore, our future research endeavors will concentrate on augmenting detection and tracking capabilities and the development of an integrated framework that seamlessly performs detection, tracking, and motion prediction, thereby allowing for gradients to flow across each component.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[2] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE international conference on computer vision*, 2015. 2

[3] Deva Ramanan Shu Kong Neehar Peri, Achal Dave. Towards long-tailed 3d detection. In *Conference on Robot Learning*, 2022. 1, 3

[4] Mengtian Li Aljoša Ošep Laura Leal-Taixé Deva Ramanan Neehar Peri, Jonathon Luiten. Forecasting from lidar via future object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

[5] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. *IROS*, 2020. 1, 3

[6] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. 1

[7] Xinge Zhu Qingqiu Huang Yilun Chen-Hongbo Fu Xuyang Bai, Zeyu Hu and Chiew-Lan Tai. TransFusion: Robust Lidar-Camera Fusion for 3d Object Detection with Transformers. *CVPR*, 2022. 3

[8] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021. 1, 3