Practical Self-Supervised Contrastive Driver Maneuver Interaction Learning via Augmenting Inertial Measurement Unit Signals

Yawen Deng Suining He[*] Hao Wang
{yawen.deng, suining.he, hao.3.wang}@uconn.edu
School of Computing
University of Connecticut

***Abstract*— Driver maneuver interaction learning (DMIL), i.e., learning and classifying the maneuver types (e.g., left or right turns, acceleration and deceleration) of a driver, is essential for developing advanced driver assistance systems and understanding driver behaviors in complex traffic environments. In this study, we propose $\texttt{S}^2\texttt{-DMIL}$, a self-supervised contrastive DMIL framework. We focus on the inertial measurement unit (IMU) sensors, i.e., accelerometer and gyroscope, and aim to (a) reduce the reliance on large-scale labeled or annotated IMU for DMIL; and (b) capture crucial and meaningful representations for accurate DMIL in complex traffic environments. To this end, we have performed data augmentation upon time series of the unlabeled IMU signal data (e.g., through cropping, permutation), and designed self-supervised contrastive learning to capture the important representations. Within the contrastive learning process, we have implemented efficient convolution blocks as the feature encoding module, and pre-train it for the subsequent maneuver classification. Based on the pre-trained feature encoding module from the self-supervised contrastive learning, we further fine-tune $\texttt{S}^2\texttt{-DMIL}$ based on labeled driver maneuvers toward classification of the complex driver behaviors. We have conducted extensive data-driven studies upon a total of 5,799 samples from the open-sourced dataset (Berkeley Deep Drive-X). Our results demonstrate that our $\texttt{S}^2\texttt{-DMIL}$ outperforms the other baseline approaches (e.g., by about 12% on average in terms of accuracy) in learning the complex driver maneuvers.**

***Index Terms*— self-supervised learning, driver maneuver interaction learning, inertial measurement unit.**

I. Introduction

Accurate driver maneuver interaction learning (DMIL), i.e., learning and classifying the driver maneuvers, such as left or right turns, is crucial for developing advanced driver assistance systems and connected autonomous vehicles. Furthermore, recognizing and categorizing driver maneuvers can help mitigate potential hazards on the roads [1]. DMIL can serve as the fundamental basis for understanding how to reduce the inattentive, irresponsible, or aggressive driving situations as well as human errors. Drivers, particularly those novice drivers (learners), can also receive timely and potentially helpful feedback, and promote more accountable driving in the complex traffic environments [2].

To this end, in this work, we aim to develop a driver maneuver interaction learning approach, and, in particular, focus on studying inertial measurement unit (IMU) signals, i.e., accelerometer and gyroscope, for DMIL. Unlike other modalities (say, GPS, vision, or physiological sensors), IMU signals are pervasive on smartphones and many on-board devices, and can support more ubiquitous DMIL under complex traffic environments (e.g., GPS-less environments like tunnels or near skyscrapers, low-lights). While prior studies [2], [3] have investigated learning driver maneuvers, there are two important challenges for practical DMIL. First, realizing accurate DMIL hinges on the driver maneuver data. However, IMU signals, despite being easy to harvest, can also be tedious and labor-intensive to label and annotate (i.e., providing the classes or categories of turnings) in practice. How to further leverage the unlabeled maneuver data to extract their potential values for DMIL remain largely underexplored. Second, DMIL also requires the extraction of meaningful and important representations from the complex and largely noisy maneuver data (say, the IMU signals in our case). Existing driver maneuver learning approaches [2], [3] largely rely on extensive and dedicated feature engineering. How to effectively and adaptively capture the essential representations from these IMU signals for accurate DMIL is worth further investigation.

To overcome the above challenges, we propose $\texttt{S}^2\texttt{-DMIL}$, a Self-Supervised contrastive Driver Maneuver Interaction Learning approach based on IMU signal augmentation. In this paper, we have made the following three major contributions. First, we have designed the IMU signal augmentation mechanism for DMIL. We have investigated multiple augmentation methods, including adding Gaussian noise, permutation, cutout, and stretch and squeeze, based on the multivariate time series of IMU signals. Our goal is to construct the semantic and contextual correlations between the IMU data and the driver maneuvers. Second, based on the IMU signal augmentation mechanism, we have designed the self-supervised contrastive learning based on efficient convolution blocks. Our practical designs further identify and extract the important and meaningful representations. Third, we have performed extensive experimental studies upon a total of 5,799 samples from the open-sourced dataset (Berkeley Deep Drive-X) [4]. Our experimental results have corroborated that our $\texttt{S}^2\texttt{-DMIL}$ outperforms the other baseline approaches, say, by about 12% on average in terms of accuracy, in learning the complex driver maneuvers.

The rest of the paper is organized as follows. We first review the related work in Section II. We then present

[*] Contact Author (suining.he@uconn.edu)

2993-396X/24/$31.00 ©2024 IEEE
DOI 10.1109/SWC62898.2024.00154

the system framework and data processing in Section III. After that, we present the self-supervised contrastive DMIL designs in Section IV. We show the experimental results in Section V, and finally conclude the paper in Section VI.

II. Related Work

We briefly review the related studies in terms of DMIL. In order to accurately estimate the driver maneuvers, various machine learning approaches have been studied for capturing the features of the driver maneuver behaviors. For instance, the Hidden Markov model [5] and Bayesian network [6] have been studied to characterize the driver maneuver behaviors. With the emergence of deep learning [7], [8], methods such as deep neural network [9], recurrent neural network [10], long short-term memory [11], and bidirectional long short-term memory [10], [12], have been explored for various DMIL scenarios (e.g., lane changes, intent estimation, and context awareness). For instance, Olabiyi et al. [12] designed the sequence learning techniques to capture the correlations between the input sensor information and future driver maneuvers to achieve maneuver behavior prediction. Recent designs regarding the learning paradigms, such as meta learning [3] and federated learning [13], have also been investigated for ubiquitous DMIL in terms of model adaptation and distributed model training. In addition to learning methods, various sensing modalities [11], [14], such as steering wheel angles [9], gaze recognition, in-vehicle camera [10], wireless channel state information [14], and on-board diagnostics dongles [10], have been taken into account for DMIL. For instance, Kim et al. [9] studied various on-board sensor measurements, including steering wheel angles, yaw rates, and throttle positions, to discern the road conditions and predict the driver intention and behaviors (e.g., lane changes). However, these designs largely rely upon dedicated sensing hardware within the vehicles, and may limit the ubiquitousness of the DMIL deployment.

Our work here differs from these prior studies in the following aspects. First, our studies here focus on leveraging multivariate IMU signals, particularly the unlabeled or unannotated ones for DMIL, aiming to reduce the reliance upon labor-intensive data labeling for large-scale and more ubiquitous DMIL. Second, we have designed the self-supervised contrastive learning to automatically capture the meaningful and important features for DMIL. Third, while contrastive learning have been explored for sleep staging [15] and general human activity recognition [16], how to adapt to the complex multivariate IMU signals for DMIL remains largely unexplored. Our studies and designs in developing the learning paradigm of S^2-DMIL can fill this gap and unveil the practical niches in augmenting and learning from the complex IMU signals.

III. System Framework and Data Processing

We first overview the data studied and our data processing in Sec. III-A, and then present the workflow of S^2-DMIL in Sec. III-B.

A. Driver Maneuver Data Studied and Data Processing

Our study here focuses on identifying the following typical driver maneuvers, i.e., left turns, right turns, acceleration, deceleration, and U-turns, from the open-sourced Berkeley Deep Drive-X (BDD-X) dataset [4]. In particular, we have extracted the accelerometer and gyroscope time series that correspond to the intended driving maneuvers based on the action and scene descriptions available in the BDD-X dataset. We note that the accelerometer and gyroscope readings are sampled at 50Hz by iPhone 5, provide the 6-D time series data with nanosecond precision.

We note that as the IMU signals are collected when the smartphone is mounted on the vehicle's windshield with a phone holder, we have further rotate the IMU signal readings from the local coordinate system to align with the global (earth) coordinate system, and obtain the transformed IMU signals for DMIL. In our current studies, the resulting x, y, and z axes point toward the front, left, and upward directions [4].

Specifically, each sample, indexed by i, of the rotated acceleration time series, denoted as $\mathbf{A}_i \in \mathbb{R}^{3\times L}$ ($i \in \{1,\ldots,N\}$), consists of three time series that span over a total of L timestamps, i.e.,

$$\mathbf{A}_i = [a_i^{x1}, a_i^{x2}, \ldots, a_i^{xL}; a_i^{y1}, a_i^{y2}, \ldots, a_i^{yL}; a_i^{z1}, a_i^{z2}, \ldots, a_i^{zL}]. \quad (1)$$

Similarly, we have the rotated gyroscope time series $\mathbf{G}_i \in \mathbb{R}^{3\times L}$ ($i \in \{1,\ldots,N\}$) as

$$\mathbf{G}_i = [g_i^{x1}, g_i^{x2}, \ldots, g_i^{xL}; g_i^{y1}, g_i^{y2}, \ldots, g_i^{yL}; g_i^{z1}, g_i^{z2}, \ldots, g_i^{zL}]. \quad (2)$$

Each driver maneuver sample, denoted as $\mathbf{T}_i \in \mathbb{R}^{L\times 6}$ ($i \in \{1,\ldots,N\}$), consists of $\mathbf{A}_i$ and $\mathbf{G}_i$, i.e., $\mathbf{T}_i = \left[\mathbf{A}_i^\intercal | \mathbf{G}_i^\intercal\right]$. We provide the label under one-hot encoding for each maneuver sample as $\mathbf{Y}_i$ ($i \in \{1,\ldots,N\}$) that correspond to the types of driver maneuver behaviors. More specifically, we have $\mathbf{Y}_i[c] = 1$ if a maneuver sample i belongs to a class c (say, left turn), and 0 otherwise. In this study, we empirically set the sliding window of each maneuver as 2.5s, and $L = 125$.

To summarize, we have identified a total of 5,799 driver maneuver samples, i.e., 1,408 left turns (LTs), 1,870 right turns (RTs), 1,595 acceleration (AC) samples, 741 deceleration (DC) samples, and 185 U-Turns (UTs). The goal of S^2-DMIL is to return the types of driver maneuvers $\hat{Y}_i$ ($i \in \{1,\ldots,N\}$) such that their differences with the ground-truth labels Y_i ($i \in \{1,\ldots,N\}$) are minimized.

In preparing our maneuver data, we have empirically evaluated filters such as the Kalman filter [3], [17] and the low-pass filter such as the moving average, and select the Kalman filter for smoothing the IMU sensor readings (following the practices in [3]). We have also examined the time-series data processing methods [18] such as the min-max standardization and the z-score standardization, and choose the z-score standardization in our current studies.

B. Workflow Overview of S^2-DMIL

Fig. 1 illustrates the entire work flow of S^2-DMIL, whose framework consists of the following three major modules, i.e., the IMU signal augmentation, the feature encoder, and the maneuver classifier. The IMU signal augmentation

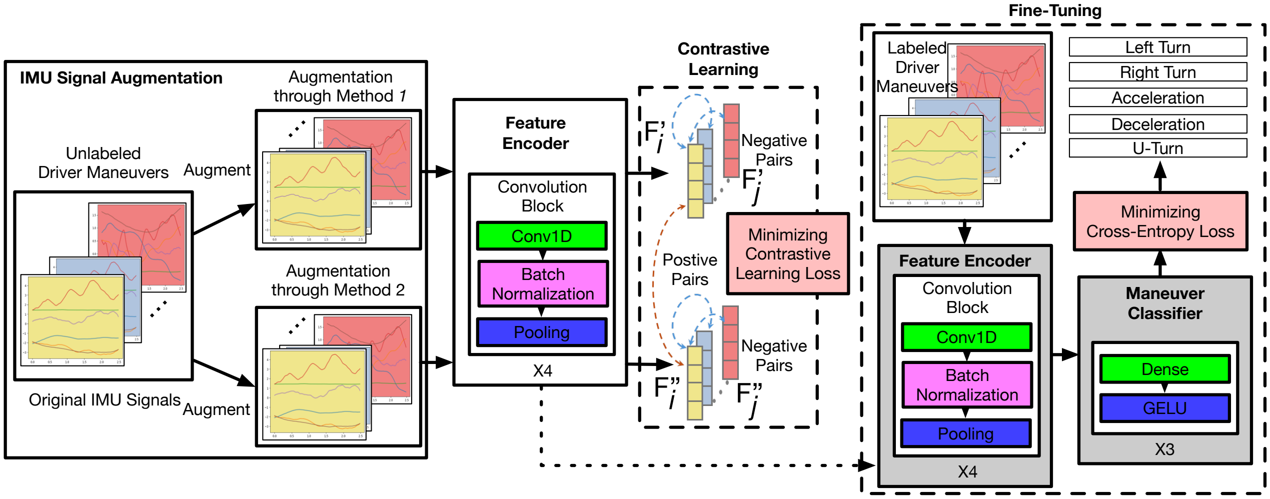

Fig. 1: Overall workflow of S^2-DMIL, that consists of the IMU signal augmentation, the feature encoder, and the maneuver classifier. Contrastive learning and fine-tuning are implemented to extract meaningful representations and classify the driver maneuvers. In the self-supervised contrastive learning, the transformed IMU signals from two augmentation methods are respectively fed to the feature encoder, and we obtain the encoded features. We consider the embeddings $\mathbf{F}'_i$ is close to the embedding of $\mathbf{F}''_i$ since they are transformed from the same original driver maneuver sample, but should be distant from the embeddings of other samples ($j \neq i$).

module first transforms the original time series of the input IMU signals into two kinds of time series. Then, S^2-DMIL passes the two kinds of augmented IMU signals through the feature encoder, and performs the contrastive learning based on the output embeddings (differentiating the positive and negative pairs based on the contrastive learning loss). The key idea is to capture the shared useful representations across the augmentations. After that, we further fine-tune the feature encoder and the maneuver classifier using the labeled driver maneuvers (based on the cross-entropy loss).

IV. Self-Supervised Contrastive Learning on Driver Maneuvers

We first overview the data augmentation phase in Sec. IV-A, followed by the detailed designs of the feature encoder and the maneuver classifier in Sec. IV-B.

A. IMU Signal Augmentation

In this study, we have designed and examined the following IMU data augmentation method for S^2-DMIL, which are presented as follows. We note that after a certain data augmentation method upon a driver maneuver sample, denoted as $\mathbf{T}_i$, we obtain the augmented one, denoted as $\mathbf{T}'_i$.

1) **Gaussian Noise**: We impose the Gaussian noise following (μ, σ^2) upon each reading of the six axes of the original IMU signals. Through the imposed Gaussian noise, we can examine the robustness of the discovered features within the IMU signals and strengthen the self-supervised learning performance.
2) **Permutation**: For each of the six axes, we divide the original IMU signals into M consecutive segments of equal length, i.e., $\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_M\}$. We then shuffle these M segments through permutation, and concatenate them into a time series.
3) **Cutout**: For each of the six axes, given the divided M consecutive segments of equal length, i.e., $\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_M\}$, we randomly discard one segment, concatenate the rest, and resize the time series of length of $(L - L/M)$ back into the length L.
4) **Stretch and Squeeze**: For each of the six axes, given the divided M consecutive segments of equal length, i.e., $\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_M\}$, we randomly stretch or squeeze each segment through a scaling factor β. The resulting segments are then concatenated and resized back into a time series of the original length L.
5) **Crop**: For each of the six axes, we randomly select a starting index t_0 from $\{1, 2, \ldots, L - K + 1\}$, find a segment of readings $\{t_0, t_0 + 1, \ldots t_0 + K - 1\}$, and resize the segment from K to L through interpolation.
6) **Inversion**: For each of the six axes, we reverse the time series and obtain the inverted IMU signal. For instance, we convert $\{a_i^{x1}, a_i^{x2}, \ldots, a_i^{xL}\}$ into $\{a_i^{xL}, a_i^{x(L-1)}, \ldots, a_i^{x1}\}$, and similarly for the other axes.

B. Model and Learning Designs

Overview: We further present the designs of the feature encoder and the maneuver classifier as follows. We note that the feature encoder will serve as the pre-trained module that will be first pre-trained during the contrastive learning based on the unlabeled maneuver data that undergo the IMU signal augmentation mechanism. Then we will fine-tune both the feature encoder and the maneuver classifier upon the labeled maneuver data to perform the DMIL and identify the correct maneuvers.

Feature Encoder: Our feature encoder consists of four consecutive convolution blocks. Within each convolution block, we have designed two sets of 1-D convolution layers $\texttt{Conv1D}(\cdot)$. Each $\texttt{Conv1D}(\cdot)$ consists of a kernel width of 5 dilated to a width of 9 using a dilation rate of 2. The two sets of $\texttt{Conv1D}(\cdot)$ are interleaved by the $\texttt{GELU}$ activation function $\sigma(\cdot)$, batch normalization $\texttt{BN}(\cdot)$, and the max pooling operation $\texttt{PL}(\cdot)$ for downsampling. In other words, given the input $\mathbf{T}_i \in \mathbb{R}^{L \times 6}$, we consider each sensor axis as a channel, and each block indexed by j ($j \in 1, \ldots 4$) performs the following operation, and obtain the feature embeddings $\mathbf{F}_i^j$, i.e.,

$$\mathbf{F}_i^j = \texttt{PL}(\texttt{BN}(\sigma(\texttt{Conv1D}(\texttt{PL}(\texttt{BN}(\sigma(\texttt{Conv1D}(\mathbf{F}_i^{j-1})))))))). \tag{3}$$

In the first block, we feed the input tensor of IMU signals $\mathbf{F}_i^1 = \mathbf{T}_i$, and we denote the output from the last block as $\mathbf{F}_i$ ($\mathbf{F}_i^4$) for ease of description in the subsequent contrastive learning.

Contrastive Learning on the Feature Encoder: We present the details of the contrastive learning designs as follows. In our current work, regarding each driver maneuver sample $\mathbf{T}_i$ ($i \in \{1, \ldots, N\}$), we perform the IMU signal augmentation based on any two methods presented in Sec. IV-A, and obtain $\mathbf{T}_i'$ and $\mathbf{T}_i''$, respectively. We note that the same augmentation method can be applied twice to one maneuver sample. Our aim here is to leverage the two augmented samples to compare against other samples, and obtain the contrasted features for DMIL.

The transformed IMU signals $\mathbf{T}_i'$ and $\mathbf{T}_i''$ are respectively fed to the feature encoder, and we obtain the encoded features, $\mathbf{F}_i'$ and $\mathbf{F}_i''$. We consider the embeddings $\mathbf{F}_i'$ is close to the embedding of $\mathbf{F}_i''$ since they are transformed from the same original driver maneuver sample, but should be distant from the embeddings of other samples. Then, our contrastive learning loss, denoted as ℓ_{cl}, is formally given by

$$\ell_{\mathrm{cl}} = -\sum_{i=1}^{N} \log \frac{\exp\left(\mathrm{sim}\left(\mathbf{F}_i', \mathbf{F}_i''\right)/\tau\right)}{\sum_{* \in \{', ''\}} \sum_{j=1}^{N} \mathbb{1}_{i \neq j} \exp\left(\mathrm{sim}\left(\mathbf{F}_i^*, \mathbf{F}_j^*\right)/\tau\right)}, \tag{4}$$

where $\mathrm{sim}\left(\mathbf{F}_i', \mathbf{F}_i''\right) = \frac{\mathbf{F}_i' \cdot \mathbf{F}_i''}{\|\mathbf{F}_i'\| \cdot \|\mathbf{F}_i''\|}$ represents the cosine similarity between the two feature embeddings, $\mathbb{1}_{i \neq k}$ is an indicator function that is 0 when $i = k$ and 1 otherwise, and τ is a temperature parameter used to adjust the scale. This loss function encourages the feature encoder to generate similar feature embeddings for positive pairs as close as possible, and differentiate the feature embeddings for negative pairs.

Maneuver Classifier and Fine-Tuning: Given the pre-trained feature encoder after the contrastive learning, we will connect it with the maneuver classifier to perform fine-tuning and subsequent classification. In our maneuver classifier, we have implemented three consecutive linear layers (denoted as $\texttt{LN}$) integrated with with the $\texttt{GELU}$ activation function $\sigma(\cdot)$. Specifically, with the embeddings output from the feature encoder, denoted as $\mathbf{F}_4$, we have the maneuver classifier output as

$$\hat{\mathbf{Y}}_i = \texttt{Softmax}(\texttt{LN}(\sigma(\texttt{LN}(\sigma(\texttt{LN}(\mathbf{F}_4)))))), \tag{5}$$

where $\texttt{Softmax}(\cdot)$ represents the softmax function for multi-class classification, and $\hat{\mathbf{Y}}_i$ represents the label of a driver maneuver sample i under one-hot encoding (i.e., $\hat{\mathbf{Y}}_i[c] = 1$ if a maneuver sample i belongs to a class c, and 0 otherwise). In our work, the fine-tuning of $\texttt{S}^2\texttt{-DMIL}$ aims to minimize the cross entropy of the labeled samples, i.e.,

$$\ell_{\texttt{ft}} = \sum_{i=1}^{N} \texttt{CEL}\left(\hat{\mathbf{Y}}_i, \mathbf{Y}_i\right), \tag{6}$$

where the function $\texttt{CEL}(\hat{\mathbf{Y}}_i, \mathbf{Y}_i)$ represents the cross entropy between the classification $\hat{\mathbf{Y}}_i$ and the ground-truth $\mathbf{Y}_i$.

V. Experimental Studies

We first overview the following baselines compared and our experimental settings in Sec. V-A. After that, we present the experimental results in Sec. V-B.

A. Experimental and Evaluation Settings

We compare our $\texttt{S}^2\texttt{-DMIL}$ with the following baseline approaches: support vector machine ($\texttt{SVM}$) with radial basis function ($\texttt{RBF}$) kernel (with the hyperparameter C = 1,000), k-nearest neighbors ($\texttt{KNN}$; with 20 nearest neighbors), long short-term memory ($\texttt{LSTM}$), bidirectional long short-term memory ($\texttt{BiLSTM}$), gated recurrent unit ($\texttt{GRU}$), convolution neural network ($\texttt{CNN}$), and $\texttt{Transformer}$ [19]. We also compare $\texttt{S}^2\texttt{-DMIL}$ under various combinations of IMU signal augmentation, with $\texttt{S}^2\texttt{-DMIL}$ without the self-supervised learning (denoted as $\texttt{S}^2\texttt{-DMIL}$ w/o SSL).

We present the parameter settings of IMU signal augmentation as follows. In terms of Gaussian Noise, we set $\mu = 1.0$ and σ is randomly selected from [10, 20]. In terms of Permutation and Cutout, we set $M = 5$. In terms of Stretch and Squeeze, we set the scaling factor β either in a range of [1.5, 4] for the stretch operation, or [0.25, 0.7] for the squeeze operation. In terms of Crop, we set $K = 25$.

In our experimental studies, in the feature encoder, we set the numbers of filters in each of the four convolution blocks (from the first to the fourth) as 16, 32, 64, and 128, respectively, with the sizes of pooling windows of 5, 5, 2, and 2. In terms of experimental parameters, for the total 5,799 data samples, we retrieve 90% for self-supervised contrastive learning. We leverage 60% of the self-supervised contrastive learning samples for further fine-tuning of the feature encoder and the maneuver classifier. We use the rest 10% of the entire data for the model testing.

We have adopted the $\texttt{Adam}$ optimizer [20] with a learning rate of 5×10^{-5}, and a batch size of 128 for our contrastive learning. In fine-tuning, we adopt the batch size of 16 minimize the cross-entropy cost function. We have adopted 1,000 epochs and 60 epochs as the early stopping critieria in the contrastive learning and fine-tuning, respectively. To mitigate overfitting, we have applied $L-2$ regularization with a regularization parameter of 10^{-5}. We have leveraged the overall classification accuracy, F1 score, precision, recall, and F1 per class as the evaluation metrics to evaluate the performance of the proposed model.

B. *Experimental Results*

We present the experimental results as follows.

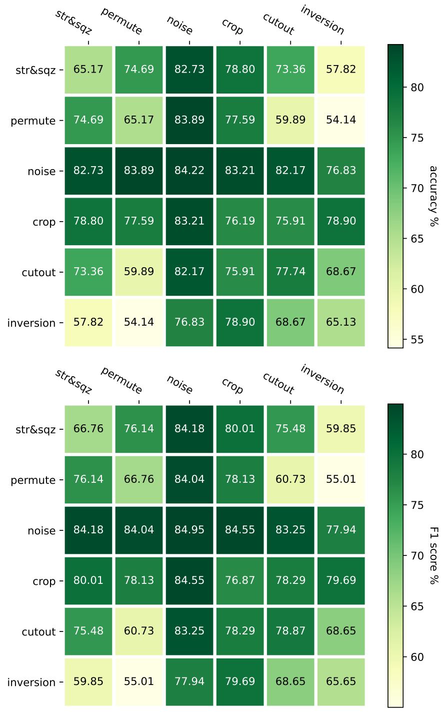

Fig. 2: Comparison of augmentation compositions

Combination of IMU Signal Augmentation. To evaluate the representation quality of self-supervised learning with combinations of augmentation methods, we train the maneuver classifier using the frozen feature encoder and measure the accuracy and F1 score upon the test data. Fig. 2 showcases the performance of S^2-DMIL in terms of accuracy and F1 given combinations of different augmentation methods. We have combined the methods of stretch and squeeze (denoted as str&sqz), permutation (denoted as permute), Gaussian noise (denoted as noise), crop, cutout, and inversion. We can first see that each diagonal element in Fig. 2 represents the same augmentation method for both sets of input IMU signals. We can observe that the method of imposing Gaussian noise achieves the best performance in terms of F1 and accuracy, mainly because such a noise help enhance the model's discernability regarding the latent noise of the IMU signals. Augmentation methods like stretch and squeeze (denoted as str&sqz) and inverse will, however, largely diminish the performance of S^2-DMIL.

In terms of combining different augmentation methods, we can also see that imposing Gaussian noise benefits the DMIL. We note that other combinations could only achieve sub-optimal or even inferior performance. For instance, we can observe that the crop method, when combined with other augmentation methods, can only reach the sub-optimal performance. We observe that inversion and permutation may damage the temporal and physical features within the IMU signals, and hence lead to performance drops. In what follows, we further showcase and focus on the performance regarding S^2-DMIL with Gaussian noise, and its combinations with shuffle (denoted as noise × shuffle), crop (denoted as noise × crop), and stretch and squeeze (denoted as noise × str&sqz).

Overall Performance. We first show the overall performance of S^2-DMIL under different settings, and other baseline approaches in Table I, in terms of accuracy, F1 score, precision, and recall. We also show the F1 per class regarding the detailed performance in classifying the five maneuvers. We can observe that S^2-DMIL (with augmentation using Gaussian noise) achieves overall the best performance. We can see that, on average S^2-DMIL (with augmentation using Gaussian noise) outperforms the other baseline approaches by 12.22% in terms of accuracy, 12.78% in terms of F1 score, 10.58% in terms of precision, and 12.22% in terms of recall. In terms of the F1 score for each class, we can observe that S^2-DMIL (with augmentation using Gaussian noise) outperforms other baselines methods 7.93% in left turn, 7.37% in right turn, 4.21% in acceleration and 6.01% in deceleration. We also note that the U-turn maneuver samples can be considered as few-shot settings due to the relative low frequency of observations. We can observe that S^2-DMIL achieves a 38.52% improvement compared to other baseline models, which demonstrates the effectiveness of S^2-DMIL in a few-shot setting.

Model Sensitivity Analysis. Table II shows the model sensitivity on the percentage of labeled samples during (a) S^2-DMIL under self-supervised learning; and (b) S^2-DMIL under supervised setting. Compared with supervised learning, self-supervised learning provides overall better results in every percentage of training samples, which demonstrates the effectiveness of self-supervised settings. In terms of the self-supervised settings, it reaches the sweet point with only 60% of driver maneuvers compared with the supervised learning settings. This indicates that S^2-DMIL is overall effective and efficient in terms of sample needs, which meets the practical requirements as the available labeled IMU signals are often limited due to data privacy concerns.

We further show the final contrastive learning losses ℓ_{cl}'s after convergence of our self-supervised contrastive learning under different batch sizes in Table III. We can observe that the contrast learning loss is smaller with a larger batch size. This is mainly because when using a larger batch size, there are more negative samples in contrastive learning. This helps S^2-DMIL improve the contrastive learning process upon the unlabeled maneuver data, and leads to faster convergence.

VI. Conclusion

We propose S^2-DMIL, a practical self-supervised contrastive driver maneuver interaction learning approach via augmenting the inertial measurement unit (IMU) signals. We

Model	Overall Metrics				F1 per Class				
	Accuracy	F1	Precision	Recall	LT	RT	UT	AC	DC
SVM(RBF)	0.79	0.79	0.79	0.79	0.80	0.83	0.65	0.90	0.75
KNN	0.76	0.77	0.90	0.76	0.88	0.94	0.26	0.96	0.79
LSTM	0.77	0.79	0.81	0.77	0.84	0.88	0.57	0.91	0.74
BiLSTM	0.75	0.77	0.82	0.75	0.84	0.87	0.53	0.89	0.70
GRU	0.80	0.80	0.81	0.80	0.87	0.90	0.55	0.91	0.79
CNN	0.74	0.76	0.80	0.74	0.88	0.90	0.36	0.94	0.71
Transformer	0.80	0.78	0.76	0.80	0.85	0.83	0.60	0.88	0.71
S^2-DMIL w/o SSL	0.84	0.82	0.82	0.84	0.89	0.89	0.71	0.89	0.75
S^2-DMIL (noise × noise)	0.89	0.90	0.91	0.89	0.93	0.95	0.86	0.95	0.79
S^2-DMIL (noise × shuffle)	0.88	0.88	0.87	0.88	0.92	0.95	0.81	0.93	0.79
S^2-DMIL (noise × crop)	0.86	0.87	0.90	0.86	0.92	0.95	0.76	0.95	0.78
S^2-DMIL (noise × str & sqz)	0.86	0.86	0.87	0.86	0.91	0.96	0.68	0.96	0.79

TABLE I Overall performance of all settings and baselines. We also show the F1 per class of left turn (LT), right turn (RT), U-turn (UT), acceleration (AC), and deceleration (DC).

Percentage of Samples	S^2-DMIL in Self-Supervised Learning				S^2-DMIL in Supervised Learning			
	Accuracy	F1	Precision	Recall	Accuracy	F1	Precision	Recall
20%	0.84	0.86	0.90	0.84	0.72	0.75	0.88	0.72
40%	0.87	0.88	0.90	0.87	0.82	0.82	0.82	0.82
60%	0.89	0.89	0.90	0.89	0.83	0.83	0.83	0.83
80%	0.89	0.89	0.90	0.89	0.83	0.84	0.84	0.83
100%	0.88	0.89	0.90	0.88	0.85	0.83	0.82	0.85

TABLE II Comparison of S^2-DMIL under Self-Supervised Learning and Supervised Learning settings.

Batch Size	16	32	64	128
Contrastive Learning Loss ℓ_{cl}	0.49	0.29	0.17	0.03

TABLE III Learning losses w/ different batch sizes.

have introduced the unlabelled IMU data to pre-train convolution blocks, and further fine-tune upon the labeled data for maneuver classification. This way, we reduce the reliance upon extensive labeling of the IMU signals and retrieve the important and meaningful representations for accurate DMIL. We have performed extensive experimental studies upon the driver maneuvers, i.e., left/right turns, acceleration and deceleration, and U-turns, from the open-sourced BDD-X dataset. Our experimental results have demonstrated that S^2-DMIL outperforms the baseline approaches and provides accurate DMIL results.

VII. Acknowledgment

This project is supported, in part, by the National Science Foundation (NSF) under Grants 2239897 and 2303575, the Google Research Scholar Award Program (2021–2022), and the NVIDIA Applied Research Accelerator Program Award (2021–2022). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] S. Timothy, "Overview of motor vehicle crashes in 2020 (Report No. DOT HS 813 266). National Highway Traffic Safety Administration." 2022.

[2] M. Tabatabaie and S. He, "Driver maneuver interaction identification with anomaly-aware federated learning on heterogeneous feature representations," *Proc. ACM IMWUT*, vol. 7, no. 4, pp. 1–28, 2024.

[3] M. Tabatabaie, S. He, and X. Yang, "Driver maneuver identification with multi-representation learning and meta model update designs," *Proc. ACM IMWUT*, vol. 6, no. 2, jul 2022.

[4] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata, "Textual explanations for self-driving vehicles," *Proc. ECCV*, 2018.

[5] S. Zabihi, S. S. Beauchemin, and M. A. Bauer, "Real-time driving manoeuvre prediction using IO-HMM and driver cephalo-ocular behaviour," in *Proc. IEEE IV*. IEEE, 2017, pp. 875–880.

[6] S. Tezuka, H. Soma, and K. Tanifuji, "A study of driver behavior inference model at time of lane change using Bayesian networks," in *Proc. IEEE International Conference on Industrial Technology*. IEEE, 2006, pp. 2308–2313.

[7] C. Ou and F. Karray, "Deep learning-based driving maneuver prediction system," *IEEE TVT*, vol. 69, no. 2, pp. 1328–1340, 2019.

[8] M. Tabatabaie, S. He, and X. Yang, "Reinforced feature extraction and multi-resolution learning for driver mobility fingerprint identification," in *Proc. ACM SIGSPATIAL*, 2021, pp. 69–80.

[9] I.-H. Kim, J.-H. Bong, J. Park, and S. Park, "Prediction of driver's intention of lane change by augmenting sensor information using machine learning techniques," *Sensors*, vol. 17, no. 6, p. 1350, 2017.

[10] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture," in *Proc. IEEE ICRA*. IEEE, 2016, pp. 3118–3125.

[11] N. Khairdoost, M. Shirpour, M. A. Bauer, and S. S. Beauchemin, "Real-time driver maneuver prediction using LSTM," *IEEE T-IV*, vol. 5, no. 4, pp. 714–724, 2020.

[12] O. Olabiyi, E. Martinson, V. Chintalapudi, and R. Guo, "Driver action prediction using deep (bidirectional) recurrent neural network," *arXiv preprint arXiv:1706.02257*, 2017.

[13] M. Tabatabaie and S. He, "Driver maneuver interaction identification with anomaly-aware federated learning on heterogeneous feature representations," *Proc. ACM IMWUT*, vol. 7, no. 4, jan 2024.

[14] X. Xie, K. G. Shin, H. Yousefi, and S. He, "Wireless csi-based head tracking in the driver seat," in *Proc. ACM CoNext*, 2018, pp. 112–125.

[15] X. Jiang, J. Zhao, B. Du, and Z. Yuan, "Self-supervised contrastive learning for eeg-based sleep staging," in *Proc. IJCNN*, 2021, pp. 1–8.

[16] H. Haresamudram, I. Essa, and T. Plötz, "Assessing the state of self-supervised human activity recognition using wearables," *Proc. ACM IMWUT*, vol. 6, no. 3, pp. 1–47, 2022.

[17] M. Tabatabaie and S. He, "Naturalistic e-scooter maneuver recognition with federated contrastive rider interaction learning," *Proc. ACM IMWUT*, vol. 6, no. 4, pp. 1–27, 2023.

[18] J. Han, J. Pei, and H. Tong, *Data mining: concepts and techniques*. Morgan kaufmann, 2022.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Proc. NeurIPS*, vol. 30, 2017.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.