Gated Recurrent Fusion to Learn Driving Behavior from Temporal Multimodal Data

Athma Narayanan, Avinash Siravuru, and Behzad Dariush

***Abstract*—The Tactical Driver Behavior modeling problem requires an understanding of driver actions in complicated urban scenarios from rich multimodal signals including video, LiDAR and CAN signal data streams. However, the majority of deep learning research is focused either on learning the vehicle/environment state (sensor fusion) or the driver policy (from temporal data), but not both. Learning both tasks jointly offers the richest distillation of knowledge but presents challenges in the formulation and successful training. In this work, we propose promising first steps in this direction. Inspired by the gating mechanisms in Long Short-Term Memory units (LSTMs), we propose Gated Recurrent Fusion Units (GRFU) that learn fusion weighting and temporal weighting simultaneously. We demonstrate it's superior performance over multimodal and temporal baselines in supervised regression and classification tasks, all in the realm of autonomous navigation. On tactical driver behavior classification using Honda Driving Dataset (HDD), we report 10% improvement in mean Average Precision (mAP) score, and similarly, for steering angle regression on TORCS dataset, we note a 20% drop in Mean Squared Error (MSE) over the state-of-the-art.**

***Index Terms*—Sensor fusion, intelligent transportation systems, computer vision for transportation.**

I. Introduction

RECENTLY, the domain of autonomous driving has emerged as one of the hotbeds for deep learning research, bolstered by strong industry support and availability of large real-world datasets (such as KITTI [1], Berkeley Driving Dataset [2], Honda Driving Dataset [3], Argoverse [4]) and physically/visually realistic simulators ([5], [6], [7], [8]). Multi-sensor[1] temporal data, provided by these datasets, offers more leverage to understand optimal driving actions.

Manuscript received September 10, 2019; accepted January 9, 2020. Date of publication January 20, 2020; date of current version January 31, 2020. This letter was recommended for publication by Associate Editor E. Ricci and Editor E. Marchand upon evaluation of the reviewers' comments. This work was supported by Honda Research Institute, USA. *(Corresponding author: Athmanarayanan Lakshmi Narayanan.)*

A. Narayanan and B. Dariush are with Honda Research Institute, Sunnyvale, CA 94086 USA (e-mail: anarayanan@honda-ri.com; bdariush@honda-ri.com).

A. Siravuru is with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 500032 USA (e-mail: avinashs@cmu.edu).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/LRA.2020.2967738

[1]In this letter, we use the terms *sensor* and *mode* interchangeably. The term *sensor* is meaningful to interpret in the autonomous driving setting whereas *mode* is generally used in literature to indicate the various forms of state representation. This may come directly from sensors (image, speech signal) or after some meaningful post-processing (depth map, n-grams, etc.).

The current strategy for multimodal temporal data is to either pre-concatenate (concatenation followed by recurrent modules) [3], [9], [10] or post-concatenate (parallel recurrent modules for each sensor followed by concatenation) [10]–[14]. While they both offer unique merits and challenges, in neither of the two choices, the multi-sensor data is *fused explicitly*. Moreover, such design choices may lead to networks with much bigger parametric spaces resulting in training difficulties. Hence designing efficient architectures to exploit this rich source of data is still an open research problem.

Research on temporal fusion is popular in multimedia domains where either text or audio is combined with video [11], [13]. However, in the autonomous driving community, while non-temporal fusion is popular ([15]–[17]), temporal fusion has garnered much lesser attention. Compared to multimedia datasets, autonomous driving datasets present additional challenges, namely,

- There can be much more number of sensor inputs or multiple copies of each sensor type (multi-camera [18] and multi-radar, etc.)
- The individual sensor data sizes could be disproportionate leading to undesirable biases towards a select few.
- Intermittent data quality degradation or sensor failures could occur (e.g., motion blur and occlusion in image, LiDAR failures in snow, etc.)

This makes autonomous navigation a more general and challenging setting for developing temporal fusion models. Therefore, we first validate our models on autonomous driving related tasks. Given these complex inter-dependencies that emerge from learning on multimodal temporal data, it is essential to ensure that the models are interpretable to verify and correct for any over-fitting. Hence, in this work, fusion is formulated as the problem of finding the optimal linear interpolation between the sensors. The interpolation weights (also learned using *gating* functions) can be interpreted as each sensor's percentage contribution to the fused state.

Contributions of this work: We introduce a novel recurrent neural network unit, called the Gated Recurrent Fusion Unit (GRFU) that can jointly learn fusion and target prediction from temporal data. This new formulation, designed to learn a linear interpolation of sensor encodings, offers superior performance two tasks (*driver behavior classification* and *steering angle regression*) on two challenging datasets (one *real-world* and one *simulated*, respectively). By using global average pooling along the time dimension, we can explain the individual sensor's contribution to the fused representation. Such interpretable fusion

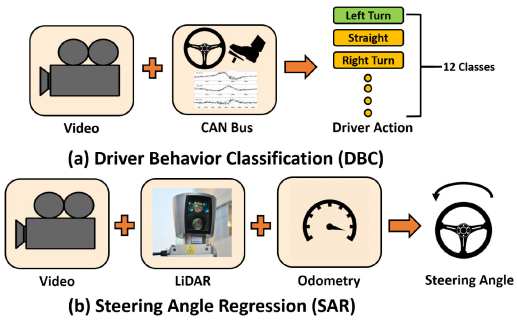

Fig. 1. In (a) **Driver Behavior Classification**, the objective is to correctly classify driver action based on given video and CAN data streams into one of the twelve ground-truth labels and, in (b) **Steering Angle Regression**, we predict the correct steering angle from video, LiDAR and odometry data streams. Both are used to understand and compare driving styles.

allows for higher-level intervention in the case of sensor failures. To the best of our knowledge, this is the first time this has been attempted in the autonomous navigation domain.

An overview of each task is shown in Fig. 1. On the classification task, we report a 10% improvement in the mAP score over the current state-of-the-art, and on the regression task, we note a 20% drop in test error.

The paper is structured as follows: In Section II, we review prior work in autonomous navigation and temporal fusion. In Section III, we formulate our Gated Recurrent Fusion Unit (GRFU) starting from a vanilla Long Short-Term Memory unit. Later, in Section IV, we provide task-specific dataset details and analyze GRFU performance when compared to existing benchmarks. Finally, Section V, summarizes the key ideas in the paper and lists future work avenues.

II. Related Work

A. Temporal Fusion in Machine Learning

Learning using temporal multimodal data is a major sub-branch in deep learning research with applications that span video ([10], [19]), audio ([11], [13]), text processing([14], [20]) and robot navigation([21], [22]), to name a few. Various neural network architectures, like LSTM [23], GRU [24], have been proposed to model and learn underlying data patterns temporally. However, training complication becomes severe in a multimodal setting as complex correlations between the modalities and their history has to be disentangled.

Some of the major strategies for fusion include Kalman filtering [21], and Fuzzy logic [22], just to name a few. These methods have shown great promise in robot navigation and are focused on combating sensor noise and sensor redundancy to do better state estimation. We, however, wish to propose an alternative strategy that can learn fusion using a more data-driven approach for driver behavior understanding.

Most similar to our work in this are attention based fusion [25] or LSTM modifications [11] used in multimedia data. Unlike explicit attention mechanisms that fuse non-temporal information outside the LSTM cell as in [25], we wish to focus on LSTM modifications within the LSTM cell unit. This we believe can model complex temporal correlations in a unified way (using state/information sharing between sensors).

B. Learning in Autonomous Navigation

Sensor fusion in autonomous navigation is a very active research topic [21], [22], [26], [17], [27]. However, temporal fusion has received much lesser attention even though they generate a lot of multimodal temporal data coming from a range of sensors like Camera, LiDAR (Light Detection and Ranging), wheel odometry, etc. Typical driving automation tasks of interest are learning driver behavior [3] and intent [12], motion forecasting [28], [29], object detection [16], learning affordances [30], action regression [15], [31], semantic segmentation [32], [33] among others.

Majority of research attacks these tasks in a non-temporal fashion, mainly using either RGB or RGB-D data [15], [16], [18], [30], [32]. Prior work on using fusion for autonomous navigation is either non-recurrent [34] in the reinforcement learning setting or recurrent unsupervised [35] in the motion forecasting setting. We present experiments showing that incorporating multimodal fusion, in a recurrent supervised setting is beneficial to model driver behavior.

III. Methodology

In this Section, we describe the new temporal fusion architectures that we build over the standard LSTM model. We first review the LSTM model and simple fusion ideas in Section III-B. Next, in Section III-C, we introduce three new models that uses linear interpolation to find the optimal fused state to pass through the recurrent units.[2]

A. Preliminaries

Assume we are given a set of modalities $\tilde{\boldsymbol{S}}^{\mathbf{1}}, \tilde{\boldsymbol{S}}^{\mathbf{2}}, \tilde{\boldsymbol{S}}^{\mathbf{3}} \ldots \tilde{\boldsymbol{S}}^{\boldsymbol{M}}$ where M is the number of sensors, and the sensor signal for an arbitrary sensor, $i \in [1, M]$, is a time-series $\tilde{\boldsymbol{S}}^i = \{\tilde{s}_1^i, \tilde{s}_2^i \ldots \tilde{s}_{\mathbf{t}}^i \ldots \tilde{s}_T^i\}$. The objective is to jointly learn the optimal temporal and modal composition to correctly predict the desired classification/regression target. Further, we make no additional assumptions like sensors having similar structure or dimensions, having similar forms of occlusions and noise ranges, or to be temporally correlated always. We do however pre-process all sensor inputs, $\tilde{s}_t^i$ using appropriate encoders to bring them to the same dimension prior to temporal fusion (we call the processed sensor inputs as *sensor encodings* and denote them as $s_{\mathbf{t}}^i$). This is done for all proposed models and baselines for fair comparison.

The LSTM setup most commonly used in literature [23], [36] features three gated states (input i_t, forget f_t, output o_t) along with the hidden and candidate cell states (h_t, c_t). Cell state represents memory while the hidden state is the output of the model at time t. The gated states control how much of the current and the past information need to be fused and transmitted to the next state in time. The two hidden states

[2]For easier visualization, all the model figures depicted in this Section are for two sensor case, but the equations are defined for an M sensor case.

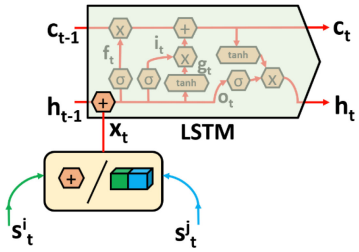

Fig. 2. Early Fusion (Add/Concat) LSTM Unit.

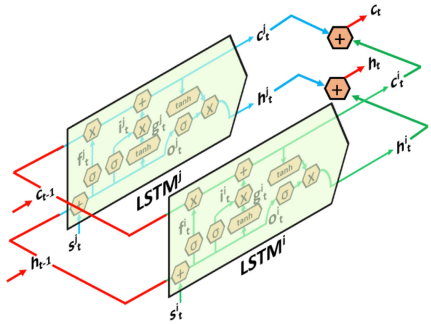

Fig. 3. Late Recurrent Summation (LRS) LSTM Unit.

perform important functions namely: slow-state c_t that fights vanishing gradient problems, and a fast-state h_t that allows the LSTM to make complex decisions over short periods of time. Each gated state performs a unique task of modulating the exposure and combination of the cell and hidden states. For a detailed overview of LSTM inner-workings and empirically evaluated importance of each gate, refer to [37], [38].

B. Early Recurrent Fusion (ERF)

A simple way to extend LSTMs to multimodal settings is by first summing or concatenating all the sensor encodings [3], [9], [10] and passing that as an input to the LSTM, ie., $X = \{x_1, x_2, x_3 \dots x_t\}$, where each $x_t = (s^1 \oplus s^2 \dots s^i \oplus s^j \dots s^M)$. From a temporal standpoint, one can view this as a type of early fusion. A simple ERF unit is shown in Fig. 2.

$$x_t = \dots s_t^i + s_t^j \dots \text{ (or) } \dots s_t^i \oplus s_t^j \dots, \tag{1}$$

$$\begin{aligned} f_t &= \sigma(W_f * x_t + U_f * h_{t-1} + b_f), \\ i_t &= \sigma(W_i * x_t + U_i * h_{t-1} + b_i), \\ o_t &= \sigma(W_o * x_t + U_o * h_{t-1} + b_o), \\ g_t &= \tanh(W_g * x_t + U_g * h_{t-1} + b_g), \end{aligned} \tag{2}$$

$$c_t = c_{t-1} \odot f_t + i_t \odot g_t,$$

$$h_t = o_t \odot \tanh(c_t). \tag{3}$$

Remark: Concatenation, while providing individual sensor inputs to the LSTM to extract useful information, bloats up the cell and hidden state size. On the other hand, summation reduces the cell size but naively combines all sensor encodings with equal emphasis. This may not be a good idea always, especially at time steps where one or more sensors provide noisy information to the fused state (for example, when a car is driving through a tunnel, camera information is unreliable). Temporal fusion architectures must be provided with sufficient tuning choice to learn how to fuse and use temporal data. This is necessary in driving datasets and both ERF models lack the explicit structures to learn them. Example scenarios where fusion needs to be *dynamic* are,

1) *Occlusion in a sensor subset:* While approaching an intersection a huge object in the form of a truck occludes the entire view in one of the image frames rendering image features unreliable. The model should rely on CAN data history to classify driver action correctly.
2) *Action dependency:* Actions like lane branching are subtle steering actions. If the steering signal does not offer sufficient information, video features like lane markers and road curvature could supplement to avoid inter-class confusion.
3) *Loss of temporal correlation across sensors:* As alluded to previously, when a car is going through a dark tunnel, optical flow for odometry maybe hard to obtain and might at best be weakly correlated to the data stream obtained from the CAN bus or LiDAR. Similarly, LiDAR gets really noisy and unreliable in snow, rain and grass [39].

C. Proposed Temporal Fusion Models

We identify two important ways to mitigate the above mentioned issues, a) delay fusion and pass each sensor parallely through M LSTM cells, allowing each sensor to individually decide how much of their respective histories to utilize with the current sensor input (we term this *late recurrent sensor summation*), b) define gates for each sensor to determine the contribution of each sensor encoding to the fused cell and output states (we term this *early gated recurrent fusion*). In the following Section, we first define both the modifications separately and finally define our main model which combines the two (this leads to the *late gated recurrent fusion* model). Moreover, we use the *late recurrent sensor summation* and *early gated recurrent fusion* models also as baselines to evaluate the individual contributions (*ablation study*) of the two modifications.

1) Late Recurrent Summation (LRS): In this model, we use M distinct LSTM units in total (one for each sensor). For each modality separate *forget*, *input*, *output* and *cell* states are computed. Model schematic with equations are shown in Fig. 3 and Eqns. (4)–(6) respectively.

$$\begin{aligned} f_t^i &= \sigma(W_f^i * s_t^i + U_f^i * \boldsymbol{h_{t-1}} + b_f^i), \\ i_t^i &= \sigma(W_i^i * s_t^i + U_i^i * \boldsymbol{h_{t-1}} + b_i^i), \\ o_t^i &= \sigma(W_o^i * s_t^i + U_o^i * \boldsymbol{h_{t-1}} + b_o^i), \\ g_t^i &= \tanh(W_g^i * s_t^i + U_g^i * \boldsymbol{h_{t-1}} + b_g^i), \end{aligned} \tag{4}$$

$$c_t^i = \boldsymbol{c_{t-1}} \odot f_t^i + i_t^i \odot g_t^i,$$

$$h_t^i = o_t^i \odot \tanh(c_t^i), \tag{5}$$

$$\boldsymbol{c_t} = \sum_{i=1}^{M} c_t^i, \quad \boldsymbol{h_t} = \sum_{i=1}^{M} h_t^i. \tag{6}$$

The weights, W_*, U_*, and biases, b_*, that transform the input space for each gate are unique for each modality but are shared

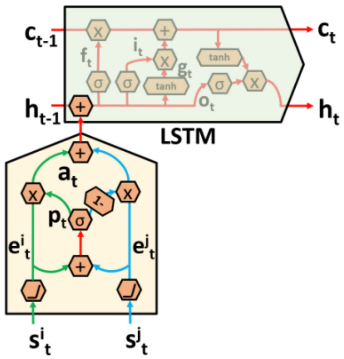

Fig. 4. Early Gated Recurrent Fusion (EGRF) LSTM Unit.

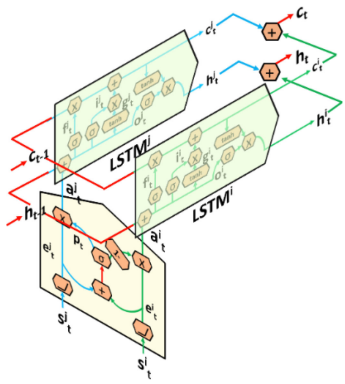

Fig. 5. Late Gated Recurrent Fusion (LGRF) LSTM Unit.

across time. As summarized in the previous Section, each LSTM unit receives information from the states of the past time step (c^i_{t-1}, h^i_{t-1}) and the input from the current time step, s^i_t. Now, instead of having separate states of each LSTM unit of a sensor, all the copies receive the same states (c_{t-1}, h_{t-1}) obtained from the previous time-step. Through this modelling choice we can propagate fused representations temporally. In contrast, in [11], the weights are shared between modalities but not states. By sharing the past cell state (c_{t-1}) across all sensors, the model can individually decide whether to retain or discard memory for each modality. Finally, all the hidden (h^i_t) and cell (c^i_t) states are added to produce a combined representation h_t and c_t that is sent to the next time step (hence the prefix *late* to indicate late fusion in the model name).

2) Early Gated Recurrent Fusion (EGRF): Late fusion offers the model some flexibility to separately control the memories of individual sensors, but even here summation at the end fuses all sensors assuming equal importance. However, we wish to also learn from the data the extent of each sensor's contribution to the final fused states. Inspired by the gating mechanisms used in the LSTM [23], [36] and GRU [24], we propose a similar exposure control in the sensor fusion module as well. For *M* sensors, we define *M-1* gates (p^*) that control the exposure of the sensor encoding, s^i_t, in the final state a_t. Similar to [24], we define the gating for the last sensor as $1 - \sum_i^{M-1} p^i$. This makes the joint representation a linear interpolation of individual sensor encodings. The model schematic and equations are shown in Fig. 4 and Eqns. (7)–(11) respectively. Firstly, the sensors embeddings are converted to the same dimension using a non-linear operation as in Eqn. (7). Then $M-1$ gates are computed as shown in Eqn. (8). As shown in Eqn. (9), the final fusion is performed where each gate is multiplied to the corresponding sensor encoding and summed to form the fused state a_t. Temporal Modelling is performed with a_t as the input as shown in Eqns. (10)–(11).

$$e^i_t = \text{relu}(W^i_e * s^i_t), \tag{7}$$

$$p^k_t = \sigma\left(\sum_{i=1}^{M} W^i_p * e^i_t\right), \forall k \in [1, M-1], \tag{8}$$

$$\boldsymbol{a_t} = \left(\sum_{k=1}^{M-1} p^k_t \odot e^k_t\right) + \left(1 - \sum_{k=1}^{M-1} p^k_t\right) \odot e^k_t, \tag{9}$$

$$\begin{aligned} f_t &= \sigma(W_f * \boldsymbol{a_t} + U_f * h_{t-1} + b_f), \\ i_t &= \sigma(W_i * \boldsymbol{a_t} + U_i * h_{t-1} + b_i), \\ o_t &= \sigma(W_o * \boldsymbol{a_t} + U_o * h_{t-1} + b_o), \\ g_t &= \tanh(W_g * \boldsymbol{a_t} + U_g * h_{t-1} + b_g), \end{aligned} \tag{10}$$

$$c_t = c_{t-1} \odot f_t + i_t \odot g_t,$$

$$h_t = o_t \odot \tanh(c_t). \tag{11}$$

The gating functions are valuable to draw insights and explain the nature of fusion occurring within the model. Once learnt, the user can interpret the gating values as contributions of each sensor and verify if they match human insight for some arbitrary sample in the dataset. This explainability feature is crucial for scenarios involving safety-critical tasks.

3) Late Gated Recurrent State Fusion (LGRF): Finally, we describe our Late Gated Recurrent Fusion model, which combines the best aspects of both late recurrent fusion (independent control of memory for each sensor) and gated recurrent fusion (learning how to fuse) in order to improve learning performance of temporal fusion models.

$$e^i_t = \text{relu}(W^i_e * s^i_t). \tag{12}$$

$$p^k_t = \sigma\left(\sum_{i=1}^{M} W^i_p * e^i_t\right). \quad \forall k \in [1, M-1]. \tag{13}$$

$$\boldsymbol{a^i_t} = \begin{cases} p^i_t \odot e^i_t & \text{if} \quad i \in [1, M-1], \\ \left(1 - \sum_{k=1}^{M-1} p^k_t\right) \odot e^i_t & \text{if} \quad i = M \end{cases}$$

$$\begin{aligned} f^i_t &= \sigma(W^i_f * \boldsymbol{a^i_t} + U^i_f * \boldsymbol{h_{t-1}} + b^i_f), \\ i^i_t &= \sigma(W^i_i * \boldsymbol{a^i_t} + U^i_i * \boldsymbol{h_{t-1}} + b^i_i), \\ o^i_t &= \sigma(W^i_o * \boldsymbol{a^i_t} + U^i_o * \boldsymbol{h_{t-1}} + b^i_o), \\ g^i_t &= \tanh(W^i_g * \boldsymbol{a^i_t} + U^i_g * \boldsymbol{h_{t-1}} + b^i_g), \end{aligned} \tag{14}$$

$$c^i_t = \boldsymbol{c_{t-1}} \odot f^i_t + i^i_t \odot g^i_t,$$

$$h^i_t = o^i_t \odot \tanh(c^i_t), \tag{15}$$

$$\boldsymbol{c_t} = \sum_{i=1}^{M} c^i_t, \boldsymbol{h_t} = \sum_{i=1}^{M} h^i_t. \tag{16}$$

TABLE I
MAP OF DRIVER BEHAVIOR CLASSIFICATION ON HDD DATASET

Models	Fusion	Inputs	intersection passing	left turn	right turn	left lane change	right lane change	left lane branch	right lane branch	crosswalk passing	railroad passing	merge	u-turn	mAP
Non-Fusion	-	CAN	36.41	66.25	74.27	26.17	13.39	8.03	0.20	0.30	0.07	3.59	33.57	23.84
	-	Img	65.74	57.79	54.43	27.84	26.11	25.76	1.77	16.08	2.56	4.86	13.65	26.96
	Early-Concat [3]	CAN+Img	74.93	71.68	76.80	32.68	37.02	29.78	5.51	9.91	3.62	3.44	13.90	32.66
	Early-Add	CAN+Img	77.08	75.90	76.40	23.38	19.14	16.62	2.51	10.19	2.67	8.90	15.91	29.88
	Late-Concat	CAN+Img	70.04	77.43	75.82	29.98	13.37	21.89	3.08	4.70	1.98	2.88	30.06	30.11
	Late-Add	CAN+Img	79.03	76.84	76.64	28.61	21.72	20.40	4.29	12.15	2.50	7.66	31.30	32.83
	TCN [45]	CAN+Img	79.98	71.91	73.80	28.14	23.60	30.69	4.63	8.46	2.99	7.96	32.11	33.16
Fusion LSTM	Look,Listen & Learn [11]	CAN+Img	81.11	78.46	79.01	43.20	25.29	30.17	**7.79**	13.94	3.56	8.92	33.39	36.80
	EGRF(ours)	CAN+Img	84.29	82.36	80.21	48.26	29.67	**40.39**	5.69	18.23	3.06	9.70	**37.93**	39.98
	LRS(ours)	CAN+Img	82.40	77.04	76.49	51.77	35.14	32.85	6.66	18.29	**3.82**	**11.00**	36.74	39.29
	LGRF(ours)	CAN+Img	**86.83**	**85.39**	**82.95**	**57.76**	**37.79**	37.42	3.85	**20.38**	3.35	10.60	37.07	**42.13**

The model schematic is shown in Fig. 5. Similar to the early gated recurrent fusion model, we compute fusion gates p_t^* as a function of all the sensor encodings e_t^*, but instead of doing the linear interpolation of all sensor inputs to get a joint input state, a_t, we use the gates to control the exposure of each encoding that is passed into sensor specific LSTM cells. The final joint cell and hidden states are computed by summing all the final cell and hidden state outputs. Having described the new temporal fusion designs, in the next Section, we test the models on two challenging autonomous driving datasets.

IV. EXPERIMENTS

A. Tactical Driver Behavior Classification

1) HDD Dataset: Recently, HDD [3] was proposed to stimulate research on learning driver behavior in interactive situations. The dataset includes a 104-hour synchronized multi-sensor naturalistic driving data. We focus our attention on the **goal-oriented** driver behavior classification task which involves temporally classifying the multimodal data involving video stream and CAN signal data into driver actions. The 104-video-hour data corresponds to 137 sessions. Each frame contains one label from the twelve behavior classes such as *left turn, right turn, intersection passing, lane change, etc.*

We follow the prior work [3] and obtain our training (100 driving sessions) and testing splits (37 driving sessions). CAN signal includes: car speed, accelerator and braking pedal positions, yaw rate, steering wheel angle, and the rotation speed of the steering wheel,turn signals (eight dimensional stream). The images are of dimension $720 \times 1280 \times 3$. The image representation is extracted from conv$2d_7b_{1x1}$ layer of InceptionResnet-V2 [40] pre-trained on ImageNet [41]. The features are convolved with a 1×1 convolution to reduce the dimension from $8 \times 8 \times 1536$ to $8 \times 8 \times 20$ and flattened to 1×1280. Raw sensor signals are passed through a fully-connected layer to transform 1×8 size signal to obtain a feature vector of size 1×20.

2) Results: In this task, the input is untrimmed, egocentric sequences of video and CAN signals. The output is the tactical driver behavior label of each frame. We follow the evaluation protocol as in [3], [42], [43] to compute mean Average Precision (mAP) over all classes. We use the Adam optimizer [44] to learn the network parameters with the sequence length set to 90 video frames. To fairly compare with the baseline methods [3], we use the same batch size set to 40. The training is performed using truncated back-propagation through time. The training process is terminated after 50 epochs, with a fixed learning rate 5×10^{-4}. The results for the same test split as used in [3] are showcased in Table.I.

Non-Fusion Architecture: We first perform experiments only on the *CAN* signal and *Img* (Image) sensors separately. The embeddings are directly sent to a standard LSTM with hidden size of dimension 2000. The output h_t is directly fed into a fully connected layer then squashes the dimension to 12 classes including *background class.* The CAN signal outperforms in certain classes such as *left turn, right turn, U-turn* while Image performs better in classes such as *lane change, lane branch, intersection passing and crosswalk passing*. TCN [45] performs slightly better than LSTM as showcased in Table.I. A successful sensor fusion should outperform these results benefiting from each sensor separately.

Early Fusion LSTM: As baseline architectures we use the early sensor fusion where sensor embeddings are either concatenated (*Early-Concat*) or element wise summed (*Early-Add*) as explained in Section III-B. *Early-Concat* is similar to the technique used in [3]. In the early fusion stage the *Early-Concat* outperforms *Early-Add* (mAP of 32.66 vs 29.88) as the LSTM has access to individual sensor information, and can choose to discard noisy sensor readings. However each sensor has different ranges and are normalized individually. For example, steering angle is between $+360^\circ$ to -360° and image has pixel values between 0 to 255. Features added from a normalized 0 pixel value and 10° steering input can result in the same output when the values are interchanged. Several such combination of values can be created. Hence adding could corrupt the fused encoding resulting in the LSTM operating on a corrupted feature space.

Late Fusion LSTM: Here we have two separate LSTM cells that do not share any weights or hidden states between the modalities. Concatenation or summation happens after the LSTM cell. More precisely $h_t^{Img} \oplus h_t^{CAN}$ is sent to a single fully connected layer for classification. The fully connected layer operates on a 2000×2 dimension vector in the case of *Late-Concat* or 2000 dimension vector in the case of *Late-Add* respectively. Interestingly *Late-Add* (which is essentially LRS without cell state sharing) outperforms all other types of baseline fusion as the addition of cell states allows the model to focus more on the temporal aspects of each sensor.

Look, Listen and Learn [11]: The most similar baseline to our *LRS* model described in Section III-C1 is the Look, Listen and Learn architecture presented in [11]. We re-implement the architecture in Pytorch for the HDD dataset. We add auxiliary losses to both the modalities and sum in the predicted output results with weight sharing. This results in a huge improvement over baseline model over every class. We call this model *Look Listen* in Table.I

EGRF, LRS, LGRF: We replace the standard LSTM with each of our fusion modules explained in Section III-C2 (*EGRF*), Section III-C1 (*LRS*), Section III-C3 (*LGRF*). Each of our hypothesised fusion architectures outperform the state of the art on almost all of the classes. Our *EGRF* and *LRS* models increase the mAP by 7% over the standard fusion LSTMs while benefiting distinct class labels as shown in Table.I. Finally, we hypothesize that our combined model *LGRF* attempts to combine the benefits of both *LRS* and *EGRF* and therby increasing the mAP by 10%. The main driver for the performance boost is the added flexibility in learning afforded by the gating functions which allow the network to modulate the fusion process at each time step and best optimize the data being input from individual sensors.

3) Discussion: One of the limitations of most sensor fusion architectures is the inability to provide visual explanations for the decision-making process. For example, when in the case of a noisy sensor signal, the model needs to adapt to another sensor and gate the noise. LGRF model is uniquely positioned to give class specific reasoning for the sensor weighting. For this, we apply global average pooling on the pre-gate layer p_k^{si} along the sensor dimension and display its value. For example, a value of 0.7 for sensor one means that sensor one had a higher weighting than the 0.3 for sensor two. We additionally visualize the class activation maps [46] to show the localization ability of our models by using Grad-CAM [47] on the last convolution layer of the image input.

We get explainable results that validate our assumptions about which sensor is important for which action, as showcased in Fig. 6. The heat map falls on image locations such as lane markers for lane actions, road extensions for turns or intersection passing. Turns have higher CAN weighting as they capture the motion better. An interesting observation is the truck occluding the view in the last example. Our model not only improves the attention region by localizing to the cross-walk but also shows equal weighting for both images and CAN signals, thereby correctly classifying the action.

4) Failure Cases: We focus on the driver behavior classification task and visualize the attention maps on the image sensor to better understand the failure cases. We conjecture that the majority of failure cases occur due to: 1) Wrong sensor being weighted higher, 2) Fusion model's inability to resolve inter class confusion. Such cases, as shown in Fig. 7, have to be investigated further to resolve unexplained attention regions.

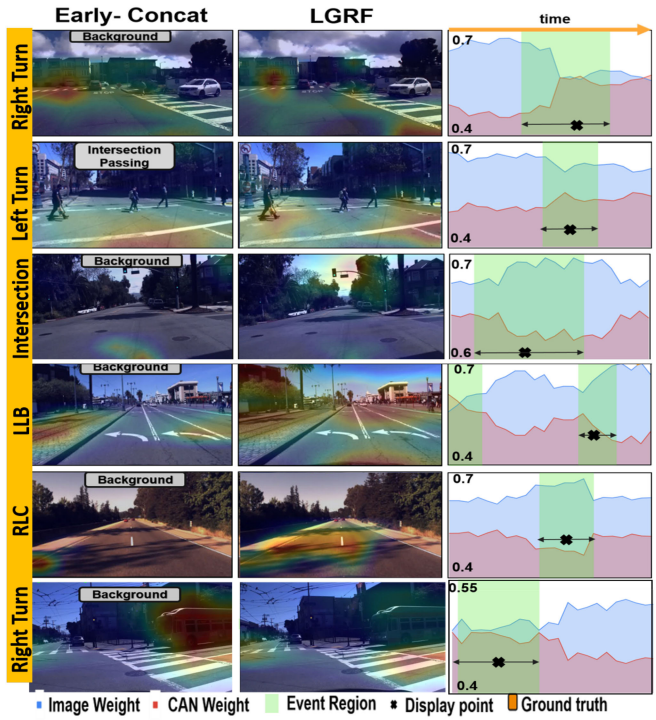

Fig. 6. Sensor Attention visualized for different actions where baseline models fail, and our model succeeds. In most of the actions we can see a shift in the attention to more meaningful parts of the image by LGRF model. For Right Lane Change (RLC) focus is on lane marker, Left lane branch (LLB) on branch arrow, Intersection on traffic lights and road extension. Moreover, we can also determine the weighting scheme the model used to predict the correct results using global average pooling on the fusion gates as shown in the last column. (Shown in Eq. 8).

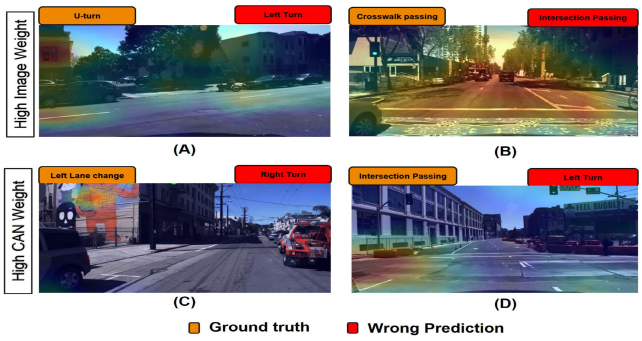

Fig. 7. Failure modes of our models. *High Image weight:* (A) Inter class confusion between Left turn and U-Turn due to improper long term CAN signal weighting. (B) Crosswalk and Intersection confusion is high in spite of expected higher Img weight. *High CAN weight:* (C) The Attention appears on graffiti. Higher CAN weight does not resolve issue. (D) Attention falls on plausible road extension but is misclassified due to higher CAN weight.

B. Steering Action Regression

To show the generality of our methodology we test our models on steering angle regression as well. Given a set of sensor signals the task is to determine the appropriate steering control action to successfully drive in a race track. One method of addressing this problem is to perform end to end regression. A better temporal fusion could provide richer features to deal with the challenging task of understanding vehicle dynamics just by observing sensors. We also showcase an extension of our work to a three sensor setting.

1) TORCS Dataset: TORCS driving simulator is capable of simulating physically realistic vehicle dynamics as well as

TABLE II
MSE OF STEERING ANGLE REGRESSION ON TORCS DATASET

Model	Image	LiDAR	Odometry	Early-Concat[3]	TCN[45]	Late-Add	Look Listen[11]	**EGRF**	**LRS**	**LGRF**
MSE	1.3973	1.5610	2.10	0.997	0.71	1.1051	0.810	0.7087	0.635	**0.619**

multiple sensing modalities to build sophisticated AI agents that can complete race tracks. The following sensing modalities for our state description include : (1) odometry (*SpeedX, SpeedY, SpeedZ*) substituting CAN signals (2) laser scans consisting of 19 LiDAR points. (3) color images capturing the ego-view. We collect 1000 time steps from 32 different tracks that vary in the form of complicated loops to different road conditions. To collect the steering action ground truths we use the standard PID controller that successfully completes navigating one lap on each track without veering off the road. Out of the 32 tracks, we divide the training-test into 20–12 track split. We perform 5 fold cross validation on training data (80 – 20 split) and display the best model results for the test split in Table II. For the encoding we employ multiple convolution layers of kernel size 3×3. More specifically, the following layers are used: *layer1,layer2* has 32 filters, followed by max pooling with kernel size 2×2, followed by *layer3,layer4* with 64 filters each. Finally an additional max pooling layer with 2×2 to downsample the feature space to a feature size of 1×1638. The 1×19 velodyne points and 1×3 odometry signals are embedded using separate linear layers to 1×30 and 1×20 respectively. The flattened feature embeddings are used as inputs to our model. The output of the LSTM is sent to a linear layer followed by $\tanh$ activation.

2) Results and Discussion: In this task the input is a batch of images, odometry and LiDAR points with a time history of four time steps. We train the model to directly regress the steering action. We compute the average Mean squared error (MSE) between the predicted action and ground truth action value over all the data in test set. After ablation study we found that history above four time steps does not provide significant information for temporal modeling. We set the batch size to 128 for all experiments and use Adam optimizer to train the weights. For the final activation we choose $\tanh$ activation function to squish the last linear layer output to a range $(-1, 1)$. We perform a grid search on the learning rate from $1e-2$ to $1e-5$. Overall $5e-3$ performs best. The comparison of the overall MSE for the test set is shown in Table II.

Baselines: We extend our previous baselines from HDD experiments to a three sensor setting and report the results in Table II. A similar trend in results is obtained with [11] outperforming other baselines with the lowest error of 0.810. Most of the models are not able to handle the huge disparity in sensor embedding dimensionality with image embedding size of 1×1638 overshadowing others.

EGRF, LRS, LGRF: We extend our models to a three sensor setting. This involves 1) modifying Eqn. 4 in LRS to support 12 gates (4 for each modality) 2) compute two pre-gates for EGRF as in Eq. 8 for images (p_t^{img}), LiDAR (p_t^{lidar}) and odometry ($1 - p_t^{img} \times p_t^{lidar}$) 3) Combine both for LGRF. Our models outperform all other baselines. LGRF gives an overall best performance with an additional +20% drop over the state-of-the-art. An interesting note is the huge variation in error between EGRF and LRS. We suspect that this may be due to the highly correlated sensors in a simulated setting as opposed to the real world setting in HDD dataset. Hence the benifit of early noise rejection from EGRF does not play as important a role as learning to fuse the best aspects of each sensor as in LRS.

V. CONCLUSION

In this work we presented a novel temporal fusion architecture that we termed *Gated Recurrent Fusion Unit* to learn from large-scale multi-sensory temporal data. Gating functions modulate the exposure of individual sensor data at each time step to determine optimal fusion strategy. GRFU eliminates the need to design and train separate network blocks for pre-processing sensor data, learning intermediate representations, or driver behavior modeling assuming full state information. As GRFU is end-to-end differentiable, all these building blocks can be learned together.

For future work we plan to evaluate GRFU's effectiveness on other challenging temporal multimodal settings not limited to autonomous driving domain. Moreover additional extensions using TCN backbones may be considered to determine the best temporal abstraction for fusion. Additionally, for situations where the data is not synchronized, or, actions depend on tracking of particular objects in the scene, we could add attention mechanisms [14] to GRFU to provide more context.

REFERENCES

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
[2] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2174–2182.
[3] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 7699–7707.
[4] M.-F. Chang *et al.*, "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2019, pp. 8740–8749.
[5] L. Cardamone, D. Loiacono, and P. L. Lanzi, "Learning drivers for torcs through imitation using supervised methods," in *IEEE Symp. Comput. Int. Games*, 2009, pp. 148–155.
[6] N. Yoshida, "Gym-torcs," 2016, [Online] Available: https://github.com/ugo-nama-kun/gym_torcs
[7] Udacity, "Udacity's self-driving car simulator," 2017, [Online] Available: https://github.com/udacity/self-driving-car-sim
[8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conf. Robot Learn.*, 2017, pp. 1–16.
[9] Y. Chen *et al.*, "Lidar-video driving dataset: Learning driving policies effectively," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 5870–5878.

[10] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, "Actionvlad: Learning spatio-temporal aggregation for action classification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 971–980.

[11] J. Ren *et al.*, "Look, listen and learn—a multimodal LSTM for speaker identification," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 3581–3587.

[12] 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 Int. Conf. Robot. Autom.*, 2016, pp. 3118–3125.

[13] X. Yang, P. Ramesh, R. Chitta, S. Madhvanath, E. A. Bernal, and J. Luo, "Deep multimodal representation learning from temporal data," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 5447–5455.

[14] S. S. Rajagopalan, L.-P. Morency, T. Baltrusaitis, and R. Goecke, "Extending long short-term memory for multi-view structured learning," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 338–353.

[15] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1989, pp. 305–313.

[16] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun, "Off-road obstacle avoidance through end-to-end learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 739–746.

[17] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using CNN-based fusion of vision and lidar in autonomous vehicle environment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4224–4231, Sep. 2018.

[18] M. Bojarski *et al.*, "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*.

[19] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, 2016, Art. no. 115.

[20] I. Gallo, A. Calefati, S. Nawaz, and M. K. Janjua, "Image and encoded text fusion for multi-modal classification," in *Proc. Digital Image Comput.: Techniques Appl.*, 2018, pp. 1–7.

[21] M. Kam, X. Zhu, and P. Kalata, "Sensor fusion for mobile robot navigation," *Proc. IEEE*, vol. 85, no. 1, pp. 108–119, Jan. 1997.

[22] J. Sasiadek and Q. Wang, "Sensor fusion based on fuzzy kalman filtering for autonomous robot vehicle," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999, vol. 4, pp. 2970–2975.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Emp. Methods Natural Lang. Proc. (EMNLP)*, 2014, pp. 1724–1734.

[25] C. Hori *et al.*, "Attention-based multimodal fusion for video description," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 4193–4202.

[26] Q. Li, L. Chen, M. Li, S.-L. Shaw, and A. Nüchter, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 540–555, Feb. 2014.

[27] V. De Silva, J. Roche, and A. Kondoz, "Fusion of lidar and camera sensor data for environment sensing in driverless vehicles," 2017, *arXiv preprint arXiv:1710.06230*.

[28] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 3569–3577.

[29] N. Radwan, A. Valada, and W. Burgard, "Multimodal interaction-aware motion prediction for autonomous street crossing," 2018, *arXiv:1808.06887*.

[30] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 2722–2730.

[31] A. Kendall *et al.*, "Learning to drive in a day," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 8248–8254.

[32] N. Radwan, A. Valada, and W. Burgard, "Vlocnet++: Deep multitask learning for semantic visual localization and odometry," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4407–4414, Oct. 2018.

[33] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *Int. J. Comput. Vis.*, pp. 1–47, 2018.

[34] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor, "Learning end-to-end multimodal sensor policies for autonomous navigation," in *Proc. Conf. Robot Learn.*, 2017, pp. 249–261.

[35] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2174–2182.

[36] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.

[37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *NIPS 2014 Workshop Deep Learn.*, Dec. 2014, *arXiv:1412.3555*.

[38] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2342–2350.

[39] R. H. Rasshofer, M. Spies, and H. Spies, "Influences of weather phenomena on automotive laser radar systems," *Adv. Radio Sci.*, vol. 9, no. B. 2, pp. 49–60, 2011.

[40] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2009, pp. 248–255.

[42] K. Nakamura, S. Yeung, A. Alahi, and L. Fei-Fei, "Jointly learning energy expenditures and activities using egocentric multimodal signals," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 6817–6826.

[43] Z. Shou *et al.*, "Online action detection in untrimmed, streaming videos-modeling and evaluation," *ECCV*, vol 1, no. 2, p. 5, 2018.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent.*, May 7–9, 2015, *arXiv:1412.6980*.

[45] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[46] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2921–2929.

[47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 618–626.