*machines*

MDPI

*Article*

# View-Invariant Spatiotemporal Attentive Motion Planning and Control Network for Autonomous Vehicles

**Melese Ayalew [1], Shijie Zhou [1,*], Imran Memon [2], Md Belal Bin Heyat [3,4,5], Faijan Akhtar [6] and Xiaojuan Zhang [7]**

[1] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[2] Department of Computer Science, Shahdadkot Campus, Shah Abdul Latif University, Khairpur 66111, Pakistan
[3] IoT Research Center, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[4] Centre for VLSI and Embedded System Technologies, International Institute of Information Technology, Hyderabad 500032, India
[5] Department of Science and Engineering, Novel Global Community Educational Foundation, Hebersham, NSW 2770, Australia
[6] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[7] Zhejiang CloudNeedle Information Technology Co., Ltd., Hangzhou 311121, China
* Correspondence: sjzhou@uestc.edu.cn

**Abstract:** Autonomous driving vehicles (ADVs) are sleeping giant intelligent machines that perceive their environment and make driving decisions. Most existing ADSs are built as hand-engineered perception-planning-control pipelines. However, designing generalized handcrafted rules for autonomous driving in an urban environment is complex. An alternative approach is imitation learning (IL) from human driving demonstrations. However, most previous studies on IL for autonomous driving face several critical challenges: (1) poor generalization ability toward the unseen environment due to distribution shift problems such as changes in driving views and weather conditions; (2) lack of interpretability; and (3) mostly trained to learn the single driving task. To address these challenges, we propose a view-invariant spatiotemporal attentive planning and control network for autonomous vehicles. The proposed method first extracts spatiotemporal representations from images of a front and top driving view sequence through attentive Siamese 3DResNet. Then, the maximum mean discrepancy loss (MMD) is employed to minimize spatiotemporal discrepancies between these driving views and produce an invariant spatiotemporal representation, which reduces domain shift due to view change. Finally, the multitasking learning (MTL) method is employed to jointly train trajectory planning and high-level control tasks based on learned representations and previous motions. Results of extensive experimental evaluations on a large autonomous driving dataset with various weather/lighting conditions verified that the proposed method is effective for feasible motion planning and control in autonomous vehicles.

**Keywords:** autonomous vehicles; deep learning; invariant representation learning; motion planning; spatiotemporal attention; vehicle control

check for updates

**Citation:** Ayalew, M.; Zhou, S.; Memon, I.; Heyat, M.B.B.; Akhtar, F.; Zhang, X. View-Invariant Spatiotemporal Attentive Motion Planning and Control Network for Autonomous Vehicles. *Machines* **2022**, *10*, 1193. https://doi.org/10.3390/machines10121193

Academic Editor: Domenico Mundo

Received: 9 November 2022
Accepted: 7 December 2022
Published: 9 December 2022

## 1. Introduction

Autonomous driving vehicles (ADVs) have become the subject of much research because of their potential to transform transportation, reduce traffic accidents, and alleviate congestion [1,2], as well as assist soldiers with various tasks, including surveillance, fire fighting, and monitoring [3]. For ADVs to efficiently operate and reach their full potential, they must understand their environment, recognize static and dynamic obstacles, generate feasible trajectories toward the goal position, and then execute the desired driving behavior.

However, achieving fully autonomous driving, particularly feasible motion planning and decision-making remains challenging due to the high complexity of driving scenarios. Following the taxonomies used in [4–6], existing motion planning and control methods can be divided into three alternative approaches: mediated perception (modular), end-to-end (behavioral reflex), and intermediate (direct perception). The modular paradigm [7,8] decomposes complex ADV tasks into several more tractable sub-modules, such as perception, planning, and control modules; each module is solved sequentially and serves as an input for the next. Such a decomposition of complex ADV systems enables solving each problem independently with less effort. However, such a decomposition of tasks may add extra computational burden due to duplicated feature extraction for each task. In addition, such systems cause error accumulation and propagation from the upstream module, e.g., perception to subsequent modules that may cause an overall system failure [9]. Although deep learning has helped advance the perception module of ADVs, more crucial motion planning and control tasks still rely on classical rule-based algorithms [7,8], which are time-consuming and inefficient in satisfying all driving scenarios in a dynamic environment.

Recently, the end-to-end deep learning approach integrates perception, planning, and control tasks and has shown impressive results. These approaches use neural networks to directly map raw sensor data as input into control commands [10–15] or future trajectories [16–22]. However, end-to-end ADV approaches have poor generalization toward the unobserved driving environment due to distribution shift problem [23]. Consequently, learned decision-making in such approaches is typically limited to the driving environment or tasks in which it was trained [16]. Moreover, driving decisions made in such an approach lack interpretability due to the black-box nature of end-to-end learning models [24]. To improve the robustness of these methods, many researchers have proposed more advanced deep learning models that can leverage multi-modal and multi-view information [25–28]. Nevertheless, a naive addition of very complex (various raw sensor) inputs and directly mapping it into driving decisions may lead to increased sampling and training complexity [29,30]. In addition, it increases memory requirements in decision-making units of ADVs [29] and may lead to worse generalization/performance [31], especially in the presence of multiple sensors that can cause a distributional shift.

To further improve the efficiency, generalization, and interpretability of end-to-end driving models, many researchers also proposed intermediate representation learning [6,32–34] and multi-task learning (MTL) [35] methods. Unlike the aforementioned explicit manual task decomposition [7,8] or blind direct mapping [12] approaches, intermediate (direct perception) approaches [6] first learn a mapping from raw perceptual input to intermediate representations and then use this representation to make more generic driving decisions [30,32–34]. These approaches, however, require predefined representations, which can lead to limited robustness and may introduce spurious inputs. Unlike all aforementioned methods, MTL approaches typically have a shared backbone network that learns shared representations that allows the model to simultaneously learn multiple tasks related to motion planning [36–38] and control [16,39–41]. Recently, attention mechanisms [42–52] have shown great success in further improving the efficiency and interpretability of end-to-end ADVs by learning salient representations while filtering out irrelevant inputs [12].

Despite the success of deep learning-based methods in enabling ADVs to perform and generalize well, the methods still have several shortcomings: (1) learning of invariant representation, which can enhance the generalization capability of generated trajectories or driving actions are under-explored so far; (2) various end-to-end learning methods neglect joint learning of spatiotemporal representations and focus on single-task learning, which limits the robustness of ADVs under a changing environment; (3) several works on intermediate [30,32–34] and MTL [36–38] approaches for motion planning heavily rely on post-processed detail environmental maps for driving decision making, which are costly to create and transfer to new driving scenarios; (4) fnally, most works only evaluate their ADV models in simple environments with limited complexity without considering diverse weather/lighting conditions and dynamic obstacles.

To address the above-mentioned challenges, we propose **V** iew-invariant **S**patio-**T**emporal **A**ttentive **M**otion **P**lanning and **C**ontrol **N**etwork (ViSTAMPCNet) for autonomous vehicles. In summary, our contributions are:

- We propose an end-to-end motion planning and control network for ADVs based on imitation learning. The proposed ViSTAMPCNet takes front and top-view image sequences and first learns view-invariant spatiotemporal representations, which are more robust to domain shift and more interpretable than directly mapping raw camera images or using detailed environment map post-processing. Then, use the learned shared invariant representations to predict feasible future motion plans and control commands simultaneously.
- We conducted ablation studies to verify the benefit of shared view-invariant spatiotemporal representations for joint motion planning and high-level control.
- In order to demonstrate its superiority, we also compared ViSTAMPCNet to other baselines and existing state-of-the-art methods on a large-scale driving dataset with dynamic obstacles and weather/lighting conditions (e.g., clear, rainy, and foggy).

The rest of the article is organized as follows. Section 2 introduces related works, in particular, some elements of end-to-end learning, multi-task learning, attention mechanisms, and invariant feature learning methods. Section 3 presents a basic overview, problem formulation, and architectural components of the proposed method, i.e., ViSTAMPCNet. Section 4 describes the experiment settings and then presents the results and discussion to verify the performance of ViSTAMPCNet. Section 5 concludes the paper.

## 2. Related Work

### *2.1. End-to-End Learning Methods*

End-to-end (behavior reflex) approaches directly map perceptual input into driving decisions. The pioneering work of Pomerleau [10] proposed using a single neural network that directly outputs a driving control command. Following this work, Lecun et al. [11] proposed DAVE, an end-to-end obstacle avoidance system. This system learns obstacle avoidance directly from low-resolution images using a six-layer convolutional neural network (CNN). In light of these works, Bojarski et al. [12] developed a driving model named DAVE-2 that takes front-view camera images and predicts steering commands using a nine-layer CNN and achieves autonomous lane following in relatively simple real-world scenarios, such as flat or barrier-free roads. Following this work, end-to-end control models have been explored in [13,14]. However, these approaches do not consider temporal information that is critical for self-driving. To address this, many works introduced temporal information as input into their driving decision-making models. For example, Chi and Mu [15] used carefully designed recurrent layers (e.g., LSTM and Conv-LSTM) to jointly utilize spatial and temporal cues to predict wheel angle or other steering control operations. Xu et al. [16] proposed dilated fully convolutional networks (FCNs) and long short-term memory (LSTM) to predict future ( discrete and continuous) moving paths given trajectory history and image frames. In a similar architecture with [16], Song et al. [17] proposed an end-to-end motion network but used VGG-16 for extracting more rich visual features instead of a dilated FCN as in [16]. Fernando et al. [18] also adopted a pre-trained VGG-16 backbone with multiple LSTM modules to extract spatial features and temporal dependencies from visual and motion history inputs for generating path plans with good performance.

Besides their success in end-to-end control, CNN and RNN architectures also showed promising results in end-to-end planning [19–22]. Bergqvist et al. [20] also designed several CNN and RNN-based path planning networks that use various input types, including gray-scale images and ego-motions. They showed LSTM or CNN-LSTM's ability to generate smooth and feasible path plans in many situations, although they only considered lane-following tasks in simple scenarios. Recently, Cai et al. proposed a similar CNN-LSTM architecture that computes a future trajectory for autonomous vehicles using image sequences and three discrete command values (Turn Left, Turn Right, and Keep Straight), which uses different sub-networks [21]. In their following work, they proposed a similar network architecture that takes image sequences, trajectory histories, and three discrete commands and estimates uncertainty and trajectory planning [22].

All of the above works on end-to-end learning-based methods employ one specific input and output module, which makes it inefficient and unscalable for self-driving in a dynamic environment. In addition, these systems have poor generalization toward unseen environments and lack interpretability.

### *2.2. Multi-Task Learning Methods*

Instead of having a separate network for each task, the multi-task learning paradigm [35] aims to simultaneously learn several autonomous driving tasks by sharing parameters and computations, while achieving state-of-the-art performance. Yang et al. [39] proposed a multi-modal multi-task learning (MMTL) network for vehicle control that predicts both steering angle and vehicle speed instead of having two distinct networks for each task. Their encoder network comprises five convolutional layers, LSTM, and FCN layers to process single front-facing camera images and current vehicle and past vehicle motion inputs. These are then passed through an FC layer to predict future vehicle speed. Similarly, Codevilla et al. [40] proposed an MMTL network that predicts both steering angle and acceleration. Besides visual inputs, they used high-level navigation commands (i.e., keep straight and turn left) as auxiliary input representations while training their CNN. In their following work [41], they modified their network architecture to use a residual network as a perception module to extract a richer representation and prediction speed in addition to the prior acceleration and steering angle outputs. In both cases, the use of secondary high-level command inputs allows ambiguity in a change of driving behavior and makes the network more flexible and adaptive to the unseen situation. A work by Xu et al. [16] can also be considered an MMTL model as it jointly learns (discrete and continuous) motion control and image segmentation auxiliary tasks and takes multi-modal input, sequence of camera images, and trajectory history. Different from other works, they have used spatiotemporal information and large-scale crowd-sourced video data, making the system more robust toward unseen scenarios.

The works mentioned above generally focus on jointly training the main and auxiliary tasks to enhance the training performance and robustness of final control commands. However, these approaches are not trained for challenging sequential decision-making tasks such as motion planning. In addition, these methods still lack interpretability as they follow direct mapping techniques and may suffer over-fitting problems, which is most common in soft parameter sharing [35]. Recent studies have adopted bird's eye view (BEV)-centric cascaded multi-task learning approaches that jointly learn several interpretable intermediate representations, e.g., the object detection and prediction results or the egocentric semantic maps in BEV space, and then use it to perform motion planning [36–38]. These approaches generally require expensive sensors such as LiDAR and HD maps, which are time-consuming to process and have a large gap with perspective (front)-view space. In contrast to these methods, we use easily accessible front and top-view camera images (RGB image sequences) to learn view-invariant spatiotemporal representations from which we predict more interpretable motion planners and controllers while maintaining robustness and efficiency.

### *2.3. Attention-Based Methods*

Due to its potential to improve model efficiency, robustness, and interpretability in deep learning models, the attention mechanism has a great promotion in learning various tasks such as caption generation [42], classification [43,44], etc. For ADVs, several attention mechanisms were designed to point out important factors that could assist in correcting the prediction and classification of driving behavior. For instance, Kim and Canny [45] investigated using a visually attentive CNN to learn steering angles from images and showed the importance of simpler visual attention maps to learn and maintain control accuracy. Mehta et al. [46] introduced soft attention for extracting observation attention from route-planer, residual block, and speed inputs. Then, they apply the same attention module to efficiently learn primitive action and affordance of sub-driving policies, which are used as input to enhance the final control command outputs. Motivated by [42,43], which applies the attention mechanism to RNNs and LSTM, temporal attention mechanisms have also been adopted in autonomous driving [19,22,50]. Deo and Trivedi [19] applied temporal attention mechanisms for vehicle trajectory prediction. Cai et al. [22] introduced a self-attention-based LSTM module for trajectory planning. Zhao et al. [50] applied separate spatial and temporal attention mechanisms to capture driver speed and steering decision information for vehicle speed and angle prediction. However, the methods mentioned above require different modules for learning intermediate representations in channel, spatial, and temporal dimensions.

Recently, module-based attention mechanisms [47–49], which can easily be applied to existing CNNs and simultaneously learns attentive feature maps in different dimensions (e.g., channel, spatial, temporal) without requiring additional modules, have been proposed. for example, Mori et al. [51] introduced attention block network (ABN) [47] into the autonomous driving model to obtain attention maps, which enables not only improved control performance but also intuitively analyzes it. Ishihara et al. [52] adopted CBAM [48] to generate a task-specific channel and spatial attention-weighted latent feature maps for multitask learning. Although these methods dealt with learning attentive features, which are mostly applied to classification and segmentation tasks, they do not take into account the simultaneous acquisition of spatiotemporal attention for sequential decision-making problems. Recently, STAMPNet [24] applied the squeeze-and-excitation (SE) module [49] in their feature extractor and 3D-ResNet [53] to learn attended intermediate features of the video and trajectory history for trajectory planning. Following this work, we introduced the SE module into our 3DCNN feature extractor, but instead of using a single backbone, we use a Siamese backbone to simultaneously learn intermediate spatiotemporal features that are invariant across (front and top) driving views. Then, the learned intermediate representations are utilized to co-optimize and train LSTM trajectory planning and CNN-FC controller modules. To further enhance spatiotemporal information and make it applicable to sequential decision-making, we introduce an attention mechanism into the LSTM module to produce feasible future trajectory plans.

### 2.4. Invariant Representation Learning

Using deep learning to train autonomous driving systems has seen many successes. However, learning representations that generalize across domains and tasks remains challenging due to the inherent domain shift problem. To deal with the issue of domain shift (e.g., variation in road views, weather, and lighting conditions), most approaches use data augmentation as well as diverse data collection methods [12,30]. For example, authors in [12] augment driving center road view training data by adding corresponding shifts (off-center), i.e., left and right road view camera images from onboard cameras. This increases the driving model's generalization ability and tackles the domain shift problem. Bansal et al. [30] add synthetic perturbation to the expert trajectory by training a driving policy on the semantic segmentation as well as the image to increase the robustness of the network towards domain shift. However, the data distribution shift problem remains challenging even with data augmentation.

Other approaches are domain knowledge transfer [6,33] and privileged information [16,54], which provide intuitive representations and can be applied to domain adaptation, transfer learning, or multitask learning scenarios. The authors in [6,33] proposed to predict "affordances" such as the distance between lanes and vehicles, or the status of the streetlights and use it as domain knowledge to improve driving performance. Learning such representations may add robustness, interpretability, and efficiency into models; however, they have limited scale in a dynamic environment. As such, they are predefined and may introduce spurious inputs. Xu et al. [16] proposed a method that exploits a privileged/side-task (e.g., segmentation) training paradigm, which shares the same labels but differs in their input modalities; thus, input modality from the source task is privileged. This enables the model to learn a relevant scene representation feature that improves motion prediction performance. The data-driven multi-source domain adaptation [55] is closest to ours. The recent popular domain adaptation methods have maximum mean discrepancy loss (MMD loss) [56,57], which projects different views of data into a shared subspace to minimize their discrepancy.

Motivated by these works, we incorporate multi-view metric learning (MMD loss) into our attentive 3DCNN network decision-making spatiotemporal representations that are invariant across front and top road views.

## 3. Proposed Approach

### 3.1. Overview

This section describes the general overview of the proposed ViSTAMPCNet for autonomous vehicles, shown in Figure 1. In general, the ViSTAMPCNet is based on imitation learning using CNN-LSTM models [16,22,24], which involves a mapping from expert observations to view-invariant spatiotemporal representations. Then, the view-invariant spatiotemporal representations are used for driving decision making, i.e., driving control command and future trajectory generation. The proposed ViSTAMPCNet comprises Siamese backbones to extract the discriminative spatiotemporal representations from two views of driving scenes. To minimize the representation distribution shift between two driving views, we adopt maximum mean discrepancy (MMD) metric learning [56], which takes representative features from Siamese backbones as input to capture the view-invariant information from center- and top-views of video input. Finally, the learned view-invariant representations are fed into the trajectory planner and controller modules.

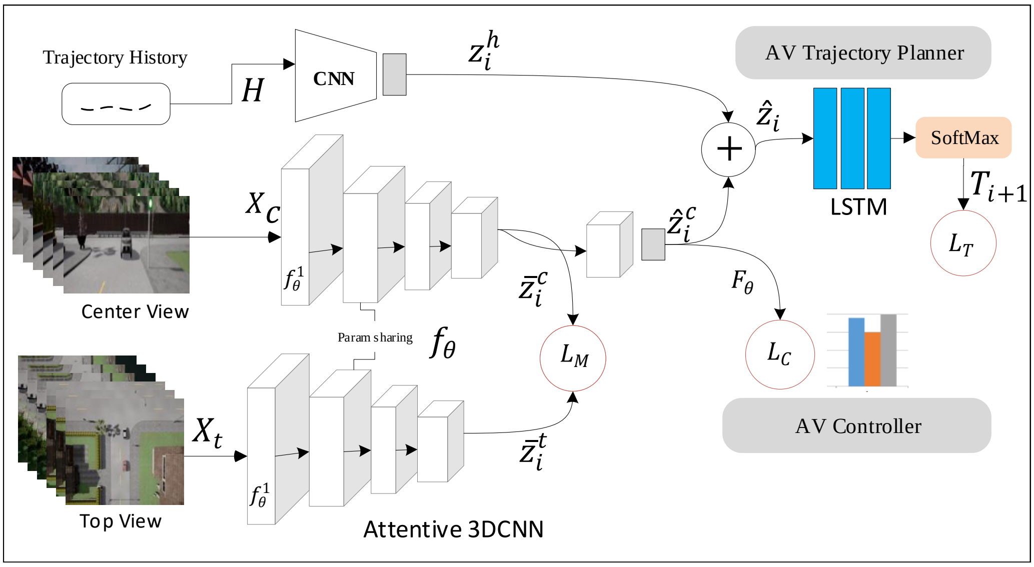

**Figure 1.** The proposed ViSTAMPCNet is comprised of two parts: representation learning and driving decision making. The representation learning uses Siamese 3DCNN, which is responsible for learning a mapping from raw image sequences directly to view-invariant spatiotemporal representations. The driving decision making part is responsible for learning the mapping from the learned representation to future trajectories and control output using LSTM and CNN, respectively.

### 3.2. Problem Formulation

Considering view-invariant trajectory planning and control tasks, given a set of recorded videos from human driver in the center and top views of the road: (1) let $\mathcal{X}_c = \{(x_i^c, y_i^c, T_i^c)\}_{i=1}^{\mathcal{N}_c}$ denote the center views, where $x_i^c$ is an instance of $\mathcal{X}_c$, $y_i^c$ is the corresponding control command (ground truth label), and $T_i^c$ is the ground truth trajectory; (2) let $\mathcal{X}_t = \{x_j^t\}_{j=1}^{\mathcal{N}_t}$ denote the top views, where $\{x_i^c, x_j^t\} \in \mathcal{R}^{T\times H\times W\times C}, i = j$ and

$\{\mathcal{N}_c, \mathcal{N}_t\}$ represent the total number of videos in center and top views, respectively. Additionally, corresponding to each center view video input $x_i^c$, there exists a trajectory history $\mathcal{H} = \{h_1, h_2, \ldots h_{\mathcal{N}_c}\}$, where $h_i \in [1, \mathcal{N}_c]$ is a single trajectory instance.

Subsequently, we adopt 3D-ResNet [53] as a backbone encoder. Spatiotemporal attention is incorporated in the backbone encoder, which we denote as an attentive 3D Siamese backbone $f_\theta(.)$ to encode discriminative spatiotemporal features from the center and top views of video inputs. We also define a convolutional layer followed by the fully connected layer as $g_\theta(.)$ to extract the previous trajectory information $h_i$. Suppose $f_\theta^k$ is the residual blocks (layers) of the 3D ResNet18 $f_\theta$, where $k \in [1, 5]$. The center view $x_i^c$ and top view $x_i^t$ are encoded as $\bar{z}_i^c = f_\theta^k(x_i^c)$ and $\bar{z}_i^t = f_\theta^k(x_j^t)$ from the penultimate layer of the Attentive 3D-ResNet-18 $f_\theta(.)$ backbone, where $k = \{1, 2, 3, 4\}$. Then, these $\mathcal{R}^{512}$ dimensional feature embeddings are used to compute the MMD loss $\mathcal{L}_M$ ($\bar{z}_i^c$,$\bar{z}_i^t$) to minimize the spatiotemporal feature discrepancy between the center and top driving view image sequences. Since the inter-channel information across the temporal dimension is aggregated and extracted using the Squeeze-and-Excitation [49] attention module, these spatiotemporally attended feature representations are considered to be more discriminative. Therefore, MMD loss reduces the distributional discrepancy between these low-dimensional spatiotemporal features based on the assumption that if the mean of the corresponding driving views $\bar{z}_i^c$ and $\bar{z}_i^t$ generated by $f_\theta(.)$ are equal or similar. The trajectory history can be encoded as $z_i^h = g_\theta(h_i)$. Afterward, the feature vector from the center view $\hat{z}_i^c = f_\theta^k(\bar{z}_i^c)$ is further utilized for attentive trajectory planing jointly with motion features extracted from the trajectory history as $\hat{z}_i = z_i^h + \hat{z}_i^c$. A three-layer LSTM module $\mathcal{G}_\theta(.)$ is defined to generate the future trajectories from highly representative spatiotemporal features as $T_{i+1} = \sigma(\mathcal{G}_\theta(\hat{z}_i))$, where $\sigma(.)$ is a SoftMax function to further refine trajectory features. Similarly, the vehicle controller $F_\theta(.)$ takes in the feature $\hat{z}_i^c$ and predicts high-level driving commands as $\hat{y}_i = F_\theta(\hat{z}_i^c)$.

### 3.3. View-Invariant Representation Learning Module

*Attentive spatiotemporal feature extractor.* For driving decision-making, autonomous vehicles need to encode the surrounding environment's information from the perceptual input. To achieve this, the proposed Siamese 3D networks take center $x_i^c$ and top $x_i^t$ view image sequences as input and extract spatiotemporal features $\bar{z}_i^c$ and $\bar{z}_i^t$ using the residual block shown in Figure 2. However, naively passing all extracted information from the 3D-ResNet-18 $f_\theta(.)$ backbone may increase the computation burden and lack interpretability. Instead of directly passing these extracted features to the MMD module, attention is applied to the residual block to selectively learn more salient spatiotemporal representations while filtering out irrelevant inputs for driving decision-making. Attention block guides the model to selectively learn spatiotemporal representations from the center and top-view image sequences while adding interpretability to the model. As in Figure 2, the channel-wise attention module $S_{se}$ aggregates the inter-channel relationship to attend meaningful patterns and perform two major tasks: the squeeze operation performs Global Average Pooling (GAP) to squeeze the input tensor $T \times C \times H \times W$ into $T \times C \times 1 \times 1$ and then extract the mean values across each channel $H \times W$.

As shown in Figure 2, the squeeze-and-excitation block $S_{se}$ takes the output feature map $\{z_i^c, z_i^t\} \in R^{T \times H \times W \times C}$ as an input and transform them into attentive intermediate feature vectors $\bar{z}_i^c$ and $\bar{z}_i^t$, where $\{\bar{z}_i^c, \bar{z}_i^t\} \in R^{T'H' \times W' \times C'}$ dimensions, which can be obtained as:

$$\bar{z}_i^c = S_{se}(\sigma(W_2\rho(W_1 \frac{1}{W \times H} \sum_{k=1}^{H} \sum_{L=1}^{W} z_i^c(k, l)))) \tag{1}$$

$$\bar{z}_i^t = S_{se}(\sigma(W_2\rho(W_1 \frac{1}{W \times H} \sum_{k=1}^{H} \sum_{L=1}^{W} z_i^t(k, l)))) \tag{2}$$

where $S_{se}$ is a squeeze-and-excitation block, $\sigma$ is a sigmoid activation function, $\rho$ is a ReLU activation, $W_1 \in R^{\frac{c}{r} xC}$ and $W_2 \in R^{\frac{c}{r} xC}$ are learned weights from two fully connected layers and added non-linearity. $C$ denotes channel, $r$ is reduction ratio The attended output features obtained from the $S_{se}$ block are re-weighted through the re-scale operation as:

$$\bar{z}_i^c = f_\theta(F_{scale}(z_i^c, \bar{z}_i^c) + x_i^c)) = f_\theta(z_i^c \times \bar{z}_i^c) + x_i^c) \tag{3}$$

$$\bar{z}_i^t = f_\theta(F_{scale}(z_i^t, \bar{z}_i^t) + x_i^t)) = f_\theta(z_i^t \times \bar{z}_i^t) + x_i^t) \tag{4}$$

where $F_{scale}$ is a scaling factor and $\bar{z}_i^c$ and $\bar{z}_i^t$ are the attentive feature vectors from the center and top driving views, respectively.

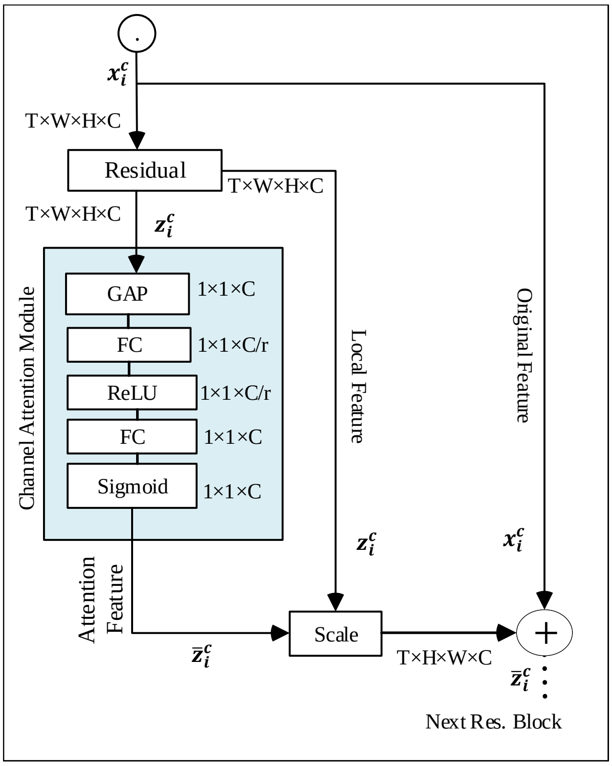

**Figure 2.** An illustration of the residual and attention block for center view input $x_i^c$.

*View-invariant feature learning.* To minimize the feature distribution shift between the center and top driving view training samples, the learned intermediate features $\bar{z}_i^c$ and $\bar{z}_i^t$ are used as input to MMD loss $L_M$. The $L_M$ loss is used to penalize the whole Siamese backbone network and encourage shared feature learning, while minimizing the spatiotemporal representation discrepancy.

Then $L_M$ loss function is given by:

$$\begin{aligned} L_M &= \psi(\bar{z}_i^c, \bar{z}_i^t) \\ &= \mathbb{E}_{\bar{z}_i^c \sim c}[\psi(\bar{z}_i^c)], \mathbb{E}_{\bar{z}_i^t \sim t}[\psi(\bar{z}_i^t)] \\ &= \mathbb{E}_{\sim c}[\psi(\bar{z}_i^c)] + \mathbb{E}_{\bar{z}_i^t \sim t}[\psi(\bar{z}_i^t)] - 2\mathbb{E}_{\bar{z}_i^c \sim c, \bar{z}_i^t \sim t}[\psi(\bar{z}_i^c, \bar{z}_i^t)] \end{aligned} \tag{5}$$

where $\psi(.)$ is a kernel function.

### 3.4. Trajectory Planning and Control Module

*High-level control.* We introduce the methods for predicting discrete control commands. Most of the existing end-to-end ADV approaches [6,12,16] rely on single modal input sequences, which are insufficient to make robust driving control decisions. Moreover, features learned by these methods have limited robustness towards domain-shift and interpretability [41]. To address these challenges, we introduce view-invariant attention-based motion prediction models that provide more generalizable features, which further improves interpretability of the driving commands. Therefore, given discriminative spatiotemporal features $\hat{z}_i^c$ from the Siamese backbone $f_\theta$, we further transform it using the control module $\hat{y}_i = F_\theta(\hat{z}_i^c)$. Then, calculate high-level control commands $\mathcal{L}_c$ with $N$-class cross-entropy loss using the following equation:

$$\mathcal{L}_C = -\sum_i^N y_i \log F_\theta(\hat{z}_i^c) = -\sum_i^N y_i \log \hat{y}_i \tag{6}$$

where $N$ is the number of high-level control commands defined as predicting three feasible actions (i.e., go-straight, turn-left, turn-right).

*Trajectory planner.* Given learned attentive view-invariant features $\hat{z}_i^c$ extracted from the complementary Siamese backbones, and the corresponding sequence of trajectory histories $z_i^h$ as input, our LSTM network predicts its future trajectories $\mathcal{G}_\theta(.)$, which can be expressed as follows:

$$\begin{aligned} \hat{z}_i &= z_i^h + \hat{z}_i^c \\ T_{i+1} &= \Sigma(\mathcal{G}_\theta(\hat{z}_i)) \end{aligned} \tag{7}$$

where $T_{i+1}$ is the predicted future trajectory. To optimize the trajectory planning model, we use the MSE [24] objective function over $T_i^c$ and the predicted trajectory $T_{i+1}$ as follows:

$$\mathcal{L}_T = \|T_i^c - T_{i+1}\|_2^2 \tag{8}$$

To better leverage sequential information and encourage discriminative learning, we also introduced the SoftMax function $\Sigma(.)$ into the LSTM module, which produces refined spatiotemporal features for the trajectory planner.

*The overall objective.* To optimize the proposed ViSTAMPCNet, we aggregate all the losses, namely MMD loss $\mathcal{L}_m$, trajectory loss $\mathcal{L}_T$, and control loss $\mathcal{L}_C$ as follows:

$$\mathcal{L}_{total} = \gamma\mathcal{L}_m + \mathcal{L}_C + \mathcal{L}_T \tag{9}$$

where $\gamma = 1$ is a hyper-parameter to control the MMD loss.

## 4. Experiments

In this section, we evaluate the performance of the proposed ViSTAMPCNet. To this end, we introduce the experimental settings, including datasets, implementation details, and experiment results.

### *4.1. Dataset and Evaluation Metrics*

We implement the proposed ViSTAMPCNet with PyTorch and train it on the VTG-Driving dataset [22]. The VTG-Driving dataset consists of three driving situation sub-datasets, i.e., go-straight, turn-left, and turn-right, each containing front (center) and top road view image sequences in different weather/lighting conditions. Our final dataset contains 88,558 image sequences that cover five different weather/lighting (e.g., clear day, sunset, night, foggy day, rainy day) driving environments as illustrated in Figure 3, each having a unique style of visual appearance and difficulty level for autonomous driving as mentioned in [22].

For evaluation, we adopt the $L_2$ loss (Equation (8)) and top-1 accuracy metrics to validate the trajectory generation performance and high-level control command classification accuracy following VTG-Net [22] and FCN-LSTM [16], respectively. We use trajectory generation results and control accuracy obtained from prior works, such as FCN-LSTM [16], CNNState-FCN [20], VTGNet [22], and STAMPNet [24] as baselines.

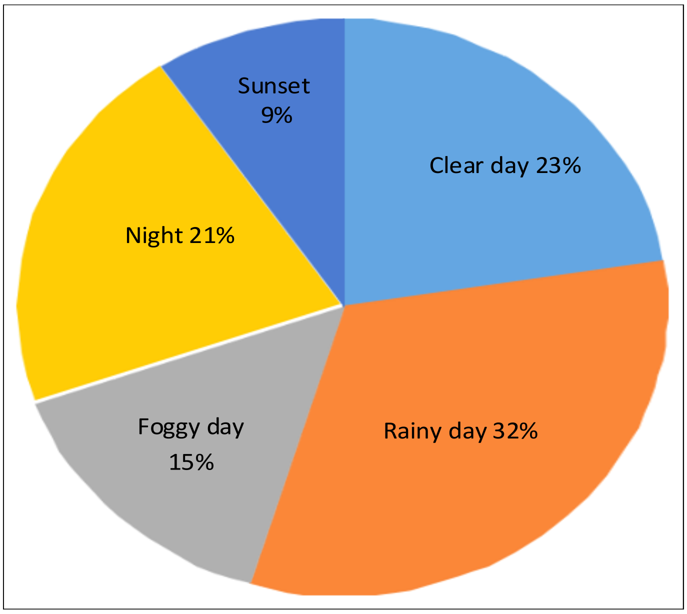

**Figure 3.** Dataset distribution across various weather conditions.

### *4.2. Implementation Settings*

We split the training and test dataset 80:20. We sampled 12 frames per video clip with a frame size of 3 × 112 × 112 resolutions. The experiments were conducted on two RTX2080Ti 11GB GPUs with a batch size of 64 for 100 max epochs. The Adam optimizer with learning rate of 0.001 was used for model optimization. As shown in Table 1, the proposed method employed 3D ResNet18 Siamese backbone network integrated with channel-wise attention module to extract intermediate spatiotemporal features from center and top road views as well as trajectory history. While both the center and top views are used during model training, only center view is used during testing. In addition, we employed GradCam [58] to generate the attention heatmap from the proposed backbone weight and 2D-tSNE [59] to project learned features in the embedding space for high-level control commands.

### *4.3. Experiment Results and Discussion*

#### 4.3.1. Ablation Study

In this subsection, we conduct extensive experiments to validate various components of the proposed ViSTAMPCNet across dataset with diverse weather conditions. In order to demonstrate the contribution of each component of the proposed ViSTAMPCNet, we evaluate the effectiveness of the 3DResNet blocks, spatiotemporal attention mechanisms, and MMD loss components on trajectory planning and control under various weather and lighting conditions.

*Effectiveness of each 3DResNet18 block.* To show the effectiveness of the 3DResNet18 layers in learning intermediate representations for trajectory planning and control, we trained the proposed backbone 3DResNet18 by removing the last layer as 3DResNet18(-1), two layers as 3DResNet18(-2), and three layers as 3DResNet18(-3), and show the evaluation results in Table 2.

As shown in Table 2, the 3DResNet18(-2) backbone has a trajectory error of 1.23, which is lower than the of 3DResNet18(-1) and 3DResNet18(-3) backbone networks, which are 5.58 and 3.22. In addition, 3DResNet18(-2) achieved a high-level control accuracy of 93.348, which is higher than 3DResNet18(-1) and 3DResNet18(-3), which are 79.890 and 92.97, respectively. This result indicates that the 3DResNet18(-2) can capture better intermediate representation that is beneficial for joint optimization of planning and control tasks compared to other blocks. Hence, the backbone for the proposed ViSTAMPCNet is constructed by removing the last two layers from the attentive 3DResNet18 Siamese backbone, which achieves the best performance in trajectory planning and control compared to other blocks.

**Table 1.** A detailed description of the different layers of ViSTAMPCNet is provided, including (i) Siamese 3DResNet Video encoder Backbone, which extracts discriminative intermediate spatiotemporal features via the residual and attention blocks, (ii) feasible future trajectory generation module, and (iii) high-level control command classification module.

| Modules | Blocks | Conv Layers | Number of Blocks |
|---|---|---|---|
| (i) Spatiotemporal Feature Extractor (3DResNet) × 2 | Input | Conv3d (7,7), 110, stride = 2<br>MaxPool3d, 3 × 3, stride = 2 | |
| | BasicBlock | Conv3d, (3,3), 144, stride = 1<br>Conv3d, (1,1), 64, stride = 1<br>Conv3d, (3,3), 144, stride = 1<br>Conv3d, (1,1), 64, stride = 1 | 2 |
| | AttentionBlock | FC(64,32), FC(32,64), Sigmoid | 2 |
| | BasicBlock | Conv3d, (3,3), 230, stride = 1<br>Conv3d, (1,1), 128, stride = 1<br>Conv3d, (3,3), 288, stride = 1<br>Conv3d, (1,1), 128, stride = 1 | 2 |
| | AttentionBlock | FC(128,64), FC(64,128), Sigmoid | 2 |
| | BasicBlock | Conv3d, (3,3), 460, stride = 1<br>Conv3d, (1,1), 256, stride = 1<br>Conv3d, (3,3), 576, stride = 1<br>Conv3d, (1,1), 256, stride = 1 | 2 |
| | AttentionBlock | FC(256,128), FC(128,256), Sigmoid | 2 |
| | BasicBlock | Conv3d, (3,3), 921, stride = 1<br>Conv3d, (1,1), 512, stride = 1<br>Conv3d, (3,3), 1152, stride = 1<br>Conv3d, (1,1), 512, stride = 1 | 2 |
| | AttentionBlock | FC(512,256), FC(512,256), Sigmoid | 2 |
| (ii) Trajectory | LSTM | LSTM(704, 512)<br>FC(512,66), FC(512,1) | 3<br>1 |
| | Trajectory History | Conv1d,(1,1), 256, stride = 1<br>FC(256,256), FC(256,256) | 1 |
| (iii) Control | Output | FC(8192,128), FC(128,Command = 3), Softmax | 1 |

**Table 2.** Effectiveness of attentive Residual Blocks. − indicates the number of removed last layers in attentive 3DResNet18 when learning spatiotemporal representations.

| Attentive 3DCNN | Block Layers | L2 Loss | Accuracy (%) |
|---|---|---|---|
| Attentive 3DResNet18 | −1 | 5.580 | 79.890 |
| Attentive 3DResNet18 | −2 | **1.230** | **93.348** |
| Attentive 3DResNet18 | −3 | 3.220 | 92.970 |

*Effectiveness of attention mechanism.* To see the effectiveness of the attention mechanism in learning discriminative spatiotemporal representation for planning and control, we trained the proposed method with and without the attention module. As shown in Table 3, the trajectory planning error of Vi STAMPCNet with spatiotemporal attention was 2.944, 2.4, 1.23, which is significantly lower than that of ViSTAMPCNet without spatiotemporal attention: 3.123, 2.74, 2.491, respectively, in foggy, rainy, and clear days. Similarly, with attention, the proposed method achieved much higher high-level control accuracy: 88.242, 91.857, and 93.34 than ViSTAMPCNet without attention—85.205, 87.164, and 87.424, respectively, across foggy, rainy, and clear weather conditions.

*Effectiveness of MMD loss.* To show the importance of the MMD component in the proposed method, we also trained ViSTAMPCNet with and without MMD loss. The experiment results presented in the middle of the Table 3 indicate that ViSTAMPCNet with MMD has a much lower trajectory planning error (L2 loss) 2.944, 2.401, and 1.23 compared to ViSTAMPCNet without MMD 4.82, 3.404, and 1.61, respectively, in clear, rainy, and foggy weather conditions. Similarly, ViSTAMPCNet with MMD also achieved much better control accuracy (Acc.) 88.242, 91.857, and 93.348 compared to ViSTAMPCNet without MMD 78.950, 85, and 93.2, respectively, in clear, rainy, and foggy weather conditions.

Considering the experimental results presented in Table 3, it is evident that jointly training both spatiotemporal attention and MMD loss components highly contribute to the proposed method's generalization and performance improvement in motion planner and controller regardless of change in environmental (weather/lighting) conditions.

**Table 3.** Evaluation of the performance of ViSTAMPCNet with and without spatiotemporal attention mechanism and MMD in motion planning and control under different weather/lighting conditions. Att., w/o, and Acc. denote attention, without, and accuracy, respectively.

| Weather | ViSTAMPCNet w/o Att. | | ViSTAMPCNet w/o MMD | | ViSTAMPCNet | |
|---|---|---|---|---|---|---|
| | L2 Loss | Acc. | L2 Loss | Acc. | L2 Loss | Acc. |
| Foggy day | 3.123 | 85.205 | 4.820 | 78.950 | 2.944 | 88.242 |
| Rainy day | 2.741 | 87.164 | 3.404 | 85.001 | 2.401 | 91.857 |
| Clear Day | 2.491 | 87.424 | 1.601 | 93.207 | **1.23** | **93.348** |

### 4.3.2. Qualitative Analysis

*Effectiveness and interpretability analysis.* To evaluate the effectiveness and interpretability of the proposed method for planning and control, we generated attention heatmaps using ViSTAMPCNet and visualized them using Grad-CAMs [58].The Grad-CAM visualization results shown in Figure 4 illustrate a comparison of the learned spatiotemporal attention using the proposed method (Figure 4d) and baselines without MMD (Figure 4b) and without attention (Figure 4c). As shown in Figure, the proposed model with attention and MMD (Figure 4d) clearly focuses on the dynamic agents (e.g., vehicles and pedestrians) and the road ahead across different environments while giving some attention to distant road markings and vehicles. This ability to dynamically pay attention in important driving regions is why our model outperforms the baseline, which indiscriminately pays equal attention, e.g., ViSTAMPCNet (w/o Att.) (Figure 4c), which is less effective in capturing critical driving road regions due to the absence of spatiotemporal attention mechanism. As expected, the other baseline model, STAMPNet (ViSTAMPCNet without MMD; Figure 4b) showed the worst visualization performance compared to ViSTAMPCNet with MMD loss. This shows the importance of MMD loss in assisting the proposed method to capture invariant representation, which thereby improves model ability to handle distribution shifts due to dynamic changes in driving views across weather and lighting conditions.

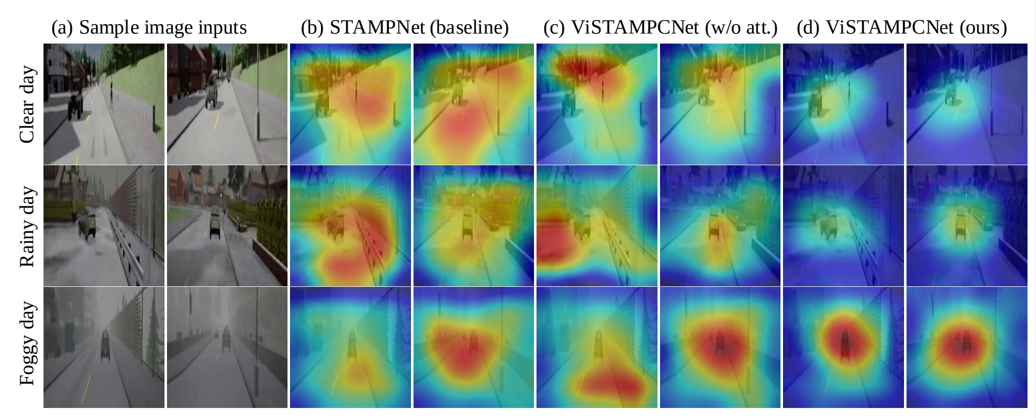

**Figure 4.** Attention heat map visualization of our ViSTAMPCNet and baselines. (**a**) Sample image sequence across different weather conditions. (**b**–**d**) Attention heat map visualization of STAMPNet baseline (without MMD loss), ViSTAMPCNet(without attention), and ours (ViSTAMPCNet with attention and MMD loss), respectively.

In summary, the proposed ViSTAMPCNet (Figure 4d) achieves better attention heat map visualization results than baseline methods without MMD loss as well as attention mechanisms. While the attention mechanism enables ViSTAMPCNet to learn more important regions of the driving environment, MMD loss minimizes spatiotemporal discrepancy due to dynamics in view and weather/lighting condition shifts in a driving environment. The results are further confirmed and illustrated in Figures 5 and 6, where our model generates control commands and future trajectories which are consistently closer to demonstrated driving behavior (ground truths) compared to other baselines.

*Two-dimensional tSNE visualization of embedding space for control.* We use the tSNE [59] to visualize the effectiveness of the proposed method in classifying high-level control commands in the embedding space earned with the proposed ViSTAMPCNet. Specifically, the 2D tSNE projection of 5.9 k test samples shown in Figure 5, illustrates the ability of proposed ViSTAMPCNet (right) in clearly semantically classifying control commands (turn-left, turn-right, and go-straight) compared to the other baselines: ViSTAMPCNet without MMD(left) and attention (middle). The visualized qualitative result indicates the effectiveness of joint MMD loss and attention mechanisms in our method for improved control classification performance and robustness.

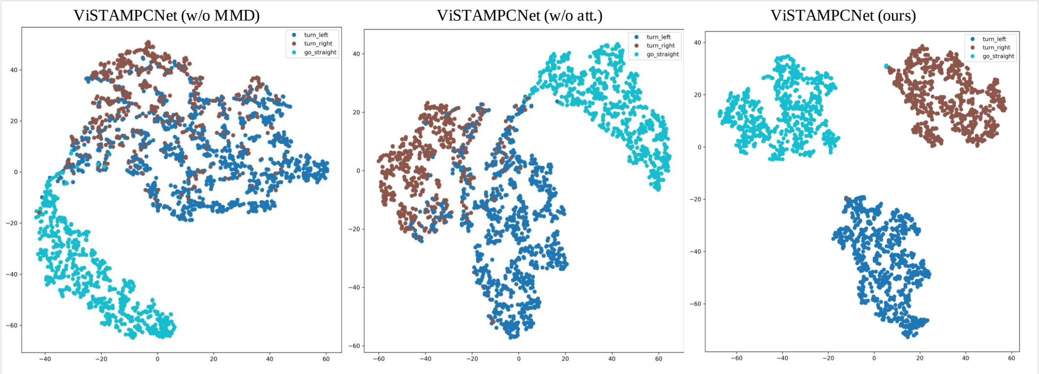

**Figure 5.** tSNE visualization for ViSTAMPCNet on 5.9 k test dataset taken from different weather conditions.

*Trajectory prediction analysis.* To further visualize the effectiveness of the proposed method on the trajectory prediction downstream task, we predict mid- and long-range future trajectories in three challenging driving scenes, including a clear day, a foggy day, and a rainy day as shown in Figure 6.

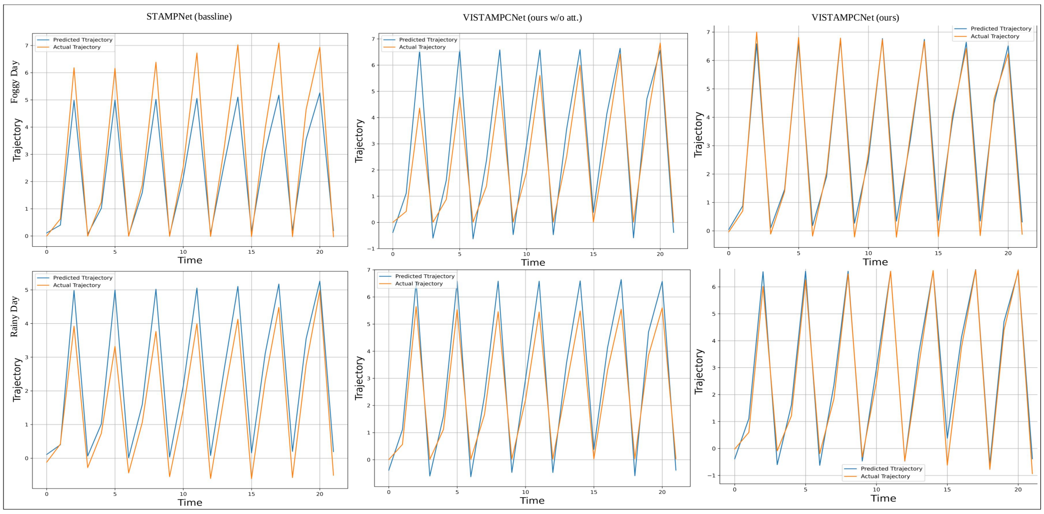

**Figure 6.** Visualization of future trajectory prediction for 22 way points.

The generated trajectories indicate that the proposed ViSTAMPCNet (ours) generates 22 trajeconfirmaypoints and has demonstrated that the predicted future trajectory is consistently close to the actual (ground truth) regardless of changes in driving views and weather conditions. On the other hand, the generated future trajectory waypoints with baseline models STAMPNet (without MMD) and the ViSTAMPCNet without attention deviated from the actual trajectories in both rainy and foggy weather conditions, demonstrating low performance. This illustrates the ViSTAMPCNet's ability to learn discriminative invariant representations that improve the robustness in driving decision-making ( e.g., motion planning) toward changes in driving environment, e.g., views and weather/lighting conditions.

### 4.3.3. Comparison with State-of-the-Art Methods

To show the superiority of VISTAMPCNet in motion planning and high-level control tasks, we compare the proposed VISTAMPCNet with several competing algorithms: CNN-LSTM [20], FCN-CNN [16], VTGNet [22], and STAMPNet [24]. For a fair comparison, Refs. [16,20] are re-implemented on VTG-Driving dataset [24] by replacing their backbone network with 3DResNet and results reported in [22,24] are directly compared as we use the same dataset and following similar setting with these works. As shown in Table 4, VISTAMPCNet has shown better planning performance and control accuracy improvement when compared with other multitask leaning networks FCN-CNN [16] and STAMPNet [24]. For example, our method outperforms the FCN-LSTM [16] model without attention and MMD loss by 20.948 and 1.677, respectively, in control and planning tasks. Compared to attentive 3D-CNN-LSTM [24], VISTAMPCNet has also shown better performance in planning (1.23 vs. 1.601) and control accuracy (93.207 vs. 93.348). The proposed method also achieved better performance in planning (2.491 vs. 1.230) and control (87.424 vs. 93.348) when compared against our baseline Siamese 3D-CNN-LSTM model without attention.

**Table 4.** Comparison with State-of-The-Art planning and control methods. $^\dagger$ indicates our implementation on VTG-Driving dataset [22]. Att., $T_p$, and $L_c$ denote motion planning, high-level control, and attention, respectively.

| Approaches | Task | | Architecture | L2 Loss | Accuracy (%) |
|---|---|---|---|---|---|
| | $T_p$ | $L_c$ | | | |
| Bergqvist [20] $^\dagger$ | ✓ | – | CNNState-FCN | 1.444 | – |
| VTGNet [22] | ✓ | – | 2D-CNN-LSTM | 1.036 | – |
| STAMPNet [24] | ✓ | – | Att. 3D-CNN-LSTM | 1.015 | – |
| Xu et al [16] $^\dagger$ | ✓ | ✓ | FCN-LSTM | 2.907 | 72.400 |
| STAMPNet + $L_c$ [24] $^\dagger$ | ✓ | ✓ | Att.3D-CNN-LSTM | 1.601 | 93.207 |
| ViSTAMPCNet (without Att.) | ✓ | ✓ | Siamese 3D-CNN-LSTM | 2.491 | 87.424 |
| ViSTAMPCNet **(Ours)** | ✓ | ✓ | Siamese Att.3D-CNN-LSTM | **1.230** | **93.348** |

Compared to single-task learning models, ViSTAMPCNet achieved better performance in motion planning against the CNN-LSTM model without attention and MMD [20], whereas it achieves competitive results with VTGNet [22] and STAMPNet [24] architectures that use attention mechanism without considering MMD loss. Although these single-task learning models are good at performing a single task, they perform worse when trained for other driving skills. For example, the attentive 3D-CNN-LSTM model [24] scored a higher planning error of 1.601 and lower control accuracy of 93.207 compared to ViSTAMPCNet's planning error of 1.23 and higher control accuracy of 93.348.

Overall, the experiment results provided in Figures 4 and 5, as well as comparison against state-of-the-art planning and control models shown in Table 4, confirm the importance of proposed spatiotemporal attention and MMD loss in improving the generalization of the capability of ViSTAMPCNet to new driving environments and tasks.

## 5. Conclusions

In this paper, we proposed a view-invariant spatiotemporal attentive motion planning and control network (ViSTAMPCNet) for autonomous vehicles. The proposed ViSTAMPCNet consists of invariant representation learning and driving decision-making modules. The representation learning module uses Siamese 3DCNN, which is responsible for learning a mapping from raw image sequences directly to view-invariant spatiotemporal representations. The driving decision-making module is responsible for learning the mapping from the learned representation to future trajectories and control output using LSTM and CNN, respectively. We demonstrate the effectiveness of the proposed ViSTAMPCNet through extensive experiments on a large-scale driving dataset with dynamic obstacles and weather/lighting conditions (e.g., clear, rainy, and foggy). Results from the evaluation and comparison against state-of-the-art methods confirm that invariant representations learned via the ViSTAMPCNet enable more generalizable motion planning and control in autonomous vehicles.

Although ViSTAMPCNet has shown promising results in motion planning and high-level control, it still has some limitations. (1) Even with learned invariant representation learning, the proposed method's robustness/scalability is limited, as it requires expert demonstrations in every scenario for training the network. Therefore, more studies are needed to improve the system's scalability, for example, by leveraging self-supervised learning approaches. (2) For autonomously driving in complex road scenarios, ADVs require more than discrete high-level commands. Therefore, more studies are needed to integrate low-level vehicle control tasks such as steering angle and speed control. (3) For good motion planning and control in extreme weather conditions, we also aim to incorporate multi-modal data from complementary sensors (such as lidar, radar, and thermal cameras) in future work while maintaining efficiency. This would increase the system's robustness in challenging environments.

**Author Contributions:** Conceptualization, M.A.; methodology, M.A.; software, M.A.; validation, M.A.; formal analysis, M.A., F.A. and M.B.B.H.; investigation, M.A. and S.Z.; resources, M.A., S.Z., M.B.B.H. and X.Z.; data curation, M.A., I.M. and M.B.B.H.; writing—original draft preparation, M.A.; writing—review and editing, M.A., S.Z., M.B.B.H., I.M., F.A. and X.Z.; visualization, M.A., I.M., M.B.B.H. and F.A.; supervision, S.Z.; project administration, S.Z. and X.Z.; and funding acquisition, S.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is funded by the Sichuan Science and Technology Program under Grant 2020YFG0472.

**Data Availability Statement:** Data are available upon request through Melese Ayalew (meleawima@gmail.com).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Paden, B.; Čáp, M.; Yong, S.Z.; Yershov, D.; Frazzoli, E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans. Intell. Veh. (T-IV)* **2016**, *1*, 33–55. [CrossRef]
2. Pendleton, S.D.; Andersen, H.; Du, X.; Shen, X.; Meghjani, M.; Eng, Y.H.; Rus, D.; Ang, M.H., Jr. Perception, planning, control, and coordination for autonomous vehicles. *Machines* **2017**, *5*, 6. [CrossRef]
3. Chen, H.Y.; Zhang, Y. An overview of research on military unmanned ground vehicles. *Acta Armamentarii* **2014**, *35*, 1696–1706.
4. Schwarting, W.; Alonso-Mora, J.; Rus, D. Planning and decision-making for autonomous vehicles. *Annu. Rev. Control Robot Auton. Syst.* **2018**, *1*, 187–210. [CrossRef]
5. Ly, A.O.; Akhloufi, M. Learning to drive by imitation: An overview of deep behavior cloning methods. *IEEE Trans. Intell. Veh. (T-IV)* **2020**, *6*, 195–209. [CrossRef]
6. Chen, C.; Seff, A.; Kornhauser, A.; Xiao, J. DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 2722–2730.
7. Thrun, S.; Montemerlo, M.; Dahlkamp, H.; Stavens, D.; Aron, A.; Diebel, J.; Fong, P.; Gale, J.; Halpenny, M.; Hoffmann, G.; et al. Stanley: The robot that won the DARPA Grand Challenge. *J. Field Robot* **2006**, *23*, 661–692. [CrossRef]
8. Fan, H.; Zhu, F.; Liu, C.; Zhang, L.; Zhuang, L.; Li, D.; Zhu, W.; Hu, J.; Li, H.; Kong, Q. Baidu apollo em motion planner. *arXiv* **2018**, arXiv:1807.08048.

9. McAllister, R.; Gal, Y.; Kendall, A.; Van Der Wilk, M.; Shah, A.; Cipolla, R.; Weller, A. Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning. In Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI), Melbourne, Australia, 19 August 2017; pp. 4745–4753.
10. Pomerleau, D.A. Alvinn: An Autonomous Land Vehicle in a Neural Network. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Denver, CO, USA, 27–30 November 1989.
11. Muller, U.; Ben, J.; Cosatto, E.; Flepp, B.; Cun, Y. Off-road obstacle avoidance through end-to-end learning. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vancouver, BC, Canada , 12 May–12 August 2005.
12. Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L.D.; Monfort, M.; Muller, U.; Zhang, J.; et al. End to end learning for self-driving cars. *arXiv* **2016**, arXiv:1604.07316.
13. Jhung, J.; Bae, I.; Moon, J.; Kim, T.; Kim, J.; Kim, S. End-to-end steering controller with CNN-based closed-loop feedback for autonomous vehicles. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 617–622.
14. Kocić, J.; Jovičić, N.; Drndarević, V. An end-to-end deep neural network for autonomous driving designed for embedded automotive platforms. *Sensors* **2019**, *19*, 2064. [CrossRef] [PubMed]
15. Chi, L.; Mu, Y. Deep steering: Learning end-to-end driving model from spatial and temporal visual cues. In Proceedings of the 24th International Conference on Pattern Recognition, Beijing, China, 20 August 2018.
16. Xu, H.; Gao, Y.; Yu, F.; Darrell, T. End-to-end learning of driving models from large-scale video datasets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21 July 2017; pp. 2174–2182.
17. Song, S.; Hu, X.; Yu, J.; Bai, L.; Chen, L. Learning a deep motion planning model for autonomous driving. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1137–1142.
18. Fern, O.T.; Denman, S.; Sridharan, S.; Fookes, C. Going deeper: Autonomous steering with neural memory networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW),Venice, Italy, 22–29 October 2017.
19. Deo, N.; Trivedi, M.M. Convolutional social pooling for vehicle trajectory prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1468–1476.
20. Bergqvist, M.; Rödholm, O. Deep Path Planning Using Images and Object Data. Master's Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2018.
21. Cai, P.; Sun, Y.; Chen, Y.; Liu, M. Vision-based trajectory planning via imitation learning for autonomous vehicles. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019.
22. Cai, P.; Sun, Y.; Wang, H.; Liu, M. VTGNet: A Vision-based Trajectory Generation Network for Autonomous Vehicles in Urban Environments. *IEEE Trans. Intell. Veh. (T-IV)* **2021**, *6*, 419–429. [CrossRef]
23. Ross, S.; Gordon, G.; Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics(AISTATS), Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 627–635.
24. Ayalew, M.; Zhou, S.; Assefa, M.; Yilma, G. spatiotemporal Attentive Motion Planning Network for Autonomous Vehicles. In Proceedings of the 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 17 December 2021; pp. 601–605.
25. Hecker, S.; Dai, D.; Van Gool, L. End-to-end learning of driving models with surround-view cameras and route planners. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 435–453.
26. Xiao, Y.; Codevilla, F.; Gurram, A.; Urfalioglu, O.; López, A.M. Multimodal end-to-end autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 537–547. [CrossRef]
27. Huang, Z.; Lv, C.; Xing, Y.; Wu, J. Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding. *IEEE Sens. J.* **2020**, *21*, 11781–11790. [CrossRef]
28. Hawke, J.; Shen, R.; Gurau, C.; Sharma, S.; Reda, D.; Nikolov, N.; Kndall, A. Urban driving with conditional imitation learning. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 251–257.
29. Sallab, A.E.; Abdou, M.; Perot, E.; Yogamani, S. Deep reinforcement learning framework for autonomous driving. *Electron. Imaging* **2017**, *2017*, 70–76. [CrossRef]
30. Bansal, M.; Krizhevsky, A.; Ogale, A. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv* **2018**, arXiv:1812.03079.
31. De Haan, P.; Jayaraman, D.; Levine, S. Causal confusion in imitation learning. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vancouver, DC, USA, 10–12 December 2019.
32. Rhinehart, N.; Kitani, K.M.; Vernaza, P. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 772–788.
33. Sauer, A.; Savinov, N.; Geiger, A. Conditional affordance learning for driving in urban environments. In Proceedings of the 2nd Conference on Robot Learning (CoRL), Zurich, Switzerland, 29–31 October 2018; pp. 237–252.
34. Müller, M.; Dosovitskiy, A.; Ghanem, B.; Koltun, V. Driving policy transfer via modularity and abstraction. *arXiv* **2018**, arXiv:1804.09364.
35. Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv* **2017**, arXiv:1706.05098.

36. Luo, W.; Yang, B.; Urtasun, R. Fast and furious: Real-time end-to-end 3d detection, tracking, and motion forecasting with a single convolutional net. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
37. Zeng, W.; Luo, W.; Suo, S.; Sadat, A.; Yang, B.; Casas, S.; Urtasun, R. End-to-end interpretable neural motion planner. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA 15 June 2019; pp. 8660–8669.
38. Sadat, A.; Casas, S.; Ren, M.; Wu, X.; Dhawan, P.; Urtasun, R. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In Proceedings of the European Conference on Computer Vision(ECCV), Glasgow, UK, 23–28 August 2020; pp. 414–430.
39. Yang, Z.; Zhang, Y.; Yu, J.; Cai, J.; Luo, J. End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 2289–2294.
40. Codevilla, F.; Müller, M.; López, A.; Koltun, V.; Dosovitskiy, A. End-to-end driving via conditional imitation learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 4693–4700.
41. Codevilla, F.; Santana, E.; López, A.M.; ; Gaidon, A. Exploring the limitations of behavior cloning for autonomous driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9329–9338.
42. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 2048–2057.
43. Mnih, V.; Heess, N.; Graves, A. Recurrent models of visual attention. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014) , Montreal, QC, Canada, 8–13 December 2014.
44. Gedamu, K.; Yilma, G.; Assefa, M.; Ayalew, M. Spatio-temporal dual-attention network for view-invariant human action recognition. In Proceedings of the Fourteenth International Conference on Digital Image Processing (ICDIP 2022), Wuhan, China, 12 October 2022; Volume 12342, pp. 213–222.
45. Kim, J.; Canny, J. Interpretable learning for self-driving cars by visualizing causal attention. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 25 December 2017; pp. 2942–2950.
46. Mehta, A.; Subramanian, A.; Subramanian, A. Learning end-to-end autonomous driving using guided auxiliary supervision. In Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing, Hyderabad, India, 18–22 December 2018; pp. 1–8.
47. Fukui, H.; Hirakawa, T.; Yamashita, T.; Fujiyoshi, H. Attention branch network: Learning of attention mechanism for visual explanation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
48. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
49. Hu, J., Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
50. Zhao, X.; Qi, M.; Liu, Z.; Fan, S.; Li, C.; Dong, M. End-to-end autonomous driving decision model joined by attention mechanism and spatiotemporal features. *IET Intell. Transp. Syst.* **2021**, *15*, 1119–1130. [CrossRef]
51. Mori, K.; Fukui, H.; Murase, T.; Hirakawa, T.; Yamashita, T.; Fujiyoshi, H. Visual explanation by attention branch network for end-to-end learning-based self-driving. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1577–1582.
52. Liu, S.; Johns, E.; Davison, A.J. End-to-end multi-task learning with attention. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 5–20 June 2019; pp. 1871–1880.
53. Hara, K.; Kataoka, H.; Satoh, Y. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 19–21 June 2018.
54. Lopez-Paz, D.; Bottou, L.; Schölkopf, B.; Vapnik, V. Unifying distillation and privileged information. *arXiv* **2015**, arXiv:1511.03643.
55. Stojanov, P.; Gong, M.; Carbonell, J.; Zhang, K. Data-driven approach to multiple-source domain adaptation. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS), Naha, Japan, 16–18 April 2019; pp. 3487–3496.
56. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. *J. Mach. Learn. Res (JMLR)* **2012**, *13*, 723–773.
57. Yilma, G.; Gedamu, K.; Assefa, M.; Oluwasanmi, A.; Qin, Z. Generation and Transformation Invariant Learning for Tomato Disease Classification. In Proceedings of the 2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML), Chengdu, China, 16 July 2021; pp. 121–128.
58. Chattopadhay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 839–847.
59. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res. (JMLR)* **2008**, *9*, 2579–2605.