# Multi-task Learning with Future States for Vision-based Autonomous Driving

Inhan Kim[1 [0000−0002−1426−108X]], Hyemin Lee[1 [0000−0002−1899−7211]], Joonyeong Lee[1 [0000−0003−3217−2168]], Eunseop Lee[1 [0000−0002−6027−2863]], and Daijin Kim[1 [0000−0002−8046−8521]]
{kiminhan, lhmin, joonyeonglee, eunseop90, dkim}@postech.ac.kr

Department of Computer Science and Engineering, POSTECH, Korea[1]

**Abstract.** Human drivers consider past and future driving environments to maintain stable control of a vehicle. To adopt a human driver's behavior, we propose a vision-based autonomous driving model, called Future Actions and States Network (FASNet), which uses predicted future actions and generated future states in multi-task learning manner. Future states are generated using an enhanced deep predictive-coding network and motion equations defined by the kinematic vehicle model. The final control values are determined by the weighted average of the predicted actions for a stable decision. With these methods, the proposed FASNet has a high generalization ability in unseen environments. To validate the proposed FASNet, we conducted several experiments, including ablation studies in realistic three-dimensional simulations. FASNet achieves a higher Success Rate (SR) on the recent CARLA benchmarks under several conditions as compared to state-of-the-art models.

**Keywords:** Vision-based Autonomous Driving, Controller with Future Actions, Multi-task Learning based Autonomous Driving

## 1 Introduction

Traditionally, an autonomous vehicle can drive itself using localization information and motion equations defined by the kinematic vehicle model. With advances in deep learning, the classical rules for manipulating the vehicle can be learned by deep neural networks. Recently, we have witnessed wide and significant progress in autonomous driving, especially in the field of computer vision [1–5]. Furthermore, starting with a Conditional Imitation Learning (CIL) [6], several successive studies [6–11] apply high-level navigational commands (i.e., Follow Lane, Go Straight, Turn Right, and Turn Left) as provided by a navigation system to guide the global optimal path to reach the final destination.

Despite the impressive progress, vision-based autonomous vehicles face unexpected situations that can reduce driving accuracy and stability. For example, the autonomous vehicle can turn abruptly or stop and never move. Once an abnormal behavior is caused by an incorrectly predicted control value, unsafe situations are likely to occur and it may take a long time to recover to a safe

driving situation. In addition, several authors [12, 11, 13, 7, 9] have highlighted limited capability of a model trained with a single task learning manner or single RBG image as an input. To tackle this problem, we focus on the ways to reduce unstable control and increase the generalization ability of the vehicle controller.

First, we observed that human drivers manipulate vehicles safely by anticipating future situations. For example, when human drivers approach an intersection, they anticipate when to slow down and when to turn the steering wheel. This is the main reason why human drivers can usually control a vehicle with stability. However, most autonomous driving research uses only current information for decision making, which can lead to unstable longitudinal or lateral control. Motivated by this observation, we utilize the generated future frame using the concept of Deep Predictive-Coding Network (PredNet) [14] and calculated future localization information using kinematic motion equations [15].

Second, it improves the generalization and regularization abilities of the trained model with Multi-task Learning (MTL). MTL is a good solution for generalization by leveraging domain-specific information contained in training representations of other tasks through a joint optimization strategy [16]. Furthermore, learning with auxiliary tasks in MTL will improve the generalization for the main tasks without introducing any extra computational burden at inference time. Finally, MTL acts as a regularizer, which reduces the risk of overfitting, by introducing an inductive bias. With these benefits, MTL has displayed success in the field of autonomous driving [12, 13, 9, 7].

Our key contributions are summarized as follows: 1) We present a successful vision-based driving architecture based on the weighted average of future actions by preventing the situation wherein a single incorrect control action makes the vehicle's movement unstable; 2) We designed the MTL model, including auxiliary tasks, by generating future representations and future localization information without additional knowledge compared with a baseline model; and 3) We achieve enhanced and stable driving results, especially in unseen environments by applying these concepts to base networks.

## 2 Related Work

### 2.1 Deep Learning Based Autonomous Driving

There has been a steady release of various driving models that map camera pixels directly to the longitudinal and lateral controls. End-to-end driving models automatically learn important representations of the necessary processing steps, such as recognizing useful road features. Bojarski et al. [3] demonstrated that a vehicle could potentially be controlled using a deep Convolutional Neural Network (CNN) based solely on a front-facing camera. Following this research, various CNN-based end-to-end networks have been investigated [17, 18, 4, 6, 7, 19]. Other researchers are investigating Reinforcement Learning (RL) for controlling vehicles using, which trains a deep neural network based on a policy-based reward [20, 8, 21]. These systems can be generalized for unseen scenarios, and their

performance will be coverage of human driving. In these studies, the main aim is to learn human-controllable policy in a trial-and-error manner without explicit human supervision. However, RL-based approaches are difficult to apply in the real world because the training process is not reliable and fatal accidents may occur.

Recently, many studies have employed MTL to achieve the optimal information exchange between highly correlated tasks [22, 23, 12, 9, 7, 21]. Furthermore, based on these approaches, some studies employ auxiliary tasks, such as the result of segmentation network and optical flow as well as raw image [11, 13]. One of the reasons for this success is attributed to the inbuilt sharing mechanism, which allows deep neural networks to learn representations shared across different categories. This insight naturally extends to sharing between tasks and leads to further performance improvements.

Many studies have contributed to significant driving performance improvements; however, these studies may predict the unexpected control values that can reduce driving accuracy and stability. While driving, the vehicle can become quite unstable with just one wrong control value, particularly at higher speeds. Thus, we focus on a stable driving model and propose FASNet, which employs a weighted average action based on multiple predicted actions.

### 2.2 Action Prediction with Generated Representation

Various methods to predict realistic pixel values in future frames have been investigated [24, 25]. Most state-of-the-art approaches use generative neural networks to represent the pixel-wise appearance of a given image in a video sequence. The performance of these networks is undoubtedly impressive.

However, future frame prediction is a challenging task due to the complex appearance and motion dynamics. To address this problem, joint learning is the most commonly explored approach to model complex motion fields [26, 27]. In [26], the authors attempt to jointly train the network to resolve the prediction problems that derive from complex pixel-level distributions in natural images. In addition, an effective future frame prediction method for complex urban driving scenes utilizing video and optical flow sequences has been proposed [27].

Furthermore, recent research shows that learning correlated tasks simultaneously can enhance the performance of individual tasks [28]. In [29], the authors proposed a MTL approach by jointly predicting steering angles and future frame sequences. In addition, the latent variables of a multi-task generative network can be effectively used to predict vehicle steering angles [14, 30]. In this paper, we utilize an enhanced PredNet that has a specialized neural network branches, similar to the conditional branch in [6]. The branches are also selectively trained via high-level commands that indicate where the vehicle will go. With the conditional branches, we can generate more accurate future images and latent variables for angle prediction.

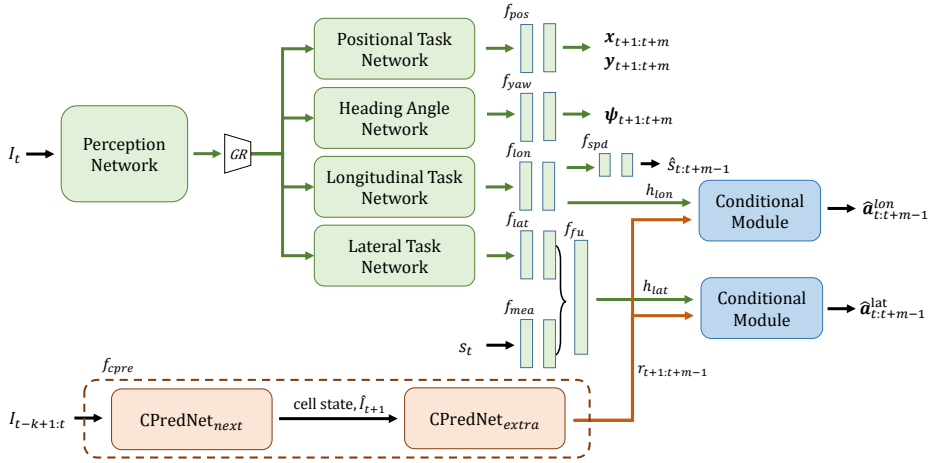


**Fig. 1.** Overall architecture of the FASNet. A perception network processes an input image to a latent space followed by four task-specific networks: two for localization tasks and two for control tasks. The subnetwork $f_{cpre}$, which has two submodules CPredNet$_{next}$ and CPredNet$_{extra}$, generates future representations from past sequential images. The future representations and embedded features by FC layers fused and fed into the conditional module, which predicting future longitudinal and lateral control values according to the navigational command.

## 3 Method

### 3.1 Task Definition

Conditional Imitation Learning (CIL) [6] for autonomous driving is a form of supervised learning that can learn a driving policy from human driver experts using given $N$ video sequences $v_i$, $i \in (1, \ldots, N)$ with observations $\boldsymbol{o}_{i,t}$, actions $\boldsymbol{a}_{i,t}$, and high-level navigational commands $\boldsymbol{c}_{i,t}$. Basically, all datasets have at least one command: follow the lane (Follow Lane). Various additional commands are added according to the driving scenarios, such as drive straight (Go Straight), turn left at the next intersection (Turn Left), and turn right at the next intersection (Turn Right). The observations consist of tuples, each of which contains an image ($I_{i,t}$) and measured signals, such as speed ($s_{i,t}$). Here, $\boldsymbol{a}_{i,t}$ actions include steering angle ($a^{str}_{i,t}$), an acceleration value ($a^{acc}_{i,t}$) and braking value ($a^{brk}_{i,t}$) of the human driver to manipulate the vehicle. The dataset is defined $D = \{(\boldsymbol{o}_{i,t}, \boldsymbol{a}_{i,t}, \boldsymbol{c}_{i,t})\}_{i=1}^{N}$.

The expanded strategy of the CIL is that CIL with a ResNet backbone Speed prediction (CILRS) model [7]. The CILRS objective function, which minimizes the parameters $\theta$ of the action prediction network ($F$), is expressed as follows:

$$\min_{\theta} \sum_{i}^{N} \sum_{t}^{T_i} L_a(F(I_{i,t}, s_{i,t}, G(c_{i,t})), \boldsymbol{a}_{i,t}), \tag{1}$$

where $L_a$ is L1 loss function for three predicted actions $\hat{\boldsymbol{a}}_t$. The command $c_{i,t}$ acts as branch selector that controls the selective activation of a branch via a gating function $G(c_{i,t})$. Each branch has Fully Connected (FC) layers to encode the distinct hidden knowledge for corresponding command and thus selectively used for action prediction. Further details can be found in the literature, e.g., Codevilla et al. [6] and Liang et al. [8].

Following the definition in [10], we define sequential notation. In the case of observation, the sequential notation $\{\boldsymbol{o}_{t-k+1}, \ldots, \boldsymbol{o}_t, \ldots, \boldsymbol{o}_{t+m-1}\}$ are represented as $\boldsymbol{o}_{t-k+1:t+m-1}$ where $k$ is the total number of past data and $m$ is the total number of future data. The proposed $F$ differs from other driving architectures in that it predicts future actions from current time step $t$ to $t+m-1$ using temporal images ($I_{i,t-k+1:t}$). The objective function can be rewritten as follows:

$$\min_{\theta} \sum_{i}^{N} \sum_{t}^{T_i} L_a(F(I_{i,t}, s_{i,t}, G(c_{i,t})), \boldsymbol{a}_{i,t:t+m-1}), \tag{2}$$

where $\boldsymbol{a}_{i,t:t+m-1}$ is the sequential ground truth of actions, $T_i$ is total time at each video sequence, and $L_a$ is also the L1 loss function for predicted sequential actions $\hat{\boldsymbol{a}}_{t:t+m-1}$:

$$L_a = \sum_{t}^{t+m-1} ||\hat{a}_{i,t}^{str} - a_{i,t}^{str}||^1 + ||\hat{a}_{i,t}^{acc} - a_{i,t}^{acc}||^1 + ||\hat{a}_{i,t}^{brk} - a_{i,t}^{brk}||^1. \tag{3}$$

### 3.2 Conditional Future Actions and States Prediction Network

We propose a novel architecture to predict future actions ($\hat{\boldsymbol{a}}_{t:t+m-1}$). Note that for simplicity, hereafter, the index of video sequences will be omitted in the definitions. We use a ResNet34 architecture [31] pretrained on the ImageNet as a backbone network for generalized driving on learning reactions to dynamic objects in complex environments [7]. As shown in Fig. 1, the ResNet34 architecture is divided into two groups: 1) Convolution stages from *Conv1* to *Conv4* are utilized for the perception network. The output feature map of the perception network is shared between all related task-specific subnetworks on hard parameter sharing manner. 2) The last convolution stage (*Conv5*) is used for task-specific networks. The feature maps, which are fed into each subnetworks, are trained to represent features that focus more on each task. Specifically, the task-specific network has a FC layer to embed the extracted features.

The positional task and heading angle networks in Fig. 1 estimate the future localization states such as positions of the vehicle in global coordinates ($x_{t+1:t+m}, y_{t+1:t+m}$) and heading angles ($\psi_{t+1:t+m}$). These localization tasks, which are closely related to target control tasks, are considered the auxiliary tasks to improve the generalization of the main tasks. Therefore, the positional task and heading angle networks are utilized only at the training time. The loss function of the localization states is defined as follows:

$$L_l = \sum_{t}^{t+m-1} ||\hat{x}_{t+1} - x_{t+1}||^2 + ||\hat{y}_{t+1} - y_{t+1}||^2 + ||\hat{\psi}_{t+1} - \psi_{t+1}||^2, \tag{4}$$

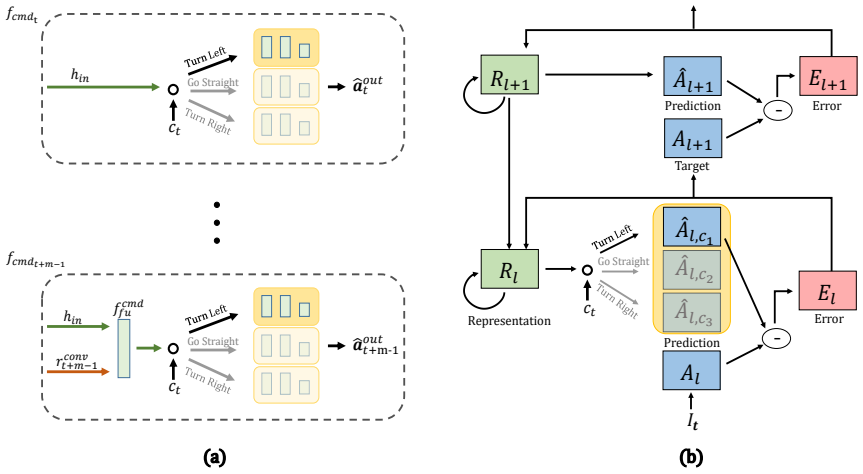


**Fig. 2. (a)** Architecture of the conditional module. The $h_{in}$ can be the $h_{lon}$ or $h_{lat}$. Additionally, the $\hat{\boldsymbol{a}}^{out}$ can be the $\hat{\boldsymbol{a}}^{lon}$ or $\hat{\boldsymbol{a}}^{lat}$ as well. **(b)** Information flow in two level modules in CPredNet. Each module consists of a recurrent representation layer ($R_l$), an input convolutional layer ($A_l$), a prediction layer ($\hat{A}_l$), and an error representation unit ($E_l$). In addition, a gating function choose activation layer in the first level using a command ($c_t$).

The longitudinal task network considers longitudinal outputs by processing the latent space followed by two prediction branches: one for hidden variables $h_{lon}$ to predict longitudinal actions ($\hat{a}^{acc}_{t:t+m-1}$, $\hat{a}^{brk}_{t:t+m-1}$) and one for speed sequence ($\hat{s}_{t:t+m-1}$) to achieve an effect of regularization [7]. The output, which passes through the lateral task network and $f_{lat}$, is fuse with the output of $f_{mea}$ by $f_{fu}$ in Fig. 1.

$f_{cpre}$ comprises two modified PredNets, which we refer to as CPredNets. The CPredNets generate future representations using sequential past images ($I_{t-k+1:t}$) and high-level command ($c_t$) as inputs. The outputs of $f_{cpre}$ are defined as $r_{t+1:t+m-1}$, where each $r_t$ is the lowest representation layer in the CPredNet, which learned latent variables to generate future frames $\hat{I}_{t+1:t+m-1}$. Note that the model can make optimal actions when there is no temporal gap between $\boldsymbol{o}$ and $\hat{\boldsymbol{a}}$. Therefore, predicting $\hat{\boldsymbol{a}}_{t+1:t+m-1}$ without $\boldsymbol{o}_{t+1:t+m-1}$ may reduce the accuracy. To prevent such loss of accuracy, the representational features $r_{t+1:t+m-1}$ act as the observation of the corresponding time step.

Each conditional module of $f_{cmd_{t+n}}$ has FC layers and a gating function to encode the distinct hidden knowledge for the corresponding command and thus is selectively used for prediction. However, in the case of $n > 0$, there is additional FC layer $f^{cmd}_{fu}$ in Fig. 2 (a) to combine inputs. Before fusing these features, we apply some $3 \times 3$ convolutional layers to $r_{t+1:t+m-1}$ to extract meaningful information named $r^{conv}_{t+n}$.

The proposed architecture has multi-task regression heads for taking maximized generalization capability using hard parameter sharing. According to the

[32], the hard parameter shared layers have unstable backpropagation flow and converge slowly without gradient rescaling, because the loss variances of multiple heads are unlimited. To make the variance always limited, a gradient rescaling module $GR$, which was proposed in [32], is added between the perception backbone network and task-specific networks in Fig. 1.

### 3.3 Future State Generation

In this study, we utilize two future states to achieve the benefits of MTL and eliminate the temporal gap between current and the future observations.

First of all, we employ the enhanced PredNet called CPredNet and architecture is diagrammed in Fig. 2. (b). Compared to the PredNet, the conditional branches in the CPredNet are applied to the first generative module to improve future frame generation capability. The role of high-level command ($c_t$) is a clear indicator of the directional change and acts as a switch that selects which representation branch is used at any given situation. The CPredNet follows all the PredNet training rules, except for the conditional learning. The further details and full set of update rules can be found in supplementary material.

The second way to generate the future state is to employ motion equations defined by the kinematic vehicle model. In this study, a discrete kinematic bicycle model with the desired point on the center of the front axle is used to describe the vehicle dynamics. According to the kinematic analysis, the vehicle localization information, which the position of the vehicle in global coordinates $(x, y)$ and consists of heading angle $(\psi)$, at that time $t+1$ are given as follows:

$$x_{t+1} = x_t + s_t cos(\delta_t + \psi_t)\Delta t \tag{5}$$

$$y_{t+1} = y_t + s_t sin(\delta_t + \psi_t)\Delta t \tag{6}$$

$$\psi_{t+1} = \psi_t + \frac{s_t tan(\delta_t)}{l}\Delta t \tag{7}$$

where $\delta$ is the wheel steering angle and $l$ is the wheel base.

Because we consider the offset of localization information from current position and heading angle, the first current states $x_t$, $y_t$, and $\psi_t$ can be set to zero. And then we can calculate the future coordinates and heading angles recursively. In addition, we set the constant values $\Delta t$ and $l$ to 0.1 and 3, respectively.

### 3.4 Kinematic Relation based Learning

The main goal of kinematic-relation based learning is that leveraging useful representation contained in multiple related task-specific networks to help improve the generalization performance of the control tasks. In order to utilize the effect of this relation to the proposed architecture, we define implicit relation loss functions.

The equation (5) - (7) can be expressed with respect to the speed $s_t$ and steering angle $\delta_t$.

$$s_t = \frac{x_{t+1}}{cos(\delta_t)} = \frac{y_{t+1}}{sin(\delta_t)} \tag{8}$$

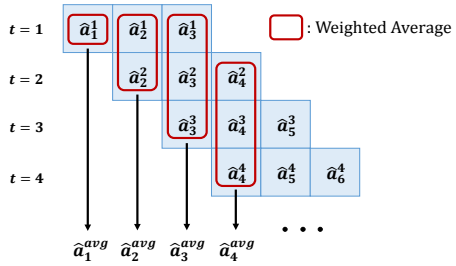


**Fig. 3.** Illustration of weighted average concept.

$$\delta_t = tan^{-1}(\frac{\psi_{t+1}}{s_t}) \tag{9}$$

We can reformulate the equations (5) and (6) to define the formula expressed with respect to the steering angle $\delta_t$, but we use the equation (7). Because if we use the equations (5) and (6), an arccosine and arcsine functions are included. Kinematic motion equations are utilized in training process with predicted outputs from the network. In this case, as the domain of the arccosine and arcsine functions are from -1 to +1 inclusive, the inputs of both functions should be restricted by clipping or scaling. This human intervention in the learning process can produce unexpected results.

The equation (8) and (9) indicate that the current vehicle control value and localization state of $t+1$ can be determined by each other. According to the relation, assuming that the steering angle $\delta_t$ and next position $x_{t+1}, y_{t+1}$ are well estimated, we can calculate the accurate speed $s_t$. In addition, the $\delta_t$ is calculated as well as the predicted $\psi_{t+1}$ and $s_t$ using equation (9). The output from the each task-specific head has a relation with other outputs. To add the relationship between the prediction heads directly, the explicit relational loss function is defined as follows:

$$L_r = \sum_{t}^{t+m-1} ||\frac{\hat{y}_{t+1}}{sin(\hat{\delta}_t)} - s_t||^2 + ||\frac{\hat{x}_{t+1}}{cos(\hat{\delta}_t)} - s_t||^2 + ||tan^{-1}(\frac{\hat{\psi}_{t+1}}{\hat{s}_t}) - \delta_t||^2, \tag{10}$$

where $\hat{\delta}_t$ can be replaced by $\hat{a}^{str}$. The pair of related regression heads can be jointly optimized with backpropagation to estimate accurate prediction.

### 3.5 Weighted Average Action

In order to make a stable action value, we utilize a weighted average method. Here, we redefine a predicted action as $\hat{\boldsymbol{a}}_q^p$, where $p$ is a time step index and $q$ is an index of the predicted sequence corresponding to the time step.

In Fig. 3, we show an example when $m$ is 3. In that case, for each time step $t$, the FASNet predicts $\hat{\boldsymbol{a}}_{t:t+2}^t$. The weighted average of actions ($\hat{\boldsymbol{a}}_t^{avg}$) is calculated using predicted actions with the same index $q$ and will be used as the control

signal at that time $t$. To make the most recently predicted value contributes more to the final result, we define weight as follows:

$$w_q^p = \frac{e(log_b(q-p+1))}{\sum_{i=1}^{min(q,m)} e(log_b(i)))}, \tag{11}$$

where $b$ is greater than 0 and less than 1. The contribution of the predicted actions can be adjusted using $b$.

## 4 Experiments

### 4.1 Implementation in Detail

Before training whole network, we first update the weights of $f_{cpre}$. During the training stage, CPredNets are trained using the same experimental setting as PredNet [14]. CPredNet$_{next}$ is pretrained using sequences of the previous $k$ frames, and then the weights are fine-tuned for extrapolation. To prevent a decrease in computation speed, the CPredNets comprise three-level generative modules with a channel size of (3, 48, and 96). In addition, the input images are resized to 100 $\times$ 44 by interpolation.

As mentioned previously, the representation layer $r_t$ passes through convolutional layers and then fused by $f_{fu}^{cmd}$. We use six convolutional layers, and all layers are followed by Batch Normalization (BN) and Rectified Linear Unit (ReLU) activation. The detailed setting of each layer for $r_t$ is shown in Table 1.

The number of hidden units of the $f_{pos}$, $f_{yaw}$, $f_{lon}$, $f_{lat}$, $f_{mea}$, $f_{spd}$, and $f_{fu}$ layers are same as the CILRS network [7]. Additionally, the sizes of the last FC layer of $f_{pos}$ and $f_{yaw}$ are $2 \times m$ and $m$, respectively. Finally, the size of $f_{fu}^{cmd}$ is 512. In our experiment, sequences of 9 frames are sampled for training, and our model predicts 3 consecutive actions ($k = 6$, $m = 3$). In addition, we set $b$ to 0.4 for the weighted average. Our model is trained using the Adam solver with mini-batches of 200 samples. The learning rate is set to 0.0001 at the beginning and then it decreases by a factor 0.1 at 50 % and 75 % of the total number of training epochs. The total multi-task loss function is defined as:

$$L_t = \lambda_a L_a + \lambda_l L_l + \lambda_c L_c + \lambda_r L_r + \lambda_s L_s, \tag{12}$$

where $L_c$ is the $f_{cpre}$ loss and $L_s$ is a loss for speed prediction branch. We set the $\lambda_a$, $\lambda_l$, $\lambda_c$, $\lambda_r$, and $\lambda_s$ to 0.5, 0.15, 0.15, 0.1 and 0.1, respectively.

### 4.2 Benchmark and Dataset

The CARLA simulator has a large variety of driving environments, such as traffic lights and dynamic obstacles, including dynamic vehicles and pedestrians. We employ the original CARLA benchmark [33], *NoCrash* benchmark [7], and *Any-Weather* benchmark[19] on CARLA simulator to evaluate the proposed method. For a fair comparison with the other methods, we follow the benchmark polices

**Table 1.** Network architecture applied to representation layers $r_{t+1:t+m-1}$.

| Layer Type | Kernel Size | Feature Maps | Stride |
|---|---|---|---|
| Conv-BN-ReLU | (3,3) | 32 | 2 |
| Conv-BN-ReLU | (3,3) | 32 | 1 |
| Conv-BN-ReLU | (3,3) | 64 | 2 |
| Conv-BN-ReLU | (3,3) | 64 | 1 |
| Conv-BN-ReLU | (3,3) | 128 | 2 |
| Conv-BN-ReLU | (3,3) | 128 | 1 |

[33, 7]. All benchmarks evaluate the driving performance under four environments "Training Conditions", "New Weather", "New Town", and "New Town & Weather" in terms of Success Rate (SR).

We collect the training data while using the autopilot [34] of the CARLA simulator for approximately 100 hours. For augmentation and teaching the model how to recover from a poor position [3], we use three RGB cameras by adjusting the roll values of the left and right cameras by -20 and 20 degrees from the middle camera, respectively. As most of the data are collected from straight driving scenes, we refer to the idea of sampling [10] to solve the data imbalance problem. Additionally, we perform extensive data augmentation by adding Gaussian blur, additive Gaussian noise, additive and multiplicative brightness variation, contrast variation, and saturation variation to prevent overfitting.

### 4.3 Experimental Result

We compare our FASNet model with the recent state-of-the-art approaches: the Conditional Imitation Learning (CIL) [6] extension with a ResNet backbone and speed prediction model (CILRS) [7], Learning By Cheating (LBC) [11] model, Implicit Affordances based Reinforcement Learning (IARL) [21] model, and Learning Situational Driving (LSD) [19] model. The closest baseline model to ours is the CILRS model. We set the same experimental settings, such as dataset conditions (including image resolution) and perception network. The LBC and IARL models rely on different prior information such as the 3D position of all external dynamic agents or semantic segmentation. Although the LSD uses an RGB image only, it utilizes a deeper backbone network (ResNet50) and a larger image resolution ($256 \times 256$).

Table 2 reports the quantitative comparison on the original CARLA 0.8.4 benchmark with state-of-the-art networks. This benchmark consists of four driving tasks "Straight", "One Turn", "Navigation", and "Navigation with Dynamic Obstacles" [33]. Compared to the baseline model, the proposed FASNet exhibits significant improvements especially under the "New Town" environment. In addition, we achieved the state-of-the-art driving performance among the models, which use only RGB images for training and evaluation.

The *NoCrash* benchmark [7] measures the capability of the controller to react to dynamic objects for three driving tasks: "Empty", "Regular Traffic", and

**Table 2.** Comparison with the state-of-the-art networks on the original CARLA 0.8.4 benchmark in terms of Success Rate (SR) in each condition. The results are percentage (%) of SR and higher values are better.

| | Training Conditions | | | | | New Weather | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Task | CILRS[7] | LSD[19] | FASNet | LBC[11] | IARL[21] | CILRS[7] | LSD[19] | FASNet | LBC[11] | IARL[21] |
| Straight | 96 | - | **100** | **100** | **100** | 96 | - | **100** | **100** | **100** |
| One Turn | 92 | - | **100** | **100** | **100** | 96 | - | **100** | 96 | **100** |
| Navigation | 95 | - | **100** | **100** | **100** | 96 | - | **99** | **100** | **100** |
| Nav. Dynamic | 92 | - | **100** | **100** | **100** | 96 | - | **98** | 96 | **100** |
| | New Town | | | | | New Town & Weather | | | | |
| Task | CILRS[7] | LSD[19] | FASNet | LBC[11] | IARL[21] | CILRS[7] | LSD[19] | FASNet | LBC[11] | IARL[21] |
| Straight | 96 | **100** | **100** | **100** | **100** | 96 | **100** | **100** | **100** | **100** |
| One Turn | 84 | 99 | **100** | **100** | **100** | 92 | **100** | **100** | **100** | **100** |
| Navigation | 69 | **99** | **99** | **100** | **100** | 92 | **100** | **100** | **100** | **100** |
| Nav. Dynamic | 66 | 98 | **99** | **99** | 98 | 90 | 98 | **100** | **100** | **100** |

**Table 3.** Comparison with the state-of-the-art networks on the *NoCrash* CARLA benchmarks in terms of success rate in each condition.

| | Training Conditions | | | | | New Weather | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Task | CILRS[7] | LSD[19] | FASNet | LBC[11] | IARL[21] | CILRS[7] | LSD[19] | FASNet | LBC[11] | IARL[21] |
| Empty | **97±2** | - | 96±0 | 97 | **100** | 96±1 | - | **98±0** | **87** | 36 |
| Regular | 83±0 | - | **90±1** | 93 | **96** | 77±1 | - | **80±1** | **87** | 34 |
| Dense | 42±2 | - | **44±2** | **71** | 70 | **47±5** | - | 38±4 | **63** | 26 |
| | New Town | | | | | New Town & Weather | | | | |
| Task | CILRS[7] | LSD[19] | FASNet | LBC[11] | IARL[21] | CILRS[7] | LSD[19] | FASNet | LBC[11] | IARL[21] |
| Empty | 66±2 | 94±1 | **95±1** | **100** | 99 | 90±2 | **95±1** | 92±2 | **70** | 24 |
| Regular | 49±5 | 68±2 | **77±2** | **94** | 87 | 56±2 | 65±4 | **66±4** | **62** | 34 |
| Dense | 23±1 | 30±4 | **37±2** | **51** | 42 | 24±8 | **32±3** | **32±4** | **39** | 18 |

"Dense Traffic". Quantitative comparisons on the *NoCrash* benchmark are reported in Table 3. The LBC and IARL, which utilize additional prior knowledge, are evaluated on the version of CARLA 0.9.6. Compared to the RGB-based models, we achieve state-of-the-art performances in the "New Town" conditions and "Regular" tasks. We established that training with various related tasks could make a more generalized model under unseen environments. Additionally, we observe that the accidents caused by unexpected single control value are alleviated through the weighted average action of future predictions. Moreover, our proposed model significantly improves over the baseline model except for the "Dense Traffic" tasks in the training town. Most of our failure, in the "Dense Traffic" task, is that pedestrians and other vehicles crash into the ego-vehicle. In some cases, the intersection is already blocked by an accident.

The *AnyWeather* benchmark is a new benchmark to quantify the ability of drastically diverse weather conditions. The evaluation condition is a new town under all ten weathers unseen in training. The results are presented in Table 4. It is observable that FASNet achieves state-of-the-art SRs under all tasks. This aspect means that the proposed architecture has higher generalization capability and robustness to unseen environments. We observe that most of our failure cases are under the "MidRainSunset" and "HardRainSunset". As the lane on the road is invisible, predicting stable steering control value is difficult.

**Table 4.** Experimental results the harsh environments on the *AnyWeather* benchmark in terms of success rate.

| | New Town & Weather | | |
|---|---|---|---|
| Task | CILRS | LSD | FASNet |
| Straight | 83.2 | 85.6 | **93.2** |
| One Turn | 78.4 | 81.6 | **87.0** |
| Navigation | 76.4 | 79.6 | **82.8** |
| Nav. Dynamic | 75.6 | 78.4 | **81.2** |

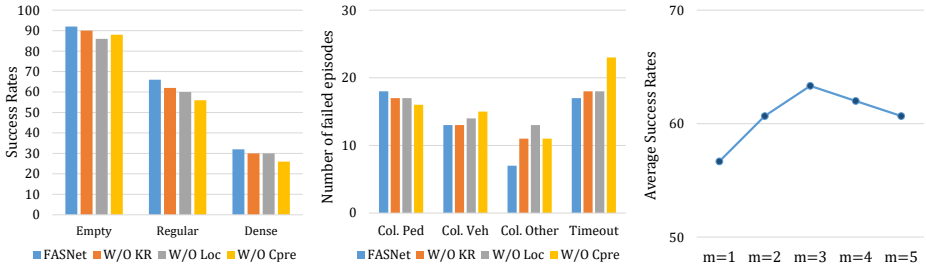


**Fig. 4.** Results of the ablation studies on the CARLA *NoCrash* benchmark under "New Town & Weather" conditions, which requires the highest generalization. (a) success rates, (b) number of failed episodes, and (c) average success rates of FASNet with different number of future actions.

During the inference stage, the computation time required by FASNet is approximately $0.04 \sim 0.06$ $ms$ on a Titan RTX.

### 4.4 Ablation Studies

To evaluate whether our approaches improve accuracy, we conducted another experiment without the explicit kinematic relational loss (W/O KR), the localization task-specific networks (W/O Loc), and the $f_{cpre}$ network (W/O Cpre) in Fig. 4. In the case of "W/O Cpre", future representations are not used, including future actions. In summary, the absence of any module causes the Success Rate (SR) to decrease, as shown in Fig. 4. (a). As seen in Fig. 4. (b), the number of collisions with others (Col. Other) shows a considerable increase even in the absence of a single component. This indicates that the vehicle has come off the road owing to unexpected behavior with a lack of generalization ability. Further, the high failure rates of "Col. Other" confirm the usefulness of the localization tasks for safe driving. The major cause of failure in the case of "W/O Cpre" is "Timeout". This happens because the vehicle stops and never moves caused by an unexpected longitudinal control value. In addition, few input images prevent a car from operating normally, and this has led to a suspension from driving on the simulator. The weighted average action of future predictions can overcome this problem by degrading an effect of the incorrect prediction. The average SR

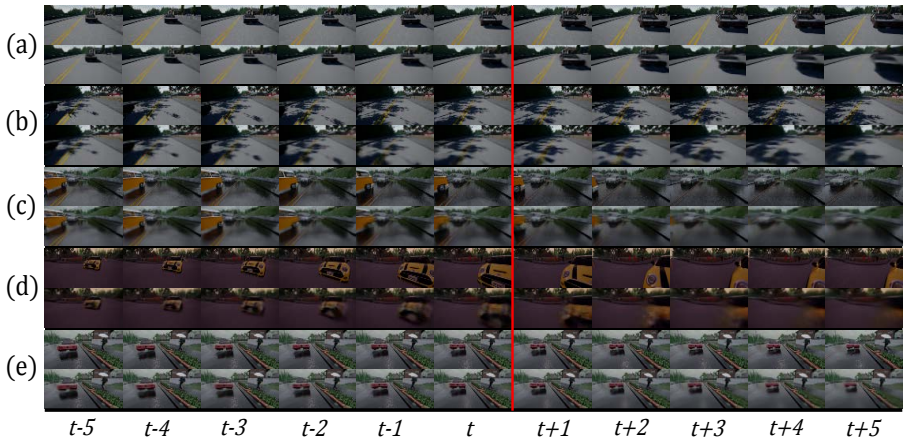


**Fig. 5.** The results of $f_{cpre}$ (for next and extrapolation frames) in dynamic objects with various weather: (a) and (b) clear noon, (c) wet cloudy noon, (d) clear sunset, and (e) rainy noon. Every odd rows contain actual frames, and the even rows contain predictions. Left of the red line: Generated by CPredNet$_{next}$; Right: Generated by CPredNet$_{extra}$.

values of FASNet for different number of future states to use ($m$) are shown in Fig. 4. (c). Parameter $m$ indicates the number of future states employed. In the case of $m = 1$, the model equals to the "W/O Cpre" case. Our observation can be proven, because when $m > 1$, every average SR value is greater than that when $m = 1$.

### 4.5 Future State Generation

We make future representations using the CPredNets and localization information using a kinematic vehicle model without additional annotations. To verify that these methods can generate the knowledge successfully, we show the results for both approaches.

Fig. 5 shows the qualitative predictions on the test scenarios on the CARLA simulator. The CPredNets can generate fairly accurate frames under the various environments. The results of various weather and scenarios are represented in Fig. 5: (a) sunny/straight, (b) sunny/go straight with dynamic object, (c) wet cloudy/turn left, and (d) sunset/straight with dynamic object. Sometimes the CPredNets generate blurry objects and backgrounds (i.e., shadow of tree in (c) and yellow vehicle in (d)). However, as can be seen in Fig. 5, primarily activated factors for a making decision [3], such as road, lane, and curb are perfectly generated. This indicates that the generated future frame will not negatively affect the prediction of the vehicle's action. Additionally, we report the comparison results between the CPredNet and PredNet in supplementary material.

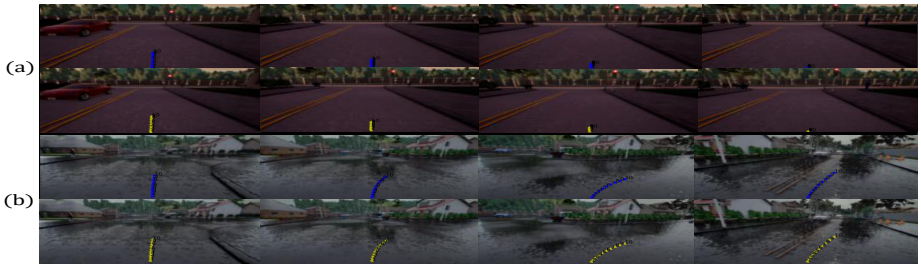


**Fig. 6.** Each image shows the coordinate changes from current position for 10 time steps: (a) stop scenario, (b) right turn scenario. Every odd row contains the ground truth coordinates and the even row contains the calculated coordinates with vehicle motion equations.

Fig. 6 shows the coordinates change from current to future positions qualitatively. Note that we calculate the positions of the vehicle in global coordinates with speed and the steering angle. As observable in Fig. 6, future locations are successfully calculated. Furthermore, the quantitative errors of calculated coordinates and vehicle heading are under 0.04 in terms of the mean square error.

## 5 Conclusion

In this study, we investigated a stable end-to-end vision-based autonomous driving model by weighted averaging of predicted future actions. We attempted to prevent a situation wherein a single incorrect control action renders the vehicle's movement unstable. To achieve enhanced generalization ability, we designed multi-head networks that are supervised by task-specific objectives including auxiliary localization tasks. During the training, the related tasks are jointly optimized with the shared layers, which serve as the regularizer. Thus, we generated training knowledge without any additional annotations. We have empirically shown that such a strategy can improve the generalization and driving performance of the base model through various experiments.

## Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2014-0-00059, Development of Predictive Visual Intelligence Technology), (No.2017-0-00897, Development of Object Detection and Recognition for Intelligent Vehicles) and (No.2018-0-01290, Development of an Open Dataset and Cognitive Processing Technology for the Recognition of Features Derived From Unstructured Human Motions Used in Self-driving Cars)

## References


1. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32** (2013) 1231–1237
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 3213–3223
3. 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 preprint arXiv:1604.07316 (2016)
4. 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). (2018) 435–453
5. Huang, Z., Zhang, J., Tian, R., Zhang, Y.: End-to-end autonomous driving decision based on deep reinforcement learning. In: 2019 5th International Conference on Control, Automation and Robotics (ICCAR), IEEE (2019) 658–662
6. Codevilla, F., Miiller, M., López, A., Koltun, V., Dosovitskiy, A.: End-to-end driving via conditional imitation learning. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE (2018) 1–9
7. Codevilla, F., Santana, E., López, A.M., Gaidon, A.: Exploring the limitations of behavior cloning for autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 9329–9338
8. Liang, X., Wang, T., Yang, L., Xing, E.: Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 584–599
9. Sauer, A., Savinov, N., Geiger, A.: Conditional affordance learning for driving in urban environments. arXiv preprint arXiv:1806.06498 (2018)
10. Wang, Q., Chen, L., Tian, B., Tian, W., Li, L., Cao, D.: End-to-end autonomous driving: An angle branched network approach. IEEE Transactions on Vehicular Technology (2019)
11. Chen, D., Zhou, B., Koltun, V., Krähenbühl, P.: Learning by cheating. arXiv preprint arXiv:1912.12294 (2019)
12. Li, Z., Motoyoshi, T., Sasaki, K., Ogata, T., Sugano, S.: Rethinking self-driving: Multi-task knowledge for better generalization and accident explanation ability. arXiv preprint arXiv:1809.11100 (2018)
13. Chowdhuri, S., Pankaj, T., Zipser, K.: Multinet: Multi-modal multi-task learning for autonomous driving. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE (2019) 1496–1504
14. Lotter, W., Kreiman, G., Cox, D.: Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint arXiv:1605.08104 (2016)
15. Kong, J., Pfeiffer, M., Schildbach, G., Borrelli, F.: Kinematic and dynamic vehicle models for autonomous driving control design. In: 2015 IEEE Intelligent Vehicles Symposium (IV), IEEE (2015) 1094–1099
16. Zhang, Y., Yang, Q.: A survey on multi-task learning. arXiv preprint arXiv:1707.08114 (2017)
17. 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. (2017) 2174–2182

18. Chi, L., Mu, Y.: Deep steering: Learning end-to-end driving model from spatial and temporal visual cues. arXiv preprint arXiv:1708.03798 (2017)
19. Ohn-Bar, E., Prakash, A., Behl, A., Chitta, K., Geiger, A.: Learning situational driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 11296–11305
20. Yu, A., Palefsky-Smith, R., Bedi, R.: Deep reinforcement learning for simulated autonomous vehicle control. Course Project Reports: Winter (2016) 1–7
21. Toromanoff, M., Wirbel, E., Moutarde, F.: End-to-end model-free reinforcement learning for urban driving using implicit affordances. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 7153–7162
22. Tai, L., Yun, P., Chen, Y., Liu, C., Ye, H., Liu, M.: Visual-based autonomous driving deployment from a stochastic and uncertainty-aware perspective. arXiv preprint arXiv:1903.00821 (2019)
23. 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: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE (2018) 2289–2294
24. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440 (2015)
25. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: International conference on machine learning. (2015) 843–852
26. Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual motion gan for future-flow embedded video prediction. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 1744–1752
27. Wei, H., Yin, X., Lin, P.: Novel video prediction for large-scale scene using optical flow. arXiv preprint arXiv:1805.12243 (2018)
28. Ranjan, R., Patel, V.M., Chellappa, R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **41** (2017) 121–135
29. Du, L., Zhao, Z., Su, F., Wang, L., An, C.: Jointly predicting future sequence and steering angles for dynamic driving scenes. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE (2019) 4070–4074
30. Jin, X., Xiao, H., Shen, X., Yang, J., Lin, Z., Chen, Y., Jie, Z., Feng, J., Yan, S.: Predicting scene parsing and motion dynamics in the future. In: Advances in Neural Information Processing Systems. (2017) 6915–6924
31. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
32. Song, G., Chai, W.: Collaborative learning for deep neural networks. In: Advances in Neural Information Processing Systems. (2018) 1832–1841
33. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. arXiv preprint arXiv:1711.03938 (2017)
34. felipecode: Carla 0.8.4 data collector. https://github.com/carla-simulator/data-collector (2018)