Self-Supervised Interpretable Sensorimotor Learning via Latent Functional Modularity

Hyunki Seong[1], **David Hyunchul Shim**[1]

[1]School of Electrical Engineering
Korea Advanced Institute of Science and Technology, South Korea
hynkis, hcshim@kaist.ac.kr

Abstract

We introduce MoNet, a novel method that combines end-to-end learning with modular network architectures for self-supervised and interpretable sensorimotor learning. MoNet is composed of three functionally distinct neural modules: Perception, Planning, and Control. Leveraging its inherent modularity through a cognition-guided contrastive loss function, MoNet efficiently learns task-specific decision-making processes in latent space, without requiring task-level supervision. Moreover, our method incorporates an online post-hoc explainability approach, which enhances the interpretability of the end-to-end inferences without a trade-off in sensorimotor performance. In real-world indoor environments, MoNet demonstrates effective visual autonomous navigation, surpassing baseline models by 11% to 47% in task specificity analysis. We further delve into the interpretability of our network through the post-hoc analysis of perceptual saliency maps and latent decision vectors. This offers insights into the incorporation of explainable artificial intelligence within the realm of robotic learning, encompassing both perceptual and behavioral perspectives.

Introduction

One of the main objectives in end-to-end learning for autonomous navigation is to develop complex policies through human demonstrations. This is achieved by an end-to-end network that learns the hierarchical pipeline of perception, planning, and control in robotic systems via imitation learning (IL). Given that IL facilitates safe and efficient policy learning in an offline, supervised manner, end-to-end networks have been widely used in the design of learning-based applications (Tampuu et al. 2020).

However, although studies on IL have shown preliminary successes, designing an end-to-end sensorimotor network that can scale up to complex driving scenarios remains challenging. Traditional end-to-end networks often exhibit a less clear decision-making process, which complicates learning entangled tasks from demonstrations. To address this, recent conditional learning methods (Huang et al. 2020) employ multiple branching networks for each task, with outputs that switch based on task-level conditional inputs. However, this conditional input often corresponds to the outcome of an internal decision-making process in human demonstrations, which is typically implicit and difficult to identify. As a consequence, this approach necessitates extra task-level annotations (e.g., *go-straight*, *turn-left*), making it more demanding than simply collecting sensorimotor pairs.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

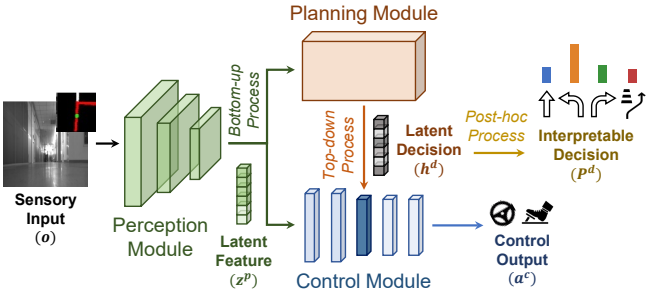

Figure 1: Our approach incorporates a functionally modular end-to-end network architecture, which includes a post-hoc process for an interpretable latent decision-making process.

Moreover, conventional networks, which directly compute control commands from sensory inputs, lack a transparent inference process. This obscurity makes it unclear what behavioral decision was intended for the resulting control output without direct execution. Several studies have sought to enhance interpretability by reconstructing various modalities (Chen, Li, and Tomizuka 2021; Zeng et al. 2019), or by visualizing attention maps (Kim and Canny 2017). However, they mainly focus on perceptual insights, still leaving the high-level decisions behind sensorimotor outputs largely obscure. This lack of clarity leads to insufficient task specificity and interpretability during the sensorimotor process, ultimately diminishing the reliability and trustworthiness of end-to-end networks in practical applications.

To tackle those limitations, we adopt the insight of functional modularity in the neuroscience field for the design of an end-to-end architecture. Neuroscientists explain the organization of the human brain with two fundamental principles: functional segregation and integration (Meunier, Lambiotte, and Bullmore 2010). The brain has separate submodules that are structurally connected but functionally segregated according to their specific functions, such as sensory, cognitive, or motor processes, to perform complex behaviors. This suggests that employing a functionally modularized end-to-end architecture would be more suitable for ap-

proximating an autonomous driving stack comprising multiple sub-functions.

In this paper, we present a modular end-to-end network, MoNet, inspired by the functional modularity of interconnected brain networks. MoNet is divided into perception, planning, and control modules, which are functionally separated but explicitly connected to form a single end-to-end structure (Fig. 1). Our network contains an internal latent decision process to facilitate task-oriented guidance for behaviorally relevant sensory-motor processes. Simultaneously, it employs a self-attention mechanism to extract salient spatial features from sensory input.

Leveraging the inter-modular relationships in MoNet, we design a novel cognition-guided contrastive (CGC) loss function. Directed by the perception module with task-oriented contexts, this function encourages the planning module to make consistent decisions in similar driving contexts while differentiating responses in varied situations. Furthermore, we integrate a post-hoc multi-class classification method to decode task-relevant latent decisions into understandable representations. The internal hierarchy, combined with the contrastive scheme and augmented by the post-hoc explainability method, not only promotes functional specialization but also enables the emergence of a task-relevant and interpretable decision mechanism through self-supervision.

In our evaluation, our method effectively demonstrates visual autonomous driving across multiple tasks, including corridor navigation, intersection navigation, and collision avoidance. We present empirical experiments conducted on a real-world robotic RC platform, showcasing the network's capability to perform task-specific sensorimotor inference without requiring task-level labeling. We further explore spatial saliency maps and latent decisions during end-to-end inference. Specifically, by decoding latent decisions into explainable posterior probabilities, we gain the ability to visualize sequential high-level internal decisions alongside task uncertainty during continuous end-to-end sensorimotor control. These analyses highlight the significant interpretability and transparency of our end-to-end model, showcasing its effectiveness from both perceptual and behavioral perspectives in real-world continuous control applications.

Our main contributions can be summarized as follows:

- We propose MoNet, a modular end-to-end network that incorporates a post-hoc explainable method, enabling interpretable sensorimotor control.
- We design a novel cognition-guided contrastive loss function to enhance the task-relevant and interpretable decision mechanism within the end-to-end network.
- We examine the perceptual and behavioral interpretability, as well as the sensorimotor performance of our network, showcasing the potential benefits of integrating the explainability method into robotic learning.

Related Works

End-to-End Sensorimotor Learning: In autonomous driving, end-to-end methods employ single neural networks to directly map sensory inputs to control outputs. ALVINN, the initial model for steering angle inference, utilized a multilayer perceptron (Pomerleau 1988). This approach has evolved to include convolutional neural networks (CNNs), mainly focused on lane-following tasks (Bojarski et al. 2016). Recent advancements have incorporated conditional imitation learning to cover a broader range of driving tasks (Gao et al. 2017; Codevilla et al. 2018; Huang et al. 2020; Zhang, Huang, and Ohn-Bar 2023). These methods use multiple branched layers switched by conditional inputs for navigating environments, such as *'go-straight'*, *'turn-left'*, or *'turn-right'*. While such methods reduce task-level ambiguity, they necessitate additional human-engineered labeling for the navigational inputs and are constrained to predefined tasks. Moreover, interpreting the perceptual and behavioral processes within end-to-end networks remains a challenge, which affects confidence in the network's reliability for real-world deployment.

Interpretable Methods: Recent studies have concentrated on designing interpretable end-to-end networks to address existing limitations. In this context, researchers using segmentation methods (Chen, Li, and Tomizuka 2021; Teng et al. 2022) have indirectly shown how a network can comprehend surrounding contexts by generating semantic masks from hidden features. Similarly, studies involving multi-head networks (Zeng et al. 2019) have evaluated the effectiveness of their planning methods by examining interpretable representations across various modalities, such as object detection or cost map generation. In contrast, attention mechanisms (Vaswani et al. 2017; Kim and Canny 2017) in recent studies have explicitly facilitated a deeper understanding of the areas within given feature elements where the network predominantly focuses during feedforward processing. Specifically, in the realm of autonomous driving, methods leveraging attention aim to accentuate critical aspects in driving scenarios, such as lane following (Shi et al. 2020), lane changing (Chen et al. 2019), or navigating intersections (Seong et al. 2021). However, the majority of research has primarily focused on the cognitive interpretations of how networks perceive contexts. Our work takes this a step further by investigating how to interpret the task-oriented intentions of the network in an explainable way. This approach enables both perceptual and behavioral interpretations online during end-to-end inference.

Modular End-to-End Network

Latent Functional Modularity

Our main idea is to embed functional modularity with internal hierarchy into an end-to-end network, allowing functionalities of the robotic sub-modules in latent space. As shown in Fig. 1, our modular end-to-end network, MoNet, has three distinct neural modules: *Perception* ($\mathcal{P}$), *Planning* ($\mathcal{Q}$), and *Control* ($\mathcal{R}$), which are the major components of the robotics system (Schwarting, Alonso-Mora, and Rus 2018). Each module 1) encodes raw observations o into a fused perception feature vector z^p, 2) infers a latent decision h^d, and 3) computes a sensorimotor command a^c, respectively. The modules are functionally separated yet structurally connected in latent space, enabling them to constitute

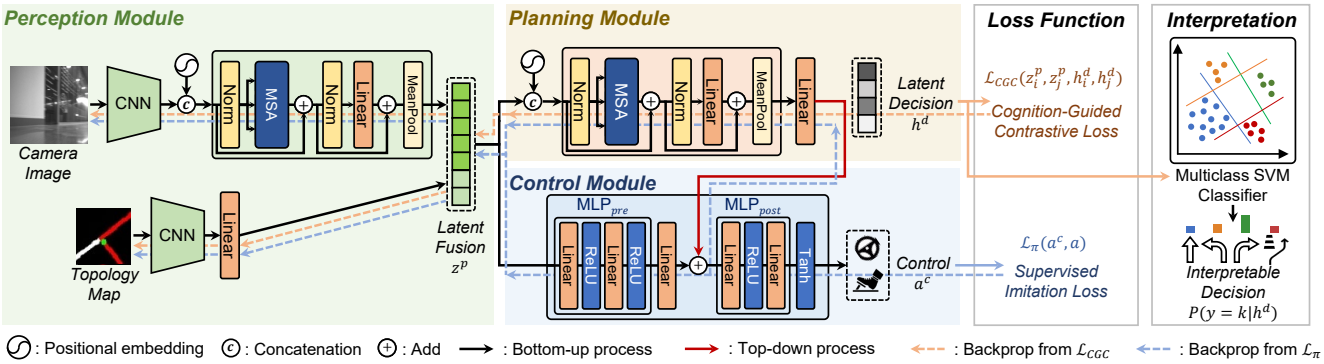

Figure 2: Overview of our method. While the entire end-to-end network is optimized by the supervised imitation loss $\mathcal{L}_\pi$, the planning module is updated by the cognition-guided contrastive loss $\mathcal{L}_{CGC}$, which is directed by the latent fusion vector z^p.

an end-to-end policy network π, parameterized by θ:

$$\begin{aligned} &\textit{Perception:} && z^p = \mathcal{P}(o;\theta) \\ &\textit{Planning:} && h^d = \mathcal{Q}(z^p;\theta) \\ &\textit{Control:} && a^c = \mathcal{R}(z^p, h^d;\theta) \end{aligned} \tag{1}$$

To encourage functional specialization of the modules in the network, we utilize two distinct mechanisms: bottom-up and top-down neural processes (Baluch and Itti 2011; Anderson et al. 2018). Specifically, the bottom-up mechanism is a stimulus-driven, exogenous process, while the top-down mechanism is a behavior-relevant endogenous process (Katsuki and Constantinidis 2014). Considering their properties, the perception ($\mathcal{P}_\theta$) and planning ($\mathcal{Q}_\theta$) modules configure with self-attention mechanisms (Vaswani et al. 2017) to extract salient spatial features from sensory input o and to obtain contextual importance from the features z^p, respectively. In contrast, the control ($\mathcal{R}_\theta$) module is designed with a top-down mechanism to internally modulate the sensory-motor signals based on the context-oriented behavioral decision h^d from the planning module. This internal hierarchy enables the network to generate spatial attention maps and high-level latent decisions that are explicitly accessible during end-to-end inference. Employing a post-hoc approach allows these to be transformed into interpretable salient maps and behavioral decisions, respectively.

Network Details

Perception module: Our network receives a high-dimensional observation $o = [I, M]$ that includes a front camera image $I \in \mathbb{R}^{224\times224\times1}$ and a topology map $M \in \mathbb{R}^{64\times64\times3}$. This observation includes visual sensory data with navigational information, providing driving contexts in the ego-centric area for navigating complex environments, such as corridors with intersections. To effectively process high-dimensional camera images, we employ a hybrid architecture that combines the Vision Transformer encoder with CNN blocks (Dosovitskiy et al. 2020). In the perception module, the image I and the topology map M are first encoded into a hidden image feature $z_I \in \mathbb{R}^{6\times6\times64}$ and a hidden route feature $z_M \in \mathbb{R}^{1\times1\times64}$, respectively, using ResNet-inspired CNN blocks (He et al. 2016). The image feature is first reshaped into a flattened embedding $z_I^{\text{flat}} \in \mathbb{R}^{(6\times6)\times64}$, serving as a tokenized embedding for $N = 6\times6$ image patches. This reshaped embedding is then concatenated with a positional embedding and fed into a Transformer encoder network (see the appendix for details). Subsequently, the Transformer model processes this input to produce an attention matrix $A(\cdot,\cdot)$, which integrates with the global context of the image feature z_I via the self-attention mechanism. The attention matrix is computed as follows:

$$A(Q, K) = softmax(\frac{QK^T}{\sqrt{d_k}}) \tag{2}$$

$$Q, K, V = ZW_Q, ZW_K, ZW_V \tag{3}$$

where $Q, K, V \in \mathbb{R}^{N\times D_k}$ refer to queries, keys, and values consisting of N data nodes with D_k dimension size by following common terminology (Dosovitskiy et al. 2020). $W_Q, W_K, W_V \in \mathbb{R}^{D_k\times D}$ are weight matrices for an arbitrary input feature $Z \in \mathbb{R}^{N\times D}$ with D dimension size to compute Q, K, V. The attention matrix $A \in \mathbb{R}^{N\times N}$ discerns the spatial significance of feature elements in the input image, offering valuable information to interpret the module's bottom-up neural processing from a perceptual standpoint. Finally, by applying mean pooling, we derive the attention-integrated feature z_I^{att} with reduced dimensionality. This feature vector is then concatenated with the flattened route feature $z_M^{\text{flat}} \in \mathbb{R}^{64}$, yielding a latent feature fusion vector $z^p = [z_I^{\text{att}}, z_M^{\text{flat}}]$. The fused feature is subsequently fed into the planning and control modules without nonlinearity.

Planning module: This module extracts contextual features from the fused vector z^p and produces a latent decision h^d to modulate the neural signals of the control module in a top-down manner. We construct the planning module using another Transformer network, mirroring the encoder of the perception module. Here, the fusion vector is expanded and tokenized into an input embedding, corresponding to the embedding z_I^{flat} in perception. This input embedding is then concatenated with a positional embedding, following a similar process to that in the perception module. To derive the latent decision in continuous space, we apply a linear layer to the output of the Transformer encoder without using a nonlinear activation function.

Control module: The control module computes a low-level control command incorporating the high-level deci-

sion through bottom-up and top-down processes (Fig. 2). The module initially extracts a pre-sensory-motor feature $x_{pre} \in \mathbb{R}^{N_d}$ from a given perceptual feature z^p using the MLP_{pre} block (Eq. 4). The motor feature is then passed through a linear fully-connected layer (FC) and modulated in a top-down fashion via elementwise addition with the task-oriented latent decision $h^d \in \mathbb{R}^{N_d}$ (Eq. 5). The module finally converts the modulated feature into the control command $a^c \in \mathbb{R}^2$ through the MLP_{post} block followed by a tanh activation function (Eq. 6).

$$x_{pre} = \text{MLP}_{pre}(z^p) \tag{4}$$

$$x_{mod} = \text{FC}(x_{pre}) + h^d \tag{5}$$

$$a^c = \tanh(\text{MLP}_{post}(x_{mod})) \tag{6}$$

Here, the command $a^c = [\delta^c, \tau^c]$ contains a normalized steering angle δ^c and throttle value τ^c. This self-modulated hierarchy facilitates the independent computation of sensorimotor and contextual data from perceptual inputs, resulting in a control signal guided by latent decision-making. As a result, MoNet is capable of learning task-specific sensorimotor policies even from task-agnostic demonstrations.

Training Details

To train the network, we first introduce the supervised loss function $\mathcal{L}_\pi$, defined as the absolute deviation (L1) between the network's prediction and the demonstration data:

$$\mathcal{L}_\pi(a^c, a) = |\delta^c - \delta| + \lambda_\tau |\tau^c - \tau| \tag{7}$$

where the loss term for throttle control τ^c is weighted by the parameter $\lambda_\tau \in [0, 1]$. This weighting aims to emphasize supervision on steering control in visual autonomous navigation (Codevilla et al. 2018). Given that we collect noisy demonstrations from a real robot platform, we choose the L1 loss to reduce the penalty for large errors and be more robust to outliers compared to the L2 loss.

Furthermore, to enhance the distinctiveness of top-down latent decisions, we design a cognition-guided contrastive (CGC) loss function using a self-supervised approach, leveraging the modular characteristics of our end-to-end network. Generally, the output of planning is influenced by the context of the driving scene. This implies that similar situations lead to analogous decisions, while different scenarios result in distinct plans. Building on the observation that planning outputs are context-dependent, we define a cognition-guided contrastive loss, denoted as $\mathcal{L}_{\text{CGC}}(z_i^p, z_j^p, h_i^d, h_j^d)$. Here, the latent decision h^d is guided by the output feature fusion z^p of the perception module as follows:

$$\mathcal{L}_{\text{CGC}} = \begin{cases} 1 - \cos(h_i^d, h_j^d) & \text{if } \cos(z_i^p, z_j^p) >= \kappa \\ \max(0, \cos(h_i^d, h_j^d)) & \text{if } \cos(z_i^p, z_j^p) < \kappa \end{cases} \tag{8}$$

where $\cos(\alpha, \beta) = \frac{\alpha \cdot \beta}{|\alpha||\beta|}$ is cosine similarity that is widely used for similarity and clustering analysis in data science (Larose and Larose 2014). It calculates the distance between two vectors based on their relative orientations, rather than their absolute distance, within the bounded range [-1, +1]. The subscript j represents the index of a sample within a mini-batch, different from the current sample index i. By minimizing Eq. 8, we aim to reduce the intra-cluster distance for latent decisions in demonstrations where perceptual feature similarity exceeds κ. Conversely, we strive to increase the inter-cluster distance when the similarity is below κ. This approach incentivizes the planning module to generate more consistent latent decisions for scenarios with comparable perceptual contexts, while ensuring diverse decisions for scenarios with differing contexts. Moreover, by leveraging perceptual features, our method eliminates the need to define positive or negative samples, thereby enabling contrastive learning through a self-supervised approach.

The overall per-sample loss function is given by the weighted summation:

$$\mathcal{L} = \mathcal{L}_\pi(a_i^c, a_i) + \lambda_{\text{CGC}} \mathcal{L}_{\text{CGC}}(z_i^p, z_j^p, h_i^d, h_j^d) \tag{9}$$

where λ_{CGC} is a weight parameter. During the training phase, the supervised loss function $\mathcal{L}_\pi$ propagates the gradient flow across all modules $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$, while the latent-guided loss function $\mathcal{L}_{\text{CGC}}$ targets only the planning and perception modules $(\mathcal{P}, \mathcal{Q})$, promoting functional distinction between the planning and control modules $(\mathcal{Q}, \mathcal{R})$.

Interpretation Details

Cognitive Interpretation: We use the attention matrix of the perception module to create a saliency map S, which highlights the spatial regions in the current driving scene that the network focuses on from a cognitive perspective. The module generates the attention matrix $A \in \mathbb{R}^{N \times N}$ corresponding to the flattened vector of the encoded feature map $z_I \in \mathbb{R}^{h \times w \times c}$, where $N = h \times w$ is a resulting size of attention, (h, w) is a reduced resolution of the image $I \in \mathbb{R}^{224 \times 224 \times 1}$, and c is the feature dimension of z_I. Thus, we initially aggregate weights along the first dimension of A to obtain the averaged attention weights $\bar{A} \in \mathbb{R}^{1 \times N}$:

$$\bar{A}_j = \frac{1}{N} \sum\nolimits_{i=1}^{N} A_{ij} \quad \text{for} \quad j = 1, ...N \tag{10}$$

where $\bar{A}_j$ represents the central tendency of the weights in each column. Subsequently, we reshape the averaged weights into a two-dimensional matrix $\bar{S} \in \mathbb{R}^{h \times w}$. This is then upscaled to form the saliency map $S \in \mathbb{R}^{224 \times 224}$.

Behavioral Interpretation: Considering that the latent decision contains task-oriented features, we decode the decision vector h^d into an understandable, task-wise probability score vector to facilitate the behavioral interpretation of our network. We employ a multiclass linear Support Vector Machine (SVM) classifier (Suthaharan 2016) that is computationally efficient and less prone to overfitting. Utilizing sample decisions and their corresponding task labels (h_i^d, y_i), linear SVM is designed to learn binary classification through the following optimization (Tang 2013):

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{M} max(0, 1 - y_i(w^T h_i^d + b))^2 \tag{11}$$

where w is the weight vector, b is the bias term, and C is the regularization parameter. The linear SVM is extended to

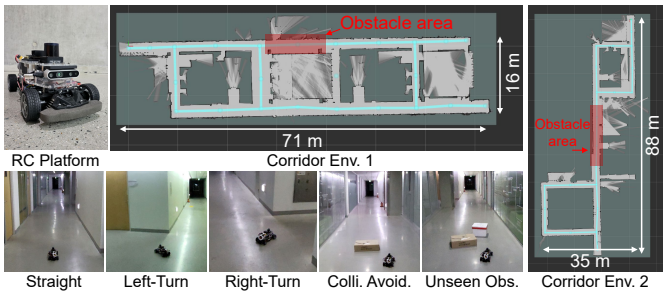

Figure 3: Hardware and experimental setup.

multiclass classification using the one-vs-rest scheme (Tang 2013). This adaptation enables SVMs to maximize the margin between input data belonging to different classes. After training the multiclass linear SVMs, we transform their output into a posterior probability score vector $P(y_i = k|h_i^d)$ for each class k, using a calibration method (Niculescu-Mizil and Caruana 2005):

$$P(y_i = k|h_i^d) = \frac{1}{1 + exp(E_k f_k(h_i^d) + F_k)} \tag{12}$$

where $f_k = w_k^T h_i^d + b_k$ is the SVM's output for class k, and E_k, F_k are parameters fitted using maximum likelihood estimation from sample data set $[f_k(h_i^d), y_i]$.

We carry out behavior interpretation in a post-hoc manner. Initially, we generate sample latent decisions for each task that human engineers aim to interpret, using the trained modular network. Subsequently, we train the multiclass SVMs, along with parameters E_k and F_k for the calibration method. Finally, during sensorimotor inferencing, we transform MoNet's latent decisions into probability score vectors using Eq. 12. This approach allows us to interpret the end-to-end model without sacrificing sensorimotor performance.

Our approach is comparable to concept-based interpretation methods in explainable artificial intelligence (Ghorbani et al. 2019). These studies focus on understanding how high-level concepts are represented and utilized by models in decision-making. In our case, the concept vector corresponds to the latent decision that encapsulates the *driving situation*. Consequently, to interpret the decision intent during the sensorimotor process, we quantify the alignment of a given decision vector with the specific tasks' concept (driving situation). This is performed by decoding the latent decision into the understandable probability score vector.

Experiments

Experimental Setup

Fig. 3 shows the overall setup for hardware, environment, and scenarios in this work. We apply MoNet on a wheeled, car-like platform, modeled after the F1TENTH vehicle (O'Kelly et al. 2020). Our platform consists of a 1/10 scale racing car chassis (TT-02) equipped with an embedded computer (Jetson Xavier NX) and a controller (Arduino). The Xavier NX receives front camera images and range measurements from sensors mounted on the platform. The range measurements are utilized to estimate the current pose of the ego vehicle and compute ego-centric coarse topology map (Amini et al. 2019). Detailed hardware setup and the coarse map processing are provided in the appendix.

Method	Success Rate (Count/Total)				
	ST	SI	LT	RT	CA
MoNet	1.00 (76/76)	1.00 (32/32)	1.00 (8/8)	**1.00 (8/8)**	**0.95 (18/19)**
ViTNet	1.00 (76/76)	1.00 (32/32)	1.00 (8/8)	0.63 (5/8)	0.89 (17/19)

Table 1: Success rate results for each driving task.

Our platform performs visual autonomous navigation with multiple driving tasks such as *straight* (ST), *left-turn* (LT), *right-turn* (RT), and *collision avoidance* (CA). Data collection is carried out under controlled conditions in two indoor environments: Corridor Environment 1 (Env. 1, $71m \times 16m$) and Environment 2 (Env. 2, $88m \times 35m$). Box-shaped obstacles are randomly positioned within specific areas in these environments. The training dataset comprises data from scenarios that feature either a single obstacle or no obstacles. However, scenarios involving multiple obstacles are introduced as new, unseen challenges during the evaluation phases. Our method is evaluated in Env. 1, characterized by more frequent intersection situations during autonomous navigation. For further details on data collection and processing (see the appendix for details).

Quantitative Evaluation

Sensorimotor Performance We evaluate the sensorimotor performance by measuring the success rate for each driving task in the evaluation environment. In addition to our method, we have implemented ViTNet, a baseline model designed as a conventional Vision Transformer-based end-to-end architecture without the planning module ($\mathcal{Q}$) from the MoNet's architecture. Under the same hardware and environmental conditions, each model performed 16 episodes in the real-world environment, totaling 143 driving tasks. Table 1 summarizes the performance results. These results demonstrate that our model exhibits stronger generalization ability across multiple sensorimotor tasks compared to the baseline model. Both models show safe navigation performance in straight driving scenarios. However, ViTNet often struggles to overcome unseen obstacle scenarios and particularly fails in turning right at intersections, where it records its lowest success rate of 63%. Although there was a situation where our model had a mild touch with a wall while avoiding cluttered obstacles, MoNet succeeded in all trials of navigating intersections and generally performed well in obstacle avoidance scenarios.

Planning Performance To assess the planning-level performance of the end-to-end network, we quantify the task specificity during sensorimotor inference using a t-SNE map (Van der Maaten and Hinton 2008) and a Representational Similarity Matrix (RSM) (Popal, Wang, and Olson 2019). In consideration of the data distribution, we sampled 318, 64, 59, 59, and 67 pieces of data, respectively, for the five tasks (ST, SI, LT, RT, CA) for these assessments.

The t-SNE visualization (Fig. 4 (A)) demonstrates that our planning module generates distinct and well-structured de-

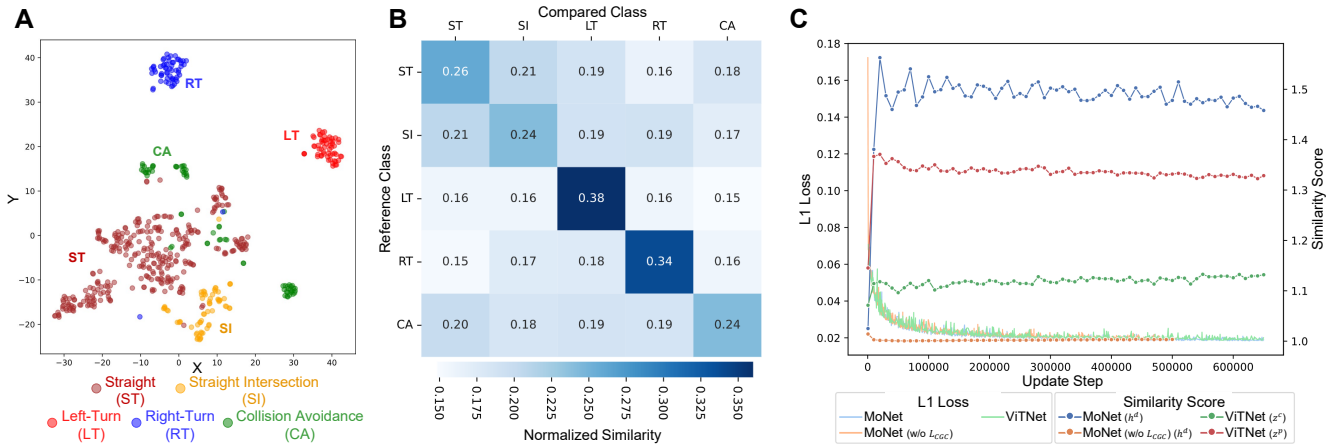

Figure 4: Results of the planning-level quantitative evaluations. **(A):** The t-SNE map illustrates clusters of latent decisions among different tasks. **(B):** The similarity matrix shows the quantitative similarity of the decision vectors between different classes. **(C):** The learning curve indicates performance improvement in L1 loss and similarity score across different models.

cisions in the latent space for various tasks. It effectively differentiates between the LT and RT tasks from the ST scenario and recognizes the directional variations in navigating intersections. The decisions for CA are positioned between the ST, LT, and RT clusters, indicating a need for moderate planning that involves both intersection-turning and corridor-following behaviors in collision avoidance scenarios. Furthermore, the data for SI exhibits a high similarity to that of ST, reflecting similar driving contexts, despite their differing inputs from topological maps.

For a more quantitative analysis of similarity in latent planning, we further examine the RSM of the learned decisions (Fig. 4 (B)). We measure the cosine similarity between the latent decisions of each task using the average linkage method (Nielsen and Nielsen 2016). The similarities among the five task classes are then normalized to a range of [0, 1] through a row-wise softmax operation. The results demonstrate that our method effectively differentiates between various tasks while clustering similar driving situations. The matrix shows strong diagonal values, indicating that the latent decisions effectively distinguish various driving tasks based on contextual features derived from sensory data, without task-level inputs. Specifically, the decisions related to navigating intersections (LT, RT) and avoiding collisions (CA) are clearly differentiated from other tasks, ensuring minimal confusion. Additionally, the decisions from similar contexts, such as ST and SI, although showing higher similarities than the average value of 0.2, remain distinguishable from each other. This analysis provides the user with a clear understanding of the network's performance in discriminating between different tasks. It also allows for the consolidation of similar task classes (ST and SI in our case), which increases the efficiency of decision interpretation.

Learning Curves We evaluate the learning curves of various baseline models based on their control and planning performance (Fig. 4 (C)). For sensorimotor control performance, we calculate the L1 loss using the validation dataset. To assess planning-level performance, we compute a similarity score, which is the sum of the diagonal values in the RSM results. In addition to our method, we configure ViTNet, which comprises only perception and control modules, to investigate the necessity of the planning module. For ViTNet's similarity score computation, we select the perceptual feature (z^p) and the control-level hidden states (z^c), where z^c is the output of Eq. 5, computed without the top-down decision process. Additionally, we introduce another version of MoNet (MoNet w/o L_{CGC}), trained without the contrastive loss L_{CGC}, to analyze the effectiveness of the self-supervised scheme. Our approach achieves notable improvement in top-down latent planning over other models, without compromising sensorimotor learning capabilities. The L1 loss curves show minimal changes with the addition of an extra planning module or a contrastive learning scheme. Meanwhile, in the similarity score curves, our method outperforms other approaches, demonstrating 11%-47% higher final performance. MoNet (w/o L_{CGC}), which only includes a planning module, does not exhibit significant discrimination performance (1.00) in the latent space. In contrast, the latent decision (h^d) of our modular network (MoNet), utilizing the self-supervised contrastive scheme, achieves the highest terminal score (1.47), outperforming the perception-level (1.32) and control-level (1.13) hidden features. This indicates that our method substantially enhances the task-specificity of end-to-end inferences without affecting policy learning. Moreover, it reveals that our latent decision more effectively embeds contextual characteristics compared to perceptual or low-level control features, thereby improving its suitability for behavioral interpretation.

Analysis of Interpretability

We investigate the interpretability and transparency of our model while performing end-to-end sensorimotor processing by decoding the top-down latent decisions. Fig. 5 illustrates the quantitative results, including latent decisions, decoded interpretable decisions, and control output, during an autonomous navigation episode encompassing multiple tasks. Since the decoded decisions are represented as probabilistic score vectors, we further compute the entropy of

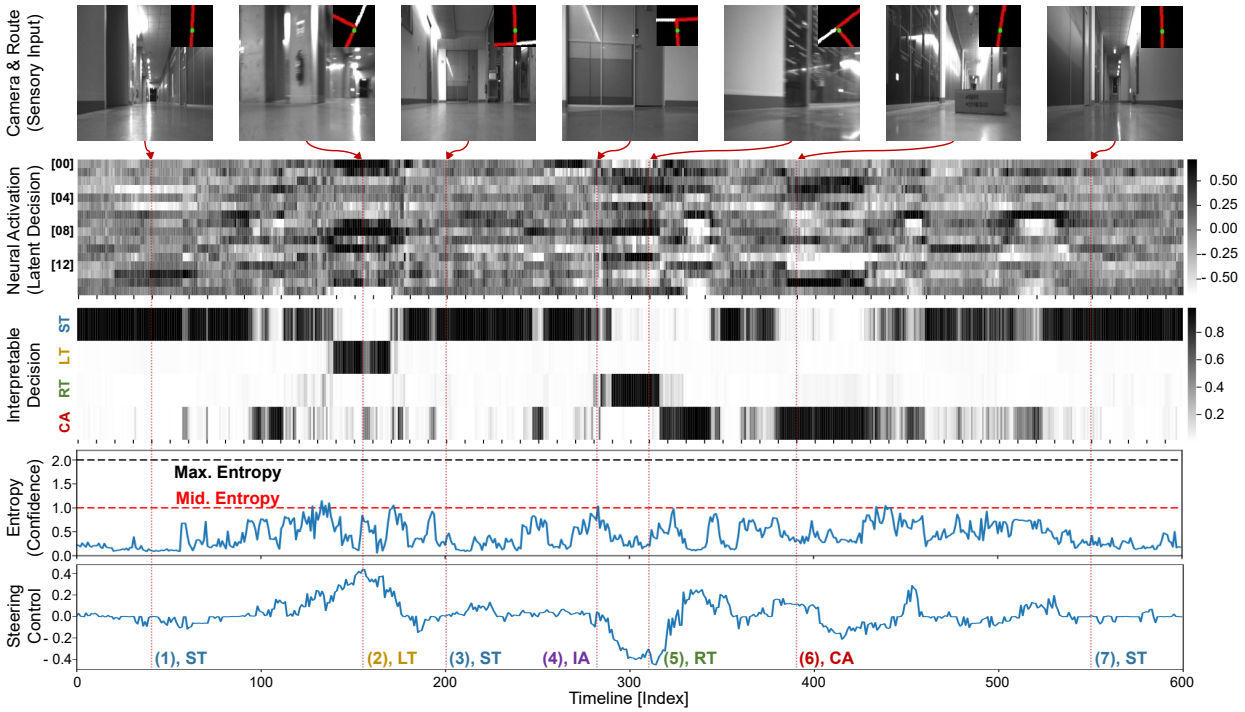

Figure 5: Quantitative results showing given sensory inputs, latent decisions, decoded interpretable decisions, entropy, and control output during an autonomous navigation episode.

these decisions. This entropy represents the confidence level of the internal decision-making during sensorimotor processing. The results show that our method can provide interpretable sensorimotor processes through decoded decisions that validly reflect the driving situation based on sensory inputs. In the early phase, our robot followed a straight corridor using minimal steering control, demonstrating strong probability scores for the task decision ST. However, when navigating intersections, our model produced latent decisions that were decoded as high scores for corresponding turns (LT, RT), necessitating large steering angle commands. In the case of approaching a wall or obstacle, our network generates different patterns (CA) of neural activations, resulting in unique decision responses compared to straight driving (ST) and turns (LT, RT).

Moreover, by analyzing the entropy of the decoded probability score vector, we can assess the confidence level of internal decision-making during end-to-end control. Whenever the robot needed to alter its current driving decisions, such as when approaching intersections or obstacles, the entropy of the decision increased to more than 1.0, indicating mid-level entropy values. Since latent decision-making is the causal process leading to robot control, our method can provide the internal confidence of the end-to-end inference prior to executing the robot's control actions. This shows the significant interpretability and transparency of our model from a behavioral perspective in practical applications.

We further delve into the perceptual and behavioral interpretation across various driving tasks and include these results in the appendix for brevity.

Conclusion

We introduced MoNet, a modular end-to-end network for self-supervised and interpretable sensorimotor learning. The network leverages functional modularity to enable a novel cognition-guided contrastive scheme. This scheme allows the network to learn task-specific sensorimotor control without task-level supervision. Furthermore, our network incorporates a self-attention mechanism and an internal decision process, both of which can be decoded into a spatial saliency map and an explainable decision. In real-world autonomous navigation, our model demonstrates effective sensorimotor performance with interpretability among multiple tasks.

Our approach to interpretable sensorimotor learning with functional modularity offers several advantages for the use of end-to-end networks. Firstly, it enables more reliable and less uncertain end-to-end processes in robotics. Our method allows human engineers to comprehend the network's intent and the rationale behind specific control outputs from perspectives beyond control-level observation, including perception and planning. Such enhancement is particularly valuable in real-world deployments where safety is critical. Secondly, our approach facilitates the integration of learning-based, black-box modules with nonlearning-based, white-box ones into a hybrid architecture. By leveraging decoded interpretable decisions from our modular network, it becomes feasible to conditionally apply either network-based policies or conventional controllers during deployment. We hope that our work contributes to integrating robotic sensorimotor processes with explainable artificial intelligence. We plan to validate our method across a more diverse range of robotic platforms and scenarios.

Acknowledgments

This work was supported by the Technology Innovation Program (RS-2023-00256794, Development of drone-robot cooperative multimodal delivery technology for cargo with a maximum weight of 40kg in urban areas) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea).

References

Amini, A.; Rosman, G.; Karaman, S.; and Rus, D. 2019. Variational end-to-end navigation and localization. In *2019 International Conference on Robotics and Automation (ICRA)*, 8958–8964. IEEE.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6077–6086.

Baluch, F.; and Itti, L. 2011. Mechanisms of top-down attention. *Trends in neurosciences*, 34(4): 210–224.

Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Muller, U.; Zhang, J.; et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Chen, J.; Li, S. E.; and Tomizuka, M. 2021. Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 5068–5078.

Chen, Y.; Dong, C.; Palanisamy, P.; Mudalige, P.; Muelling, K.; and Dolan, J. M. 2019. Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Codevilla, F.; Müller, M.; López, A.; Koltun, V.; and Dosovitskiy, A. 2018. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, 4693–4700. IEEE.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Gao, W.; Hsu, D.; Lee, W. S.; Shen, S.; and Subramanian, K. 2017. Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation. In *Conference on robot learning*, 185–194. PMLR.

Ghorbani, A.; Wexler, J.; Zou, J. Y.; and Kim, B. 2019. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hess, W.; Kohler, D.; Rapp, H.; and Andor, D. 2016. Real-time loop closure in 2D LIDAR SLAM. In *2016 IEEE international conference on robotics and automation (ICRA)*, 1271–1278. IEEE.

Huang, Z.; Lv, C.; Xing, Y.; and Wu, J. 2020. Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding. *IEEE Sensors Journal*, 21(10): 11781–11790.

Katsuki, F.; and Constantinidis, C. 2014. Bottom-up and top-down attention: different processes and overlapping neural systems. *The Neuroscientist*, 20(5): 509–521.

Kim, J.; and Canny, J. 2017. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, 2942–2950.

Larose, D. T.; and Larose, C. D. 2014. *Discovering knowledge in data: an introduction to data mining*, volume 4. John Wiley & Sons.

Meunier, D.; Lambiotte, R.; and Bullmore, E. T. 2010. Modular and hierarchically modular organization of brain networks. *Frontiers in neuroscience*, 4: 200.

Niculescu-Mizil, A.; and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, 625–632.

Nielsen, F.; and Nielsen, F. 2016. Hierarchical clustering. *Introduction to HPC with MPI for Data Science*, 195–211.

O'Kelly, M.; Zheng, H.; Karthik, D.; and Mangharam, R. 2020. F1tenth: An open-source evaluation environment for continuous control and reinforcement learning. *Proceedings of Machine Learning Research*, 123.

Pomerleau, D. A. 1988. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1.

Popal, H.; Wang, Y.; and Olson, I. R. 2019. A guide to representational similarity analysis for social neuroscience. *Social Cognitive and Affective Neuroscience*, 14(11): 1243–1253.

Schwarting, W.; Alonso-Mora, J.; and Rus, D. 2018. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1: 187–210.

Seong, H.; Jung, C.; Lee, S.; and Shim, D. H. 2021. Learning to drive at unsignalized intersections using attention-based deep reinforcement learning. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 559–566. IEEE.

Shi, W.; Huang, G.; Song, S.; Wang, Z.; Lin, T.; and Wu, C. 2020. Self-supervised discovering of interpretable features for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5): 2712–2724.

Suthaharan, S. 2016. Machine learning models and algorithms for big data classification. *Integr. Ser. Inf. Syst*, 36: 1–12.

Tampuu, A.; Matiisen, T.; Semikin, M.; Fishman, D.; and Muhammad, N. 2020. A survey of end-to-end driving: Architectures and training methods. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4): 1364–1384.

Tang, Y. 2013. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*.

Teng, S.; Chen, L.; Ai, Y.; Zhou, Y.; Xuanyuan, Z.; and Hu, X. 2022. Hierarchical interpretable imitation learning for end-to-end autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 8(1): 673–683.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zeng, W.; Luo, W.; Suo, S.; Sadat, A.; Yang, B.; Casas, S.; and Urtasun, R. 2019. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8660–8669.

Zhang, J.; Huang, Z.; and Ohn-Bar, E. 2023. Coaching a Teachable Student. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7805–7815.

Supplementary Materials

A demo video, codes, and dataset for quantitative results and interpretation are available at https://drive.google.com/drive/folders/1zREu8PdCf1D6hre06dnSpni8FPGwU6vT.

Perception Module in Details

The perception module utilizes the Transformer encoder to generate a saliency map, which is then integrated with the global context of the input image I through the self-attention mechanism. Following the description in (Dosovitskiy et al. 2020), the encoder network comprises: 1) a multi-head self-attention block (MSA), and 2) an MLP block, both equipped with layer normalizations (LN) and residual connections. After feature extraction by the CNN block (Eq. 13), the input embedding undergoes preprocessing (Eq. 14-15) before being fed into the Transformer encoder process (Eq. 16-18).

$$z_i = \text{CNN}_i(o_i), \quad i = \{I, M\} \tag{13}$$

$$z_i^{\text{flat}} = \text{reshape}_i(z_i), \quad z_I^{\text{flat}} \in \mathbb{R}^{N_p \times D_p}, z_M^{\text{flat}} \in \mathbb{R}^{D_p} \tag{14}$$

$$z_0^I = [z_I; z_I^{pos}] \in \mathbb{R}^{N_p \times (D_p+1)}, \quad z_I^{pos} \in \mathbb{R}^{N_p \times 1} \tag{15}$$

$$z'_l = \text{MSA}(\text{LN}(z_{l-1})) + z_{l-1}, \quad l = 1, ..., L \tag{16}$$

$$z_l = \text{MLP}(\text{LN}(z'_l)) + z'_l, \quad l = 1, ..., L \tag{17}$$

$$z^p = [z_I^{\text{att}}; z_M^{\text{flat}}], \quad \text{where } z_I^{\text{att}} = \text{MeanPool}(z_L) \tag{18}$$

where $N_p = 6 \times 6$ and $D_p = 64$. The MLP block includes ReLU for nonlinearity. Considering the limited computing resources available for on-board implementation, we construct a single-stack Transformer encoder ($L = 1$) for each module.

Hardware System and Coarse Topology Map

Our platform consists of a 1/10 scale racing car chassis equipped with an embedded computer, Jetson Xavier NX, and a controller, Arduino. The Xavier NX receives front camera images with the Realsense D435i camera sensor and acquires range measurements using a 2D LiDAR sensor (Hokuyo UST-20LX). These measurements are utilized to estimate the current pose of the ego vehicle through on-board localization (Hess et al. 2016) in GPS-denied indoor environments. The Xavier NX then computes an ego-centric coarse topology map (Amini et al. 2019), which includes a highlighted routed map alongside an unrouted map, based on the ego vehicle's pose and a globally routed path. This path is planned using the Dijkstra algorithm, utilizing a sparse topological roadmap of the indoor corridor environments. The Arduino receives commands from either the Xavier NX or a human driver, converting them into PWM signals for the steering and speed control motors of the platform.

Data Collection and Processing

While collecting data, we record camera images, topology maps, and corresponding command signals for steering and throttle control from the human driver. These control signals are normalized to a range of [-1, +1]. We collect data for a total of 2 hours, amounting to 88,326 pairs of sensory input and labels in the environments of Env. 1 and Env. 2. The data is split into training and validation sets at a ratio of $80 : 20$. The camera image is cropped to a size of 440×240 pixels and then resized to 224×224 pixels for use in our network. For data augmentation, we apply random image shifts and corresponding steering angle adjustments, as outlined in (Bojarski et al. 2016).

Perceptual and Behavioral Interpretability

Fig. 6 illustrates the results of perceptual and behavioral interpretation among various driving tasks. Using the spatial saliency map, we can explicitly interpret where the network focuses during autonomous navigation in real-world indoor environments. While the network does not specifically focus on any areas when driving straight through corridors, it shows strong spatial attention on the boundaries of intersections during turns, areas crucial for navigating the desired route. Similarly, upon encountering obstacles, the network generates spatial attention on the obstacle regions, further emphasizing critical areas for avoiding collisions. These results show that our network effectively identifies the regions with spatial importance in the visual sensory input during end-to-end autonomous driving, offering human engineers understandable insights into its perceptual processes.

Our model can also provide explainable decisions while performing end-to-end sensorimotor processing by decoding task-specific top-down latent decisions. In the experiments, our method yields explainable decoded decisions, which are validly recognized as corresponding to the driving situation based on the sensory inputs (Fig. 6). Even with varying environmental conditions in the driving scene, the top-down latent decision produces a similar distribution of neural values when the task-level context is analogous, resulting in accurate interpretations of behavioral intents.

Moreover, our method demonstrates both flexibility and scalability in interpretability. Drawing on previous quantitative results, we have consolidated straight driving tasks (ST, SI) into a single category, ST, by reconfiguring samples for the refitting of the SVMs. This underscores our method's ability to tailor the interpretation method to meet the specific

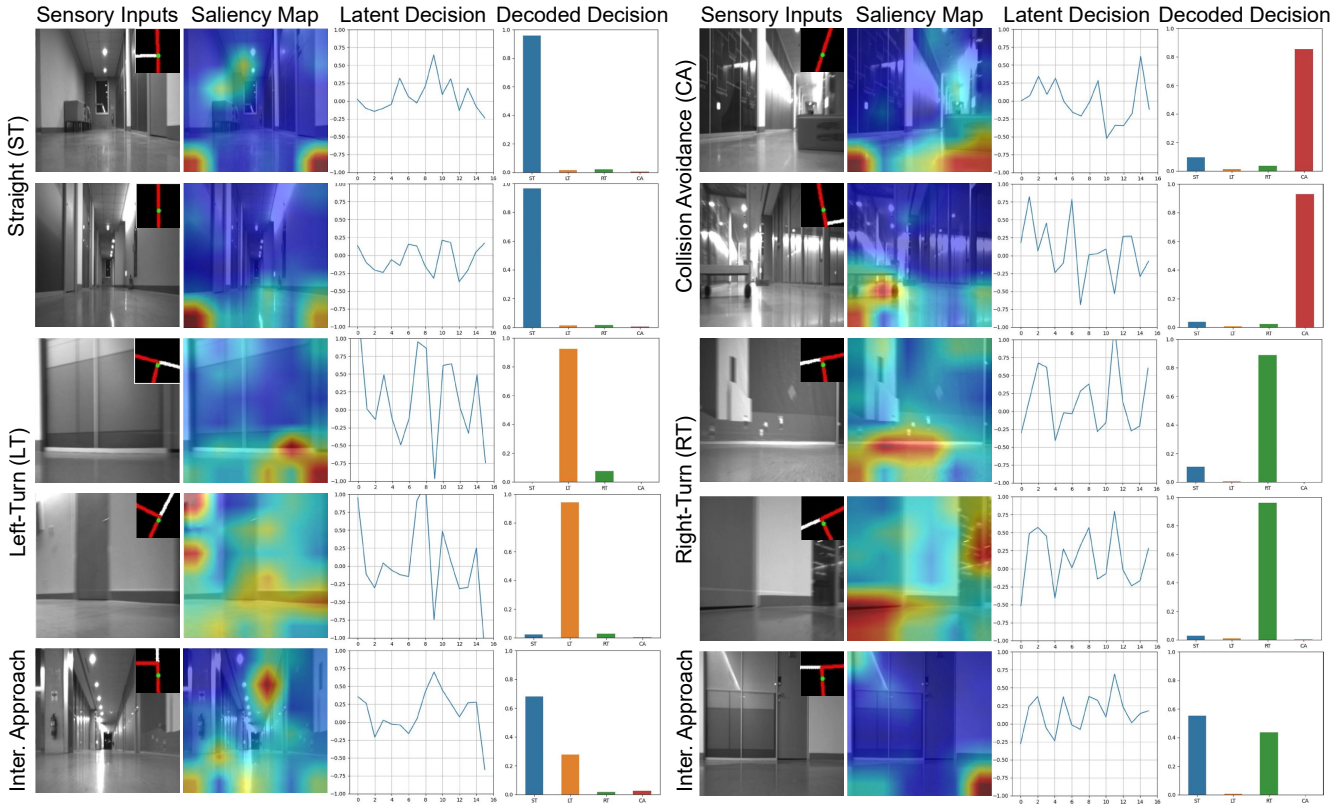

Figure 6: Results of the spatial saliency maps, latent decisions, and decoded interpretable decisions, corresponding to given sensory inputs from various driving tasks. Latent decisions are plotted to represent their distribution, while the decoded decisions are presented as posterior probabilities between ST, LT, RT, and CA.

needs of human engineers without having to retrain the original end-to-end network. Additionally, during navigation, we observe transition zones when the robot approaches intersections (*Inter. Approach*), shifting from straight driving (ST) to turning (LT/RT). This transition presents a unique pattern, with both ST and LT/RT exhibiting high posterior probabilities simultaneously. Standing apart from the five predefined tasks (ST, SI, LT, RT, CA), this pattern suggests our method's capacity to uncover new behavioral tasks not previously identified by human engineers. As mentioned, SVM samples can be restructured if necessary to facilitate interpretation of these newly identified tasks.