Goal-Guided Transformer-Enabled Reinforcement Learning for Efficient Autonomous Navigation

Article *in* IEEE Transactions on Intelligent Transportation Systems · September 2023

DOI: 10.1109/TITS.2023.3312453

4 authors:

Wenhui Huang
Nanyang Technological University
20 PUBLICATIONS **68** CITATIONS
SEE PROFILE

Yanxin Zhou
Nanyang Technological University
5 PUBLICATIONS **11** CITATIONS
SEE PROFILE

Xiangkun He
Nanyang Technological University
58 PUBLICATIONS **757** CITATIONS
SEE PROFILE

Chen Lv
Nanyang Technological University
341 PUBLICATIONS **7,593** CITATIONS
SEE PROFILE

Goal-guided Transformer-enabled Reinforcement Learning for Efficient Autonomous Navigation

Wenhui Huang, *Student Member, IEEE,* Yanxin Zhou, Xiangkun He, *Member, IEEE,* and Chen Lv, *Senior Member, IEEE*

***Abstract*—Despite some successful applications of goal-driven navigation, existing deep reinforcement learning (DRL)-based approaches notoriously suffers from poor data efficiency issue. One of the reasons is that the goal information is decoupled from the perception module and directly introduced as a condition of decision-making, resulting in the goal-irrelevant features of the scene representation playing an adversary role during the learning process. In light of this, we present a novel Goal-guided Transformer-enabled reinforcement learning (GTRL) approach by considering the physical goal states as an input of the scene encoder for guiding the scene representation to couple with the goal information and realizing efficient autonomous navigation. More specifically, we propose a novel variant of the Vision Transformer as the backbone of the perception system, namely Goal-guided Transformer (GoT), and pre-train it with expert priors to boost the data efficiency. Subsequently, a reinforcement learning algorithm is instantiated for the decision-making system, taking the goal-oriented scene representation from the GoT as the input and generating decision commands. As a result, our approach motivates the scene representation to concentrate mainly on goal-relevant features, which substantially enhances the data efficiency of the DRL learning process, leading to superior navigation performance. Both simulation and real-world experimental results manifest the superiority of our approach in terms of data efficiency, performance, robustness, and sim-to-real generalization, compared with other state-of-the-art (SOTA) baselines. The demonstration video [1] and the source code [2] are also provided.**

***Index Terms*—Autonomous navigation, deep reinforcement learning, goal guidance, transformer, data efficiency.**

I. Introduction

REINFORCEMENT Learning (RL) algorithms have significantly contributed to a wide range of domains over the past years, including but not limited to autonomous driving [1, 2, 3], unmanned ground vehicle (UGV) navigation [4, 5], and computer games [6]. With the representative capability of handling high-dimensional states, recent RL algorithms, e.g., deep Q-learning (DQN) [7, 8], deep deterministic policy gradient (DDPG) [9, 10], and soft actor-critic (SAC) [11, 12] are increasingly adopted by the robotics community to address decision-making problems, especially for autonomous navigation.

W. Huang, Y. Zhou, X. He, and C. Lv are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, 639798. (E-mail: wenhui001@e.ntu.edu.sg, yanxin001@e.ntu.edu.sg, xiangkun.he@ntu.edu.sg, lyuchen@ntu.edu.sg)

Corresponding author: Chen Lv. (E-mail: lyuchen@ntu.edu.sg)

[1]https://www.youtube.com/watch?v=aqJCHcsj4w0

[2]https://github.com/OscarHuangWind/DRL-Transformer-SimtoReal-Navigation

Conventional autonomous navigation methods that rely on prior knowledge of maps have been well-studied thanks to the Simultaneous Localization and Mapping (SLAM) technique [13, 14]. In reality, however, such an approach significantly depends on the map's precision and might even fail in an unknown environment. Therefore, developing a simple mapless navigation strategy directly utilizing sensor input, such as laser scan [15, 16] or visual images [17, 18] is an emerging field that is garnering significant attention in current UGV research. Especially having the advantage of visual fidelity, depth image-based autonomous navigation has been intensively studied by several works [19, 20]. Similarly, segmentation images [21] are often employed in mapless end-to-end navigation as well due to their powerful representative capability [22, 23]. In order to approach a target position, the approaches mentioned above train their models in a goal-conditional learning manner [24], directly concatenating the physical goal information (i.e., goal location in polar coordinate) with the latent states from the perception system (i.e., convolutional neural network) and feed into subsequent networks. Despite various degrees of success, these methods decouple the goal information from the scene representation, leading to poor data efficiency. For instance, latent states from the goal information-less scene encoder may include certain mismatched features that are unnecessary for reaching a goal position and thus play an adverse role during the RL training process.

Self-attention-based approaches, especially Transformers [25], have become the dominant model of choice in the natural language processing field. Motivated by adapting a standard Transformer architecture to images with the fewest modifications, a variant named Vision Transformer (ViT) [26] that can deal with image input is proposed in the computer vision community and has been applied to various domains, such as robotic manipulation [27] and autonomous driving [28]. However, there has yet to be an existing work that develops ViT-enabled DRL algorithms to UGV for realizing mapless autonomous navigation, especially for goal-driven tasks.

In light of this, we present a novel Goal-guided Transformer-enabled reinforcement learning (GTRL) approach by considering the physical goal states as an input of the scene encoder for guiding the scene representation to couple with the goal information and achieving efficient autonomous navigation. To realize a ViT architecture that treats both physical and visual states as the input, we propose a novel ViT variant with minimal modifications, which we call Goal-guided Transformer (GoT) for the rest of the paper, as the backbone of our

perception system. Then, we instantiate a GoT-enabled actor-critic algorithm, namely GoT-SAC, for the decision-making system, receiving the goal-oriented scene representation from the perception system and generating decision commands for the UGV. To boost the data efficiency, we pre-train the GoT with expert priors and then learn the decision-making with the subsequent RL process. As a result, our method makes the scene representation more interpretable in terms of reaching the goal information, which is confirmed through qualitative and quantitative evaluations. Most importantly, such an approach motivates the scene representation to concentrate mainly on goal-relevant features, which substantially enhances the data efficiency of the DRL learning process, leading to superior navigation performance. Therefore, the proposed approach is an efficient DRL-based autonomous navigation method for UGV from the goal-driven task perspective. We summarize the main contributions of this paper as follows:

1) A novel and Transformer architecture-based DRL approach, Goal-guided Transformer-enabled reinforcement learning (GTRL), is realized to achieve an efficient goal-driven autonomous navigation for the UGV.
2) A novel Transformer architecture is proposed, which we call Goal-guided Transformer (GoT) in this paper, through minimal modifications of ViT to handle the multimodal input: physical goal states and visual states. Most importantly, the GoT enables the scene representation to concentrate mainly on the goal-relevant features, significantly enhancing the data efficiency of the DRL learning process from the goal-driven task perspective.
3) As for the practical contribution, we instantiated a concrete GoT-enabled DRL algorithm for the proposed method and validated it both in simulation and the physical world. The experimental results demonstrate the clear superiority of the proposed approach in data efficiency and performance compared with other SOTA baselines. Moreover, the investigation of goal-driven navigation in the unknown environment confirms our approach's robustness and sim-to-real transferability.

II. Related Works

Due to the powerful representative capability to high dimensional states and superior data efficiency, DRL algorithms are gaining increasing attention among the robot community [29], especially for autonomous navigation of the UGV. Several works that infer the decision and control commands from laser scans have been proposed thanks to their robust transfer performance from the simulation to the real world. For instance, [30] trains the Asynchronous Advantage Actor-Critic (A3C) algorithm with intrinsic reward signals measured by curiosity to achieve mapless navigation. In [31], the steering angle is discretized into seven actions and trained together with the forward commands by the Advantage Actor-Critic (A2C) algorithm, and the trained model is applied to real-world obstacle avoidance. Similarly, [32] presents goal-driven mapless navigation based on an asynchronous Deep Deterministic Policy Gradient (DDPG) algorithm and successfully generalizes the learned model to the physical environment. In order to reduce the training time, [4] proposes to pre-train the Constrained Policy Optimization (CPO) algorithm with imitation learning (IL) and then continuously train it in the RL manner.

Nevertheless, laser scans cannot provide sufficient information to describe the environment in some cases [18], and thus scholars turn their attention to visual sensors-based approaches. In [20], a DDPG algorithm that considers depth images as input is employed to train the control policy by switching the different controllers. Similarly, work from [33] utilizes the same depth-based information but combines Behavior Cloning (BC) and Generative Adversarial Imitation Learning (GAIL) to demonstrate the enhanced performance of the social force-driven path planning. [22] presents a deep learning model consisting of the Convolutional Networks (ConvNets) and Long Short-Term Memory (LSTM) network to make a decision among seven commands based on semantic segmentation images in real-time. However, existing works mainly learn goal-driven tasks in a goal-conditional learning manner which is mentioned in [24]. Though this work focuses on conditional imitation learning (CIL), the logic behind utilizing goal information is similar to the DRL-based methods, treating the physical goal information as a condition of the decision-making and directly concatenating to the latent states provided by the perception system. On the contrary, we consider the physical goal information as an input of the scene encoder, rather than a condition, to extract matching features w.r.t. the goal-driven autonomous navigation and improve the DRL data efficiency.

ViT-based architecture has been a dominant choice not only for computer vision (CV) tasks but also for robotic research to achieve better scene representation and analysis. In [27], the ViT is utilized as one of the encoders to measure the stability of their manipulation approach. Similarly, [34] proposes a temporal multi-channel ViT to classify the hand motions for achieving better control of the bionic hands. To learn a more effective global context of the scene, [28] presents a perception utilizing ViT instead of ConvNets architecture, handling with the birds-eye-view (BEV) images. The simulation results indicate that such an encoder can identify the significant surrounding cars for the ego car to learn a safe and effective policy in complex environments. Another ViT-based work related to Vehicle-to-Everything (V2X) is presented in [35]. They propose a robust cooperative perception framework by means of building a holistic attention model, effectively integrating information across road users. Despite various degrees of success in the above domains, to our best knowledge, no current work develops ViT-enabled DRL algorithms for UGV's mapless autonomous navigation, especially for goal-driven tasks. Despite the superior representation capability, ViT is insufficient for achieving goal-driven autonomous navigation, as our task requires the scene encoder to handle the multi-modality of sensors. Several works related to multimodal ViT have recently been proposed to address computer vision tasks and achieve SOTA performance [36, 37]. The primary motivation of these approaches for employing multimodal inputs is that supplying the RGB images with rich complementary information [38]. In contrast, our approach focuses

on leveraging the input's multi-modalities to filter out goal-irrelevant information rather than enriching it. We accomplish this by fusing RGB images with physical goal states at the input level, extracting significant features oriented towards the goal.

III. Preliminaries

A. Reinforcement Learning

The objective of goal-driven autonomous navigation is that infers the linear and angular velocity of the UGV from the input states, including images and goal information. We consider such a task as a standard Markov decision process (MDP) formulated by a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} >$, where $\mathcal{S}$ is a set of states denoting the possible condition of the agent and environment, $\mathcal{A}$ represents action space, $\mathcal{P}$ models the transition of the environment, and $\mathcal{R}$ is the reward function evaluating the future overall payoff. At each time step t, the RL agent percepts the state $s_t \in \mathcal{S}$ and executes an action $a_t \in \mathcal{A}$, receiving an immediate reward $r_t = \mathcal{R}(s_t, a_t) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, as well as next state $s_{t+1} \in \mathcal{S}$ based on the transition probability $\mathcal{P}(s_{t+1}|s_t, a_t) : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$. Usually, the RL agent selects an action based on a policy $a_t \sim \pi(\cdot|s_t) : \mathcal{S} \rightarrow \mathcal{A}$, which represents a probability distribution denoting the belief that the agent holds about its decision at each time step. The target of the RL agent is to maximize the discounted total return along the future from an initial state s, i.e., $V^\pi(s)$, denoted as:

$$V^\pi(s) = \mathop{\mathbb{E}}_{s_t \sim \mathcal{P}}[\sum_{t=0}^{T} \gamma^t \cdot r_t], \tag{1}$$

where V^π is called value function and γ is the discounting factor constrained by $0 < \gamma \leq 1$. Similarly, the state-value function Q^π based on the state s_t and the action a_t at time step t is defined as:

$$\begin{aligned} Q^\pi(s_t, a_t) &= r_t + \gamma \cdot \mathop{\mathbb{E}}_{s_{t+1} \sim \mathcal{P}}[V^\pi(s_{t+1})] \\ &= r_t + \gamma \cdot \mathop{\mathbb{E}}_{s_{t+1} \sim \mathcal{P},\ a_{t+1} \sim \pi}[Q^\pi(s_{t+1}, a_{t+1})]. \end{aligned} \tag{2}$$

In the actor-critic method, an optimal policy π^* can be obtained by maximizing the overall future payoff for all states along one trajectory. Additionally, an entropy term can be augmented to the objective to prevent the policy from trapping in the local optima in the early stage [39]:

$$\max_{\pi} \mathop{\mathbb{E}}_{s_t \sim \mathcal{P}, a_t \sim \pi}[\sum_{t=0}^{T} \gamma^t (r_t + \alpha \mathcal{H}(\pi(\cdot|s_t)))] \tag{3}$$

where α is the temperature parameter that balances between overall future payoff and Shannon entropy of the policy.

B. Deep Imitation Learning

As one technique of behavior cloning method in imitation learning (IL) field, deep imitation learning (DIL) aims at directly mimicking the decision policy given a set of image state-action pairs $\mathcal{D} = \{< s_i, a_i >\}_{i=1}^{N}$, where N represents the number of samples. Therefore, it is a supervised learning problem by minimizing the statistical distance between action a_i and parameterized function approximator $\mathbf{F}(s_i; \psi)$:

$$\underset{\psi}{minimize} \sum_{i=1}^{N} \mathcal{L}(\mathbf{F}(s_i; \psi),\ a_i) \tag{4}$$

where $\mathcal{L}$ indicates loss function. Usually, we assume the action $a^{\mathbf{E}}$ is directly from a human expert which means the Eq. 4 can be reformulated as:

$$\underset{\psi}{minimize} \sum_{i=1}^{N} \mathcal{L}(\mathbf{F}(s_i; \psi),\ a_i^{\mathbf{E}}) \tag{5}$$

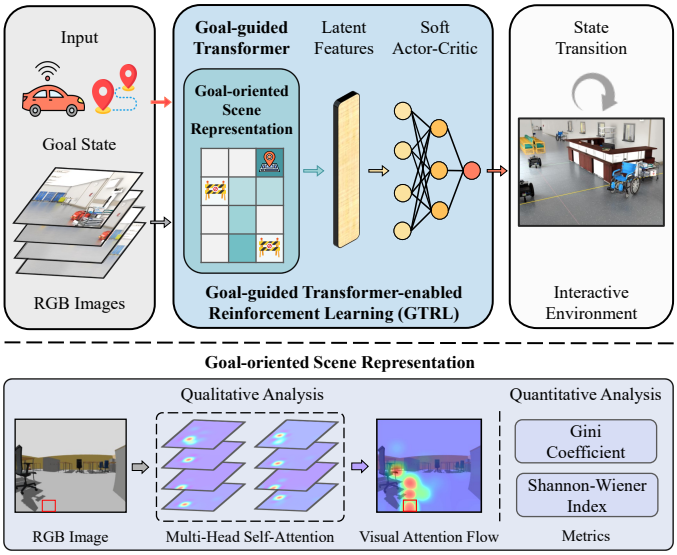


Fig. 1: The overall framework of the proposed approach. The goal state in the polar coordination system is considered as input of the proposed approach through the entire learning process, guiding the scene representation to couple with the goal information and boosting the subsequent decision-making process. The goal-oriented scene representation is evaluated through both qualitative and quantitative analysis.

C. Vision Transformer

The main idea behind ViT is splitting the images into patches and mapping them into linear embeddings in the same way the standard Transformer architecture treats tokens in natural language processing (NLP). Given an input image $x \in \mathbb{R}^{H \times W \times C}$, the ViT first reshapes it into a sequence of symbol representation $(x_1, x_2, ...x_n)$, where (H, W, C) are the resolution and channel dimension of the input image x and $x_n \in \mathbb{R}^{N \times (P^2 \cdot C)}$ is a representation of flattened 2D patches with the resolution P. Therefore, the total number of 2D patches can be calculated as follows:

$$N = \frac{H \cdot W}{P^2} \tag{6}$$

Then, the input of the ViT encoder can be obtained by augmenting the position embeddings $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$ to D-dimensional flattened 2D patches:

$$z_0 = [x_0;\ \mathbf{LP}(x_1);\ \mathbf{LP}(x_2); \cdots;\ \mathbf{LP}(x_n)] + \mathbf{E}_{pos} \tag{7}$$

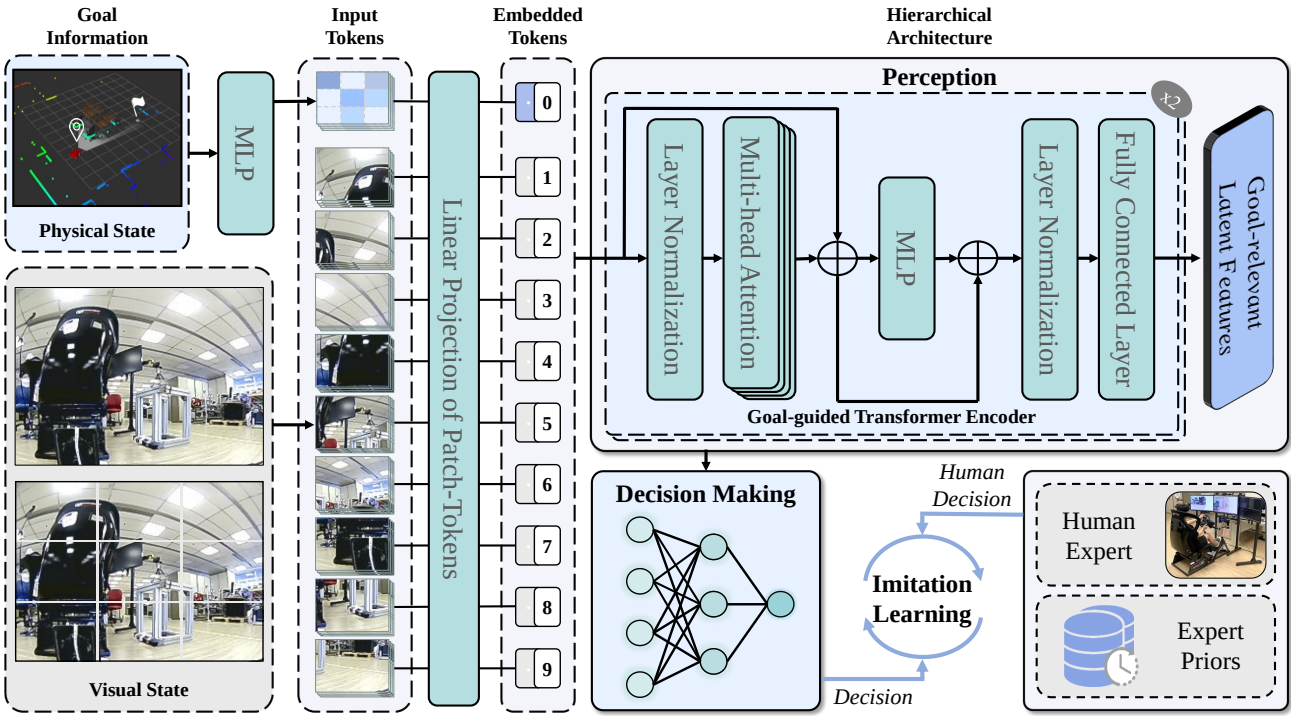


Fig. 2: Goal-guided Transformer Architecture and Pre-train with Expert Priors. The physical goal state is encoded to the goal tokens; thus, the input tokens consist of both goal and visual information. Then the final inputs of the GoT are obtained by performing position embeddings on the input tokens and fed into GoT encoder. Next, the GoT extracts goal-relevant latent features by coupling the scene representation with the goal information and delivers them to the subsequent decision-making network. Finally, the GoT is pre-trained with expert priors through the DIL technique to boost data efficiency.

where **LP** represents linear projection, and $x_0 \in \mathbb{R}^{1\times D}$ is an extra learnable embedding called class token. By feeding the embedded patches into the classic Transformer encoder, we can get multi-head self-attention (MSA) through the self-attention (SA) mechanism:

$$MSA(Q,K,V) = \mathbf{LP}([\mathbf{ATT}_1(Q,K,V);\ \mathbf{ATT}_2(Q,K,V);\cdots;\ \mathbf{ATT}_k(Q,K,V)]) \tag{8}$$

where k denotes k-th head and **ATT** indicates self-attention (SA) mechanism. As demonstrated in [25], we compute SA through the query Q, keys K, and values V:

$$\begin{aligned} \mathbf{ATT}(Q,K,V) &= softmax(\frac{QK^T}{\sqrt{d_k}})V \\ [Q,K,V] &= \mathbf{LP}(z) \end{aligned} \tag{9}$$

where z represents a set of embedded patches and d_k is a scaling factor.

IV. Methodology

A. Framework

The main aim of the goal-driven autonomous navigation task is to successfully navigate to a specific target instance within an unknown environment. Realizing such mapless navigation requires the DRL-based approach to understand and analyze the goal information. One possible solution is to treat the parameterized goal states as an input rather than a condition, feeding it together with the visual input, such as raw RGB images, to enhance the capability of the scene representation. Specifically, we learn the goal-oriented scene representation through a novel Transformer-based architecture that considers multimodal (i.e., physical goal states and visual states) input as a sequence of continuous representations. In light of this, we term the backbone of our perception system Goal-guided Transformer (GoT). Once the goal-oriented latent features are extracted, we motivate the SAC algorithm to learn the decision policy for approaching the goal position by interacting with the environment. Therefore, the two main ingredients, GoT and Transformer architecture-based SAC algorithm, complete our approach that we term Goal-guided Transformer-enabled reinforcement learning (GTRL).

The overall framework of our approach is depicted in Fig. 1. In our case, the input consists of two parts, i.e., goal position in polar coordinates and raw fisheye RGB images stacked over four frames. In the first stage, they are flattened into the same dimension and fed into the GoT encoder. Then, these embedded patches are encoded to goal-relevant latent features through the MSA and provided to the subsequent decision-making system. Finally, the GoT-SAC algorithm makes a decision according to the goal-relevant latent features, and the UGV executes the decision command to trigger the state transition of the environment. After the algorithm converges, we qualitatively (visual attention flow maps) and quantitatively (Gini coefficient and Shannon-Wiener Index) evaluate the trained model in terms of the SA mechanism to analyze and

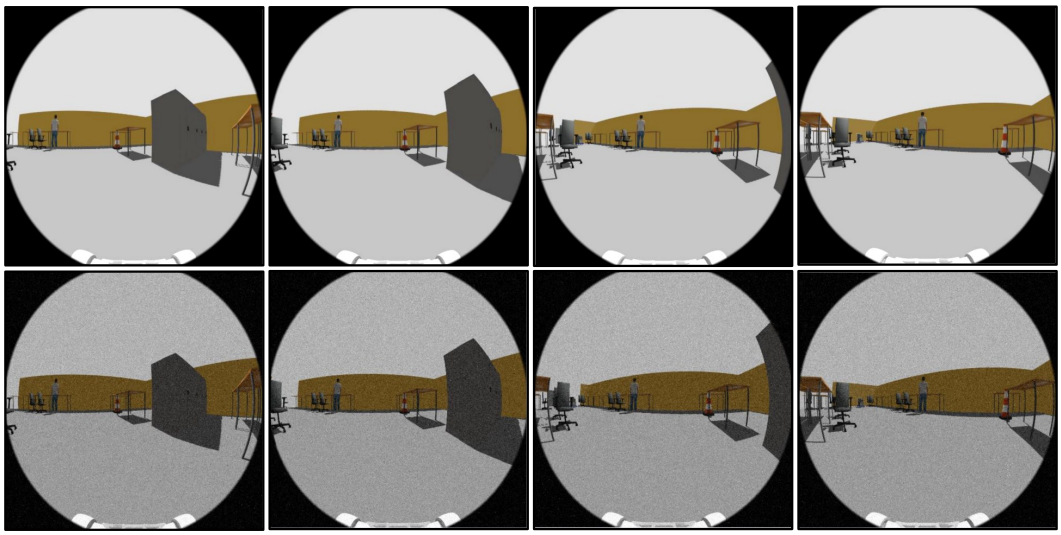


Fig. 3: RGB images from the fisheye camera stacked for most recent four frames. The upside pair of figures show the raw RGB images, whereas those on the downside illustrate pixel-level Gaussian noise-augmented images after preprocessing.

interpret the significance of the goal-oriented scene representation (Section V).

B. Goal-guided Transformer

In order to deal with the multimodality of the input, i.e., the goal states and visual images, we propose a novel variant of the ViT that we term GoT in this paper. In model design, We construct the architecture of the GoT by the minimum modification of ViT for the purpose of a simple setup. Specifically, inspired by BERT [40], we define a special goal token $\mathcal{G} \in \mathbb{R}^{1\times D}$ that is mapped from input goal states $s_{goal} \in \mathbb{R}^{1\times 2}$ through a multilayer perception (MLP) network:

$$\mathcal{G} = \mathbf{MLP}(s_{goal}) \tag{10}$$

Therefore, the embeddings of GoT can be formulated as:

$$\begin{aligned} z_0^{'} &= \mathbf{E}_{input}(s, \mathcal{G}) \\ z_0 &= z_0^{'} + \mathbf{E}_{pos} \end{aligned} \tag{11}$$

where $\mathbf{E}_{input}$ and $\mathbf{E}_{pos}$ represent input embeddings and position embeddings. By feeding the embeddings to the GoT encoder:

$$\begin{aligned} z_l^{'} &= \mathbf{MSA}(\mathbf{LN}(z_{l-1})) + z_{l-1} \\ z_l &= \mathbf{FC}(\mathbf{LN}(\mathbf{MLP}(z_l^{'}) + z_l^{'})) \end{aligned} \tag{12}$$

where l indicates the l-th block. We decide the depth of GoT as two blocks in this work, and hence, the latent features can be obtained from the output of the second block, denoted as:

$$h = \boldsymbol{GoT}(s, \mathcal{G}; \varphi) \tag{13}$$

where φ represents parameters of the GoT.

Figure 2 illustrates an overview of the GoT architecture and pre-train process. As the figure shows, the input consists of two modalities: goal information as the physical state and raw RGB images as the visual state. The physical state is fed into **MLP** network and encoded as feature patches while the visual state is decomposed to eight by eight small image patches (we illustrate this process with three by three image patches in the figure due to limited space). Therefore, we can obtain complete input tokens by integrating both kinds of patches. Furthermore, we add position embeddings for each input token and fix the one encoded from goal information to the first position in particular. As for the GoT encoder, it consists of an MSA block, the **MLP**, the fully connected layer (**FC**), the layer normalization operation [41], and the residual connections [42]. Considering the limited computational power and lightweight design, we employ two blocks of the encoder with only four heads per block. Having the latent features from perception and the subsequent decision system, we are able to perform DIL through expert demonstration data to pre-train the GoT, enabling the hot-start initialization in the subsequential training process. In a standard DIL, in terms of the goal-driven end-to-end navigation problem, the function approximator depends on the environment state s_i and goal state $s_{\{goal,\ i\}}$:

$$\underset{\psi,\ \psi_s}{minimize} \sum_{i=1}^{N} \mathcal{L}(\mathbf{F}(\mathbf{F}_s(s_i; \psi_s),\ s_{\{goal,\ i\}}; \psi),\ a_i^{\mathbf{E}}) \tag{14}$$

where ψ and N are the parameters of the function approximator and the number of samples. In our proposed approach, however, the goal state is no longer a condition but an input. Thus, the objective of goal-oriented imitation learning becomes:

$$\underset{\psi}{minimize} \sum_{i=1}^{N} \mathcal{L}(\mathbf{F}(\boldsymbol{GoT}(s_i, \mathcal{G}_i); \psi),\ a_i^{\mathbf{E}}) \tag{15}$$

In our case, such a design is essential since we aim to guide the scene representation to couple with the physical goal information so that the perception can extract goal-relevant and rational features to promote the data efficiency of the subsequent goal-driven decision process. To clearly demonstrate the point, we visualize goal-oriented scene representation through

Algorithm 1 Goal-guided Transformer-enabled Reinforcement Learning (GTRL)

Initialize Goal-guided Transformer (GoT) network with pre-trained parameters: φ^*.
Initialize actor and critic network parameters: ϕ, θ.
Initialize entropy parameters: α.
Initialize batch size N and replay buffer $\mathcal{D} \leftarrow \varnothing$.
Assign target parameters: $\theta_{target} \leftarrow \theta$.
for episode=1 to E **do**
  Initialize the environment state: $s_t \sim Env$
  Initialize the goal state: $s_{\{goal,\ t\}} \sim Env$
  for step=1 to S **do**
    Map goal token: $\mathcal{G}_t = \textbf{MLP}(s_{\{goal,\ t\}})$
    Scene Representation:
    $h_t \leftarrow \boldsymbol{GoT}(s_t, \mathcal{G}_t\ ; \varphi^*)$
    Sample an action: $a_t \leftarrow \pi_\phi(a_t|h_t)$
    Interact with the environment:
    $r_t, s_{t+1}, s_{\{goal,\ t+1\}} \sim Env$
    Store the transition:
    $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, s_{\{goal,\ t\}}, a_t, r_t, s_{t+1}, s_{\{goal,\ t+1\}})$
    If time to update critic **then**
      Sample a batch of the data:
      $(s_t^i, s_{\{goal,\ t\}}^i, a_t^i, r_t^i, s_{t+1}^i, s_{\{goal,\ t+1\}}^i)_{i=1}^N \sim \mathcal{D}$
      Compute critic (MBSE) loss: $\mathcal{L}(\theta)$ based on Eq. 23.
      Update parameters of critic network.
    end if
    If time to update actor **then**
      Sample a batch of the data:
      $(s_t^i, s_{\{goal,\ t\}}^i, a_t^i, r_t^i, s_{t+1}^i, s_{\{goal,\ t+1\}}^i)_{i=1}^N \sim \mathcal{D}$
      Compute actor loss: $\mathcal{L}(\phi)$ based on Eq. 25.
      Update parameters of actor network.
      If automatic tune is True **then**
        Update temperature parameter α.
      end if
    end if
    If time to update target network **then**
      Update target network: $\theta_{target} \leftarrow \theta$.
    end if
  end for
end for

visual attention flow maps [43] and quantitatively evaluate the reliability of our approach in section V. Additionally, this design allows us to generalize the Transformer architecture to the multimodal input while keeping the original characteristics.

C. Goal-guided Transformer-enabled Reinforcement Learning

As mentioned in section IV-A, the input of GTRL consists of two ingredients: visual states with raw RGB images and goal states in polar coordinates. In this work, we employ 160 × 120 raw RGB images from a fisheye camera with a FOV of 220 degrees and stack the four most recent frames. Additionally, we augment a pixel-level noise to the input images to learn a more robust and transferable decision policy for the sim-to-real experiments. Figure 3 demonstrates the difference between the original images and our input. The upside pair of four figures show the most recent four raw RGB images from the fisheye camera, whereas those on the downside illustrate Gaussian noise-augmented images that are utilized for training our algorithm. As for the goal state s_{goal}, we provide it in a 2-dimensional manner with the relative distance and heading error. Specifically, we define the first dimension of the goal state as the normalized relative distance and compute it as:

$$d_t = \textbf{min}(\frac{\|p_t^{<x,y>} - q^{<x,y>}\|_2}{\lambda},\ 1.0) \tag{16}$$

where $p_t^{<x,y>}$ denotes the real-time position of the UGV, $q^{<x,y>}$ indicates an arbitrary location of the goal point, $\|\cdot\|_2$ represents euclidean norm operation, and λ is a constant normalizer that maps the relative distance in the range of [0, 1]. Correspondingly, we associate the second dimension of the goal state as the heading error between UGV's orientation and the directional vector points to the goal position:

$$\Delta\varphi_t = \textbf{atan}\big((q^{<y>} - p_t^{<y>}),\ (q^{<x>} - p_t^{<x>})\big) - \psi_t \tag{17}$$

where ψ_t represents the heading angle of the UGV. Similar to the relative distance, we normalize the heading error as:

$$\Delta\varphi_t = \begin{cases} \frac{\Delta\varphi_t - 2\pi}{\pi}, & if\ \Delta\varphi_t > \pi \\ \frac{\Delta\varphi_t + 2\pi}{\pi}, & if\ \Delta\varphi_t < -\pi \\ \frac{\Delta\varphi_t}{\pi}, & otherwise \end{cases} \tag{18}$$

Receiving the above-mentioned input, the GTRL outputs decision commands $a_t = [v_t, \omega_t]$, i.e., linear velocity $v_t \in [0, 1]$ and angular velocity $\omega_t \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, and delivers them to the UGV through the Robot Operating System (ROS).

The target of autonomous navigation is to demonstrate a goal-driven decision and collision-free path planning for reaching the goal position. Therefore, we carefully design the reward function in combination with the continuous and sparse reward to boost the converge efficiency of the GTRL. More specifically, the overall payoff consists of four individual ingredients as follows:

$$r_t = r_h + r_a + r_g + r_c \tag{19}$$

where r_h denotes heuristic reward, r_a represents action reward, r_g indicates reward for arriving the goal position, and r_c is the

(a) Gazebo Environment. (b) UGV.

Fig. 4: Laboratory environment and UGV model.

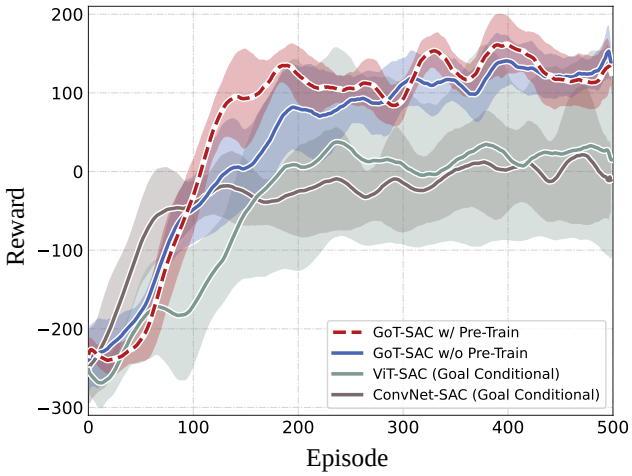


Fig. 5: Convergence curve comparison. The red dotted line and solid lines represent the average rewards of our algorithms and baselines per episode, while the shaded areas depict the variances over five runs.

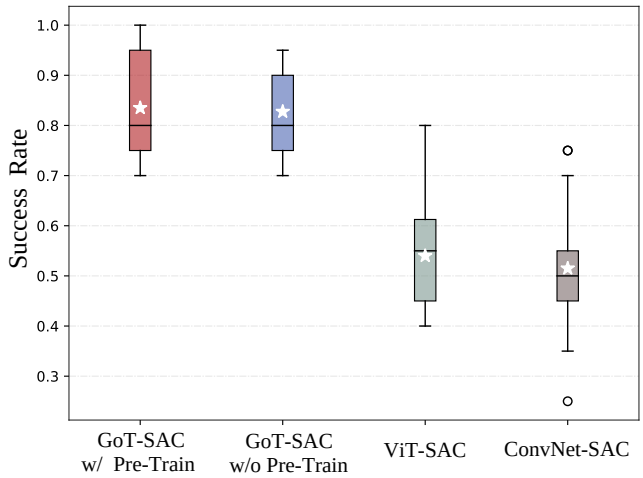


Fig. 6: Success Rate Boxplot. The black-solid line and "star" located at the box body denote the median and average, while the hollow circles represent the outliers.

collision penalty. The heuristic reward is designed to motivate the UGV to move toward the goal position:

$$r_h = \eta_h \times (\|p_{t-1}^{<x,y>} - q^{<x,y>}\|_2 - \|p_t^{<x,y>} - q^{<x,y>}\|_2) \quad (20)$$

where η_h is a constant weight. Similarly, we design the action reward to drive the UGV to approach the goal position as soon as possible but with the minimum number of steering operations:

$$r_a = v_t - \eta_a \times \mathbf{abs}(\omega_t) \quad (21)$$

where **abs** is an absolute value operation. Last but not least, two sparse rewards, i.e., the goal reach reward and the collision penalty, are designed as follows:

$$r_g = \begin{cases} 100, & if\ d_t <= \xi \\ 0, & otherwise \end{cases}$$
$$r_c = \begin{cases} -100, & if\ collision \\ 0, & otherwise \end{cases} \quad (22)$$

where ξ represents a constant margin w.r.t. the goal position.

Subsequently, given the extracted latent features h_t from GoT at a specific timestep t, the SAC algorithm learns the decision policy $\pi(a_t|h_t)$ based on the reward function mentioned above. One common technique widely utilized in the SAC algorithm is double Q-networks to tackle the over-estimation issue. Hence, the parameters of the critic network of GoT-SAC are updated by minimizing the mean bellman-squared error (MBSE) loss function:

$$\mathcal{L}(\theta_i) = \mathop{\mathbb{E}}_{h_t \sim \mathcal{P}, a_t \sim \pi} \|Q_{\theta_i}^{\pi}(h_t, a_t) - (r_t + \gamma \cdot \hat{Q}^{\pi})\|_2 \quad (23)$$

where $\hat{Q}^{\pi}$ is the state-action value of the next step from double target Q-networks and calculated by:

$$\hat{Q}^{\pi} = \mathop{\mathbb{E}}_{h_{t+1} \sim \mathcal{P}, a_{t+1} \sim \pi} \Big[\min_{i=1,2} Q_{\theta_i^{target}}^{\pi}(h_{t+1}, a_{t+1}) - \alpha \cdot log\pi(a_{t+1}|h_{t+1}) \Big] \quad (24)$$

where α is a temperature parameter that trades off between the stochasticity of the optimal policy and the state-action value. Accordingly, the actor network updates its parameters by maximizing the soft state-action function:

$$\mathcal{L}(\phi) = \mathop{\mathbf{E}}_{h_t \sim \mathcal{P}, a_t \sim \pi} \Big[\min_{i=1,2} Q_{\theta_i}^{\pi}(h_t, a_t) - \alpha \cdot log\pi_{\phi}(a_t|h_t) \Big] \quad (25)$$

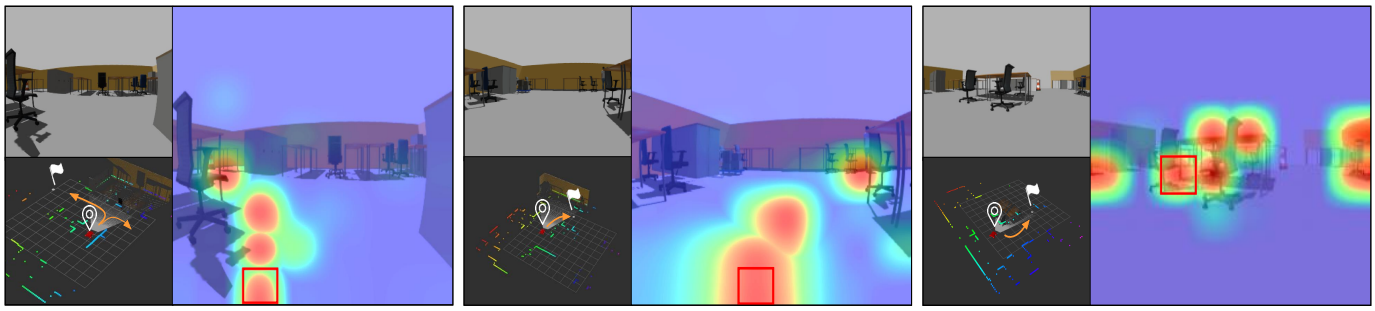
The detailed implementation of our approach is provided in Algorithm 1.

V. Experiments

A. Baseline Algorithms

To benchmark the proposed GTRL method for trustworthy end-to-end autonomous navigation, we employ state-of-the-art RL and DIL algorithms as baselines to compare the qualitative and quantitative performance both in simulation and the real world.

1) **ConvNet-SAC** [11]: A SOTA off-policy DRL algorithm that employs ConvNets as its scene representation encoder. We augment the physical goal state to the latent features encoded from ConvNets in a goal-conditional manner.
2) **ViT-SAC**: This baseline is derived from a SOTA ViT-based DRL algorithm called ViT-DQN [28], which employs ViT-DINO as the backbone of the DQN encoder. Without losing the original vital characteristics, we replace the DQN with SAC to fit the end-to-end navigation demand and call it ViT-SAC in the rest of the paper.
3) **MultiModal CIL** [44]: A SOTA conditional IL (CIL) algorithm that considers the human command or goal vector as a condition in the learning process. We select the command-input method among two architectures proposed in the original work to fit the goal-driven autonomous navigation task.
4) **MoveBase Planner**: A conventional planner widely utilized in UGV for goal-driven autonomous navigation. To



(a) Scenario I. (b) Scenario II. (c) Scenario III.

Fig. 7: Attention Flow Visualization. The left pair of diagrams for each subfigure shows the original RGB image and goal information, while the right side diagram depicts the revised RGB image masked by the attention flow. A red square highlights the queried image patch, and the attention level is represented through a color transition from blue (low) to red (high). a) Scenario I: query for 59th image patch occupied with drivable space, b) Scenario II: query for 60th image patch occupied with drivable space, c) Scenario III: query for 34th image patch occupied with obstacles.

TABLE I: Quantitative statistics of the self-attention mechanism behavior for the three goal-driven tasks. The data highlighted in bold denotes better results, i.e., higher Gini coefficient and lower Shannon-Wiener index, which depicts the attention are more concentrative on the significant image patches.

Model	Episode I		Episode II		Episode III	
	Gini Coefficient	Shannon-Wiener Index	Gini Coefficient	Shannon-Wiener Index	Gini Coefficient	Shannon-Wiener Index
GoT-SAC	**0.927**	**0.896**	**0.848**	**1.133**	**0.901**	**0.984**
ViT-SAC	0.802	1.613	0.616	1.807	0.695	1.545

be fair enough, we turn off the global map while keeping an eight-by-eight local map for real-time obstacle avoidance.

In addition, we employ vanilla GoT-SAC to learn the policy from scratch without any expert priors during the reinforcement training process to validate our proposed algorithm's data efficiency thoroughly.

B. Expert Priors

In order to pre-train the GoT, we asked the human participants to perform the demonstration in terms of the goal-driven navigation task and collected the data in image state-action pairs formation. During each episode, participants were given access to a randomly assigned goal position and were required to navigate toward it without collisions by continuously monitoring the fish-eye images displayed from a first-person perspective. More specifically, we provide the Logitech G29 driving set to the participants, allowing them to control the linear and angular velocity by manipulating the pedal/braking and steering wheel. To ensure the collection of reliable demonstrations, we selected two participants possessing valid driving licenses for the experiment. Additionally, a 10-minute training session was provided prior to the experiments to familiarize participants with the interaction devices and environments. Consequently, we obtained a total of 200 trajectories from human demonstrations, comprising 17,215 state-action pairs. These expert demonstrations were then divided into training and validation datasets, with a ratio of eight to two, resulting in 13,372 and 3,443 pairs, respectively, for use in the DIL process. The learning process terminates either when the maximum iteration limit is reached or when the validation loss turns to increase. The best model, determined by the lowest validation loss, is selected for the subsequent decision-making learning process.

C. Simulation Assessment

All algorithms are trained on a computer equipped with an Intel Core i7-10700 CPU, 64 GB of RAM, and an NVIDIA GTX 1660 SUPER graphics card. A high-fidelity autonomous navigation simulator, Gazebo, is employed to build the realistic laboratory environment and the UGV model for goal-driven mapless navigation, shown in Fig. 4. We train the instantiated algorithm for 500 episodes with a maximum of 200 steps for each. An episode ends when the goal position is reached, a collision occurs, or the UGV runs out of maximum step numbers. To well generalize the DRL-based policy and achieve better sim-to-real transferability, we not only augment a pixel-level Gaussian noise to the RGB image but also vary the initial location and goal position for each episode. Though the proposed algorithm only needs one fisheye camera for autonomous navigation, we also set a laser sensor in the simulation to detect the collision (4(b)). Furthermore, we employ the Robot Operating System (ROS) open-source platform to communicate with Gazebo and derive the goal information through subscribing to odometry messages.

Figure 5 illustrates the learning curves of GoT-SAC and all the DRL-based baselines. We run each algorithm with five different random seeds to measure statistics and evaluate the

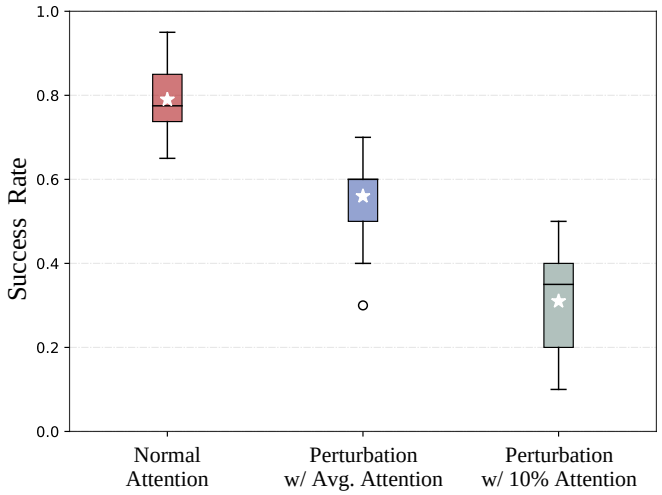


Fig. 8: Success Rate Comparison in terms of the different attention levels. The black-solid line and "star" located at the box body denote the median and average, while the hollow circles represent the outliers.

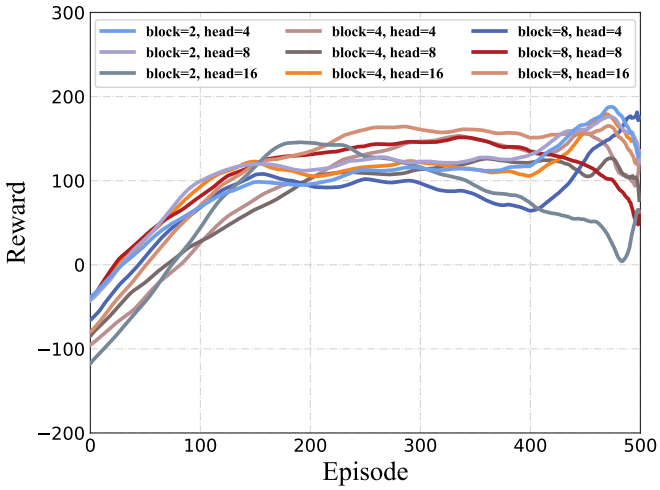


Fig. 9: Convergence curve for different combinations of SA head and encoder block parameters.

robustness. Specifically, the red dotted line and solid lines represent the average rewards of our algorithms and baselines per episode, while the shaded areas depict the variances over five runs. As the figure shows, both versions of GoT-SAC achieve higher reward levels with relatively lower variances than those of goal-conditional DRL-based algorithms, which indicates the significance of the goal-oriented scene representation. Moreover, both GoT-SAC models exhibit a faster convergence and enhance the training efficiency by over 129% and 86% compared with the ViT-SAC model. It should be noticed that though the convergence pace of ConvNet-SAC at the early stage is slightly faster due to its relatively small number of parameters, the average episode return is much lower than our proposed algorithm. Overall, the convergence curve confirms the better data efficiency of the proposed approach, that is, achieving an enhanced performance (a higher reward level) by consuming less amount of the data (fewer training episodes) compared to other baselines.

To evaluate the performance, we validate all the trained policies with 20 random seeds and run for 50 episodes for each seed. The success rate, which is obtained as the number of goal-reached episodes divided by the total runs, is employed as the metric for the evaluation. From Fig. 6, we can observe the dominant success rate and superior robustness of the proposed algorithm compared with other baselines regardless of varying a wide range of random seeds.

D. Attention Visualization and Evaluation

Besides the superior efficiency and performance, the GTRL approach also possesses a significant advantage in model interpretability thanks to the goal-oriented scene representation. To analyze the rationale behind fast convergence and excellent performance of our algorithm, we extract the attention from the GoT encoder w.r.t. randomly sampled RGB images and visualize it in Fig. 7. As the figure shows, the left pair of diagrams for each subfigure shows the original RGB image and goal information, while the right side diagram depicts the visual attention flow map [43] masked by the extracted attention. The queried image patch is highlighted by a red square, and the attention level is represented through a color transition from blue (low) to red (high). In Scenario I (Fig. 7(a)), the UGV is facing the oncoming T intersection, and the goal position locates on the left side behind the office chair and table. From the visual attention flow map, we can observe that the overall attention generates a visual path by mainly focusing on goal-oriented image patches. It should be noticed that the orientation of such a visual path obviously towards the goal position though the right turn is also feasible in this scenario, which proves that the scene representation successfully couples with the goal information. Similarly, a clear goal-driven visual path is shown in Scenario II (Fig. 7(b)). In Fig. 7(c), different from the previous two scenarios, we query for the image patch that occupies an obstacle (office chair) instead of drivable space. We surprisingly find that the attention highlights most of the adjacent obstacles, evidently pointing out the undrivable regions. Therefore, we qualitatively verify that our approach can provide a clear explanation of how the UGV analyzes the scene and arrives at the destination with a collision-free path.

Furthermore, we quantitatively evaluate and compare our approach with the ViT-SAC (goal-conditional) model to support the above conclusion. Specifically, we run each model with three random episodes and measure the statistical characteristics by averaging the whole frame. Realizing that this is an unsupervised task since there do not exist labels or ground truth for comparing, we employ two unsupervised metrics: Gini coefficient for measuring the evenness of the attention weights distribution [45] and Shannon-Wiener index for evaluating the concentration of the attention [46] in this experiment and the results are reported in Table I. The Gini coefficient and Shannon-Wiener index are widely utilized metrics for measuring statistical dispersion intended

TABLE II: Computational efficiency of selected parameter settings.

Settings	Algorithm	Complexity Metrics			
		FLOPs (M)	Params (M)	Inference (ms)	Train (hr)
2 Blocks & 4 Heads	GoT-SAC	36.9	0.46	0.24 ± 0.02	3.16 ± 0.1
	ViT-SAC	36.2	0.45	0.23 ± 0.03	3.10 ± 0.15

(a) SCOUTMINI. (b) Robotics Research Center.

Fig. 10: The real UGV platform and sim-to-real experiment environment. a) SCOUTMINI: an omnidirectional steering mobile robot from Agilex. b) Robotics research center: an indoor laboratory space in Nanyang Technological University.

to represent the evenness and diversity of distribution. From the table, we can observe that both metrics clearly reveal that the attention of the GoT-SAC model is sparser and tends to be more concentrated on task-related image patches, proving that better interpretability is achieved through goal-oriented scene representation.

Last but not least, we also investigate the impact of the significant attention through perturbation-based method [47] to observe how modifications of critical attention affect the navigation task performance. In light of this, we measure the success rate of the GoT-SAC model over another twenty random seeds with fifty episodes for each by dynamically replacing the essential attention (weights higher than 0.995) with a moving average and 10% of the original value. The boxplot illustrated in Figure 8 shows the overall result. We can observe that the performance of the GoT-SAC model, the one that decreases the significant attention to 10%, degrades catastrophically (62.5%) in terms of the success rate. Though the success rate of the model employing the average perturbation method is slightly higher than the previous one, it is still clearly lower than the normal GoT-SAC model, indicating the significance of the attention learned by our approach.

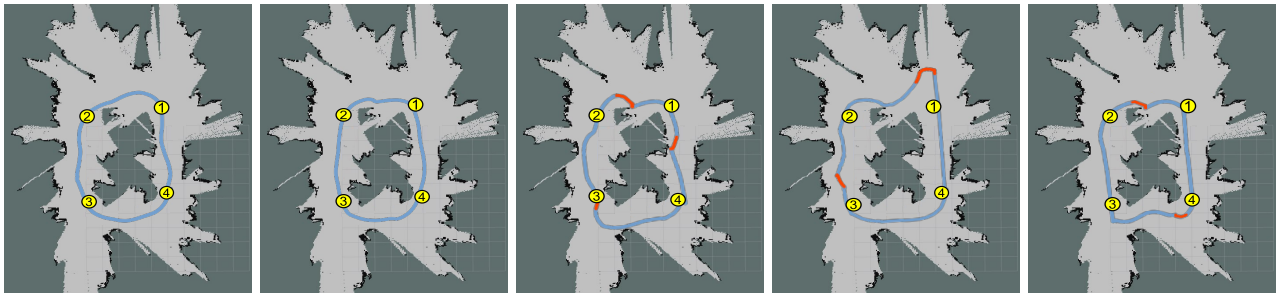
E. Ablation Study of Goal-guided Transformer Parameters

Acknowledging the significance of the attention mechanism discussed in the preceding section, it is of value to examine the influence of the architecture of the GoT, specifically the quantity of self-attention (SA) heads and encoder blocks. Thus, here we analyze the training performance of the GTRL in terms of the abovementioned two parameters. We fixed a random seed and tested the nine combinations of SA head and encoder block, shown in Fig. 9. From the figure, we can observe that all of the combinations successfully converge at a similar pace, though the overall performance slightly grows up as the number of SA heads and encoder blocks increases. More specifically, the group of eight blocks and sixteen heads demonstrates the fastest convergence speed and achieves the highest reward performance among nine combinations, while the worst one originated by the setting of two blocks and sixteen heads. Considering the principle of lightweight design and limited computation power of the hardware platform in the sim-to-real experiments, we employ two blocks of the encoder with four heads per block in this work. Table II illustrates the quantitative comparison of computational efficiency between GoT-SAC and ViT-SAC algorithms with selected parameters.

F. Sim-to-Real Assessment

In addition to evaluating the feasibility and performance of the algorithm in a virtual simulation environment, we also expect to apply our approach in real-world navigation tasks. In terms of the UGV, we use the omnidirectional steering mobile platform from Agilex called SCOUTMINI. The SCOUTMINI equips with an edge computing platform NVIDIA Jetson Xavier, a ZED2i stereo camera, an inertial measurement unit (IMU), and a fisheye camera with an ultra-wide FOV of 220 degrees (Fig. 10(a)). Regarding the software, we deliver the goal information and raw fisheye RGB images to the UGV through ROS. Then, the GoT-SAC sends the real-time decision inference to the UGV chassis via controller area network (CAN) communication to realize motion control. In this real-world experiment, all the algorithms are applied at the Robotics Research Center at Nanyang Technological University to complete a loop navigation task, as shown in Fig. 10(b). More specifically, we design four destinations that motivate the UGV to reach one by one with a small break after each arrival and finally return to the vicinity of the starting point. This experiment aims to test the algorithm's ability to avoid static obstacles and quickly navigate to given goal positions.

It should be mentioned that, unlike the evaluation in the simulation environment, our sim-to-real experiment involves the utilization of visual simultaneous localization and mapping (VSLAM) techniques to provide the ego-pose of the UGV and visualization of the environment. For this purpose, we have implemented the VINS-Fusion framework [48], which has been proven to demonstrate commendable performance within indoor settings. Notably, upon rigorous evaluation using benchmark datasets such as EuRoc [49], the VINS-Fusion framework has yielded a root square mean error (RSME)



(a) GoT-SAC. (b) ViT-SAC. (c) ConvNet-SAC. (d) MultiModal CIL. (e) MoveBase.

Fig. 11: Qualitative measurement of proposed algorithm and baselines. The solid blue line depicts the ground truth of the trajectory, while the solid red line represents the human-engaged path.

TABLE III: Quantitative performance of proposed algorithm compared with baselines. The data highlighted in bold denotes better results, i.e., relatively lower mean and variance in terms of the traveling distance and time.

Approach	Goal Position	Avg. Dist.	Var. Dist.	Avg. Time.	Var. Time	Success Rate	Engage Number
GoT-SAC	1st	6.044	± 0.049	14.048	± 0.282	100%	0
	2nd	**5.747**	± **0.110**	**11.987**	± **0.368**	100%	0
	3rd	**6.171**	± **0.036**	**12.821**	± **0.076**	100%	0
	4th	6.360	± 0.113	13.902	± 1.287	100%	0
ViT-SAC	1st	5.958	± 0.036	15.274	± 0.394	100%	0
	2nd	6.506	± 0.189	15.432	± 0.741	100%	0
	3rd	6.226	± 0.015	22.502	± 1.601	100%	0
	4th	7.274	± 0.210	16.449	± 0.722	100%	0
ConvNet-SAC[11]	1st	6.150	± 0.294	17.918	± 2.563	100%	4
	2nd	6.780	± 0.252	20.735	± 1.519	100%	5
	3rd	6.322	± 0.245	18.757	± 2.683	100%	6
	4th	5.971	± 0.123	13.550	± 0.414	100%	0
MultiModal CIL[44]	1st	5.852	± 0.021	13.458	± 0.180	100%	0
	2nd	9.868	± 0.480	57.263	± 6.026	60%	5
	3rd	6.666	± 0.134	75.121	± 12.019	20%	5
	4th	6.913	± 0.217	28.893	± 4.475	100%	0
MoveBase	1st	5.822	± 0.063	12.180	± 0.088	100%	0
	2nd	7.070	± 0.529	23.446	± 9.388	100%	4
	3rd	6.092	± 0.141	14.010	± 3.122	100%	1
	4th	6.510	± 0.517	24.098	± 5.442	100%	5

of approximately 0.1 meters for the absolute trajectory error (ATE). This level of precision signifies a considerable achievement and holds practical viability for real-world inference applications.

Figure 11 illustrates the qualitative measurement of performance for each algorithm. We plot both trajectories from the UGV and the human with two different colors, blue for ground truth and red for human engagement, respectively. As shown in the figure, the GoT-SAC policy performs smooth and collision-free navigation, while the other three algorithms (ConvNet-SAC, MultiModal CIL, and MoveBase) all need human engagement to arrive at the destinations. Surprisingly, we find that ViT-SAC policy also demonstrates an equally excellent performance despite the low average success rate during the evaluation in the simulation environment. It is reasonable since we select the best model for each algorithm for sim-to-real transfer assessment. It may also indicate the significance of the self-attention mechanism for goal-driven autonomous navigation.

To compare the performance and robustness of the policies in a deeper sight, we employ six statistical metrics for each goal-driven task: the average and variance of traveling distance, average and variance of navigation time, success rate, and engagement number. Especially the successful arrival is determined if the UGV reaches each goal position within one minute, and we actively engage the UGV control once the collision is likely to happen. A detailed quantitative measurement is reported in the Table. III. It is clear that the GoT-SAC model

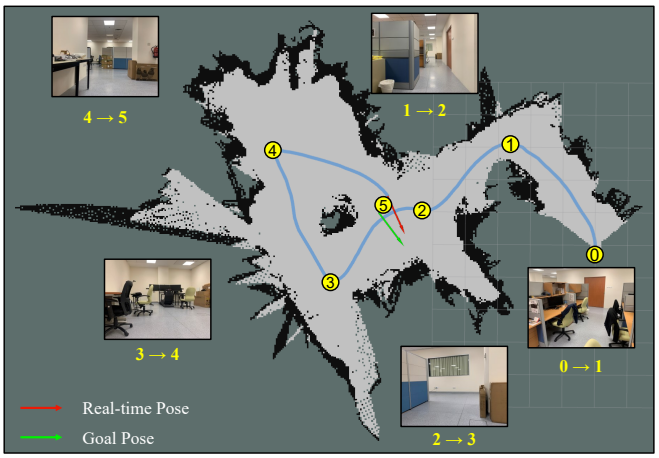


Fig. 12: Office Environment. The initial and goal positions are labeled by yellow circles, and the performed trajectory is highlighted with the solid blue line.

demonstrates dominant performance and robustness from all the domains compared with other baselines, including ViT-SAC and MoveBase planner. The performance of ViT-SAC is also comparably excellent, besides the longest navigated distance for the fourth destination and high average time for the third goal position. The worst performance is provided by the MultiModal CIL model, whose success rate is only 20% for reaching the third goal position. We can observe a similar performance from the statistical result of the ConvNet-SAC model in terms of the number of engagements, which is 15 in total. As for the MoveBase planner, the performance highly depends on the local cost-map quality, especially in the turning cases (the second and fourth goal position). For instance, the local cost-map occurs false detection frequently due to limited field of view and occlusion from the obstacles, leading to improper path planning. Overall, both quantitative and qualitative results in this sim-to-real experiment highlight the superiority of the proposed algorithm compared with other baselines, including against SOTA leaning-based approaches and the classic UGV navigation method.

Additionally, our approach is tested in an unknown environment to validate the generalization capability thoroughly. Due to the unstable connection and limitation of hardware, we select an unseen office environment rather than an outdoor space. It is worthwhile to test the generalization and transferability of the proposed algorithm in such an environment since it has a number of corner cases to be addressed, such as planning a collision-free path in narrow corridors, handling unseen obstacles (in terms of shape and color), and performing U-turn operation in order to reach the goal position. Figure 12 demonstrates the details of the experiment, where the yellow circle labels the initial and goal positions, and the performed trajectory is highlighted with the solid blue line. In particular, the UGV has to pass a narrow corridor to reach the first two destinations and perform 90-degree-turn and U-turn operations to arrive last two goal positions. Nevertheless, the GoT-SAC model can still approach all five goals without collision or engagement, indicating the excellent generalization capability and transferability of our approach.

VI. Conclusion and Discussions

This paper presents a Transformer-enabled DRL approach, namely GTRL, to realize efficient goal-driven autonomous navigation. Specifically, we first propose a novel Transformer-based architecture called Goal-guided Transformer (GoT) for the perception to consider the goal information as an input of the scene representation rather than a condition. For the purpose of boosting data efficiency, deep imitation learning is employed to pre-train the GoT. Then, a GoT-enabled soft actor-critic algorithm (GoT-SAC) is instantiated to train the decision policy based on the goal-oriented scene representation. As a result, our approach motivates the scene representation to concentrate mainly on goal-relevant features, which substantially enhances the data efficiency of the DRL learning process, leading to superior navigation performance. Both simulative and sim-to-real transfer experiments confirm our approach's superiority in data efficiency, performance, robustness, and sim-to-real generalization.

Despite the superior performance demonstrated by the proposed approach compared to other SOTA baselines, instances of failure were observed during the experiments. This can be attributed to the ground reflection resulting from the lighting conditions. Specifically, under strong lighting, the ground reflection introduces a substantial deviation between the original appearance and the RGB images captured by the fisheye camera, consequently leading to unfavorable navigation performance. Furthermore, the current approach exhibits degraded navigation performance when encountering dynamic obstacles, particularly pedestrians wearing clothes in a similar color to the ground, since it has never encountered such obstacles during the training phase.

In light of this, our future objective is to transfer the navigation environment from indoor to outdoor and incorporate a more diverse range of input modalities for our model. The navigation task becomes significantly more complex in the outdoor environment as the UGV must handle varying lighting conditions and highly dynamic pedestrians. To address these challenges, the adoption of multiple input modalities, such as occupancy flow or segmentation images, could be considered to perform interactive fusion for segmenting the drivable area explicitly, filtering out irrelevant information and thereby enhancing subsequent outdoor navigation performance. Moreover, we intend to incorporate abnormal detection and re-localization functions to mitigate the impact of position errors on navigation performance in the future. These mechanisms will help identify and handle situations where position errors exceed acceptable thresholds, maintaining a reliable and consistent navigation performance even in challenging scenarios.

References

[1] X. He, B. Lou, H. Yang, and C. Lv, "Robust decision making for autonomous vehicles at highway on-ramps: A constrained adversarial reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 4103–4113, 2022.

[2] J. Wu, W. Huang, N. de Boer, Y. Mo, X. He, and C. Lv, "Safe decision-making for lane-change of autonomous vehicles via human demonstration-aided reinforcement learning," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 1228–1233.

[3] W. Huang, F. Braghin, and Z. Wang, "Learning to drive via apprenticeship learning and deep reinforcement learning," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE Computer Society, 2019, pp. 1536–1540.

[4] M. Pfeiffer, S. Shukla, M. Turchetta, C. Cadena, A. Krause, R. Siegwart, and J. Nieto, "Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4423–4430, 2018.

[5] G. Kahn, P. Abbeel, and S. Levine, "Land: Learning to navigate from disengagements," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1872–1879, 2021.

[6] P. R. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devlic, F. Eckert, F. Fuchs *et al.*, "Outracing champion gran turismo drivers with deep reinforcement learning," *Nature*, vol. 602, no. 7896, pp. 223–228, 2022.

[7] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[8] W. Huang, C. Zhang, J. Wu, X. He, J. Zhang, and C. Lv, "Sampling efficient deep reinforcement learning through preference-guided stochastic exploration," *arXiv preprint arXiv:2206.09627*, 2022.

[9] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[10] W. Huang, F. Braghin, S. Arrigoni *et al.*, "Autonomous vehicle driving via deep deterministic policy gradient," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 3. American Society of Mechanical Engineers (ASME), 2019, pp. 1–8.

[11] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[12] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[13] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual odometry and visual slam: Applications to mobile robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.

[14] W. Huang, Y. Zhou, J. Li, and C. Lv, "Potential hazard-aware adaptive shared control for human-robot cooperative driving in unstructured environment," in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2022, pp. 405–410.

[15] R. Cimurs, I. H. Suh, and J. H. Lee, "Goal-driven autonomous exploration through deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 730–737, 2021.

[16] W. Zhu and M. Hayashibe, "A hierarchical deep reinforcement learning framework with high efficiency and generalization for fast and safe navigation," *IEEE Transactions on Industrial Electronics*, 2022.

[17] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.

[18] K. Wu, H. Wang, M. A. Esfahani, and S. Yuan, "Learn to navigate autonomously through deep reinforcement learning," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 5, pp. 5342–5352, 2021.

[19] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," *arXiv preprint arXiv:1706.09829*, 2017.

[20] L. Xie, Y. Miao, S. Wang, P. Blunsom, Z. Wang, C. Chen, A. Markham, and N. Trigoni, "Learning with stochastic guidance for robot navigation," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 166–176, 2020.

[21] X. Liu, Y. Lu, X. Liu, S. Bai, S. Li, and J. You, "Wasserstein loss with alternative reinforcement learning for severity-aware semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 587–596, 2020.

[22] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8846–8852.

[23] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklethwaite, N. Griffiths, A. Shah *et al.*, "Urban driving with conditional imitation learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 251–257.

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

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

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth

16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[27] N. Hansen, H. Su, and X. Wang, "Stabilizing deep q-learning with convnets and vision transformers under data augmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3680–3693, 2021.

[28] E. Kargar and V. Kyrki, "Vision transformer for learning driving policies in complex and dynamic environments," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 1558–1564.

[29] Z. Zhu and H. Zhao, "A survey of deep rl and il for autonomous driving policy learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, 2022.

[30] O. Zhelo, J. Zhang, L. Tai, M. Liu, and W. Burgard, "Curiosity-driven exploration for mapless navigation with deep reinforcement learning," *arXiv preprint arXiv:1804.00456*, 2018.

[31] M. Dobrevski and D. Skocaj, "Map-less goal-driven navigation based on reinforcement learning," in *23rd Computer Vision Winter Workshop*, 2018.

[32] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 31–36.

[33] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially compliant navigation through raw depth inputs with generative adversarial imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1111–1117.

[34] R. V. Godoy, G. J. Lahr, A. Dwivedi, T. J. Reis, P. H. Polegato, M. Becker, G. A. Caurin, and M. Liarokapis, "Electromyography-based, robust hand motion classification employing temporal multi-channel vision transformers," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 200–10 207, 2022.

[35] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," *arXiv preprint arXiv:2203.10638*, 2022.

[36] R. Girdhar, M. Singh, N. Ravi, L. van der Maaten, A. Joulin, and I. Misra, "Omnivore: A single model for many visual modalities," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 102–16 112.

[37] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir, "Multimae: Multi-modal multi-task masked autoencoders," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*. Springer, 2022, pp. 348–367.

[38] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen, "Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers," *arXiv preprint arXiv:2203.04838*, 2022.

[39] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *International conference on machine learning*. PMLR, 2017, pp. 1352–1361.

[40] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[41] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[43] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2942–2950.

[44] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[45] G. Letarte, F. Paradis, P. Giguère, and F. Laviolette, "Importance of self-attention for sentiment analysis," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018, pp. 267–275.

[46] J. Kim and M. Bansal, "Attentional bottleneck: Towards an interpretable deep driving network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 322–323.

[47] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of deep vision-based autonomous driving systems: Review and challenges," *International Journal of Computer Vision*, pp. 1–28, 2022.

[48] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[49] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

Wenhui Huang (Graduate Student Member, IEEE) received the B.S. degree in vehicle engineering from the Wuhan University of Technology, Wuhan, China, and the M.Sc. degree in mechanical engineering from the Polytechnic University of Milan, Milan, Italy, in 2018. He served as research engineer at HoloMatic Technology (Beijing) Co., Ltd. from 2019 to 2020. He is currently pursuing the Ph.D. degree with the school of mechanical and aerospace engineering at Nanyang Technological University, Singapore. His research mainly focuses on autonomous driving, autonomous navigation, reinforcement learning, and human-in-the-loop reinforcement learning.

Yanxin Zhou received the B.E. degree in 2019 from Harbin Institute of Technology, Weihai, China, and the Master degree in 2021 from Nanyang Technology University , Singapore, China. He is currently pursuing the Ph,D degree in mechnical and aerospace department, Nanyang Technology University, Singapore. His research interests include robotics, sensor fusion, simultaneous localisation and mapping and state estimation.

Xiangkun He (Member, IEEE) received his PhD degree in 2019 from the School of Vehicle and Mobility, Tsinghua University, Beijing, China. From 2019 to 2021, he served as a Senior Researcher at Huawei Noah's Ark Lab. He is currently a Research Fellow at Nanyang Technological University, Singapore. He has contributed as an author to over 40 peer-reviewed publications. His research interests include autonomous vehicles, reinforcement learning, trustworthy AI, decision and control. He received many awards and honors, selectively including the Tsinghua University Outstanding Doctoral Thesis Award in 2019, Best Paper Finalist at 2020 IEEE ICMA, 1st Class Outstanding Paper Award of China Journal of Highway and Transport in 2021, Huawei Major Technological Breakthrough Award in 2021, Huawei 2012 Lab Star Award in 2021, Huawei Hisilicon Chip Star Award in 2021, Best Paper Runner-Up Award at 2022 6th CAA International Conference on Vehicular Control and Intelligence, and Runner-Up at Intelligent Algorithm Final of 2022 Alibaba Global "Future Vehicle" Challenge. He is also a Reviewer for more than 40 journals and conferences, including IEEE TITS, IEEE TIV, IEEE TAI, IEEE TMech, IEEE TCYB, IEEE RAL and CoRL.

Chen Lv (Senior Member, IEEE) is a Nanyang Assistant Professor at School of Mechanical and Aerospace Engineering, and the Cluster Director in Future Mobility Solutions, Nanyang Technological University, Singapore. He received his PhD degree at Department of Automotive Engineering, Tsinghua University, China in Jan 2016. He was a joint PhD researcher at UC Berkeley, USA during 2014-2015, and worked as a Research Fellow at Cranfield University, UK during 2016-2018. He joined NTU and founded the Automated Driving and Human-Machine System (AutoMan) Research Lab since June 2018. His research focuses on intelligent vehicles, automated driving, and human-machine systems, where he has contributed 2 books, over 100 papers, and obtained 12 granted patents. He serves as Associate Editor for IEEE T-ITS, IEEE TVT, and IEEE T-IV. He received many awards and honors, selectively including the Highly Commended Paper Award of IMechE UK in 2012, Japan NSK Outstanding Mechanical Engineering Paper Award in 2014, Tsinghua University Outstanding Doctoral Thesis Award in 2016, IEEE IV Best Workshop/Special Session Paper Award in 2018, Automotive Innovation Best Paper Award in 2020, the winner of Waymo Open Dataset Challenges at CVPR 2021, Machines Young Investigator Award in 2022, and Best Paper Runner Up Award at CVCI 2022.