

Fixed-Dimensional and Permutation Invariant State Representation of Autonomous Driving

Jingliang Duan[✉], Dongjie Yu, Shengbo Eben Li[✉], *Senior Member, IEEE*, Wenxuan Wang, Yangang Ren[✉], Ziyu Lin[✉], and Bo Cheng

Abstract—In this paper, we propose a new state representation method, called encoding sum and concatenation (ESC), to describe the environment observation for decision-making in autonomous driving. Unlike existing state representation methods, ESC is applicable to the situation where the number of surrounding vehicles is variable and eliminates the need for manually pre-designed sorting rules, leading to higher representation ability and generality. The proposed ESC method introduces a feature neural network (NN) to encode the real-valued feature of each surrounding vehicle into an encoding vector, and then adds these vectors up to obtain the representation vector of the set of surrounding vehicles. Then, a fixed-dimensional and permutation-invariance state representation can be obtained by concatenating the set representation with other variables, such as indicators of the ego vehicle and road. By introducing the sum-of-power mapping, this paper has further proved that the injectivity of the ESC state representation can be guaranteed if the output dimension of the feature NN is greater than the number of variables of all surrounding vehicles. This means that the ESC representation can be used to describe the environment and taken as the inputs of learning-based policy functions. Experiments demonstrate that compared with the fixed-permutation representation method, the policy learning accuracy based on ESC representation is improved by 62.2%.

Index Terms—Permutation-invariance, state representation, autonomous driving.

I. INTRODUCTION

AUTONOMOUS driving has become a research hotspot since it can enhance road safety, ease road congestion, free human drivers, etc. Decision-making is the core component of achieving high-level autonomous driving. Although rule-based methods have been widely used to realize decision-making, manually encoding rules is not always feasible

due to the highly dynamic and stochastic nature of driving scenarios [1], [2]. The learning-based method is a promising technology to realize high-level autonomous driving by directly learning a parameterized policy that maps state representations to actions from data using supervised learning or reinforcement learning (RL) [3]. Recent learning-based decision-making research tends to use multi-layer neural networks (NNs) to represent the policy due to their remarkable fitting and generalization capabilities [4]–[6]. According to the state representation methods, the learning-based decision making can be divided into two categories: (1) end-to-end (E2E) decision making, which directly maps the raw sensors outputs to driving decisions, and (2) tensor-to-end (T2E) decision making, which describes states using real-valued representations, such as velocity and position.

The E2E decision-making method has been widely investigated during the last two decades, because it reduces the need for perception algorithms. In the late 1980s, Pomerleau built the first end-to-end autonomous driving system, called ALVINN, that took images consisting of 32×32 binary values and an 8×32 matrix from a laser range finder as inputs and output steering angles [7]. After training based on 1200 labeled samples, the NAVLAB vehicle equipped with ALVINN could drive in a 400m road without obstacles at the speed of 1m/s. Similarly, NVIDIA trained a convolutional driving policy NN for autonomous highway driving, which describes states using images from a single front-facing camera paired with the steering angles [8], [9]. In addition to supervised learning methods, Lillicrap *et al.* employed an RL algorithm, called DDPG, to learn a policy NN for lane-keeping on the TORCS simulation platform, which took simulated images as inputs and output acceleration quantity and steering wheel angles [10]. Besides, many other related works on E2E decision-making for autonomous driving can be found in [11]–[18]. Since there is a great difference between the sensor outputs of the simulated environment and the actual vehicle, the learned policy based on simulated perception is difficult to apply to real vehicles, or only applicable to simple driving tasks such as lane-keeping [7], [15]. Besides, the sensor outputs are also sensitive to the configuration of vehicle sensors, which limits the generalization of E2E decision-making methods in different vehicles.

Compared with E2E decision-making that takes raw sensors information as states, preliminary studies showed that real-valued representations perform better, due to the reduced state space being easier to learn and the real values making it

Manuscript received 20 July 2021; revised 2 November 2021; accepted 14 December 2021. Date of publication 24 December 2021; date of current version 8 July 2022. This work was supported in part by NSF China under Grant 52072213 and Grant U20A20334 and in part by the Tsinghua University-Toyota Joint Research Center for AI Technology of Automated Vehicle. (Jingliang Duan and Dongjie Yu contributed equally to this work.) (Corresponding author: Shengbo Eben Li.)

Jingliang Duan is with the State Key Laboratory of Automotive Safety and Energy, Center for Intelligent Connected Vehicles and Transportation, School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China, and also with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: duanj1@nus.edu.sg).

Dongjie Yu, Shengbo Eben Li, Wenxuan Wang, Yangang Ren, Ziyu Lin, and Bo Cheng are with the State Key Laboratory of Automotive Safety and Energy, Center for Intelligent Connected Vehicles and Transportation, School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China (e-mail: ydj20@mails.tsinghua.edu.cn; lishbo@tsinghua.edu.cn; wang-wx18@mails.tsinghua.edu.cn; ryg18@mails.tsinghua.edu.cn; linzy17@mails.tsinghua.edu.cn; chengbo@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TITS.2021.3136588

1558-0016 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

easier for the system to generalize [19]. Besides, the driving style and intention represented by pre-designed values can also be taken as policy inputs to further improve driving performance [20]. Therefore, T2E decision-making has achieved great success in autonomous driving [21]–[25]. Duan *et al.* represented driving states using a 26-dimensional vector, consisting of indicators of the ego vehicle, the road and the nearest four vehicles, realizing smooth and safe decision making on a simulated 2-lane highway via RL [26]. Guan *et al.* included a total of 16 variables from the ego vehicle and seven surrounding vehicles (position, speed, etc.) in the state representation to handle the cooperative longitudinal decision-making in a virtual intersection [27]. The information of different vehicles is sorted according to a pre-designed order to form the final state vector.

In summary, the T2E method needs to concatenate perception information of the ego vehicle, surrounding vehicles and roads into a state vector and then perform policy learning based on the vectorized state space. Although T2E has shown its advantages in terms of policy performance and generalization ability to vehicles with different sensor systems, it suffers from two challenges: (1) dimension sensitive problem and (2) permutation sensitive problem. The former means that T2E can only consider a fixed number of surrounding vehicles since the input dimension of the parameterized policy must be a predetermined value [19], [26], [27]. The latter indicates that the information of surrounding vehicles needs to be permuted according to manually designed sorting rules because different permutations lead to different state representations and policy outputs [21]–[24]. It is usually difficult to design a proper sorting order for complex driving scenarios such as intersections. These two challenges will not only limit the generality of T2E for different driving scenarios, but also hurt the performance of the learned policy.

In this paper, we propose a new state representation method, called encoding sum and concatenation (ESC), to describe the environment observation for learning-based decision making in autonomous driving. The main contributions and advantages of this paper are as follows:

- 1) The proposed ESC method introduces a feature NN to encode the real-valued feature of each surrounding vehicle into an encoding vector, and then adds these vectors up to obtain the representation vector of the set of surrounding vehicles. A fixed-dimensional and permutation-invariance state representation is obtained by concatenating the set representation with other variables, such as indicators of the ego vehicle and road. Different from the fixed-permutation representation method used in existing T2E studies [21]–[24], [26], [27], ESC is applicable to the situation where the number of surrounding vehicles is variable and eliminates the need for manually pre-designed sorting rules, leading to higher representation ability and generality.
- 2) By introducing the sum-of-power mapping, we have further proved that the injectivity of the ESC state representation can be guaranteed if the output dimension of the feature NN is greater than the number of variables of all surrounding vehicles. This means that the ESC

representation can be used to injectively describe the environment. Besides, we further show that, by taking the ESC representation as policy inputs, we can find the nearly optimal feature NN and policy NN by simultaneously optimizing them using gradient-based updating.

- 3) Function approximation experiments on six policy learning benchmarks demonstrate that, compared with the fixed-permutation representation method used in [21]–[24], [26], [27], the policy learning error based on the ESC representation is reduced by 62.2%.

In Section II, we describe the state representation problem, and analyze the effect of dimension sensitivity and permutation sensitivity on policy learning. Section III proposes the ESC state representation method. In Section IV, we present experimental results that show the efficacy of ESC. Section V concludes this paper.

Notation: \mathbb{R}^d denotes the set of d -dimensional real-valued vectors. \mathbb{N} denotes the set of natural numbers. \mathcal{O} denotes the set of all observed information. \mathcal{X} denotes the set of surrounding vehicles. x denotes the real-valued feature vector of each surrounding vehicle. s denotes the state vector. M denotes the number of surrounding vehicles within the perception range.

II. PROBLEM DESCRIPTION

In this section, we first describe the state representation problem. Then, we analyze the effect of dimension sensitive and permutation sensitive issues on the performance, generality, and sample complexity of policy learning.

A. Observation and State

We denote the observation set of driving scenarios as $\mathcal{O} \in \overline{\mathcal{O}}$, which consists of: (a) the information set of surrounding vehicles $\mathcal{X} = \{x_1, x_2, \dots, x_M\}$, where $x_i \in \mathbb{R}^{d_1}$ is the real-valued feature vector of the i th surrounding vehicle, and (b) the feature vector containing other information related to the driving task $x_{\text{else}} \in \mathbb{R}^{d_2}$, such as indicators of the ego vehicle and road geometry. Thus, $\mathcal{O} = \{\mathcal{X}, x_{\text{else}}\}$. The set size M of \mathcal{X} , i.e., the number of surrounding vehicles within the perception range of the ego car, is constantly changing due to the dynamic nature of the traffic. Assuming that the range of the number of surrounding vehicles is $[1, N] \cap \mathbb{N}$, the space of \mathcal{X} can be denoted as $\overline{\mathcal{X}} = \{\mathcal{X} | \mathcal{X} = \{x_1, \dots, x_M\}, x_i \in \mathbb{R}^{d_1}, i \leq M, M \in [1, N] \cap \mathbb{N}\}$, i.e., $\mathcal{X} \in \overline{\mathcal{X}}$. Noted that the subscript i of x_i in \mathcal{X} represents the ID of a certain surrounding vehicle. For example, $\mathcal{X} = \{x_M, x_{M-1}, \dots, x_1\}$ indicates that all vehicles are arranged in descending order according to the ID of each surrounding vehicle. Different permutations of these vehicles do not have an essential distinction since they represent the same traffic situation. But when they are input into some policy functions, the order matters.

We denote the mapping from the observation set \mathcal{O} to state representation s as $U(\mathcal{O})$, i.e.,

$$s = U(\mathcal{O}) = U(\mathcal{X}, x_{\text{else}}). \quad (1)$$

Current T2E researches usually concatenate the variables in \mathcal{O} to obtain the state representation vector s . According to the permutation of surrounding vehicles x_i in s , there are

two commonly used approaches: (1) all-permutation (AP) representation and (2) fixed-permutation (FP) representation. The AP method aims to consider all possible permutations of surrounding vehicles in s ,

$$s = U_{AP}(\mathcal{O}) = [x_{\varsigma(1)}^\top, \dots, x_{\varsigma(M)}^\top, x_{\text{else}}^\top]^\top, \quad (2)$$

where $U_{AP}(\mathcal{O}) : \overline{\mathcal{O}} \rightarrow \mathbb{R}^{Md_1+d_2}$ denotes the AP mapping and ς represents any possible permutation. For example, this representation method will take $s_1 = [x_1, \dots, x_M]$ and $s_2 = [x_M, \dots, x_1]$ as two different states although they represent the same traffic situation. Unlike the AP method, the FP method only considers one permutation, which arranges the objects in \mathcal{X} according to a pre-designed sorting rule o , i.e.,

$$s = U_{FP}(\mathcal{O}) = [x_{o(1)}^\top, \dots, x_{o(M)}^\top, x_{\text{else}}^\top]^\top, \quad (3)$$

where $U_{FP}(\mathcal{O}) : \overline{\mathcal{O}} \rightarrow \mathbb{R}^{Md_1+d_2}$ denotes the FP mapping.

According to (2) and (3), the change of vehicle number M or the permutation of surrounding vehicles may lead to different state vectors s , bringing two challenges: (1) dimension sensitivity and (2) permutation sensitivity. To find a better state representation method, it is necessary first to analyze the impact of these two issues on policy learning.

B. Dimension Sensitivity

The state dimension of AP and FP methods is $\dim(U_{AP}(\mathcal{O})) = \dim(U_{FP}(\mathcal{O})) = Md_1 + d_2$, which is proportional to the number of surrounding vehicles M . Since $M \in [1, N] \cap \mathbb{N}$ is constantly changing during driving, $\dim(s)$ is not a fixed value. However, the input dimension of the parameterized policy must be a predetermined fixed value due to the structure of the approximate functions, such as neural network (NN) and polynomial functions. This means that T2E methods based on AP or FP representation are only valid when the number of surrounding vehicles is fixed [21]–[24]. Assuming that only Z surrounding vehicles are considered, as shown in Figure 1, when $M > Z$, we need to select Z vehicles from \mathcal{X} based on pre-designed rules. When $M < Z$, we need to add $Z - M$ virtual vehicles far away from the ego to meet the input requirement of the policy function without affecting decision-making. The former will lead to information loss, while the latter will bring information redundancy. Therefore, it is crucial to select an appropriate value of Z according to the requirements of different driving tasks, which also limits the generality of AP and FP methods.

C. Permutation Sensitivity

As illustrated in Figure 2, assuming the number of surrounding vehicles M is fixed, different permutations of x_i correspond to different state vector s , thereby leading to different policy outputs. In other words, s and policy outputs are permutation sensitive to the order of surrounding vehicles. However, a reasonable driving decision should be permutation invariant to the order of objects in \mathcal{X} because all possible permutations correspond to the same driving scenario. To analyze the effect of permutation sensitivity, we first define the permutation invariant function as follows.

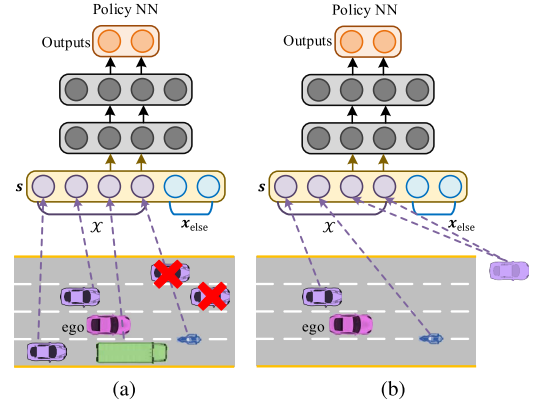


Fig. 1. Dimension sensitivity. (a) $M > Z$. (b) $M < Z$. When $M > Z$, some surrounding vehicles cannot be input into the policy function due to dimensional limitations; while when $M < Z$, we need to supplement the policy inputs with the information of virtual vehicles.

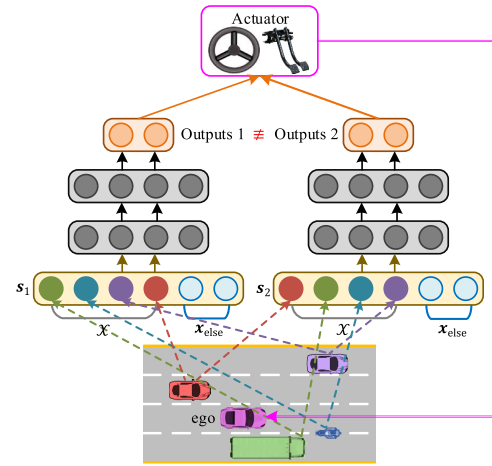


Fig. 2. Permutation sensitivity. For the same driving scenario, different permutations bring different policy inputs, which may result in different policy outputs.

Definition 1 (Permutation Invariant Function): Function $F : \overline{\mathcal{X}} \times \mathbb{R}^{d_2} \rightarrow \mathcal{Y}$ is permutation invariant to the order of objects in the set \mathcal{X} if $F(\{x_1, \dots, x_M\}, x_{\text{else}}) \equiv F(\{x_{\varsigma(1)}, \dots, x_{\varsigma(M)}\}, x_{\text{else}})$ for any permutation ς . For example, $F(\{x_1, \dots, x_M\}, x_{\text{else}}) = \|\sum_{x \in \mathcal{X}} x\|_2 + \|x_{\text{else}}\|_2$ is a permutation invariant function w.r.t. \mathcal{X} . Similarly, we define the permutation sensitive function as

Definition 2 (Permutation Sensitive Function): Function $F : \overline{\mathcal{X}} \times \mathbb{R}^{d_2} \rightarrow \mathcal{Y}$ is permutation sensitive to the order of objects in the set \mathcal{X} if $\exists \varsigma$ such that $F(\{x_1, \dots, x_M\}, x_{\text{else}}) \neq F(\{x_{\varsigma(1)}, \dots, x_{\varsigma(M)}\}, x_{\text{else}})$.

We denote the expected driving policy as $F_{PI}(\mathcal{X}, x_{\text{else}}) : \overline{\mathcal{X}} \times \mathbb{R}^{d_2} \rightarrow \mathcal{Y}$, which is permutation invariant w.r.t. \mathcal{X} . The objective of T2E decision-making methods is to learn a parameterized policy π , which takes U as inputs, such that

$$\pi(U(\{x_{\varsigma(1)}, \dots, x_{\varsigma(M)}\}, x_{\text{else}}); \psi^*) \approx F_{PI}(\mathcal{X}, x_{\text{else}}), \quad \forall \varsigma, \quad \forall \mathcal{X} \in \overline{\mathcal{X}}, \quad \forall x_{\text{else}} \in \mathbb{R}^{d_2}, \quad (4)$$

where ψ is the policy parameters and $*$ indicates that the parameters are optimal. An effective mapping U will significantly reduce the sample complexity and error of policy learning.

For the AP representation method in (2), the policy is learned by minimizing the following loss

$$\min_{\psi} \mathbb{E}_{\mathcal{X} \in \overline{\mathcal{X}}, x_{\text{else}} \in \mathbb{R}^{d_2}, \forall \mathcal{S}} (\pi(U_{\text{AP}}(\mathcal{O}); \psi) - F_{\text{PI}}(\mathcal{X}, x_{\text{else}}))^2. \quad (5)$$

The challenge faced by this method is that there are $M!$ permutations for a particular set \mathcal{X} containing M surrounding vehicles. This indicates that one driving scenario will correspond to $M!$ different state representations, which greatly increases the sample complexity.

For the FA representation method in (3), the policy can be found by minimizing

$$\min_{\psi} \mathbb{E}_{\mathcal{X} \in \overline{\mathcal{X}}, x_{\text{else}} \in \mathbb{R}^{d_2}} (\pi(U_{\text{FP}}(\mathcal{O}); \psi) - F_{\text{PI}}(\mathcal{X}, x_{\text{else}}))^2. \quad (6)$$

The pre-designed order o of FA guarantees the permutation invariance of the policy $\pi(U_{\text{FP}}(\mathcal{O}); \psi)$ w.r.t. \mathcal{X} , reducing the sample complexity compared with AP methods. However, it may break the continuity of the policy function w.r.t. each element in \mathcal{X} , i.e.,

$$\lim_{x'_1 \rightarrow x_1, \dots, x'_M \rightarrow x_M} \pi(x'_{o(1)}, \dots, x'_{o(M)}, x_{\text{else}}; \psi) \neq \pi(x_{o(1)}, \dots, x_{o(M)}, x_{\text{else}}; \psi), \quad \forall \psi. \quad (7)$$

Since the position of each surrounding vehicle is dynamically changing during driving, the position of x_i in $U_{\text{FP}}(\mathcal{O})$ may change at a certain time, resulting in a sudden change in the state s and policy output $\pi(U_{\text{FP}}(\mathcal{O}); \psi)$. For example, the rear vehicle at the current moment may become the preceding vehicle at a certain moment in the future by overtaking the ego vehicle. In particular, we will give a special case below for further explanation. Let $x_{\text{else}} \in \emptyset$, and $\mathcal{X} = \{[j, 2]^\top, [1, 5]^\top\}$, where j is a variable. The rule o sorts \mathcal{X} in increasing order according to the first element of x_i . It follows that when $j \leq 1$, $x_{o(1)} = [j, 2]^\top$ and $x_{o(2)} = [1, 5]^\top$; when $j > 1$, $x_{o(1)} = [1, 5]^\top$ and $x_{o(2)} = [j, 2]^\top$. It can be seen that the permutation of objects in \mathcal{X} has changed around $j = 1$, which may cause a sudden change in policy outputs, i.e.,

$$\lim_{j \rightarrow 1^-} \pi(U_{\text{FP}}(\mathcal{O}); \psi) = \pi([1, 2, 1, 5]^\top; \psi) \neq \lim_{j \rightarrow 1^+} \pi(U_{\text{FP}}(\mathcal{O}); \psi) = \pi([1, 5, 1, 2]^\top; \psi), \quad \forall \psi. \quad (8)$$

The policy discontinuity introduced by FA representations may bring difficulties to policy learning since the expected policy $F_{\text{PI}}(\mathcal{X}, x_{\text{else}})$ should be continuous w.r.t. to each element in \mathcal{X} . Besides, it is usually difficult to design a proper sorting rule for complex driving scenarios such as intersections.

To conclude, due to the permutation sensitivity, AP and FP methods suffer from high sample complexity and policy discontinuity respectively, which may result in poor policy learning accuracy.

III. ENCODING SUM AND CONCATENATION STATE REPRESENTATION

Both dimension sensitivity and permutation sensitivity will increase the policy learning difficulty and limit the applicability of T2E decision-making in different driving scenarios. In the last decade, permutation-invariance approximation

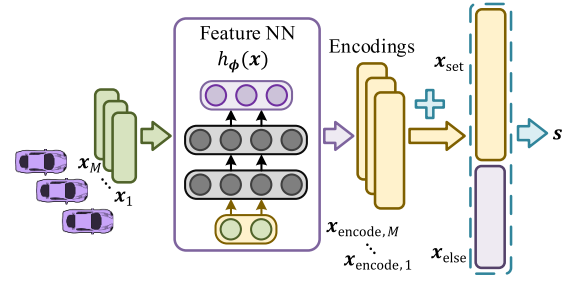


Fig. 3. ESC state representation.

methods have been extensively studied [28]–[30]. However, these methods are only applicable to (a) countable case where x_i is from a finite set, or (b) uncountable case with fixed set size M of \mathcal{X} where x_i is from a continuous space, but barely valid on uncountable case with a variable set size M . In this section, the existing permutation-invariance approximation theory is extended to the field of state representation in autonomous driving, and an encoding sum and concatenation (ESC) method is proposed to realize the fixed-dimensional and permutation invariant state representation of the observation set \mathcal{O} .

A. State Representation

As shown in Fig. 3, the mathematical description of the proposed ESC state representation is

$$s = U_{\text{ESC}}(\mathcal{O}; \phi) = \begin{bmatrix} x_{\text{set}} \\ x_{\text{else}} \end{bmatrix} = \begin{bmatrix} \sum_{x \in \mathcal{X}} h(x; \phi) \\ x_{\text{else}} \end{bmatrix}, \quad (9)$$

where $h(x; \phi) : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_3}$ is the feature NN with parameters ϕ and d_3 is the output dimension. Different from U_{AP} and U_{FP} , the ESC mapping $U_{\text{ESC}}(\mathcal{O}; \phi)$ is a parameterized function. ESC first encodes each x in the set \mathcal{X} into the corresponding encoding vector $x_{\text{encode}} \in \mathbb{R}^{d_3}$, i.e.,

$$x_{\text{encode}} = h(x; \phi). \quad (10)$$

Then, we obtain the representation vector x_{set} of the surrounding vehicles set by summing the encoding vector of each surrounding vehicle

$$x_{\text{set}} = \sum_{x \in \mathcal{X}} h(x; \phi). \quad (11)$$

From (11), it is clear that $\dim(x_{\text{set}}) = \dim(h_\phi) = d_3$ for $\forall M \in [1, N] \cap \mathbb{N}$. In other words, x_{set} is fixed-dimensional. Furthermore, the summation operator in (11) is permutation invariant w.r.t. \mathcal{X} . Thus, $U_{\text{ESC}}(\mathcal{O}; \phi) = [x_{\text{set}}^\top, x_{\text{else}}^\top]^\top$ is a fixed-dimensional and permutation invariant state representation of observation \mathcal{O} . Note that if $M = 0$, one can add a virtual surrounding vehicle that is far away from the ego vehicle, which brings no effect on the decision-making.

By taking $U_{\text{ESC}}(\mathcal{O}; \phi)$ as the inputs of π_ψ , the policy function can be expressed as

$$\pi(U_{\text{ESC}}(\mathcal{O}; \phi); \psi) = \pi\left(\sum_{x \in \mathcal{X}} h(x; \phi), x_{\text{else}}; \psi\right), \quad (12)$$

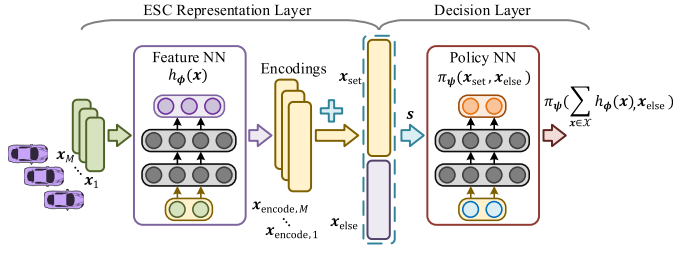


Fig. 4. Policy based on ESC state representation.

where $\pi(U_{\text{ESC}}(\mathcal{O}; \phi); \psi)$ is permutation invariant w.r.t. set \mathcal{X} . As shown in Fig. 4, the policy falls into two layers: (1) an ESC representation layer and (2) a decision layer. In the sequel, we refer to the policy function $\pi(U_{\text{ESC}}(\mathcal{O}; \phi); \psi)$ based on the ESC representations as the ESC policy.

B. Injection and Optimality Analysis

In addition to the fixed dimension and permutation invariance properties, to ensure the existence of ψ^* and ϕ^* , such that

$$\pi(U_{\text{ESC}}(\mathcal{X}, x_{\text{else}}; \phi^*); \psi^*) \approx F_{\text{PI}}(\mathcal{X}, x_{\text{else}}), \quad \forall \mathcal{X} \in \overline{\mathcal{X}}, \quad \forall x_{\text{else}} \in \mathbb{R}^{d_2}, \quad (13)$$

the ESC state representation or ESC policy also needs to be injective w.r.t. the surrounding vehicles set \mathcal{X} . If U_{ESC} is an injective mapping, for any $\mathcal{X}_1, \mathcal{X}_2 \in \overline{\mathcal{X}}$ where $\mathcal{X}_1 \neq \mathcal{X}_2$, it holds that $\sum_{x \in \mathcal{X}_1} h(x; \phi) \neq \sum_{x \in \mathcal{X}_2} h(x; \phi)$ or $U_{\text{ESC}}(\{\mathcal{X}_1, x_{\text{else}}\}; \phi) \neq U_{\text{ESC}}(\{\mathcal{X}_2, x_{\text{else}}\}; \phi)$. In contrast, if it is non-injective, there exist $\mathcal{X}_1, \mathcal{X}_2 \in \overline{\mathcal{X}}$ where $\mathcal{X}_1 \neq \mathcal{X}_2$, such that $U_{\text{ESC}}(\{\mathcal{X}_1, x_{\text{else}}\}; \phi) = U_{\text{ESC}}(\{\mathcal{X}_2, x_{\text{else}}\}; \phi)$. This indicates two different driving scenarios correspond to the identical state representation, which leads to the same policy outputs, thus impairing driving safety. Therefore, it is crucial to make sure that there $\exists \phi^\dagger$ such that $U_{\text{ESC}}(\mathcal{X}, x_{\text{else}}; \phi^\dagger)$ is injective.

Before proving the injectivity of the proposed ESC method, the following two lemmas are needed.

Lemma 1 (Universal Approximation Theorem [31]): For any continuous function $F(x) : \mathbb{R}^n \rightarrow \mathbb{R}^d$ on a compact set Ω , there exists an over-parameterized NN (i.e., the number of hidden neurons is sufficiently large), which uniformly approximates $F(x)$ and its gradient to within arbitrarily small error $\epsilon \in \mathbb{R}_+$ on Ω .

Lemma 2 (Sum-of-Power Mapping [28]): Let $z = [z_1, \dots, z_m]^\top \in [0, 1]^m$ and define a sum-of-power mapping $E_n : [0, 1]^m \rightarrow \mathbb{R}^n$ as

$$E_n(z) = \begin{bmatrix} \sum_{i=1}^m (z_i)^1 \\ \vdots \\ \sum_{i=1}^m (z_i)^n \end{bmatrix}. \quad (14)$$

The mapping $E_n(z)$ is an injection (i.e. $z_1 \neq z_2 \rightarrow E_n(z_1) \neq E_n(z_2)$) if $n \geq m$.

Then, the main theorem is given as follows.

Theorem 1 (Injectivity of the ESC State Representation): Let $\mathcal{O} = \{\mathcal{X}, x_{\text{else}}\}$, where $x_{\text{else}} \in \mathbb{R}^{d_2}$ and

$\mathcal{X} = \{x_1, x_2, \dots, x_M\}$. Denote the space of \mathcal{X} as $\overline{\mathcal{X}}$, where $\overline{\mathcal{X}} = \{\mathcal{X} | \mathcal{X} = \{x_1, \dots, x_M\}, x_i \in [c_{\min}, c_{\max}]^{d_1}, i \leq M, M \in [1, N] \cap \mathbb{N}\}$, in which c_{\min} and c_{\max} are the lower and upper bounds of all elements in $\forall x_i$, respectively. Noted that the size M of the set \mathcal{X} is variable. If the feature NN $h(x; \phi) : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_3}$ is over-parameterized with a linear output layer, and its output dimension $d_3 \geq Nd_1 + 1$, there always $\exists \phi^\dagger$ such that the mapping $U_{\text{ESC}}(\mathcal{O}; \phi^\dagger) : \overline{\mathcal{X}} \times \mathbb{R}^{d_2} \rightarrow \mathbb{R}^{d_3+d_2}$ in (9) is injective.

Proof: Let $x_i = [x_{i,1}, \dots, x_{i,d_1}]^\top$. We concatenate the j th element of each x_i into the vector $X_j = [x_{1,j}, \dots, x_{M,j}]^\top$. By normalizing X_j using the min-max scaling method, for $\forall j \in [1, d_1]$, we will get

$$X_{\text{norm},j} = \left[\frac{x_{1,j} - c_{\min}}{c_{\max} - c_{\min}}, \dots, \frac{x_{M,j} - c_{\min}}{c_{\max} - c_{\min}} \right]^\top \in [0, 1]^M. \quad (15)$$

According to Lemma 2, when $n \geq M$, the sum-of-power mapping $E_n(X_{\text{norm},j}) : [0, 1]^M \rightarrow \mathbb{R}^n$ expressed as

$$E_n(X_{\text{norm},j}) = \begin{bmatrix} \sum_{i=1}^M \left(\frac{x_{i,j} - c_{\min}}{c_{\max} - c_{\min}} \right)^1 \\ \vdots \\ \sum_{i=1}^M \left(\frac{x_{i,j} - c_{\min}}{c_{\max} - c_{\min}} \right)^n \end{bmatrix} \quad (16)$$

is injective when M is a fixed value.

Let $\overline{\mathcal{X}}_{\text{norm}} = \{[0, 1]^{Md_1} | M \in [1, N] \cap \mathbb{N}\}$. From (16), since $N \geq M$, the mapping $G : \overline{\mathcal{X}}_{\text{norm}} \rightarrow \mathbb{R}^{Nd_1+1}$ defined as

$$G \left(\begin{bmatrix} X_{\text{norm},1} \\ \vdots \\ X_{\text{norm},d_1} \end{bmatrix} \right) = \begin{bmatrix} E_n(X_{\text{norm},1}) \\ \vdots \\ E_n(X_{\text{norm},d_1}) \\ \sum_{i=1}^M 1 \end{bmatrix} \quad (17)$$

is also injective. In particular, the existence of item $\sum_{i=1}^M 1$ makes the mapping G also suitable for the case where the set size M is variable.

Furthermore, according to Lemma 1, when $d_3 \geq Nd_1 + 1$, there always $\exists \phi^\dagger$, such that

$$h(x_i; \phi^\dagger) = \begin{bmatrix} \left(\frac{x_{i,1} - c_{\min}}{c_{\max} - c_{\min}} \right)^1 \\ \vdots \\ \left(\frac{x_{i,1} - c_{\min}}{c_{\max} - c_{\min}} \right)^N \\ \vdots \\ \left(\frac{x_{i,d_1} - c_{\min}}{c_{\max} - c_{\min}} \right)^1 \\ \vdots \\ \left(\frac{x_{i,d_1} - c_{\min}}{c_{\max} - c_{\min}} \right)^N \\ 1 \\ \vdots \end{bmatrix}, \quad \forall x_i \in [c_{\min}, c_{\max}]^{d_1}. \quad (18)$$

Then, it directly follows that

$$x_{\text{set}} = \sum_{x \in \mathcal{X}} h(x; \phi^\dagger) = \begin{bmatrix} E_n(X_{\text{norm},1}) \\ \vdots \\ E_n(X_{\text{norm},d_1}) \\ \sum_{i=1}^M 1 \\ \vdots \end{bmatrix}. \quad (19)$$

Similar to (17), $\sum_{x \in \mathcal{X}} h(x; \phi^\dagger) : \overline{\mathcal{X}} \rightarrow \mathbb{R}^{d_3}$ is injective, which means $U_{\text{PI}}(\mathcal{O}; \phi^\dagger) : \overline{\mathcal{X}} \times \mathbb{R}^{d_2} \rightarrow \mathbb{R}^{d_3+d_2}$ is also injective. \square

Next, we will analyze the optimality of the ESC representation and the ESC policy.

Lemma 3 (Global Minima of Over-Parameterized NNs [32], [33]): Consider the following optimization problem

$$\min_{\psi} \mathcal{L}(\psi) = \mathbb{E}_{X_i \in \mathcal{B}} \left\{ \frac{1}{2} (\mathcal{F}(X_i; \psi) - Y_i)^2 \right\},$$

where $X_i \in \mathbb{R}^n$ is the training input, $Y_i \in \mathbb{R}^d$ is the associated label, $\mathcal{B} = \{(X_1, Y_1), (X_2, Y_2), \dots\}$ is the dataset, ψ is the parameter to be optimized, and $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}^d$ is an NN. If the NN $\mathcal{F}(X; \psi)$ is over-parameterized, simple algorithms such as gradient descent (GD) or stochastic GD (SGD) can find global minima on the training objective $\mathcal{L}(\psi)$ in polynomial time, as long as the dataset \mathcal{B} is non-degenerate. The dataset is non-degenerate if the same inputs $X_1 = X_2$ have the same labels $Y_1 = Y_2$.

Theorem 2: Given any continuous function operating on a the set \mathcal{O} , i.e., $F_{\text{PI}} : \overline{\mathcal{X}} \times \mathbb{R}^{d_2} \rightarrow \mathcal{Y}$, which is permutation invariant to the elements in \mathcal{X} . Suppose \mathcal{O} , $\overline{\mathcal{X}}$, and \mathcal{X} as described in Theorem 1. If the feature NN $h(x; \phi) : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_3}$ and policy NN $\pi(U_{\text{ESC}}(\mathcal{O}; \phi); \psi) : \mathbb{R}^{d_2+d_3} \rightarrow \mathcal{Y}$ are both over-parameterized, and $d_3 \geq Nd_1 + 1$, we can find ϕ^* and ψ^* which make (13) hold by directly minimizing $L(\psi, \phi)$ using optimization methods such as GD and SGD, where

$$L(\psi, \phi) = \mathbb{E}_{\substack{\mathcal{X} \in \overline{\mathcal{X}}, \\ x_{\text{else}} \in \mathbb{R}^{d_2}}} (\pi(U_{\text{ESC}}(\mathcal{O}; \phi); \psi) - F_{\text{PI}}(\mathcal{X}, x_{\text{else}}))^2. \quad (20)$$

Proof: From Theorem 1, there $\exists \phi^\dagger$ such that $U_{\text{ESC}}(\mathcal{O}; \phi^\dagger) : \overline{\mathcal{X}} \times \mathbb{R}^{d_2} \rightarrow \mathbb{R}^{d_3+d_2}$ in (9) is injective. Therefore, from Lemma 1, one has

$$\min_{\psi} L(\psi, \phi^\dagger) \approx 0. \quad (21)$$

In other words, there exists a pair of ψ and ϕ , which makes $\pi(U_{\text{ESC}}(\mathcal{O}; \phi); \psi)$ approximate F_{PI} arbitrarily close. Although the nearly optimal parameters may not be unique, according to Lemma 3, we can find a pair of ϕ^* and ψ^* which makes (13) hold by directly minimizing $L(\psi, \phi)$ using optimization methods such as GD and SGD. \square

The existence of the injective mapping given in Theorem 1 can ensure that $\min_{\{\psi, \phi\}} L(\psi, \phi) \approx 0$ holds. Although the solution $U_{\text{ESC}}(\mathcal{O}; \phi^*)$ found in Theorem 2 may be non-injective, $\pi(U_{\text{ESC}}(\mathcal{O}; \phi^*); \psi^*)$ still approximates the target function F_{PI} arbitrarily close.

Remark 1: The feature NN h_ϕ is only related to the space $\overline{\mathcal{X}}$ of set \mathcal{X} , but is independent of function F_{PI} . This indicates that for any different continuous permutation invariant functions $F_{\text{PI},1}$ and $F_{\text{PI},2}$ operating on set \mathcal{O} , for the same injective mapping $U_{\text{ESC}}(\mathcal{O}; \phi^\dagger)$, there exist ψ_1 and ψ_2 assuring

$$\pi(U_{\text{PI}}(\mathcal{O}; \phi^\dagger); \psi_1) \approx F_{\text{PI},1}(\mathcal{X}, x_{\text{else}})$$

and

$$\pi(U_{\text{PI}}(\mathcal{O}; \phi^\dagger); \psi_2) \approx F_{\text{PI},2}(\mathcal{X}, x_{\text{else}})$$

for $\forall \mathcal{X} \in \overline{\mathcal{X}}$ and $\forall x_{\text{else}} \in \mathbb{R}^{d_2}$, respectively.

Remark 2: In Theorem 1 and 2, we require the feature of each surrounding vehicle to satisfy $x \in [c_{\min}, c_{\max}]^{d_1}$, where c_{\min} and c_{\max} are the lower and upper bounds of all elements in x . We know that for actual autonomous driving applications, the range of different indicators, such as velocity or heading angle, may be different. However, by utilizing some normalization methods, such as min-max feature scaling, we can easily normalize each element to the same range.

IV. EXPERIMENTAL VERIFICATION

This section validates the effectiveness of the proposed ESC method in a general policy learning task based on supervised learning. We take AP and FP representation methods as baselines.

A. Experiments Design

We set the dimension of x to $d_1 = 5$, and each element of x is bounded by $c_{\min} = -5$ and $c_{\max} = 5$, i.e., $x \in [-5, 5]^5$. Similarly, we set $x_{\text{else}} \in [-5, 5]^{10}$. We assume that the maximum size of set \mathcal{X} is $N = 20$, i.e., $M \in [1, 20]$. Based on these settings, we construct six expected policy functions in Table I as benchmarks. Noted that $\min(z)$, $\max(z)$, and $\text{mean}(z)$ in Table I represent taking the minimum, maximum, and mean value of elements in z , respectively, and $\|z\|_p$ denotes the p -norm of z . $\min(z)$, $\max(z)$, and $\text{mean}(z)$ are three typical permutation invariant operators. For example, the vehicle closest to the ego vehicle is usually an important reference for decision-making. The combination of these three operators can form many representative nonlinear permutation invariant functions.

We will learn a policy to approximate each benchmark using different state representation methods. Then the performance of the ESC method can be evaluated by comparing the policy approximation accuracy of different representations. As shown in Table II, according to the set size of \mathcal{X} , the experiment for each benchmark is divided into five cases, $M = 5$, $M = 10$, $M = 15$, $M = 20$ and $M \in [1, 20]$. In particular, only ESC is applicable to variable size set \mathcal{X} , that is, case 5.

For each case of each benchmark, we randomly generated a training set $\mathcal{S}_{\text{train}}$ containing one million samples and a test set $\mathcal{S}_{\text{test}}$ containing 2048 samples. The j th sample in $\mathcal{S}_{\text{train}}$ or $\mathcal{S}_{\text{test}}$ is denoted as $\{\{\mathcal{X}_j, x_{\text{else},j}\}, y_j\}$, where \mathcal{X}_j and $x_{\text{else},j}$ are sampled uniformly within their space, and $y_j = F_{\text{PI}}(\mathcal{X}_j, x_{\text{else},j})$. Given $\mathcal{S}_{\text{train}}$, the policy NN π_ψ (and the feature NN h_ϕ for ESC) based on the AP, FP, and ESC are optimized by directly minimizing (5), (6) and (20), respectively. For the FP method, the pre-designed order o arranges the elements of \mathcal{X} in increasing order according to the first element of x_i . If the first element is equal, we will compare the second element, and so on.

Remark 3: According to Theorem 1 and 2, the proposed ESC method is a general state representation method which is suitable for the uncountable case where x_i comes from a continuous space with a variable set size M . It can be applicable in many fields, such as UAVs control and autonomous driving. The experiments provided in this section mainly focus on the evaluation of state representation ability of the proposed method for general permutation-invariance functions.

TABLE I
BENCHMARKS

| No. | Expected permutation invariant policy functions |
|-----|---|
| 1 | $F_{PI}(\mathcal{X}, x_{\text{else}}) = \text{mean}(x_{\text{else}}) - 0.2 \min(\ x_1\ _3, \dots, \ x_M\ _3) + 0.4 \text{mean}(\ x_1\ _1, \dots, \ x_M\ _1) \times \max(\ x_1\ _2, \dots, \ x_M\ _2)$ |
| 2 | $F_{PI}(\mathcal{X}, x_{\text{else}}) = 0.5 \min(x_{\text{else}}) \times \max(\max(x_1), \dots, \max(x_M)) \times \min(\ x_1\ _4, \dots, \ x_M\ _4)$ |
| 3 | $F_{PI}(\mathcal{X}, x_{\text{else}}) = 0.2 \ x_{\text{else}}\ _3 + 2 \text{mean}(\ x_1\ _1, \dots, \ x_M\ _1) \times \text{mean}(\max(x_1), \dots, \max(x_M))$ |
| 4 | $F_{PI}(\mathcal{X}, x_{\text{else}}) = 5 \ x_{\text{else}}\ _2 \times \left\ \left[\frac{\min(x_1)}{\ x_1\ _2 + 0.1}, \dots, \frac{\min(x_M)}{\ x_M\ _2 + 0.1} \right] \right\ _4$ |
| 5 | $F_{PI}(\mathcal{X}, x_{\text{else}}) = 10 \ x_{\text{else}}\ _4 \times \text{mean} \left[\frac{\text{mean}(x_1) \max(x_1)}{\ x_1\ _4 + 0.1}, \dots, \frac{\text{mean}(x_M) \max(x_M)}{\ x_M\ _4 + 0.1} \right]$ |
| 6 | $F_{PI}(\mathcal{X}, x_{\text{else}}) = 8 \ x_{\text{else}}\ _2 \times \max \left[\frac{\text{mean}(x_1) \ x_1\ _3}{\ x_1\ _2 + 0.1}, \dots, \frac{\text{mean}(x_M) \ x_M\ _3}{\ x_M\ _2 + 0.1} \right]$ |

TABLE II
FIVE EXPERIMENTAL SETTINGS OF EACH BENCHMARK

| Case | Set Size M | Representation methods |
|------|-----------------|------------------------|
| 1 | 5 | 1) ESC; 2) FP; 3) AP |
| 2 | 10 | 1) ESC; 2) FP; 3) AP |
| 3 | 15 | 1) ESC; 2) FP; 3) AP |
| 4 | 20 | 1) ESC; 2) FP; 3) AP |
| 5 | $M \in [1, 20]$ | ESC |

Therefore, the experiment is not designed based on specific driving tasks. The combination of the ESC representation and policy learning methods such as RL, and their application in autonomous driving will be studied in the future.

B. Training Details

For the ESC method, we use a fully connected NN with five hidden layers, consisting of 256 units per layer, with Gaussian Error Linear Units (GELU) as activation functions for each layer [34], for both feature NN and policy NN (See Figure 5a). The output layer of each NN is linear. According to Theorem 1, the output dimension d_3 of h_ϕ should satisfy that $d_3 \geq Nd_1 + 1 = 101$, so we set $d_3 = 101$.

Unlike the ESC method containing two NNs, AP and FP only need to learn a policy NN. To avoid the influence of different NN architectures on policy learning accuracy, the policy NN for these two methods is designed as shown in Figure 5b. This architecture comprises 11 hidden layers, in which each layer contains 256 units with GELU activations, except for the middle layer (i.e., the 6th layer). The middle layer is a linear layer containing 101 units, which is equal to the output dimension of h_ϕ . The input dimension is $5M$, which is related to the set size of \mathcal{X} . Therefore, the approximation structures in Figure 5a and 5b have the same number of hidden layers and neurons. In particular, when $M = 1$, these two architectures are identical. This design will greatly reduce the impact of network structure differences on learning accuracy. By guaranteeing the similarity of approximation architectures, we can effectively evaluate the effects of different state representation methods.

For all representation methods, we adopt Adam [35] to update NNs where the decay rate of first- and

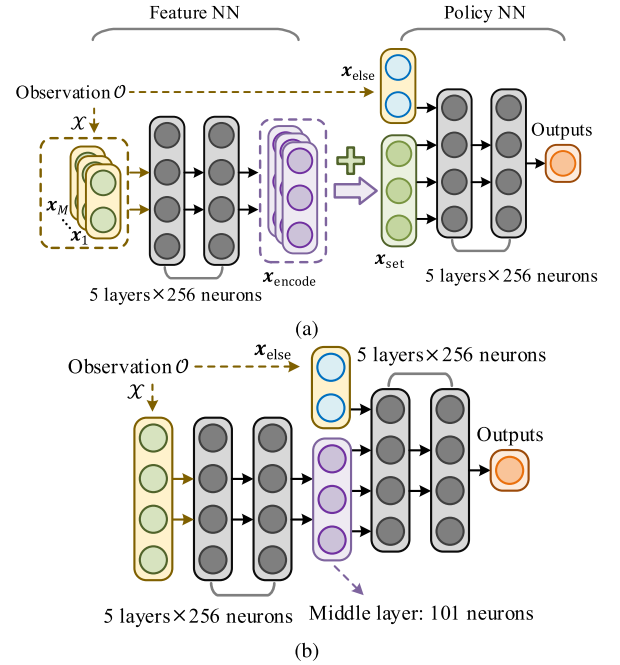


Fig. 5. NN architecture. (a) The NN architecture of the ESC method. (b) The NN architecture of AP and FP methods.

second-order moments are 0.9 and 0.999, respectively. The batchsize is 512 and the learning rate is 8×10^{-5} .

C. Results Analysis

We train 5 different runs of each representation method with different random seeds, and evaluate the policy learning accuracy by calculating the Root Mean Square Error (RMSE) based on $\mathcal{S}_{\text{test}}$. The training curves of benchmark 1 are shown in Fig. 6. In addition to the cases with fixed-size sets (case 1-4 in Table II), we also train an ESC policy based on the samples from the variable-size set (case 5). The learned ESC policy based on case 5 is evaluated when $M = 5$, $M = 10$, $M = 15$ and $M = 20$, respectively, shown as the blue solid lines in Fig. 6.

Fig. 7 and Table III display the final RMSE under each experimental setting. Results show that the proposed ESC method outperforms or matches two baselines in all benchmarks and cases. Among all the cases, the RMSE of the

TABLE III
THE AVERAGE FINAL RMSE ON S_{test} AFTER TRAINING FOR 3000 ITERATIONS. THE MINIMUM RMSE FOR EACH TASK IS BOLDED.
± CORRESPONDS TO A SINGLE STANDARD DEVIATION OVER 5 RUNS

| Benchmark | Numbers of surrounding vehicles | ESC ($M \in [1, N] \cap \mathbb{N}$) | ESC (fixed M) | FP | AP |
|-----------|---------------------------------|--|------------------------------------|--------------------|--------------------|
| 1 | $M = 5$ | 3.78 ± 0.1 | 3.77 ± 0.15 | 7.42 ± 0.13 | 8.5 ± 0.07 |
| | $M = 10$ | 3.6 ± 0.06 | 4.29 ± 0.08 | 7.68 ± 0.07 | 9.35 ± 0.06 |
| | $M = 15$ | 3.51 ± 0.02 | 4.6 ± 0.08 | 8.42 ± 0.17 | 10.36 ± 0.41 |
| | $M = 20$ | 4.19 ± 0.05 | 5.02 ± 0.06 | 9.04 ± 0.06 | 10.93 ± 0.06 |
| 2 | $M = 5$ | 36.87 ± 0.29 | 30.69 ± 0.36 | 53.63 ± 0.01 | 55.83 ± 0.02 |
| | $M = 10$ | 31.83 ± 0.57 | 27.76 ± 0.15 | 56.42 ± 0.05 | 60.14 ± 0.36 |
| | $M = 15$ | 30.15 ± 0.67 | 29.97 ± 1.05 | 54.18 ± 0.41 | 56.25 ± 0.12 |
| | $M = 20$ | 32.6 ± 0.94 | 33.9 ± 1.08 | 51.58 ± 0.19 | 53.56 ± 0.46 |
| 3 | $M = 5$ | 12.46 ± 0.73 | 10.98 ± 0.19 | 42.31 ± 1.11 | 57.56 ± 1.21 |
| | $M = 10$ | 5.56 ± 0.19 | 7.62 ± 0.11 | 33.09 ± 1.05 | 43.82 ± 0.92 |
| | $M = 15$ | 3.82 ± 0.17 | 6.13 ± 0.16 | 29.59 ± 0.23 | 44.0 ± 0.34 |
| | $M = 20$ | 6.77 ± 0.62 | 5.81 ± 0.34 | 31.6 ± 0.87 | 44.94 ± 1.43 |
| 4 | $M = 5$ | 5.96 ± 0.17 | 4.42 ± 0.36 | 10.82 ± 0.14 | 12.28 ± 0.07 |
| | $M = 10$ | 4.33 ± 0.14 | 4.7 ± 0.15 | 9.4 ± 0.21 | 10.19 ± 0.05 |
| | $M = 15$ | 3.8 ± 0.08 | 4.46 ± 0.26 | 8.3 ± 0.37 | 9.19 ± 0.1 |
| | $M = 20$ | 3.89 ± 0.11 | 4.72 ± 0.07 | 8.07 ± 0.2 | 8.53 ± 0.08 |
| 5 | $M = 5$ | 5.57 ± 0.08 | 3.95 ± 0.17 | 18.59 ± 0.31 | 24.79 ± 0.47 |
| | $M = 10$ | 2.88 ± 0.08 | 2.39 ± 0.07 | 16.93 ± 0.24 | 25.96 ± 0.35 |
| | $M = 15$ | 2.2 ± 0.06 | 1.89 ± 0.1 | 14.82 ± 0.3 | 24.16 ± 0.46 |
| | $M = 20$ | 2.1 ± 0.07 | 1.47 ± 0.03 | 13.72 ± 0.21 | 23.26 ± 0.39 |
| 6 | $M = 5$ | 40.88 ± 1.33 | 43.02 ± 3.11 | 66.01 ± 1.78 | 64.97 ± 4.91 |
| | $M = 10$ | 35.52 ± 0.28 | 42.9 ± 2.17 | 219.57 ± 16.9 | 355.9 ± 6.21 |
| | $M = 15$ | 43.59 ± 1.28 | 56.56 ± 0.87 | 349.63 ± 3.82 | 679.42 ± 25.02 |
| | $M = 20$ | 59.64 ± 1.56 | 62.02 ± 7.02 | 508.31 ± 17.91 | 832.74 ± 5.49 |

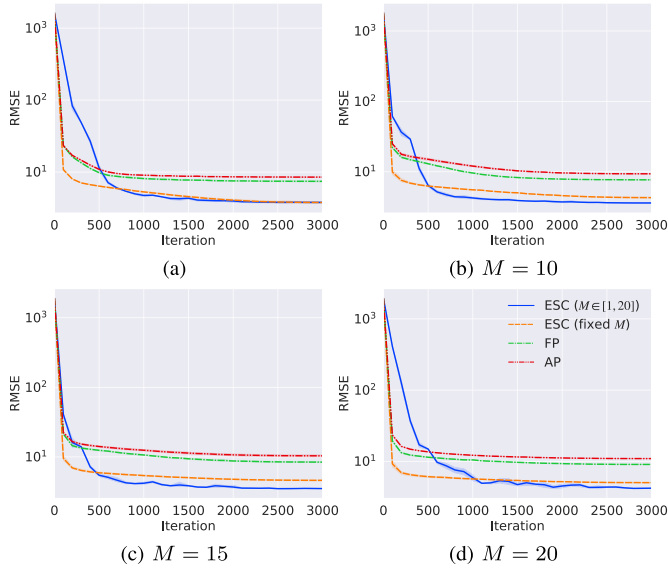


Fig. 6. Training curves of benchmark 1. The solid lines correspond to the mean RMSE and the shaded regions correspond to 95% confidence interval over 5 runs. (a) $M = 5$. (b) $M = 10$. (c) $M = 15$. (d) $M = 20$.

FP method is 20.7% lower than that of the AP method on average. This is because the predetermined order o helps to reduce the sample complexity. Compared with the AP and FP methods, ESC with fixed M achieves an average error reduction of 62.2% and 67.5%, respectively. On the one hand, it is obvious that when considering the same number of surrounding vehicles ($M = 5, 10, 15$, or 20), the performance

of ESC (fixed M) is much better than AP and FP. This indicates that ESC is more suitable to represent the surrounding vehicles set \mathcal{X} due to its permutation-invariance property and continuity. Compared with FP, ESC also eliminates the requirement of manually designed sorting rules. On the other hand, the learning accuracy of ESC with variable-size sets (ESC ($M \in [1, 20]$)) is comparable to that with fixed set size (ESC (fixed M)). Therefore, it suggests that the ESC method is capable of representing variable-size sets, thereby eliminating the burden of training different approximation NNs for scenarios with different numbers of surrounding vehicles. To conclude, experimental results indicate that the proposed ESC method improves the representation ability of driving observation.

D. Future Work

In this paper, the policy NN and feature NN are updated to approximate the designed target policy in Table I under the supervised learning framework. The proposed ESC approach has potential to be adopted in actual state representation applications of autonomous driving based on supervised learning or RL. To this end, we first need to normalize all indicators of surrounding vehicles to the same range. Then, for supervised learning based decision-making, we can learn both policy and feature NNs using the labeled data $\{\mathcal{O}_i, a_i^*\}$, where a^* is the labeled action. For RL-based decision-making, we can iteratively find nearly optimal policy and feature NNs using the samples $\{\mathcal{O}_t, a_t, r_t, \mathcal{O}_{t+1}\}$ collected from the interaction between the ego vehicle and environments, where r represents the reward function. In the future, we will focus on the

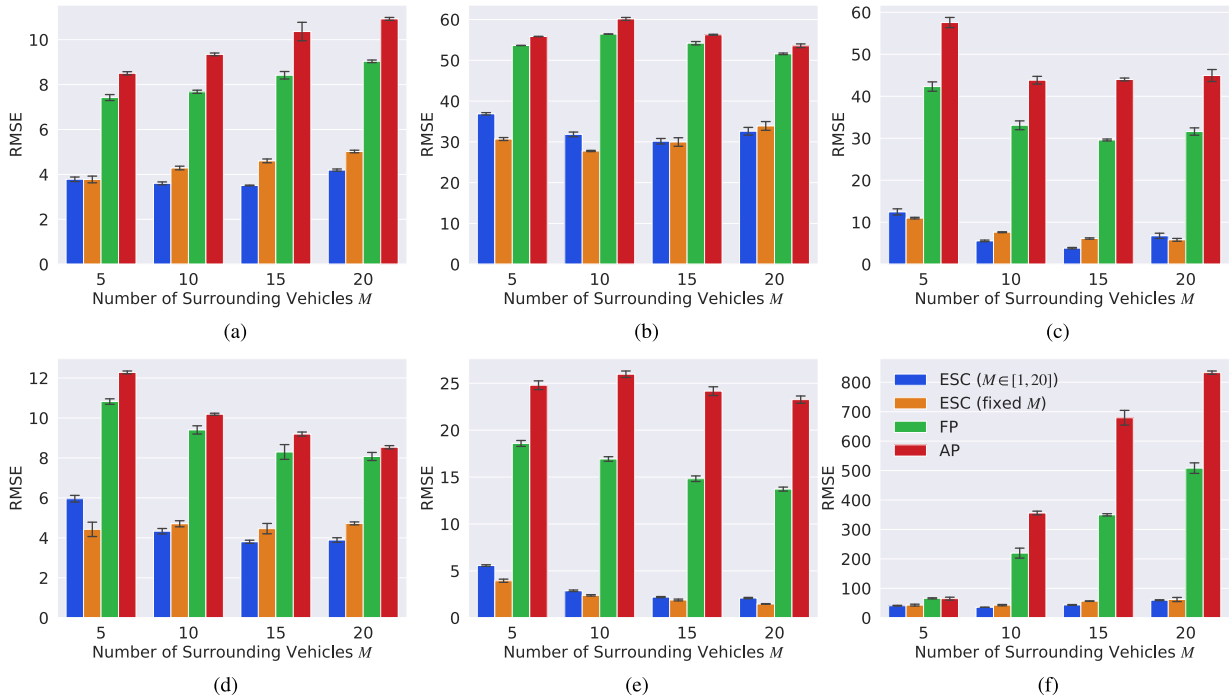


Fig. 7. Average final RMSE of different representation methods on all benchmarks. (a) Benchmark 1. (b) Benchmark 2. (c) Benchmark 3. (d) Benchmark 4. (e) Benchmark 5. (f) Benchmark 6.

applications of ESC in supervised learning based or RL-based autonomous driving.

V. CONCLUSION

In this paper, we first analyze the dimension sensitivity and permutation sensitivity issues faced by existing AP and FP representation methods. Due to dimension sensitivity, T2E methods based on AP or FP representation are only valid when the number of surrounding vehicles is fixed. Due to the permutation sensitivity, AP and FP methods suffer from high sample complexity and policy discontinuity respectively. Both dimension sensitivity and permutation sensitivity will damage the policy learning accuracy and limit the applicability of T2E decision-making in different driving scenarios.

To overcome this problem, we propose the ESC state representation method to describe the environment observation for decision-making in autonomous driving. The proposed ESC method employs a feature NN to encode the real-valued feature of each surrounding vehicle into an encoding vector, and then adds these vectors to obtain the representation vector of the set of surrounding vehicles. By concatenating the set representation with other variables, such as indicators of the ego vehicle and road, we achieve a fixed-dimensional and permutation-invariance state representation. We have further proved that there exists an over-parameterized feature NN such that the ESC state representation is injective if the output dimension of the feature NN is greater than the number of variables of all surrounding vehicles. Besides, by taking the ESC representation as policy inputs, the nearly optimal feature NN and policy NN can be found by simultaneously optimizing them using gradient-based updating. Experiments demonstrate the proposed ESC method improves the representation ability

of driving observation, leading to a reduction of 62.2% in policy learning error compared with the fixed-permutation representation method.

ACKNOWLEDGMENT

The authors are grateful to the Editor-in-Chief, the Associate Editor, and anonymous reviewers for their valuable comments.

REFERENCES

- [1] C. Katrakazas, M. Qudus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 416–442, Nov. 2015.
- [2] M. Montemerlo *et al.*, "Junior: The Stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, 2008.
- [3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Boston, MA, USA: MIT Press, 2018.
- [4] Y. Bengio, I. Goodfellow, and A. Courville, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2016.
- [5] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun, and B. Cheng, "Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 8, 2021, doi: [10.1109/TNNLS.2021.3082568](https://doi.org/10.1109/TNNLS.2021.3082568).
- [6] J. Duan, Z. Liu, S. E. Li, Q. Sun, Z. Jia, and B. Cheng, "Adaptive dynamic programming for nonaffine nonlinear optimal control problem with state constraints," *Neurocomputing*, early access, Oct. 2021, doi: [10.1016/j.neucom.2021.04.134](https://doi.org/10.1016/j.neucom.2021.04.134).
- [7] D. A. Pomerleau, "Alvin: An autonomous land vehicle in a neural network," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Denver, CO, USA: MIT Press, 1989, pp. 305–313.
- [8] M. Bojarski *et al.*, "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*.
- [9] M. Bojarski *et al.*, "Explaining how a deep neural network trained with end-to-end learning steers a car," 2017, *arXiv:1704.07911*.
- [10] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, 2016, pp. 1–14.

- [11] M. Jaritz, R. de Charette, M. Toromanoff, E. Perot, and F. Nashashibi, "End-to-end race driving with deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2070–2075.
- [12] Y. Lecun, E. Cosatto, J. Ben, U. Müller, and B. Flepp, "Dave: Autonomous off-road vehicle control using end-to-end learning," DARPA-IPTO, Arlington County, VA, USA, Tech. Rep., 2004.
- [13] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2722–2730.
- [14] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner. (2000). *TORCS, The Open Racing Car Simulator*. Accessed: Jul. 3, 2021. [Online]. Available: <http://torcs.sourceforge.net>
- [15] A. Kendall *et al.*, "Learning to drive in a day," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8248–8254.
- [16] E. Perot, M. Jaritz, M. Toromanoff, and R. D. Charette, "End-to-end driving in a realistic racing game with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 3–4.
- [17] P. Wolf *et al.*, "Learning how to drive in a real world simulation with deep Q-networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 244–250.
- [18] X. Liang, T. Wang, L. Yang, and E. Xing, "CIRL: Controllable imitative reinforcement learning for vision-based self-driving," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 584–599.
- [19] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2034–2039.
- [20] D. Li, Y. Wu, B. Bai, and Q. Hao, "Behavior and interaction-aware motion planning for autonomous driving vehicles based on hierarchical intention and motion prediction," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–8.
- [21] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, "High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2156–2162.
- [22] P. Wang and C.-Y. Chan, "Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.
- [23] P. Wang, C.-Y. Chan, and A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1379–1384.
- [24] P. Wang, H. Li, and C.-Y. Chan, "Continuous control for automated lane change behavior based on deep deterministic policy gradient algorithm," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1454–1460.
- [25] J. Ma, Z. Cheng, X. Zhang, M. Tomizuka, and T. H. Lee, "Alternating direction method of multipliers for constrained iterative LQR in autonomous driving," 2020, *arXiv:2011.00462*.
- [26] J. Duan, S. E. Li, Y. Guan, Q. Sun, and B. Cheng, "Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data," *IET Intell. Transp. Syst.*, vol. 14, no. 5, pp. 297–305, 2020.
- [27] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12597–12608, Nov. 2020.
- [28] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Proc. 31st Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA: MIT Press, 2017, pp. 3391–3401.
- [29] H. Maron, O. Litany, G. Chechik, and E. Fetaya, "On learning sets of symmetric elements," 2020, *arXiv:2002.08599*.
- [30] A. Sannai, Y. Takai, and M. Cordonnier, "Universal approximations of permutation invariant/equivariant functions by deep neural networks," 2019, *arXiv:1903.01939*.
- [31] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Netw.*, vol. 3, no. 5, pp. 551–560, 1990.
- [32] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, vol. 97, Long Beach, CA, USA, Jun. 2019, pp. 242–252.
- [33] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, PMLR, 2019, pp. 1675–1685.
- [34] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.
- [35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015, pp. 1–15.



Jingliang Duan received the B.S. degree from the College of Automotive Engineering, Jilin University, Changchun, China, in 2015, and the Ph.D. degree from the School of Vehicle and Mobility, Tsinghua University, Beijing, China, in 2021. He studied as a Visiting Student Researcher with the Department of Mechanical Engineering, University of California, Berkeley, USA, in 2019. He is currently a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include decision and control of autonomous vehicle, reinforcement learning and adaptive dynamic programming, and driver behaviour analysis.



Dongjie Yu received the B.S. degree from the School of Vehicle and Mobility, Tsinghua University, Beijing, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Vehicle and Mobility, Tsinghua University, Beijing. His research interests include decision-making of autonomous vehicles and reinforcement learning.



Shengbo Eben Li (Senior Member, IEEE) received the M.S. and Ph.D. degrees from Tsinghua University in 2006 and 2009, respectively.

He worked at Stanford University, the University of Michigan, and the University of California, Berkeley. He is currently a Tenured Professor with Tsinghua University. His active research interests include intelligent vehicles and driver assistance, reinforcement learning and distributed control, and optimal control and estimation. He is the author of over 100 journal/conference papers, and the co-inventor of over 20 Chinese patents. He was the recipient of the Best Paper Award in 2014 IEEE ITS Symposium, the Best Paper Award in 14th ITS Asia-Pacific Forum, the National Award for Technological Invention, China, in 2013, the Excellent Young Scholar of NSF China in 2016, and the Young Professorship of Changjiang Scholar Program in 2016. He serves as an Associated Editor for *IEEE Intelligent Transportation Systems Magazine (ITSM)* and *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (TITS)*.



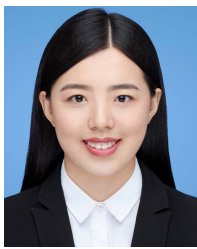
Wenxuan Wang received the B.S. degree in vehicle engineering from the Beijing Institute of Technology, Beijing, China, in 2018. He is currently a member with the State Key Laboratory of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University. His current research interests include decision-making and control of automated vehicles and reinforcement learning algorithms.



Yangang Ren received the B.S. degree from the Department of Automotive Engineering, Tsinghua University, Beijing, China, in 2018, where he is currently pursuing the Ph.D. degree with the School of Vehicle and Mobility. His research interests include decision and control of autonomous driving, reinforcement learning, and adversarial learning.



Bo Cheng received the B.S. and M.S. degrees in automotive engineering from Tsinghua University, Beijing, China, in 1985 and 1988, respectively, and the Ph.D. degree in mechanical engineering from The University of Tokyo, Tokyo, Japan, in 1998. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University, and the Dean of Tsinghua University–Suzhou Automotive Research Institute. He is the author of more than 100 peer-reviewed journal/conference papers and the co-inventor of 40 patents. His active research interests include autonomous vehicles, driver-assistance systems, active safety, and vehicular ergonomics.



Ziyu Lin received the B.S. degree in automotive engineering from China Agricultural University, Beijing, China, in 2017. She is currently pursuing the Ph.D. degree with the School of Vehicle and Mobility, Tsinghua University, Beijing. Her current research interests include model-based reinforcement learning, approximate dynamic programming, and model predictive control and distributed control. She was a recipient of the Best Paper Award at the IEEE 2020 3rd International Conference on Unmanned Systems (ICUS) and the Best Presentation Award on IEEE 2021 International Conference on Computer Control and Robotics.