Towards the Vehicular Metaverse: Exploring Distributed Inference With Transformer-Based Diffusion Model

Gaochang Xie, Zehui Xiong, *Senior Member, IEEE*,
Xinyuan Zhang, *Graduate Student Member, IEEE*,
Renchao Xie, *Senior Member, IEEE*, Yunjie Liu,
and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Generative artificial intelligence (GAI) is emerging as a promising solution for the vehicular metaverse due to its adaptable, high-quality, and multi-modal content generation capabilities. Particularly noteworthy is the recent introduction of the Sora model, a Transformer-based diffusion model, which exhibits exceptional performance in visual scenarios. However, diffusion vision transformer (DViT) models face limitations in terms of device resources, inference latency, and personalized requirements at the edge, despite their practical effectiveness in clouds. In response, we propose a DViT-enabled system to enhance vehicular metaverse services. Our approach involves a distributed DViT inference mechanism where road-side units (RSUs) and vehicles collaborate to execute the diffusion process and generate personalized content within vehicles using local prompts. Additionally, we address users' latency-sensitive service demands by formulating a distributed latency optimization problem that considers bandwidth, computation power, and dynamic positioning of heterogeneous devices. We then propose a value iteration-based distributed inference algorithm capable of adaptively determining optimal inference strategies within resource-constrained vehicular networks. Numerical simulations demonstrate that our approach achieves superior performance in reducing latency and enhancing success rates for inference tasks.**

***Index Terms*—Distributed inference, generative artificial intelligence (GAI), latency optimization, vehicular metaverse.**

I. Introduction

The development of mixed reality (MR), artificial intelligence (AI), and intelligent transportation systems (ITSs) has brought the concept of the vehicular metaverse into focus [1]. The recent emergence of generative AI (GAI) models offers flexible and high-quality content generation capabilities, making them promising for realizing the vehicular metaverse [2]. With GAI, the vehicular metaverse is expected to enable virtual interactions tailored to individual users while maintaining a shared background. In this context, the diffusion model, known for supporting multi-modal prompts and AI-generated content (AIGC), has already shown promise in exploring metaverse applications [3], [4].

The recently released state-of-the-art diffusion model, OpenAI Sora, has garnered significant interest from academia and industry due to its remarkable ability to generate virtual scenarios. Sora adopts a Vision Transformer (ViT)-based architecture, making the Transformer-based diffusion model a promising approach for realizing the vehicular metaverse [5].

However, the current diffusion-ViT (DViT) model demonstrates practical performance in cloud environments but encounters challenges in edge scenarios like vehicular networks. First, DViT struggles with latency-sensitive and computation-intensive content generation due to limitations in the execution and transmission abilities of heterogeneous vehicular devices [6]. Additionally, ensuring privacy protection for content becomes more challenging in dynamic and multi-user vehicular networks [7]. Finally, edge users have more personalized requirements for AIGC generation, potentially increasing the difficulty of flexible inference on heterogeneous vehicles [3].

To facilitate edge GAI and metaverse, the authors in [6] propose a dynamic AIGC service provider selection scheme to achieve both efficient edge network and computation resource utilization and superior quality of generated content. In [3], the authors present a blockchain-empowered framework to manage the lifecycle of edge AIGC products, which implements a multi-weight subjective logic-based reputation scheme to ensure privacy and trustworthiness. In [8], the authors enhance the vehicular metaverse through hierarchical personalized FL, which caters to the personalized requirements of ITS applications. However, while these solutions provide valuable insights into implementing an intelligence-based vehicular metaverse, to our best knowledge, there is no work that has investigated incorporating DViT into vehicular networks and its specific mechanisms.

In this paper, we focus on designing a DViT-enabled ITS and its inference workflow to fill the gap between DViT and vehicular networks. Vehicular metaverse tasks are generally characterized by multimodality, including driving services, in-vehicle applications, and specific services that drivers command [2]. We use the image generation process as a typical representative to demonstrate the generic workflow of the system. This is because (i) all of the above tasks contain requirements for image generation, and (ii) the image-generating process is the smallest unit that can completely demonstrate the workflow of DViT, and other modal content generation (e.g., video and audio) can be regarded as an overlay of this process. Specifically, the mechanism aims to explore the distributed inference executed on rode-side units (RSUs) and vehicles that collaborate with distributed heterogeneous resources to reduce latency. Furthermore, the use of token embedding and public prompts keeps both the user's local data and the intermediate results of inference from being leaked, while local prompts ensure that personalized content is generated locally. On this basis, we formulate the distributed inference latency optimization and propose a value iteration distributed inference (VIDI) method to derive inference strategies. The main contributions of this work are as follows:

- We introduce a DViT-enabled ITS and outline the DViT distributed inference workflow. We also conduct a latency analysis and consider vehicle mobility within the system.

Manuscript received 7 April 2024; revised 17 June 2024; accepted 6 August 2024. Date of publication 13 August 2024; date of current version 19 December 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 92367104, in part by BUPT Excellent Ph.D. Students Foundation under Grant CX2022115. This paper was supported in part by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research & Development Programme. The research was also supported by the Ministry of Education, Singapore, under its SMU-SUTD Joint Grant (22-SIS-SMU-048), and SUTD Kickstarter Initiative (SKI 20210204). The review of this article was coordinated by Prof. Yuan Wu. *(Corresponding author: Renchao Xie.)*

Gaochang Xie is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunication, Beijing 100876, China (e-mail: xiegaochang@bupt.edu.cn).

Zehui Xiong is with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore 487372 (e-mail: zehui_xiong@sutd.edu.sg).

Xinyuan Zhang is with Zhongguancun Laboratory, Beijing 100084, China (e-mail: zhangxy@mail.zgclab.edu.cn).

Renchao Xie and Yunjie Liu are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunication, Beijing 100876, China, and also with Purple Mountain Laboratories, Nanjing 211111, China (e-mail: renchao_xie@bupt.edu.cn; liuyj@chinaunicom.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/TVT.2024.3442292

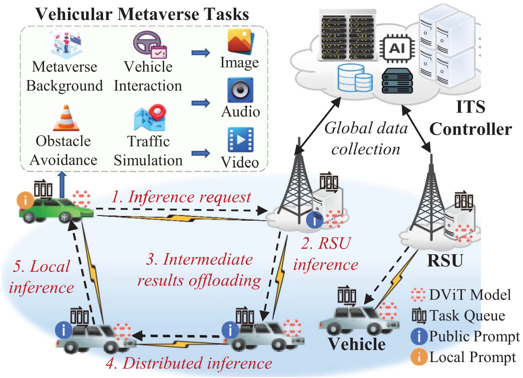

Fig. 1. The DViT-enabled ITS architecture.

- We develop a distributed inference latency optimization framework, taking into account bandwidth, computation power, and vehicle positioning, to meet users' latency-sensitive demands for metaverse services in dynamic and resource-constrained vehicular scenarios. We then propose a VIDI algorithm that dynamically determines optimal inference strategies.
- Numerical simulations demonstrate that our method achieves better performance with lower latency and a higher success rate for inference tasks.

The rest of the paper is organized as follows: Section II presents the system architecture and workflows. Section III introduces the distributed inference latency optimization and the VIDI algorithm. Section IV presents simulation results and analysis. Finally, the paper is summarized in Section V.

II. System Model

In this section, we introduce the DViT-enabled ITS. Then, we offer detailed descriptions of the distributed DViT inference workflow, alongside latency and vehicle mobility analysis.

A. System Architecture

As depicted in Fig. 1, the DViT-enabled ITS consists of three main components: vehicles, RSUs, and ITS controllers. Vehicles and RSUs, equipped with DViT models and heterogeneous computation power, support vehicular metaverse services such as background construction, vehicle interaction, obstacle avoidance, and traffic simulation, achieved through DViT diffusion for generating multi-modal content including image, audio, and video [2]. In this paper, to reflect the generalizability of the proposed methodology, we illustrate the distributed inference process with the example of image diffusion as the smallest unit of the DViT procedure. Moreover, we denote the set of vehicles as $\mathcal{D} = \{d_1, d_2, \ldots, d_D\}$, and the RSU set as $\mathcal{R} = \{r_1, r_2, \ldots, r_M\}$. All vehicles connected to a fixed RSU r_m are represented as $\mathcal{V} = \{v_1, v_2, \ldots, v_N\} \in \mathcal{D}$. Let $\mathcal{W} = \{w_1, w_2, \ldots, w_N\}$ and $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$ denote the available network bandwidth and computation power of vehicles in $\mathcal{V}$, respectively. Additionally, ITS controllers manage edge data collection to update a global status view, providing necessary information to RSUs for devising distributed inference strategies [9].

B. DViT Inference Workflow

1) Distributed DViT Inference: The DViT inference process involves a *reverse diffusion* process $p(\mathbf{x}_T)\prod p^t(\mathbf{x}_{t-1}|\mathbf{x}_t)$, aiming to approximate the original data distribution $q(\mathbf{x}_0)$ with $p(\mathbf{x}_0)$ from the noise-filled data [4] as follows:

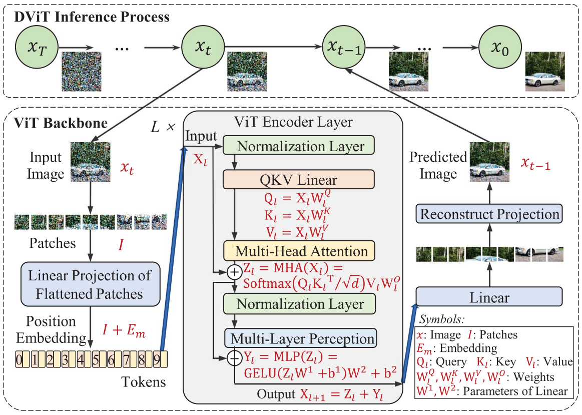

Fig. 2. DViT inference workflow.

$$p(\mathbf{x}_0) = \int \left[p(\mathbf{x}_T)\prod p^t\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)\right] d\mathbf{x}_{1:T}. \tag{1}$$

Here, during diffusion steps $t \sim \mathcal{U}(\mathrm{T} = \{1, 2, 3, \ldots, T\})$, the diffusion model is trained to learn the reverse process of a fixed-length Markov chain, where $\mathbf{x}_1$ to $\mathbf{x}_T$ are latent variables. Consequently, as depicted in Fig. 2, the inference process gradually denoises a noised image from step T to 1 and restores the real data x_0 by sequential sampling from the reverse distribution. Additionally, in the training process, the diffusion model also involves a *forward diffusion* process that continuously adds noise to the data

$$q\left(\mathbf{x}_{1:T}|\mathbf{x}_0\right) = \prod q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right). \tag{2}$$

In this paper, we focus on the sampling process of reverse diffusion associated with the inference procedure.

In our DViT-enabled ITS, all vehicles and RSUs are equipped with the same DViT model. Differing from the conventional diffusion model with a U-Net backbone, DViT employs a ViT architecture as a backbone, similar to the recent state-of-the-art OpenAI Sora model [5]. For generalization, we utilize the standard ViT structure as the backbone of the DViT model. The distributed DViT inference process includes five steps outlined in Fig. 1: (i) A vehicle $v'_n \in \mathcal{V}$ sends a metaverse task request with size D_{req} to its connected RSU $r'_m \in \mathcal{R}$. (ii) From the first inference step, after each step completion, the system determines the next execution device. Here, r'_m can execute some inference steps and then (iii) offload intermediate results to its connected vehicles in $\mathcal{V}$. For simplicity, we use set $\mathcal{O} = \{o_1, \ldots, o_k, \ldots, o_K\}$ to denote the execution device of each diffusion step, where k from 1 to K represents diffusion index T to 1, thus $|K| = |T|$, $o_k \in \mathcal{V}' = \mathcal{V} \cup \{r'_m\}$, and r'_m is designated as v_0 in $\mathcal{V}'$. We introduce a constraint $o_{k+1} \neq r'_m$, if $o_k \neq r'_m$, to prevent vehicles from transmitting intermediate results back to r'_m. Here, w_0 and c_0 represent the bandwidth and computation power of r'_m, respectively. Thus, we define $\mathcal{W}' = \mathcal{W} \cup \{w_0\}$ and $\mathcal{C}' = \mathcal{C} \cup \{c_0\}$. (iv) The distributed DViT inference utilizes public prompts generated by RSU r'_m to outline a background of the required content. The public prompts are abstracted from the common features contained in a set of local prompts via semantic analysis, which can simultaneously generate backgrounds for similar inference tasks for efficiency [6]. (v) Finally, the remaining steps are offloaded to the user v'_n for local inference using local prompts explicitly describing the required content, i.e., $o_{k+1} = v'_n$, if $o_k = v'_n$. This mechanism addresses the personalized and privacy needs of users by ensuring that the user's personalized intent for generating detailed content exists only in the local prompts. Thus, the intent is not present in the intermediate results and public prompts used for generating the background.

2) ViT Backbone Structure: As illustrated in Fig. 2, at each diffusion step, the ViT backbone transforms a high-dimensional image (i.e., the intermediate result generated by the previous step) into low-dimensional tokens. Specifically, ViT reshapes the input image $x_t \in \mathbb{R}^{H \times W}$ into a sequence of flattened patches $I \in \mathbb{R}^{n \times P^2}$ [10]. Here, (H, W) and (P, P) denote the resolution of the image and patches, respectively. $n = \frac{HW}{P^2}$ denotes the number of patches (i.e., sequence length for the transformer layers). Subsequently, position embeddings are added to patches, incorporating positional information and expanding the number of tokens to $n' = n + 1$. The resulting sequence $\mathbf{X}_l \in \mathbb{R}^{n' \times d}$ serves as the input to the encoder layers, where d represents the ViT dimension.

Following sequential processing by L encoder layers, the ViT backbone reshapes the image x_{t-1} through linear and reconstruct projection modules. Note that this process is executed once during each inference step, and we will delve into the internal computation process in Section II-C2.

C. Distributed Inference Latency Analysis

1) Communication Model: We assume the wireless channels to be independent and identically distributed (i.i.d.) block fading, i.e., the channel remains static within each time slot γ but varies from one to another. When a device o_{k-1} transmits intermediate results to the next device o_k, we denote p_{k-1} as the transmit power of o_{k-1} and $h_{k-1,k}$ as the channel gain. Thus, the transmit rate can be represented as follows:

$$B_{k-1,k}(\gamma) = w_{k-1}(\gamma)\log_2\left(1+\frac{p_{k-1} \cdot h_{k-1,k}(\gamma)}{\sigma^2 + \xi(\gamma)}\right), \ \forall o_k \neq o_{k-1}, \tag{3}$$

where o^2 represents the noise power and ξ denotes the inter-cell interference power [9]. Thus, the latency for o_k to receive intermediate results is as follows:

$$T_k^{comm} = \begin{cases} \frac{n' \cdot \mu}{B_{r'_m,k}(\gamma)} & \text{if } k = 1, \ \forall o_k \neq r'_m \\ 0 & \text{if } \forall o_k = o_{k-1} \\ \frac{n' \cdot \mu}{B_{k-1,k}(\gamma)} & \text{otherwise, } \ \forall o_k \neq o_{k-1} \end{cases} \tag{4}$$

where μ is the average size of an embedding token in $\mathbf{X}_l$.

2) Computation Model: The frequent matrix multiplication within the DViT inference results in intensive computation consumption compared to other processes such as image flattening, normalization, and reconstruct projection. As a fundamental mathematical principle, the matrix multiplication between $\mathbf{A} \in \mathbb{R}^{a' \times b'}$ and $\mathbf{B} \in \mathbb{R}^{b' \times c'}$ costs $2a'b'c'$ floating-point operations of computation power. Consequently, we can calculate the computation cost involved in the ViT backbone workflow depicted in Fig. 2. Note that the weight parameters used in ViT include $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V \in \mathbb{R}^{d \times d_a \cdot n_a}$, $\mathbf{W}_l^o \in \mathbb{R}^{d_a \cdot n_a \times d}$, $\mathbf{W}^1 \in \mathbb{R}^{d \times d_f}$, $b^1 \in \mathbb{R}^{d_f}$, and $\mathbf{W}^2 \in \mathbb{R}^{d_f \times d}$, $b^2 \in \mathbb{R}^d$, where d_a represents the dimension of one attention head, n_a denotes the number of attention heads, and d_f is the number of neurons in the multi-layer perception (MLP) layer [10].

As shown in Fig. 2, for each encoder layer, the QKV linear process involves three matrix multiplications and costs $3 \times 2dn'd_an_a$ computation power. Then, a *softmax* function is applied by the multi-head attention (MHA) mechanism, incurring a cost of $2n'^2d_an_a + 2n'^2d_an_a + 2dn'd_an_a$ computation power. Finally, the MLP process is executed using two linear layers and a Gaussian error linear units (GELU) function, with a computation cost of $2dn'd_f + 2dn'd_f$. Overall, the computation volume required by one ViT encoder layer can be calculated as follows:

$$D = 6dn'd_an_a + \left(4n'^2d_an_a + 2dn'd_an_a\right) + 4dn'd_f. \tag{5}$$

Given that a ViT backbone comprises L encoder layers, the computation volume required by one diffusion step is $L \cdot D$. Consequently, we can express the computation time of the diffusion step executed on o_k as $T_k^{comp} = \frac{L \cdot D}{c_k}$, where c_k denotes the computation power of o_k. In practical applications, it is necessary to deploy appropriate DViT models and pre-process images to limit their size according to the requirements of specific services and the available processing capacity of vehicles to avoid too many layers of the model and too large a volume of input data, resulting in an excessive computational load.

3) Queuing Model: In general ITSs, RSUs and vehicles undertake various services simultaneously. When an inference step is scheduled to o_k, it must wait for the services already queued. We assume task queues operate in a first-in-first-out (FIFO) manner, and the task arrival rate β_k follows a Poisson distribution while the average task execution rate η_k follows an exponential distribution [11]. The queuing time of o_k is

$$T_k^{queu} = \frac{\beta_k}{\eta_k\left(\eta_k - \beta_k\right)}. \tag{6}$$

Above all, the latency of each inference step executed on o_k can be calculated as follows:

$$T_k = T_k^{comm} + T_k^{comp} + T_k^{queu}. \tag{7}$$

Note that each diffusion step must be completed within the maximum one-step execution time $T_{\max}$ proposed by users; otherwise, r'_m will reschedule the inference task to another device to prevent potential prolonged congestion.

D. Vehicle Mobility Analysis

In this part, we investigate the impact of vehicle mobility on distributed DViT inference. Let $p_0(x^{r'}, y^{r'})$ denote the location of RSU r'_m, and $\vec{v}_n(\gamma)$ and $\lambda_n(\gamma)$ represent velocity and acceleration at time slot γ of a vehicle v_n. We define the relative speed and distance between v_n and the preceding vehicle v_{n-1} as $\Delta\vec{v}_n(\gamma) = \vec{v}_n(\gamma) - \vec{v}_{n-1}(\gamma)$ and $\varphi_n(\gamma) = x_{n-1}(\gamma) - x_n(\gamma) - \varpi_n$, respectively. Here, ϖ_n represents the length of v_n. Based on the dynamic properties of vehicle motion [12], we express the mobility parameters of v_n in terms of preceding vehicles as follows:

$$\lambda_n(\gamma) = \lambda_{\max}\left[1 - \left(\frac{\vec{v}_n(\gamma)}{\vec{v}_{\max}}\right)^{\delta} - \left(\frac{\varphi^*\left(\vec{v}_n(\gamma), \Delta\vec{v}_n(\gamma)\right)}{\varphi_n(\gamma)}\right)^2\right], \tag{8}$$

where $\lambda_{\max}$ denotes the maximum acceleration, $\vec{v}_{\max}$ represents the vehicle velocity of steady traffic flow, and δ is the sensitivity parameter of drivers, typically ranging from 1 to 5. φ^* denotes the desired distance and is defined as follows:

$$\varphi^*\left(\vec{v}_n(\gamma), \Delta\vec{v}_n(\gamma)\right) = \varphi_{\min} + \vec{v}_n\theta + \frac{\vec{v}_n(\gamma)\,\Delta\vec{v}_n(\gamma)}{2\sqrt{\varphi_{\max}\zeta_{\max}}}, \tag{9}$$

where $\varphi_{\min}$ represents the safe distance, θ denotes the minimum reaction time, and $\zeta_{\max} > 0$ is the maximum deceleration.

The speed and position of v_n can be represented as follows:

$$\vec{v}_n(\gamma) = \vec{v}_n(\gamma - 1) + \lambda_n(\gamma - 1) \cdot |\gamma| \tag{10}$$

$$x_n(\gamma) = x_n(\gamma - 1) + \vec{v}_n(\gamma - 1) \cdot |\gamma| + \frac{1}{2}\varphi_n(\gamma - 1) \cdot |\gamma|^2. \tag{11}$$

Thus, the remaining time during which vehicle v_n is within the coverage of r'_m can be represented as follows:

$$T_{n,m'}(\gamma)=\frac{\sqrt{{d'_m}^2-\left(y^{r'}-y_n(\gamma)\right)^2}\pm\left(x^{r'}-x_n(\gamma)\right)}{|\vec{v_n}(\gamma)|},\qquad(12)$$

where d'_m represents the coverage radius of the RSU r'_m.

The user v'_n must remain within the coverage of r'_m until it receives the required intermediate results to perform the final local inference steps. We define the time consumption of local inference steps as $T^{loc}=\sum T_k, \forall o_k=v'_n$, and $\mathcal{T}$ denotes the total latency of the distributed inference. Therefore, we have

$$\mathcal{T}-T^{loc}\leq T_{n',m'}.\qquad(13)$$

To further formulate the vehicle position model, we divide the urban road into G grids with a fixed length g'. Each vehicle is associated with a grid number in $\mathcal{H}=\{p_1,p_2,\ldots,p_G\}$, indicating its current position. Therefore, $\mathcal{H}'=\mathcal{H}\cup\{p_0\}$ contains the positions of all devices in $\mathcal{V}'$.

III. Problem Formulation and Solution

In this section, we formulate the distributed inference as a Markov decision process (MDP) and propose the VIDI algorithm to minimize inference latency.

A. Distributed DViT Inference Latency Optimization Problem

The optimization objective is to minimize the distributed inference latency, which encompasses the time of the user sending a request to the RSU, as well as the transmission, computation, and queuing latency during inference as follows:

$$\begin{aligned}\min_{\mathcal{O}}\ \ & \mathcal{T}=\frac{D_{req}}{B_{v'_n,r'_m}}+\sum_{k=1}^{K}T_k\\ s.t.\ C1:\ \ & o_{k+1}\neq r'_m,\forall o_k\neq r'_m\\ C2:\ \ & o_{k+1}=v'_n,\forall o_k=v'_n\\ C3:\ \ & T_k\leq T_{\max}\\ C4:\ \ & \mathcal{T}-T^{loc}\leq T_{n',m'}\\ C5:\ \ & f_k\in\{0,1\}\end{aligned}\qquad(14)$$

where $C1$ restricts the inference tasks offloaded to vehicles from being sent back to the RSU for processing. $C2$ ensures that the last remaining inference steps are executed on the user. $C3$ constrains the one-step maximum execution time to prevent potential prolonged congestion. $C4$ ensures that the user remains within the coverage of the RSU until it receives necessary feedback. $C5$ specifies a binary parameter indicating the type of device executing each inference step.

B. MDP Model Formulation

In the DViT-enabled ITS, existing scheduling strategies for subtasks with linear dependencies have significant limitations. Specifically, some assume negligible downlink transmission overhead, which is unsuitable for diffusion processes involving massive image and token transmission [9]. Many machine learning-based methods require extensive prior knowledge, which is difficult to obtain in variable ITSs [13]. Additionally, some works overlook the dynamic impact of inference models on computation and communication resources [11]. To address the above issues, in this paper, we model the distributed DViT inference process as an MDP for our dynamic optimization, a stochastic process evolving over time and characterized by the state space, action space, transition probability, and reward function [13]. On this basis, we solve the optimization problem using the VIDI algorithm tailored for distributed inference processes. VIDI explores optimal solutions in dynamic environments in a stochastic process that is well adapted to the dynamics of computation and communication resources and the time-varying nature of network topologies in ITSs. On the other hand, VIDI does not require prior knowledge from ITSs, nor does it need to ignore the downstream transmission process of the system, making it easy and practical to deploy.

Algorithm 1: Value Iteration Distributed Inference Strategy.

```
Input: $\rho$, $\mathcal{S}$, $\mathcal{A}$, $\mathcal{P}$, and a small positive number $\phi$
Output: Optimal strategy $\pi^*$
 1: Initialize parameter $i=0$ and $v^0(s)=0$
 2: for $s\in\mathcal{S}$ do
 3:    for $a\in\mathcal{A}$ do
 4:       $v^{i+1}(s)=\min_{a\in\mathcal{A}}\{F(s,a)+\rho\sum_{s'\in\mathcal{S}}\mathcal{P}(s'\,|\,s,a)v^i(s')\}$
 5:    end for
 6: end for
 7: if $|v^{i+1}(s)-v^i(s)|>\phi$ then
 8:    $i=i+1$, and go to line 2
 9:    Check for conflicts with constraints
10: else
11:    $\pi^*=\arg\min_{a\in\mathcal{A}}\{v(s)\}$
12: end if
13: return $\pi^*$
```

1) State Space: The successive decision epochs are represented by the sequence $k\in\{1,2,\ldots,K\}$. Within the MDP, a state comprises the bandwidth, computation power, and position grid. We define the total possible states as follows:

$$\mathcal{S}=\mathcal{W}'\times\mathcal{C}'\times\mathcal{H}',\qquad(15)$$

where '×' is the Cartesian product. At each decision epoch, the current state is denoted by $s=(w_n,c_n,p_g), s\in\mathcal{S}$.

2) Action Space: The action space defines all the possible actions. We denote $\mathcal{A}$ as the action set containing all actions that may be selected for each inference step, as follows:

$$\mathcal{A}=\{a=(a_1,a_2,\ldots,a_n,\ldots,a_N)\},\qquad(16)$$

where $a_n\in\{0,1\}$. Here, $a_n=1$ denotes the action that selects the device v_n to perform an inference step, while $a_n=0$ otherwise. Therefore, at each decision epoch, the RSU r'_m will determine an action a based on the current state s.

3) Transition Probability: The transition probability denotes the probability that the current state s could transition to the next state s' by action a, and is derived as follows:

$$\mathcal{P}(s'\,|\,s,a)=P(w'_n\,|\,w_n,a)\,P(c'_n\,|\,c_n,a)\,P(p'_g\,|\,p_g,a),\qquad(17)$$

where each $P(\cdot)$ is defined as the reciprocal of the total number of possible values that can be varied from the current value of the corresponding state parameter by action a.

4) Reward Function: The reward function $F(s,a)$ reflects the immediate reward when performing an action at the current state. Here, we define $F(s,a)=T_k(a)$ according to (7).

C. Distributed Inference Strategy Determining

The value iteration-based inference strategy, i.e., the VIDI algorithm, is shown in Algorithm 1, to solve the latency optimization issue. The value function $v^{\pi}(s)$ denotes the expected rewards. A policy function is represented by π, and Π is the set of all π that can be explored. Thus, we define $v^{\pi}(s)$ as follows:

$$v^{\pi}(s) = \mathbb{E}\left[\sum_{k=1}^{K} \rho^{k-1} F_k(s,a)\right], \tag{18}$$

where $\mathbb{E}[\cdot]$ denotes the expected value, $\rho \in (0,1]$ is the discount factor, and $F_k(s,a)$ represents the immediate reward.

Our objective is to minimize the distributed DViT inference latency by selecting the optimal devices as follows:

$$v(s) = \min_{\pi \in \Pi} v^{\pi}(s). \tag{19}$$

The optimization problem can converge into a Bellman optimality equation expressed by [13]

$$v(s) = \min_{a \in \mathcal{A}} \left\{ F(s,a) + \rho \sum_{s' \in \mathcal{S}} \mathcal{P}(s' \,|\, s,a)\, v(s') \right\}. \tag{20}$$

Algorithm 1 involves calculating rewards for all actions that can be taken for each state in each iteration until convergence, resulting in a polynomial time complexity $O(|\mathcal{S}| \cdot |\mathcal{A}| \cdot \Delta)$, where Δ denotes the maximum number of iterations. Considering the limited number of vehicles served simultaneously by r'_m and its restricted coverage, the iteration number and dimensions of the state space and action space are also significantly restricted. Thus, we consider the complexity of the VIDI algorithm acceptable for distributed DViT inference.

IV. Simulation Results

This section presents numerical simulations to evaluate the effectiveness of our proposed method. We first provide details on the simulation settings. The image resolution remains at 512 × 512, with patches of 16 × 16. The DViT models maintain a dimension of 512, and the default encoder layer number is 6. Each DViT model incorporates 8 attention heads, along with an MLP dimension of 2048. RSUs are positioned statically, equipped with computation power ranging from 15 to 20 GFLOPs and bandwidth from 100 to 200MHz. The coverage radius of an RSU remains at 500 meters, and the road grid length is fixed at 25 meters. Vehicles are randomly located in the urban area, with computation power ranging from 5 to 10 GFLOPs and bandwidth from 50 to 80MHz. The default number of vehicles and inference steps is 20 and 50, respectively. The user latency requirement is set at 0.8s per step, while the noise power remains at -100 dBm.

To comprehensively assess the performance of our method, we compare it with several benchmark methods in simulations: (i) Minimum distance distributed inference (MDDI), which selects vehicles gathered around the RSU in advance; (ii) Maximum bandwidth distributed inference (MBDI), which prioritizes devices with a higher available bandwidth for inference; (iii) Maximum computation power distributed inference (MCPDI), which prioritizes devices based on their computation power; (iv) Vehicular local inference (VLI), which leverages the vehicle with optimal computation power to perform all the inference steps without RSU participation; (v) Stochastically distributed inference (SDI).

Fig. 3 illustrates the latency performance across varying numbers of vehicles participating in distributed inference. MBDI, leveraging devices with superior communication conditions, and MDDI, offloading tasks to nearby vehicles, outperform SDI. MDDI selects vehicles in close proximity to each other for distributed inference, where the communication distance between vehicles is shorter and they are in RSU coverage for a longer period of time, but their computation ability cannot be guaranteed, leading to higher latency. In contrast, MCPDI notably reduces latency, showcasing the computational intensity of DViT inference in dynamic and device-intensive edge scenarios. VLI scheme that utilizes only the vehicle's processing power can overcome some communication bottlenecks, but the loss of the RSU's computation power makes its processing performance less impressive. This demonstrates the importance of joint vehicle-RSU processing for inference tasks. Notably, VIDI optimizes both communication and computation adaptively, reduces latency as the vehicle number increases until it stabilizes, and achieves superior performance across all methods.

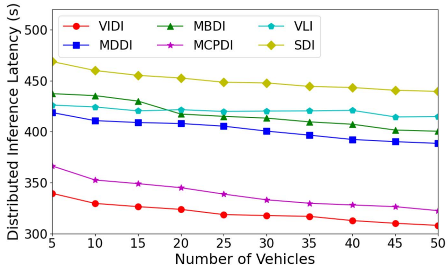

Fig. 3. Inference latency under different numbers of vehicles.

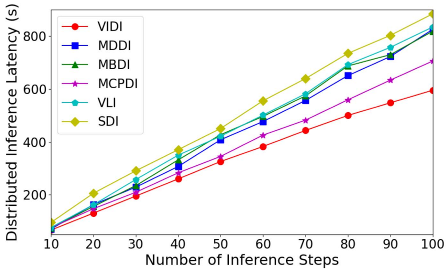

Fig. 4. Inference latency under different numbers of inference steps.

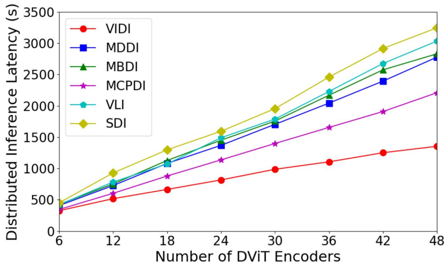

Fig. 5. Inference latency under different numbers of DViT encoders.

Similar trends can be observed in Figs. 4 and 5. In Fig. 4, as the number of inference steps grows, the advantages of VIDI become increasingly pronounced. Meanwhile, in Fig. 5, with the number of encoder layers within the DViT model increasing from 6 layers, as in a conventional transformer encoder, to 48 layers, as in the large-scale ViT-22B model [14], VIDI consistently achieves significantly lower latency, showcasing its capability to dynamically adapt to resource-constrained vehicular environments.

Fig. 6 assesses the stability of performance across all methods with the success rate, defined as the ratio of the tasks meeting both the user's latency requirements and RSU's coverage limits to the total tasks. Remarkably, both MBDI and MDDI methods demonstrate low and unstable inference success rates due to poor computation. VLI

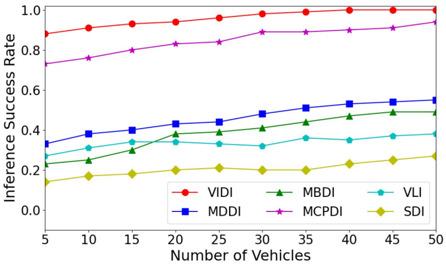

Fig. 6. Inference success rate under different numbers of vehicles.

proves that the instability in latency embodied in local inference makes it difficult to meet the needs of DViT-based services. While MCPDI aims to alleviate this issue, occasional harsh channel conditions pose challenges, leading to occasional failures. In contrast, VIDI adopts a comprehensive approach considering bandwidth, computation, and vehicle mobility, improving success rates as the vehicle number increases until it approaches 1 in dynamic and device-intensive vehicular networks.

V. Conclusion

In this paper, we have studied the implementation of DViT models in vehicular metaverse scenarios. Specifically, we have introduced the DViT-enabled ITS and its internal inference workflow and conducted latency and mobility analysis. Furthermore, we have formulated the distributed inference latency optimization problem and proposed the VIDI method, which considers bandwidth, computation power, and dynamic device positioning to address this challenge. Simulation results have demonstrated the effectiveness and efficiency of our approach in reducing latency and improving success rates.

References

[1] Y. Ren et al., “Connected and autonomous vehicles in web3: An intelligence-based reinforcement learning approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 8, pp. 9863–9877, Aug. 2024.

[2] G. Xie et al., “GAI-IoV: Bridging generative AI and vehicular networks for ubiquitous edge intelligence,” *IEEE Trans. Wireless Commun.*, early access, May 9, 2024, doi: 10.1109/TWC.2024.3396276.

[3] Y. Liu et al., “Blockchain-empowered lifecycle management for AI-generated content products in edge networks,” *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 286–294, Jun. 2024.

[4] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Proc. Annu. Conf. Neural Inf. Proc. Syst.*, Dec. 2020, pp. 6840–6851.

[5] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Dec. 2023, pp. 4172–4182.

[6] H. Du et al., “Enabling AI-generated content services in wireless edge networks,” *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 226–234, Jun. 2024.

[7] Y. Lin et al., “Blockchain-aided secure semantic communication for ai-generated content in metaverse,” *IEEE Open J. Comput. Soc.*, vol. 4, pp. 72–83, 2023.

[8] L. U. Khan, A. Elhagry, M. Guizani, and A. E. Saddik, “Edge intelligence empowered vehicular metaverse: Key design aspects and future directions,” *IEEE Internet Things Mag.*, vol. 7, no. 1, pp. 120–126, Jan. 2024.

[9] C. Tang, X. Wei, C. Zhu, Y. Wang, and W. Jia, “Mobile vehicles as fog nodes for latency optimization in smart cities,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9364–9375, Sep. 2020.

[10] F. Bao et al., “All are worth words: A ViT backbone for diffusion models,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 22669–22679.

[11] K. Li, X. Wang, Q. He, Q. Ni, M. Yang, and S. Dustdar, “Computation offloading for tasks with bound constraints in multiaccess edge computing,” *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15526–15536, Sep. 2023.

[12] C. Zhang and L. Sun, “Bayesian calibration of the intelligent driver model,” *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 8, pp. 9308–9320, Aug. 2024.

[13] L. Zhao et al., “MESON: A mobility-aware dependent task offloading scheme for urban vehicular edge computing,” *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4259–4272, May 2024.

[14] M. Dehghani et al., “Scaling vision transformers to 22 billion parameters,” in *Proc. Int. Conf. Mach. Learn.*, Jul. 2023, pp. 7480–7512.