VLASCD: A Visual Language Action Model for Simultaneous Chatting and Decision Making

Anonymous ACL submission

Abstract

Although current mainstream pre-trained large models, such as LLM models represented by ChatGPT and VLA models represented by OpenVLA, have achieved significant progress in multimodal tasks through a "Multiple-Input, Single-Output" (MISO) architecture. However, our investigation reveals that the MISO architecture exhibits fundamental limitations in "Multiple-Input, Multiple-Output" (MIMO) (e.g., parallel multi-tasks output processing): the architecture generates task mutual exclusion effects, leading to resource contention among different tasks when sharing output channels, and consequently resulting in optimization imbalance and performance. In contrast, human MIMO processing inherently enables concurrent task execution (e.g., while dialogue and decision-making) without interference. Inspired by this, in this work, we propose a unified MIMO training model with parallel multi-tasks output capabilities—the Visual Language Action Model for Simultaneously Chatting and Decision Making (VLASCD). We evaluate the model on the CARLA autonomous driving platform. The results show that, compared to LLM models with MISO dialogue capabilities, reinforcement learning models, and VLA models with MISO decision-making capabilities, VLASCD significantly outperforms existing MISO models in simultaneously handling dialogue generation and decision-making tasks within the MIMO scenario.

1 Introduction

Since ChatGPT's emergence, large language models (LLMs) have become prominent examples of large-scale pre-trained models. Trained on extensive internet text and code, LLMs encode substantial real-world knowledge, enabling superior generalization over traditional AI models—including in-context learning and reasoning abilities (e.g., via chain-of-thought (Wei et al., 2022)). A development trend in the field of large-scale pre-trained models is that their application domains are expanding from tasks like dialogue and text generation to decision-making tasks in the open physical world.

Currently, there are three primary approaches to building large-scale pre-trained models for decision-making in open physical environments. The first method serializes the decision-making process and trains sequence models like Decision Transformers (Chen et al., 2021), treating decisions similarly to text processing. This approach depends heavily on acquiring large-scale, high-quality decision-making datasets. The second strategy employs hierarchical modular systems where LLMs handle high-level planning (Chen et al., 2024; Carta et al., 2023; Hu et al., 2024; Zhou et al., 2024)—decomposing tasks and orchestrating specialized modules or tools. The third paradigm develops end-to-end Vision-Language-Action (VLA) models (Padalkar et al., 2023; Kim et al., 2024) that bypass modular systems by directly generating decisions from multimodal inputs.

In recent years, pre-trained large models have achieved remarkable progress in multimodal tasks, with notable examples including LLMs represented by ChatGPT and VLA models represented by OpenVLA. These models typically adopt a "Multiple-Input, Single-Output" (MISO) architecture, generating a single output from multiple input, and have demonstrated powerful capabilities in tasks such as text generation and image understanding. However, our investigation reveals that in "Multiple-Input, Multiple-Output" (MIMO) scenarios (e.g., parallel multi-tasks output), existing MISO LLM models (Chen et al., 2024; Liu et al., 2023), often fail to generate effective actions, and even if actions are generated, they interfere with dialogue capabilities. Similarly, existing MISO VLA models (Kim et al., 2024; Liu et al., 2024) struggle to generate effective dialogue. The fundamental limitation underlying this investigation is that task interference emerges during parallel processing, where competing tasks generate conflicts in shared output channels, leading to suboptimal resource allocation. This conflict manifests as imbalanced model optimization during training, ultimately leading to significant performance degradation in task-specific objectives. This contrasts sharply with human MIMO processing, which inherently supports non-interfering concurrent execution of tasks (e.g., simultaneous dialogue and decision-making). Inspired by this, we propose a unified MIMO training architecture with parallel multi-task output capabilities—the Visual Language Action Model for Simultaneously Chatting and Decision Making (VLASCD), with validation conducted in autonomous driving CARLA 0.9.10 (Dosovitskiy et al., 2017). Experimental results show that our MIMO architecture's task-adaptive distributed output mapping not only enables efficient multimodal collaboration but also resolves multi-task interference in MISO models.

The main contributions of this work are:

- This paper is the first to reveal that existing MISO models (e.g., LLMs and VLAs) cannot effectively handle MIMO tasks.
- we propose a unified MIMO training architecture with parallel multi-task output capabilities, termed VLASCD. It combines several experimentally validated ideas: (1) a computational module and cost function term for generating continuous action values; (2) an image reconstruction loss term added in the training cost function to ensure the exploitation of rich information from the visual modality data during text generation and decision-making processes; (3) a label smoothing strategy to maintain dialogue capabilities and enhance decision-making.
- The experimental results show that the resulting VLASCD model not only outputs more accurate real-time action decisions compared to the SOTA models but also perfectly retains real-time text-based dialogue functionality.
- We will open source our model, code, and dataset after the reviewing process.

2 Related Work

2.1 LLMs for decision-making

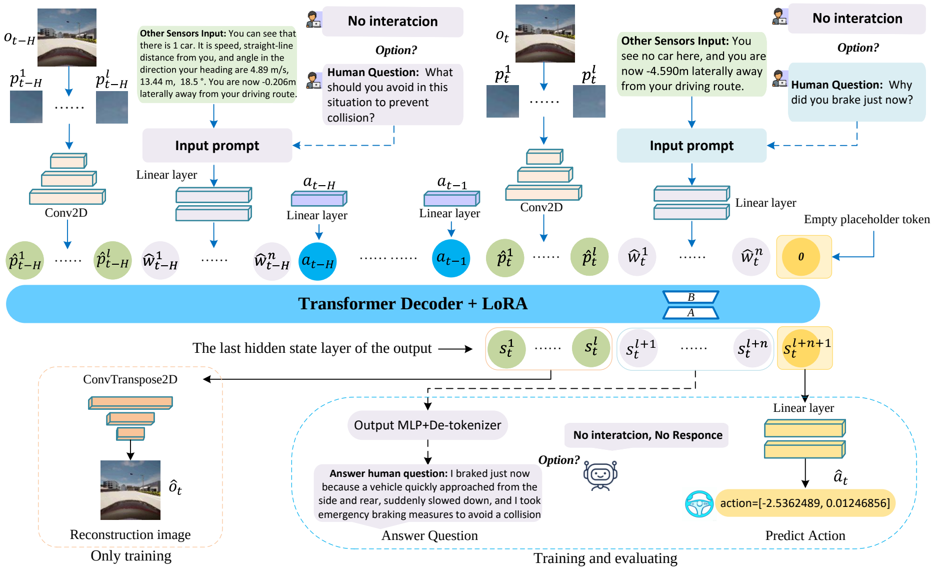
Since (Brown et al., 2020), GPT has emerged as the dominant paradigm for LLMs. Models like GPT-3.5 and GPT-4 (OpenAI, 2023a) demonstrate superior zero-shot generalization and reasoning. The open-source LLaMA series (Touvron et al., 2023a,b) further advanced LLM development. (Wei et al., 2022) introduced chain-of-thought to boost reasoning, while (Yao et al., 2022) proposed ReAct for interleaved reasoning and action generation. Additionally, recent works have used LLMs as components in building hierarchical modular decision-making agents, where they are only used to generate high-level plans and do not directly generate decisions (Ahn et al., 2022; Fu et al., 2023; Carta et al., 2023; Chen et al., 2024; Xu et al., 2024; Sha et al., 2023; Hu et al., 2024; Zhou et al., 2024). The VLASCD model proposed here can be seen as a multimodal GPT model fine-tuned for a downstream application scenario, featured by its capability to simultaneously output action decisions and textual chatting.

2.2 VLA model for decison-making

VLA models process multimodal vision-language-action inputs for embodied decision-making. Unlike conversational LLMs like ChatGPT, VLAs generate control signals for physical agents (e.g., robots) interacting with environments. They excel at instruction-following tasks by combining language understanding, visual perception, and action generation (Huang et al., 2023; Li et al., 2023b; Zhen et al., 2024; Dorka et al.). Compared to deep reinforcement learning (RL) methods, VLA has shown a remarkable performance gain in versatility, flexibility, and generality in complex environments (Padalkar et al., 2023; Brohan et al., 2023; et al, 2024; Team et al., 2024; Li et al., 2023c; Bai et al., 2023; Li et al., 2022, 2023a; Liu et al., 2024; Tan and Bansal, 2019). However, such VLA models represented by RT-X (Padalkar et al., 2023) and OpenVLA (Kim et al., 2024), typically discretize continuous action spaces into fixed intervals. This action discretization raises significant limitations for them to deal with fine-grained continuous actions that are required for capturing nuanced operations necessary for some complex tasks.

2.3 LLMs for MIMO

Existing multitasking approaches often rely on task-specific designs, lacking collaborative optimization and increasing computational costs (Geng et al., 2022; Liu et al., 2023; Ouyang et al., 2022). While recent work proposes unified frameworks like multitask fine-tuning using CGC LoRA for



Other Sensors Input: You can see that there is 1 car. It is speed, straight-line distance from you, and angle in the direction your heading are 4.89 m/s, 13.44 m, 18.5 °. You are now -0.206m laterally away from your driving route.

Human Question: What should you avoid in this situation to prevent collision?

Other Sensors Input: You see no car here, and you are now -4.590m laterally away from your driving route.

Human Question: Why did you brake just now?

Answer human question: I braked just now because a vehicle quickly approached from the side and rear, suddenly slowed down, and I took emergency braking measures to avoid a collision

action=[-2.5362489, 0.01246856]

Figure 1: An overview of VLASCD framework. First, we process expert dataset images with their text descriptions and action values through linear mappers to get feature representations. These features are combined in a set order and input to the transformer. Finally, in the LoRA-tuned model's last layer, we reconstruct sensor outputs (training only), respond to queries, and map final actions (both training and evaluation).

LLMs (Song et al., 2024), they remain MISO architectures and fail to acheive MIMO output.

3 Methodology

In this section, we present how to build VLASCD in detail, including the model architecture and the training procedure, with a focus on the loss designs in the last output hiddens layer. An overview of VLASCD is illustrated in Figure 1. To begin with, we present the problem setting of our concern.

3.1 Problem Setting

We consider a multimodal setting similar as (Xiao et al., 2020), wherein, at each time step t, upon the agent performs an action a_t, the environment returns visual and textual modalities, denoted by $\{o_t, \hat{w}_t\}$. Our objective is to build a generative model $\pi(\hat{a}_t, \hat{w*}_t | o_{t-H}, \hat{w}_{t-H}, a_{t-H}, ..., o_t, \hat{w}_t)$, which can generate both high-quality action decisions $\hat{a}_t$ and text responses $\hat{w*}_t$, given a sequence of historic trajectories. H denotes the length of the context.

3.2 Model Architecture

Our model supports three different input modalities: text, image, and numeric vector. We use Llama-7b (Touvron et al., 2023b) as the backbone model, and encode textual inputs by its pre-trained embedding layers. To encode the visual inputs, we follow the standard practice used in visual language models (VLMs) (Liu et al., 2024) and VLAs (Kim et al., 2024). Specifically, we first segment each input image o_t into L patches $p_l, l = 1, \dots, L,$, then train a 2D convolution network that directly maps the patches to the vector space. In addition, to deal with the input of the action value, we train a multi-layer perceptron (MLP) module that encodes the action values to the vector space. Finally, We concatenate encoded embeddings of all modalities together to form a sequence of embedded trajectory τ at time t as follows:

$$\begin{aligned}\tau_t = \{&(\hat{p}^1_{t-H}, \dots, \hat{p}^L_{t-H}), (\hat{w}^1_{t-H}, \dots, \hat{w}^n_{t-H}),\\ &a_{t-H}, \dots, (\hat{p}^1_t, \dots, \hat{p}^l_t), (\hat{w}^1_t, \dots, \hat{w}^n_t)\} \quad (1)\end{aligned}$$

where $\hat{p}^i_t$ and $\hat{w}^j_t$ denote the embeddings of i-th patch for visual observation and j-th token for textual observation at at time t, respectively.

During the inference stage, the transformer backbone in VLASCD generates the hidden embeddings $s^{l+1}_t, \cdots, s^{l+n+1}_t$ as shown in Figure 1, then these embeddings are decoded into the outputs of different modalities. Specifically, VLASCD supports two different output modalities: text for chatting and numeric vector for action-level decision making. For the chatting part, we use the pre-trained output MLP layers and tokenizer of the Llama-7b model to generate texts. For action decision-making, our model generates one more embedding vector after the "< EOS >", an empty placeholder token. Unlike previous work like OpenVLA (Kim et al., 2024) and RT-X (Brohan et al., 2023), in which action prediction is formalized as a token generation task by splitting the action space into discrete action bins, we train an action head consisting of multiple MLP modules. This action head directly maps the output embedding to action values. We empirically find that using our approach leads to better performance compared to discretizing action values.

3.3 Training Procedure

We fine-tune the transformer backbone with LoRA (Hu et al., 2021) and train the image encoding, text encoding, action encoding, and decoding modules using an offline dataset D_{expert} containing demonstrated driving trajectories with question-answer pairs. The model learns to predict control actions and answer driving-related questions such as "*Summarize the current driving scenario*". An auxiliary image reconstruction task is introduced where a transposed convolution layer reconstructs input image patches from the output embeddings $s_t^1, \cdots, s_t^l$ to improve feature learning. The training objective consists of three loss terms: text generation, action prediction, and image reconstruction with decoder parameters ϕ, while θ represents all other trainable parameters.

Text Generation. In our experiment, we found that merely replacing specific numerical values in the translation template (Chen et al., 2024) results in minimal representational differences caused by the sequential nature of the data, making the phenomenon of model overfitting easy to happen if we use the conventional cross-entropy loss for text generation. Refer to Appendix A.8 for details. To mitigate this, we use the label smoothing technique to regularize the training process (Szegedy et al., 2016). Specifically, the hard label for token w_i is smoothed by assigning a small portion of the probability mass to incorrect classes:

$$q_i^k = \begin{cases} 1-\epsilon & \text{if } k = y_i, \\ \frac{\epsilon}{K-1} & \text{otherwise,} \end{cases} \tag{2}$$

where ϵ is the smoothing factor and K is the number of total classes, i.e., vocabulary size. That is to say, the loss item for text generation we finally use is:

$$\mathcal{L}_{\text{language}}(\theta) = \frac{1}{N}\sum_i \sum_k q_i^k \log p(k|\tau^{:i-1}, \theta), \tag{3}$$

where $\tau^{:i-1}$ denotes the input token sequence before position i, used for predicting token i. N denotes the maximum padding length to unify the input text.

Action Prediction. To directly predicts continuous action values instead of discrete action bins, we train our model with a mean square error (MSE) loss between the ground-truth action value a_t and the predicted value, as follows:

$$\mathcal{L}_{\text{action}}(\theta) = \frac{1}{T}\sum_t^T \frac{1}{D}\sum_d^D [(a_t^d - \pi(\tau_t, \theta))^2] \tag{4}$$

where D denotes the dimension of the action space. In our experiments, the action dimension is 2, corresponding to the acceleration and steering of the vehicle, respectively.

Image Reconstruction. To better leverage the rich environmental information in visual data while avoiding information loss during training with limited data, we introduce an auxiliary image reconstruction task. This provides additional supervision for the visual modality by using a 2D transposed convolution layer f_ϕ to reconstruct image patches from their corresponding embeddings. The reconstruction loss is computed as the pixel-wise Euclidean distance between original and reconstructed patches, as follows:

$$\mathcal{L}_{\text{image}}(\theta, \phi) = \frac{1}{L}\sum_l^L \text{MSE}(o_t, f_\phi(\pi(g_\theta(\tau_t^{:p_t^l}), \theta))) \tag{5}$$

where o_t is the input image, and $\tau_t^{:p_t^l}$ is the input sequence up to this patch token, and g_θ represents a trainable 2D convolutional network that directly maps image patches $p_t^1, \cdots, p_t^l$ to the language embedding space $\hat{p}_t^1, \cdots, \hat{p}_t^l$.

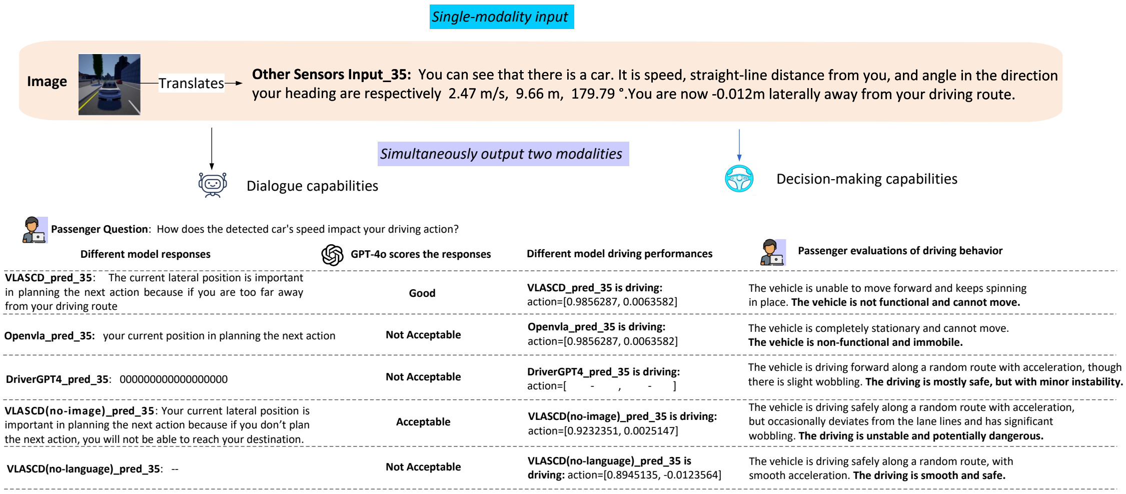
Training Loss Function. In summary, our training loss function is defined as follows:

$$\mathcal{L} = \alpha_1 \mathcal{L}_{\text{language}} + \alpha_2 \mathcal{L}_{\text{action}} + \lambda \mathcal{L}_{\text{image}} \tag{6}$$

where $\alpha_1, \alpha_2, \lambda$ are the weight hyperparameters of three components.

4 Experiments

In this section, we validate through experiments on the autonomous driving simulation platform CARLA that VLASCD can make fine-grained action decisions while maintaining dialogue functionality. Our experiments analyze: (1) the effects of different loss components on model performance, and (2) how textual data quality influences driving decisions.


Single-modality input
Image
Translates
Simultaneously output two modalities
Dialogue capabilities
Decision-making capabilities

Other Sensors Input_35: You can see that there is a car. It is speed, straight-line distance from you, and angle in the direction your heading are respectively 2.47 m/s, 9.66 m, 179.79 °.You are now -0.012m laterally away from your driving route.

Passenger Question: How does the detected car's speed impact your driving action?

Different model responses	**GPT-4o scores the responses**	**Different model driving performances**	**Passenger evaluations of driving behavior**
VLASCD_pred_35: The current lateral position is important in planning the next action because if you are too far away from your driving route	**Good**	**VLASCD_pred_35 is driving:** action=[0.9856287, 0.0063582]	The vehicle is unable to move forward and keeps spinning in place. **The vehicle is not functional and cannot move.**
Openvla_pred_35: your current position in planning the next action	**Not Acceptable**	**Openvla_pred_35 is driving:** action=[0.9856287, 0.0063582]	The vehicle is completely stationary and cannot move. **The vehicle is non-functional and immobile.**
DriverGPT4_pred_35: 000000000000000000	**Not Acceptable**	**DriverGPT4_pred_35 is driving:** action=[- , -]	The vehicle is driving forward along a random route with acceleration, though there is slight wobbling. **The driving is mostly safe, but with minor instability.**
VLASCD(no-image)_pred_35: Your current lateral position is important in planning the next action because if you don't plan the next action, you will not be able to reach your destination.	**Acceptable**	**VLASCD(no-image)_pred_35 is driving:** action=[0.9232351, 0.0025147]	The vehicle is driving safely along a random route with acceleration, but occasionally deviates from the lane lines and has significant wobbling. **The driving is unstable and potentially dangerous.**
VLASCD(no-language)_pred_35: --	**Not Acceptable**	**VLASCD(no-language)_pred_35 is driving:** action=[0.8945135, -0.0123564]	The vehicle is driving safely along a random route, with smooth acceleration. **The driving is smooth and safe.**

Figure 2: Randomly shows examples of different models (MIMO architecture) engaging in smooth conversation with humans while making real-time action decisions during driving.

4.1 Experimental setting

Our experiments were conducted in gym-carla (Chen, 2020), an OpenAI Gym-compatible environment built on CARLA 0.9.10 . For LoRA fine-tuning, we selectively updated only the Q and V projection modules (0.06% of Llama-7B's total parameters). Additional implementation details, including hyperparameters for VLASCD, linear mapping layers, and gym-carla configurations, are provided in Appendix A.1.

4.2 Comparison methods

The Behavior Cloning (BC) method performed in gym-carla was used as a baseline. The other methods involved for comparison include RL methods Dreamer (Hafner et al., 2019) and Forbes (Chen et al., 2022), Decision Transformer (DT) (Chen et al., 2021), and VLA models OpenVLA (Kim et al., 2024) and DriverGPT4 (Xu et al., 2024).

4.3 Training datasets

The training dataset D_{expert} was obtained from the EGADS framework (Tang et al., 2024), which designs RL and IL-based agent with safety constraints, demonstrating excellent performance in CARLA. Therefore, we select this agent as our experts. We let such experts drive vehicles in town03 of CARLA to collect the dataset. D_{expert} is 5.69GB in size, containing 13,761 frames. For each frame, one question out of 50 was randomly selected based on the textual description of the current frame's observation, along with its corresponding answer for that specific frame. For a more detailed description of the D_{expert} and the map, please refer to Appendix A.3 and Appendix A.2, respectively. As shown in Figure 3 (b), we used the layout of the town03 for training. In the experimental environment for data collection and online evaluation, all vehicles randomly select directions at intersections, follow randomly generated routes, slow down for preceding vehicles, and stop when the traffic light ahead turns red.

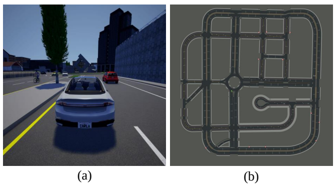


Figure 3: The (a) shows a sample view of the simulation environment, while the (b) presents a bird-eye view of our task scenario.

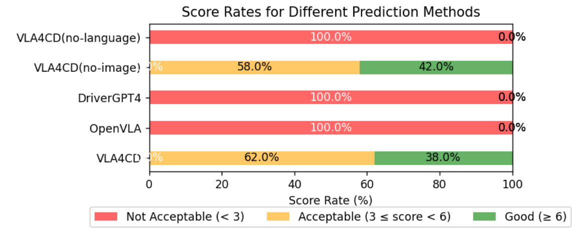


Figure 4: GPT-4o scores the answers from five methods for randomly generated inputs and question

Following Chen et al. (2024), we design a template based parser that translates sensor data (such as position and distance information, excluding vision and lidar) into natural language descriptions, as shown in *"other sensors input"* in Figures 1 and 2. For details on the templates, refer to Ap-

Table 1: Evaluation results for different methods in town03 (random), *H=1*

Method	Input	DS ↑	AR (f) ↑	ASD(m) ↑	ER(%) ↑	OR(%) ↓	CR(%) ↓
BC	image	20.21±7.46	175.34±72.86	54.21±6.41	9.08±0.56	54.86±20.04	60.00±11.23
DriverGPT4	image, text	-	-	-	-	-	-
Openvla	image, text	-13.02±4.02	-199.16±38.73	24.34±5.02	5.25±0.39	24.36±4.17	95.00±0.00
VLASCD	image, text	92.78±23.75	466.80±91.66	71.77±9.40	16.35±1.56	15.33±4.36	55.00±11.41

Table 2: Evaluation results for different methods in town03 (random), *H=4*

Method	Input	DS ↑	AR (f) ↑	ASD(m) ↑	ER(%) ↑	OR(%) ↓	CR(%) ↓
BC	image	36.39±13.37	314.66±86.02	64.08±10.48	9.04±0.62	37.56±16.44	45.00±11.41
Dreamer	image	-0.03±0.01	-14.96±0.09	0.02±0.01	0.22±0.01	0.00±0.00	0.00±0.00
Forbes	image	0.98±1.43	21.63±21.72	22.84±1.00	6.30±0.31	18.78±1.03	56.67±9.20
DT	image	7.68±3.24	51.97±29.33	23.74±2.47	9.92±0.71	10.31±2.32	65.00±10.94
DriverGPT4	image, text	-	-	-	-	-	-
Openvla	image, text	-7.84±0.67	-160.37±7.85	18.03±1.92	4.76±0.19	20.77±3.36	100.00±0.00
VLASCD	image, text	105.25±14.03	349.52±49.75	59.76± 5.04	25.02±2.57	19.93±2.11	30.00±10.51

Table 3: Evaluation the generalization for different methods in town04 (random), *H=4*

Method	Input	DS ↑	AR (f) ↑	ASD(m) ↑	ER(%) ↑	OR(%) ↓	CR(%) ↓
BC	image	39.22±11.64	358.79±79.59	63.08±9.37	8.69±0.56	5.64±1.26	60.00±11.23
Dreamer	image	-0.03±0.01	-15.03±0.07	0.02±0.01	0.01±0.21	0.01±0.00	0.00±0.00
Forbes	image	-2.63±2.75	-17.37±22.98	19.79±1.20	6.24±0.69	15.80±2.74	66.70±8.75
DT	image	10.66±3.26	85.58±27.04	24.94±2.92	10.55±0.58	11.38±2.15	55.00±11.41
DriverGPT4	image, text	-	-	-	-	-	-
Openvla	image, text	-6.74±0.88	-153.35±10.26	13.62±1.86	4.26±0.17	15.70±2.71	100.00±0.00
VLASCD	image, text	94.26±15.26	384.52±51.72	56.93±4.03	21.49±1.86	12.75±2.28	45.00±11.41

pendix A.7. Note that such *"other sensors input"* does not include any action-related information from VLASCD, such as speed and heading angle. In this way, we can test whether VLASCD can leverage informative text data to enhance the quality of action decisons.

4.4 Performance metrics

Performance metrics for evaluating the chatting ability. The study by Wang et al. (2023) confirms ChatGPT's high consistency with human judgments. Based on this finding, we employed GPT-4o (OpenAI, 2023a) to systematically compare answer quality between VLASCD and baseline models. Our evaluation procedure consisted of: (1) selecting 50 random driving environment-question pairs; (2) generating responses from baseline models for each pair; and (3) scoring responses (0-10 scale) using GPT-4o with the following criteria: Not Acceptable (< 3), Acceptable ($3 \leq$ score < 6), and Good (≥ 6). The complete evaluation prompt is provided in Appendix A.7. Additionally, to assess the impact of the language and image components on dialogue capabilities, we included VLASCD (no-language) and VLASCD (no-image), two simplified versions of VLASCD trained by removing the loss items corresponding to text generation and image reconstruction, respectively: $\mathcal{L}_{\text{action}} + \mathcal{L}_{\text{image}}$, $\mathcal{L}_{\text{action}} + \mathcal{L}_{\text{language}}$.

Performance metrics for evaluating the decision-making ability. We deployed our trained model on a vehicle for autonomous urban navigation and evaluated its performance using established metrics: Collision Rate (CR), Off-road Rate (OR), Episode Completion Rate (ER), Average Safe Driving Distance (ASD), Average Reward (AR), and Driving Score (DS). DS, a composite metric assessing overall performance, is defined as: DS = ER × AR, aligning with the CARLA Leaderboard's methodology. For AR, we adopted the reward function from Chen et al. (2019), which evaluates driving dynamics, including yaw, collisions, speeding, and lateral velocity. Model selection prioritized checkpoints optimizing both DS and AR. The remaining metrics (ER, OR, AR, ASD) were implemented following Gao et al. (2024). Further details on reward computation and metric calculations are provided in Appendices A.5 and A.6.

4.5 Results on chatting ability evaluation

As shown in Figures 2 and 4, VLASCD performs significantly better that others in terms of chatting ability. In contrast, OpenVLA performs poorly in question-answering because it focuses solely on optimizing the action loss. DriverGPT4 faces challenges as both tasks share the same decoder, causing the model to misinterpret inputs as only for action prediction, making it difficult to generate complete text. Despite having two independent loss items, the model has not effectively balanced these two losses. Furthermore, VLASCD (no language) shows a significant gap in conversational ability compared to VLASCD, while VLASCD (no image) performs similarly to VLASCD, highlighting the importance of the language loss component for enhancing chatting abilities.

4.6 Results on decision-making ability evaluation

We define the "-" entries in Tables 1-3 as system failure cases where no complete action value is generated within the 50-second threshold, which represents the maximum allowable stationary wait time before triggering system intervention. H denotes the length of the context.

As shown in Table 1, VLASCD significantly outperforms BC and OpenVLA in terms of DS, AR, and ASD at a single time step, while DriverGPT4 fails to generate precise action values. VLASCD also shows significant improvements over other methods across multiple time steps in Table 2, indicating sustained benefits over longer durations. We evaluated these models' generalization capability by training them on the town03 dataset and then evaluating them online in town04. As shown in Table 3, the primary metric DS of VLASCD significantly exceeds that of the other methods, showcasing its strong generalization ability. Tables 1, 2, and 3 indicate that DriverGPT4 faces challenges in generating precise action values for real-time control commands, highlighting the difficulties of directly generating accurate values using a detokenizer. In contrast, OpenVLA can generate precise values in experiments but produces identical action commands, causing vehicles to wander or spin in a place, resulting in significant penalties. Results in Tables 2 and 3 demonstrated that VLASCD also significantly outperforms DT, Dreamer, and Forbes in terms of decision-making and generalization.

Finally, Figure 2 illustrates how our MIMO model smoothly engages in conversation with a human while simultaneously making real-time action decisions during the driving process.

4.7 Ablation studies on the loss function design

As shown in Equation (6), our loss function is composed of three losses, namely action loss $\mathcal{L}_{\text{action}}$, language loss $\mathcal{L}_{\text{language}}$, and image loss $\mathcal{L}_{\text{image}}$. We conducted ablation studies to investigate the effect of each loss on the performance of VLASCD. The experiment result is shown in Table 4, where the action-bins loss $\mathcal{L}_{\text{action-bins}}$ denotes the action loss used by OpenVLA and RT2. They deal with continuous valued actions by value discretization. We included VLASCD (no-language) and VLASCD (no-image), two simplified versions of VLASCD trained by using $\mathcal{L}_{\text{action}} + \mathcal{L}_{\text{image}}$ and $\mathcal{L}_{\text{action}} + \mathcal{L}_{\text{language}}$, respectively.

On the effect of $\mathcal{L}_{\text{action}}$ As shown in Table 4, if we compare the performance metrics of $\mathcal{L}_{\text{image}} + \mathcal{L}_{\text{language}} + \mathcal{L}_{\text{action-bins}}$ with that of $\mathcal{L}_{\text{image}} + \mathcal{L}_{\text{language}} + \mathcal{L}_{\text{action}}$, we can see a clear advantage of using our action loss $\mathcal{L}_{\text{action}}$ over using $\mathcal{L}_{\text{action-bins}}$. This explains why VLASCD outperforms VLA models that use the type of action loss similar to $\mathcal{L}_{\text{action-bins}}$, as shown in Tables 1, 2, and 3. Specifically, from our experiments, we found that doing action discretization and tokenization as in current VLA models lead to low training loss but bad inference performance. This is because adjacent action intervals are represented by consecutive token IDs (e.g., 31830 and 31831), which are close in token space. Consequently, the model tends to output the same token (31830 or 31831) in inference, while the actual action values corresponding to them can have significant differences. In contrast, our proposed approach for continuously valued actions can avoid this phenomenon.

On the effect of $\mathcal{L}_{\text{language}}$ As shown in Table 4, if we compare performance metrics between $\mathcal{L}_{\text{image}} + \mathcal{L}_{\text{action}}$ (corresponding to VLASCD (no-language)) and $\mathcal{L}_{\text{image}} + \mathcal{L}_{\text{language}} + \mathcal{L}_{\text{action}}$ (corresponding to VLASCD), we see that including $\mathcal{L}_{\text{language}}$ in the loss function significantly enhances the quality of decision-making. As shown in Figures 2 and 4, VLASCD (no-language) has significantly different dialogue capabilities compared to VLASCD, while VLASCD (no-image) performs similarly to VLASCD. It demonstrates that $\mathcal{L}_{\text{language}}$ plays an important role for maintaining the dialogue capability. To summarize, including $\mathcal{L}_{\text{language}}$ in the loss function has beneficial impacts on both dialogue and decision-making.

On the effect of $\mathcal{L}_{\text{image}}$ As shown in Table 4,

Table 4: Ablation studies on the loss function of VLASCD in town03 (random), *H=4*

Loss function	Input	DS ↑	AR (f) ↑	ASD(m) ↑	ER(%) ↑	OR(%) ↓	CR(%) ↓
$\mathcal{L}_{\text{image}}$+$\mathcal{L}_{\text{language}}$+$\mathcal{L}_{\text{action-bins}}$	image, text	11.57±0.00	142.83±0.01	22.71± 0.01	8.10±0.05	30.87±0.10	100.00±0.00
$\mathcal{L}_{\text{image}}$+$\mathcal{L}_{\text{action}}$	image, text	45.08±10.88	234.36±52.21	39.64± 4.03	14.13±1.71	16.68±3.15	30.00±10.51
$\mathcal{L}_{\text{language}}$+$\mathcal{L}_{\text{action}}$	image, text	74.85±10.97	331.78±49.88	50.63±4.73	18.62±1.95	15.96±2.45	25.00±9.93
$\mathcal{L}_{\text{image}}$+$\mathcal{L}_{\text{language}}$+$\mathcal{L}_{\text{action}}(our)$	image, text	105.25±14.03	349.52±49.75	59.76± 5.04	25.02±2.57	19.93±2.11	30.00±10.51

Table 5: The impact of noise ratio in sensor inputs and QA content on the decision-making performance of VLASCD in town03(random)

Input	Sensor input / QA (noise ratio)	DS ↑	AR (f) ↑	ASD(m) ↑	ER(%) ↑	OR(%) ↓	CR(%) ↓
image, text	0% / 100%	74.32±24.44	288.54±74.62	62.42±7.71	25.76±1.54	11.05±1.62	50.0±0.51
image, text	0% / 0%	93.89±29.73	336.11±86.72	45.42±9.53	16.68±2.50	19.05±4.96	5.00±5.00
image, text	100% / 0%	-0.01±1.12	-5.10±0.00	0.00±0.00	0.30±0.00	0.00±0.00	0.00±0.00

when we added $\mathcal{L}_{\text{image}}$ in the loss function (corresponding to results of $\mathcal{L}_{\text{image}} + \mathcal{L}_{\text{language}} + \mathcal{L}_{\text{action}}$), all performance metrics related to decision-making are increased in value, compared to $\mathcal{L}_{\text{language}} + \mathcal{L}_{\text{action}}$. This confirms that the $\mathcal{L}_{\text{image}}$ indeed brings remarkable benefits for enhancing decision-making performance. We argue that this is because, during the decision-making, doing high-quality image reconstruction can further explore and utilize the rich information related to the current scene within the image modality data, thereby benefiting the decision-making.

4.8 How to resolve conflicts in simultaneous multi-task output ?

In DriveGPT4, the text generation and action generation tasks lack independently designed loss functions, leading to conflicts between the two tasks, particularly in complex environments where the model fails to simultaneously generate efficient text and action instructions. As shown in Figure 2, DriveGPT4 cannot guarantee fine-grained action instructions at each moment, compromising its decision accuracy and dialogue capabilities. In contrast, VLASCD establishes independent objective functions for text generation and action generation, ensuring efficient parallel processing of each task and avoiding task conflicts.

4.9 How does textual data quality in training impact model decision-making ?

Imagine a driver operating a car while conversing with a passenger. If the passenger's words are irrelevant to the driving situation, they might interfere with the driver's decision-making, though humans naturally possess some level of noise resistance. To test whether our model exhibits human-like decision-making behavior, we designed a set of experiments. The results, shown in Table 5, reveal that when noise unrelated to driving scenarios is introduced into the sensor input, the model's decision-making performance declines rapidly. However, when noise is added only to the QA content while keeping the sensor input noise-free, the performance drop is less significant. This indicates that our model maintains robustness when the sensor input remains relevant to the driving scenario, even if the QA content contains noise. These findings demonstrate that our model's decision-making performance closely resembles that of human drivers.

5 Conclusion

In this study, we explore how to develop a multimodal pre-training framework to address the inherent task mutual exclusion in MISO architectures during MIMO scenarios (e.g., parallel multi-tasks output processing), where competing conflicts arise among different tasks sharing output channels, leading to imbalanced model optimization and significant performance degradation in specific tasks. we propose a unified MIMO training architecture with parallel multi-task output capabilities-VLASCD. Experiments show that VLASCD surpasses state-of-the-art VLA models, RL, and decision transformers in decision-making while maintaining fluent dialogue, thanks to our continuous-action handling, cost function design, and label smoothing techniques.

The evolution from modular systems composed of discrete subcomponents to unified end-to-end models represents a major ongoing trend in AI research. Within the MIMO scenario, we believe this work constitutes a meaningful initial attempt in developing a unified generative model capable of simultaneously handling both dialogue and action generation in an end-to-end manner.

Limitations

This study has several limitations that warrant further exploration in future work. First, although VLASCD is designed as a general-purpose and unified MIMO training, and the experiments leverage the CARLA simulator for efficient data collection, the validation is currently limited to autonomous driving scenarios. Its generalizability to other domains (e.g., robotics, human-computer interaction) remains unverified. Second, as an initial exploration of an end-to-end multi-task generative model, the joint optimization of dialogue understanding and action generation still has room for improvement, particularly in multi-task coordination and scalability. Additionally, we observe that excessively long text prompts or large image patches can cause synchronization delays across tasks due to computational bottlenecks, highlighting the need for more efficient token processing and resource allocation strategies to enhance real-time performance. We believe that these limitations provide clear directions for our future research.

References

Michael Ahn, Yen-Ling Chen, Anthony Brohan, Mark McCarthy, Jonathan Carff, Matthew Hill, Jerry Tworek, Andrew Yuan, Michael Paster, Karol Hausman, and 1 others. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, and 1 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. 2023. Grounding large language models in interactive environments with online reinforcement learning. In *International Conference on Machine Learning*, pages 3676–3713. PMLR.

Jianyu Chen. 2020. An openai gym third party environment for carla simulator. `https://github.com/cjy1992/gym-carla?tab=readme-ov-file`.

Jianyu Chen, Bodi Yuan, and Masayoshi Tomizuka. 2019. Model-free deep reinforcement learning for urban autonomous driving. In *2019 IEEE intelligent transportation systems conference (ITSC)*, pages 2765–2771. IEEE.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.

Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. 2024. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14093–14100. IEEE.

Xiaoyu Chen, Yao Mark Mu, Ping Luo, Shengbo Li, and Jianyu Chen. 2022. Flow-based recurrent belief state learning for pomdps. In *International Conference on Machine Learning*, pages 3444–3468. PMLR.

Nicolai Dorka, Chenguang Huang, Tim Welschehold, and Wolfram Burgard. What matters in employing vision language models for tokenizing actions in robot control? In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.

A. S. et al. 2024. Introducing rfm-1: Giving robots human-like reasoning capabilitie. `Introducingrfm-1:Givingrobotshuman-likereasoningcapabilitie`.

Justin Fu, Kelvin Zhang, Utkarsh Sanyal, Lantao Yu, Collin Moses, Fan Yang, Stefano Ermon, and Zhibin Zhao. 2023. Driving with reasoning: Reinforcement learning with generalist language models for interpretable policies. *arXiv preprint arXiv:2303.00745*.

Zeyu Gao, Yao Mu, Chen Chen, Jingliang Duan, Ping Luo, Yanfeng Lu, and Shengbo Eben Li. 2024. Enhance sample efficiency and robustness of end-to-end urban autonomous driving via semantic masked world model. *IEEE Transactions on Intelligent Transportation Systems*.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2019. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.

Bin Hu, Chenyang Zhao, Pu Zhang, Zihao Zhou, Yuanhang Yang, Zenglin Xu, and Bin Liu. 2024. Enabling intelligent interactions between an agent and an llm: A reinforcement learning approach. *Reinforcement Learning Conference (RLC)*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. 2023. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*.

Chia-Chun Hung, Timothy Lillicrap, Josh Abramson, Yan Wu, Mehdi Mirza, Federico Carnevale, Arun Ahuja, and Greg Wayne. 2019. Optimizing agent behavior over long time scales by transporting value. *Nature communications*, 10(1):5223.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, and 1 others. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, and 1 others. 2023b. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023c. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*.

OpenAI. 2023a. Gpt-4 technical report.

OpenAI. 2023b. Gpt-4: Technical report. `https://www.openai.com/research/gpt-4`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, and 1 others. 2023. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*.

Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. 2023. Languagempc: Large language models as decision makers for autonomous driving. *arXiv preprint arXiv:2310.03026*.

Chao Song, Zhihao Ye, Qiqiang Lin, Qiuying Peng, and Jun Wang. 2024. A framework to implement 1+ n multi-task fine-tuning pattern in llms using the cgc-lora algorithm. *arXiv preprint arXiv:2402.01684*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Zuojin Tang, Xiaoyu Chen, YongQiang Li, and Jianyu Chen. 2024. Safe and generalized end-to-end autonomous driving system with reinforcement learning and demonstrations. *arXiv preprint arXiv:2401.11792*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faysal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Pierre-Emmanuel Albert, Amjad Almahairi, Yasmine Babaei, Dmytro Bashlykov, Subhojit Batra, Anurag Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yi Xiao, Felipe Codevilla, Akhil Gurram, Onay Urfalioglu, and Antonio M López. 2020. Multimodal end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):537–547.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. 2024. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*.

Shunyu Yao, Jeffrey Wu, Daisy Zhe Liu, Dale Schuurmans, Quoc V Le, Denny Zhou, Yuan Cao, and Andrew Dai. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 2024. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*.

Zihao Zhou, Bin Hu, Pu Zhang, Chenyang Zhao, and Bin Liu. 2024. Large language model as a policy teacher for training reinforcement learning agents. *International Joint Conference on Artificial Intelligence (IJCAI)*.

A Appendix

A.1 Hyperparameter settings

In this section, we respectively introduce the model parameters of VLASCD, the parameters of the custom linear layers, as well as the parameters of gym-carla and evaluation, as shown in Tables 6, 7, and 8. In addition, we trained the models using Python 3.8, Transformers 4.30.0, and a NVIDIA Tesla V100 GPU. The training time ranges from 5 to 13 hours, depending on the input modality and trajectory length. We also conducted experiments on the three hyperparameters of the loss function in Appendix A.12. In our experiments, we choose $\alpha_1 = 0.1$, $\alpha_2 = 10$, and $\lambda = 0.5$.

A.2 CARLA maps

In order to comprehensively evaluate the performance of our VLASCD, we utilized five maps in CARLA, including town03, town04 as shown in Figure 6. Town03 is one complex map in CARLA, closely resembling real urban road environments, including various complex scenarios such as tunnels, intersections, roundabouts, curves, and multi-turns, covering an area of 400m × 400m, with a total road length of approximately 6km.Town04 is a small town with a backdrop of snow-capped mountains and conifers. A multi-lane road circumnavigates the town in a "figure of 8".

A.3 Training datasets

We trained all comparison methods based on an expert dataset D_{expert}, which is 5.69GB in size, containing 13,761 frames. We used 90% of it as the training set and the remaining as the test set. We evaluated these comparison methods online in the random mode of CARLA town03. Following the work on DT (Chen et al., 2021), we investigated the performance of sequence fusion for both single time steps and multiple time steps. We set the context length $H = 1$, resulting in a fusion sequence length of 489. This includes dividing the 128×128 image into 64 tokens and padding the text sequences to a length of 424 tokens, including an empty placeholder token. However, due to computational constraints, we only explored trajectory sequences with a maximum length of 489*4=1956 to validate performance in a longer context. We also explored whether the decision-making ability of VLASCD is enhanced with longer context of trajectories in Appendix A.4. Additionally, we evaluated performance across different modalities and generalization capabilities in town04. For detailed information on the CARLA maps, refer to Appendix A.2. All comparison methods were tested online in the CARLA simulator. We conducted evaluations over 20 episodes, each consisting of 1000 steps, with 200 involved vehicles, whose driving routes and met scenarios are generated in random mode.

A.4 Is model decision-making ability enhanced with longer context of trajectories?

As shown in Table 9, we observed that although the context length H of input trajectories is longer, the overall DS and AR of VLASCD show some improvement, but the increase is not significant. This improvement is primarily attributed to the higher route completion and lower collision rates associated with longer time steps. According to Section 4.3, when $H = 4$, the sequence length extends to 1956, representing a fourfold increase in sequence length. Despite this, the improvement in DS and AR scores is not pronounced. Notably, in metrics such as AR and ADS, the performance of $H = 4$ is even worse than that of $H = 1$. This suggests that the input information might be redundant, and excessively long trajectories could negatively impact decision-making ability.

This result highlights several key issues. First, while longer context lengths provide the model with more historical context and information, an excessive amount of information may hinder the ability of model to effectively filter and extract useful decision signals, leading to information redundancy. Redundant information not only increases the computational complexity but also may distract the attention of model, reducing its capacity to capture critical features and thereby affecting overall decision-making. Therefore, shorter context length sequences provide more concise and precise inputs, facilitating quicker and more accurate judgments by the model. This indicates that the current fusion method has limited performance improvements. Chen et al. (2021); Hung et al. (2019) suggest that longer context lengths can bring more benefits for decision control, so we also consider how to compress historical information and efficiently fuse it in the future to enhance decision-making.

Table 6: Hyperparameters

Parameter	**Value**
batch_size	64
micro_batch_size	8
num_epochs	3
learning_rate	3e-4
cutoff_len	424
val_set_size	0.1
save_step	25
lora_r	8
lora_alpha	16
lora_dropout	0.05
lora_target_modules	{q_proj, k_proj}
Other Sensors Input_types	{obs, text}
lambda_action	10
lambda_smooth	0.1
lambda_img	0.5
horizon	1
regular_action_loss	False
img_patch_size	16

Table 7: Model Parameters and Layers

Parameter/Layer	**Details**
num_patches	64
tokenizer_vocab_size	32000
split_obs_proj	Conv2d(3, 4096, kernel_size=16, stride=16)
inverse_split_obs_proj	ConvTranspose2d(4096, 3, kernel_size=16, stride=16)
split_obs_position_embedding	Parameter(torch.randn(1, 64, 4096))
text_embedding	nn.Embedding(32000, 4096)
custom_lm_head	Linear(4096, 32000, bias=False)
actor_linear1	Linear(4096, 2048)
actor_linear2	Linear(2048, 1024)
actor_linear3	Linear(1024, 512)
actor_linear4	Linear(512, 256)
actor_linear5	Linear(256, 128)
actor_linear6	Linear(128, 64)
actor_linear7	Linear(64, 2)
reconstruction_layer	Linear(4096, micro_batch_size*3*128*128)
action_linear	Linear(2, 4096)

A.5 Reward function

We use the default reward function of the Gym-Carla benchmark (Chen et al., 2019) to evaluate all experimental methods, as follows:

$$\begin{aligned} f = 200r_c + v_{lon} + 10r_f + r_o \\ -5\alpha^2 + 0.2r_{lat} - 0.1 \end{aligned} \tag{7}$$

where r_c is the reward related to collision, which is set to -1 if the ego vehicle collides and 0 otherwise. v_{lon} is the longitudinal speed of the ego vehicle. r_f is the reward related to running too fast, which is set to -1 if it exceeds the desired speed (8 m/s here) and 0 otherwise. r_o is set to -1 if the ego vehicle runs out of the lane, and 0 otherwise. α is the steering angle of the ego vehicle in radians.

Table 8: gym-carla and evaluation Environment Parameters

Parameter	**Value**
Number of Vehicles	200
Number of Walkers	0
Random Seed	1
Other Sensors Input_names	lidar_noground
Display Size	400
Max Past Step	1
Time Step (dt)	0.1
Discrete Control	False
Continuous Acceleration Range	[-3.0, 3.0]
Continuous Steering Range	[-0.2, 0.2]
Ego Vehicle Filter	vehicle.lincoln*
Traffic Manager Port	Random integer (2000 to 9000)
Town Map	town03 or town04
Task Mode	Random
Max Time per Episode	2000
Max Waypoints	12
Observation Range	32
LiDAR Bin Size	0.25
Distance Behind Ego Vehicle	12
Lane Threshold	2.0
Desired Speed	8
Max Ego Vehicle Spawn Times	200
Display Route	True
PIXOR Grid Size	64
PIXOR Mode	False
Predict Speed	True

Table 9: Evaluation VLASCD longer context results for mulitmodal input in town03 (random)

Input	$\mathcal{L}_{\text{image}}$	H	DS ↑	AR (f) ↑	ASD(m) ↑	ER(%) ↑	OR(%) ↓	CR(%) ↓
image	×	1	29.55±6.17	226.91±42.24	54.24±4.30	11.85±0.68	20.22±5.57	70.00±10.5
image	×	4	22.38±4.96	155.79±31.87	32.45±1.74	14.41±0.59	15.93±2.65	40.00±11.23
text	×	1	37.44±10.11	248.89±52.91	47.37±5.43	15.63±1.98	17.02±2.71	40.00±11.24
text	×	4	44.16±7.39	252.10±38.94	46.96±3.23	15.66±1.06	12.86±2.45	60.00±11.23
image, text	×	1	68.10±13.20	417.24±57.41	58.81±6.55	13.71±1.26	11.39±2.41	40.00±11.24
image, text	×	4	74.85±10.97	331.78±49.88	50.63±4.73	18.62±1.95	15.96±2.45	25.00±9.93
image, text	✓	1	92.78±23.75	466.80±91.66	71.77±9.40	16.35±1.56	15.33±4.36	55.00±11.41
image, text	✓	4	105.25±14.03	349.52±49.75	59.76± 5.04	25.02±2.57	19.93±2.11	30.00±10.51

r_{lat} is the reward related to lateral acceleration, which is calculated by $r_{lat} = -|\alpha| \cdot v_{lon}^2$. The last constant term is added to prevent the ego vehicle from standing still.

A.6 Measure performance metrics

We use multiple key metrics to evaluate the performance of autonomous driving models in various driving scenarios. Collision Rate (CR): the frequency at which the vehicle collides with obstacles or other vehicles. This metric is critical for assessing the safety of the driving model. Outlane Rate (OR): the rate at which the vehicle deviates from its designated lane. This metric evaluates the ability of modes to maintain proper lane discipline. Episode Completion Rate (ER): the percentage of driving tasks or episodes that the vehicle successfully completes. Higher completion rates indicate better task performance. Average Safe Driving Distance (ASD): the average distance driven with-

(a) Town03

(b) Town04

Figure 5: CARLA maps

out incidents, such as collisions or off-road events. This metric highlights the capability to drive safely over extended periods. Average Return (AR): A metric that measures the cumulative reward collected by the vehicle during its driving tasks, often reflecting both task performance and adherence to safety guidelines. Driving Score (DS): A comprehensive metric that reflects the overall performance of the vehicle in terms of safety, efficiency, and compliance with traffic rules, aligning with the CARLA Leaderboard's methodology. For AR, we adopted the reward function f from Chen et al. (2019), which evaluates driving dynamics, including yaw, collisions, speeding, and lateral velocity. Model selection prioritized checkpoints optimizing both DS and AR. The remaining metrics (ER, OR, AR, ASD) were implemented following Gao et al. (2024).

$$CR = \frac{N_{\text{collisions}}}{N_{\text{total\_episodes}}}, OR = \frac{N_{\text{off\_road\_events}}}{N_{\text{total\_episodes}}} \tag{8}$$

$$ER = \frac{N_{\text{completed\_steps}}}{N_{\text{total\_steps}}}, ASD = \frac{\sum_{i=1}^{N_{\text{episodes}}} \text{distance}_i}{N_{\text{total\_episodes}}} \tag{9}$$

$$AR = \frac{\sum_{i=1}^{N_{\text{episodes}}} \text{rewards}_i}{N_{\text{total\_episodes}}}, DS = ER \times AR \tag{10}$$

Where $N_{\text{collisions}}$ is the number of collisions during the episode, and $N_{\text{total\_episodes}}$ is the total number of episodes in the test. Where $N_{\text{off\_road\_events}}$ is the number of times the vehicle went off-road, and $N_{\text{total\_steps}}$ is the total number of episodes. Where distance_i is the distance driven during the i-th safe driving episode, and $N_{\text{safe\_episodes}}$ is the number of episodes without incidents (such as collisions or off-road events). Where $N_{\text{completed\_steps}}$ is the number of successfully completed steps, and $N_{\text{total\_steps}}$ is the total number of steps in the episode. Where AR is the average reward f collected during the episode.

A.7 The natural language template for text input

We obtained information from the CARLA environment using other sensors (such as speed sensors and position sensors), excluding the acceleration and steering (action) of the ego vehicle). This information is transformed into a natural language template that the VLA can understand, as shown below:
<lateral_dis, delta_yaw, speed, vehicles_info> = <observation_vehicle_state>
<vehicles_num> = <len(vehicles_info)>
<multi_dis += str(vehicles_info[i][0])+"", multi_yaw += str(vehicles_info[i][1])+"", multi_speed += str(vehicles_info[i][2])+"">
*<**if vehicles_num**=1:>*
*<**new_input**="You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you're heading are respectively {**multi_speed**} m/s, {**multi_dis**} m, {**multi_yaw**}°." "You are now {**lateral_dis**}m laterally away from your driving route. ">*
*<**elif vehicles_num**>1:>*
*<**new_input**="You can see that there are **vehicles_num** cars. Their speed, straight-line distance from you, and angle in the direction you're heading are respectively {**multi_speed**} m/s, {**multi_dis**} m, {**multi_yaw**}°." "You are now {**lateral_dis**}m laterally away from your driving route. ">*
*<**elif vehicles_num**=0:>*
*<**new_input**="You see no car here, and you are now {**lateral_dis**}m laterally away from your driving route.">*

We followed Wang et al. (2023) *"Is ChatGPT a Good NLG Evaluator?"* approach. The complete evaluation prompt template for using GPT-4o

(OpenAI, 2023b) is as follows:"*The document contains 50 similar examples as described above. For each example, based on the given Input_0: and Question_0:, please evaluate and score the responses generated by the five methods (VLASCD_pred_0, Openvla_pred_0, DriverGPT4_pred_0, VLASCD_image_pred_0, and VLASCD_language_pred_0) using a 10-point scale with the following criteria:Not Acceptable* (< 3)*, Acceptable* ($3 \leq$ *score* < 6)*, and Good* (≥ 6)*. Please output the individual scores for each example. After evaluating all 50 examples, calculate the average rates for: Not Acceptable, Acceptable, Good, and Excellent performance for each method.*"

A.8 The benefits of cross-entropy loss and label smoothing loss for VLASCD

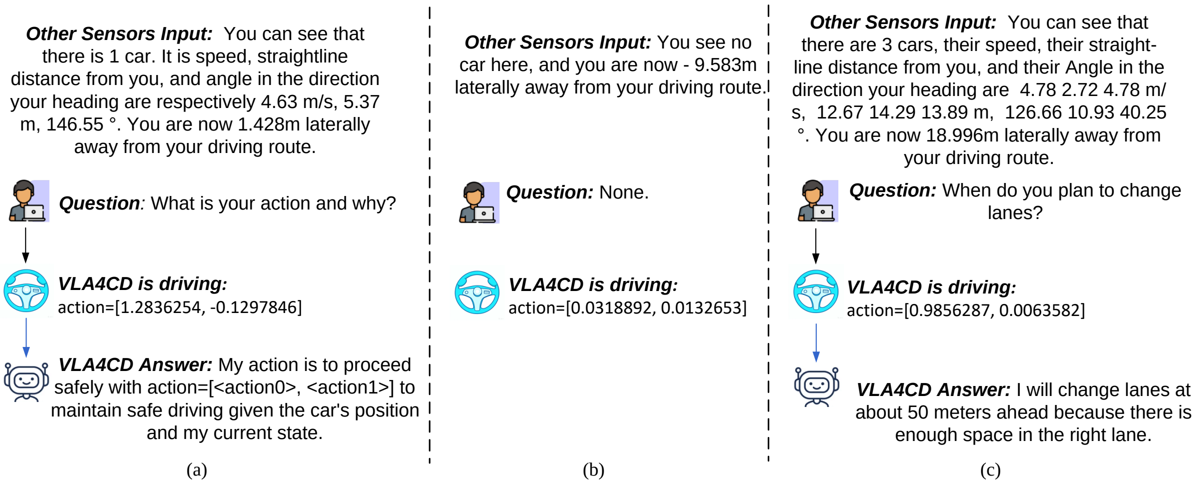
We found that merely replacing specific numerical values in the translation template (Chen et al., 2024) results in minimal representational differences caused by the sequential nature of data, making it easy for conventional cross-entropy loss to lead to overfitting in text generation tasks. As shown in Table 10, we tested on both town03 and town04, which led to a decline in the decision-making performance of model. Compared to cross-entropy loss, cross-entropy loss with smoothed labels performed better. Therefore, we chose cross-entropy loss with smoothed labels as the loss for text generation in VLASCD in our experiments.

A.9 The impact of training data-related factors on the decision performance of model

In the multimodal ablation experiments on the VLASCD model, as shown in Table 11, we systematically removed or replaced individual modalities to evaluate their contribution to decision-making. The results show that models utilizing image and text fusion significantly outperform those with only a single image or text input in terms of decision accuracy and stability. This indicates that the text modality in our dataset provides higher-level semantic abstraction to complement visual inputs, thereby enhancing overall decision-making ability. In addition, as shown in Table 11, a single text input performs better than a single image input, indicating that the information provided by the text modality in our dataset (especially from *"other sensors input"*, as shown in Figure 6) is highly beneficial for improving the decision-making ability of model.

A.10 The noise consisted of information datasets

The noise consisted of information completely unrelated to the current driving scenario as follow: {"A playful puppy brings joy and laughter to our days", "The whisper of the wind carries secrets of the universe", "A hidden garden blooms with the magic of nature's colors", "The aroma of fresh coffee awakens the senses each morning", "A handwritten letter feels like a warm hug from afar", "The glimmer of fireflies creates a magical summer night", "A spontaneous adventure can lead to unforgettable memories", "The serenity of a quiet lake reflects the beauty of the world", "A gentle touch can convey love without a single word", "The laughter of friends is the sweetest melody of all", "A warm hug is a universal language of comfort", "The dance of leaves in the breeze tells stories of change", "A cozy fire invites stories and shared moments", "The beauty of art inspires creativity and self-expression", "A day spent volunteering fills the heart with purpose", "The excitement of a new book is like embarking on a journey", "A delicious meal shared brings people closer together", "The sound of laughter can brighten even the gloomiest day", "A fleeting moment can hold the weight of a thousand memories", "The charm of small towns lies in their simple beauty", "A gentle rain nurtures the earth and inspires growth", "A colorful painting captures the essence of joy", "The peace of a mountain retreat refreshes the soul", "A favorite mug holds warmth and comfort on a chilly day", "The rustle of leaves underfoot reminds us of nature's rhythm", "A well-crafted story has the power to transport us anywhere", "The thrill of discovery keeps our spirits young and curious", "A cherished photograph holds a lifetime of memories", "The beauty of winter blankets the world in quiet calm", "A moment of kindness can change the trajectory of a day", "The aroma of spices fills the kitchen with warmth and love", "A shared joke creates bonds that laughter alone cannot", "The glow of a sunrise fills the heart with hope", "A melody can linger in the mind long after it fades", "The colors of autumn leaves create a vibrant tapestry", "A soft pillow cradles the head and invites sweet dreams", "The laughter of children brings joy and light to our lives", "A surprise visit from a friend can brighten any day", "The beauty of a flower


Other Sensors Input: You can see that there is 1 car. It is speed, straightline distance from you, and angle in the direction your heading are respectively 4.63 m/s, 5.37 m, 146.55 °. You are now 1.428m laterally away from your driving route.

Question: What is your action and why?

VLA4CD is driving:
action=[1.2836254, -0.1297846]

VLA4CD Answer: My action is to proceed safely with action=[<action0>, <action1>] to maintain safe driving given the car's position and my current state.

(a)

Other Sensors Input: You see no car here, and you are now - 9.583m laterally away from your driving route.

Question: None.

VLA4CD is driving:
action=[0.0318892, 0.0132653]

(b)

Other Sensors Input: You can see that there are 3 cars, their speed, their straight-line distance from you, and their Angle in the direction your heading are 4.78 2.72 4.78 m/s, 12.67 14.29 13.89 m, 126.66 10.93 40.25 °. You are now 18.996m laterally away from your driving route.

Question: When do you plan to change lanes?

VLA4CD is driving:
action=[0.9856287, 0.0063582]

VLA4CD Answer: I will change lanes at about 50 meters ahead because there is enough space in the right lane.

(c)

Figure 6: An example show on how VLASCD smoothly engages in conversation with a human while simultaneously making real-time action decisions during the driving process

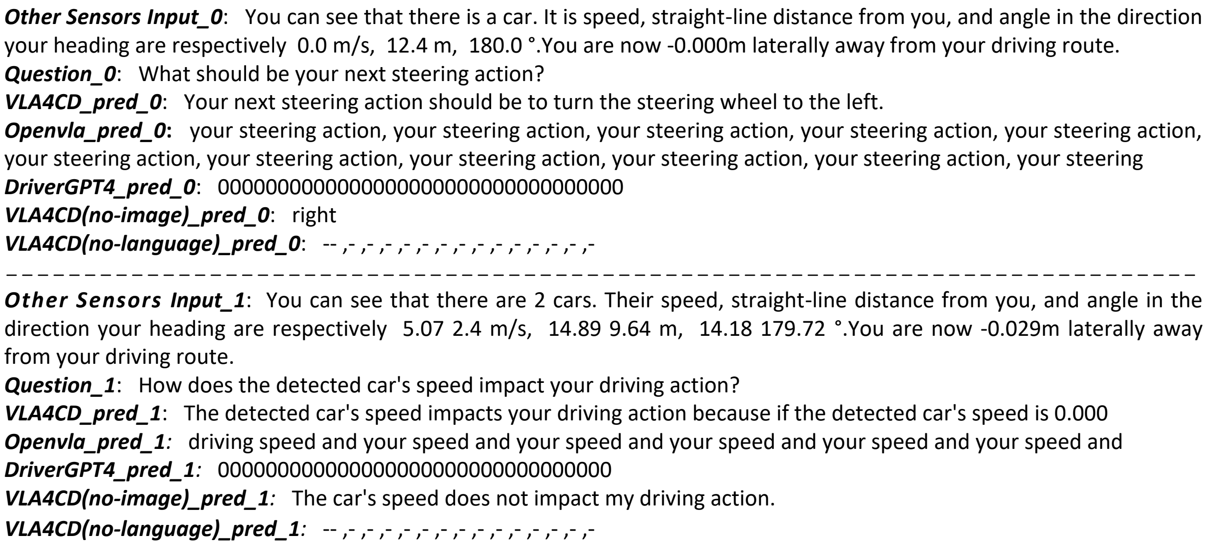
Table 10: We evaluated the performance of VLASCD using smooth label loss and cross-entropy loss functions, *H=4*

$\mathcal{L}_{\text{language}}$	Town	DS ↑	AR (f) ↑	ASD(m) ↑	ER(%) ↑	OR(%) ↓	CR(%) ↓
Cross Entropy	town03	48.97±7.60	296.53±40.72	47.10±4.87	15.37±0.85	12.41±2.73	35.00±10.94
Smooth Label	town03	105.25±14.03	349.52±49.75	59.76± 5.04	25.02±2.57	19.93±2.11	30.00±10.51
Cross Entropy	town04	66.69±16.97	358.11±61.10	52.72±5.44	15.43±1.11	9.63±1.42	55.00±11.41
Smooth Label	town04	94.26±15.26	384.52±51.72	56.93±4.03	21.49±1.86	12.75±2.28	45.00±11.41

Table 11: Evaluating the impact of different modal inputs on the decision-making of VLASCD in town03 (random), *H=4*

Input	$\mathcal{L}_{\text{image}}$	DS ↑	AR (f) ↑	ASD(m) ↑	ER(%) ↑	OR(%) ↓	CR(%) ↓
image	×	22.38±4.96	155.79±31.87	32.45±1.74	14.41±0.59	15.93±2.65	40.00±11.23
text	×	44.16±7.39	252.10±38.94	46.96±3.23	15.66±1.06	12.86±2.45	60.00±11.23
image, text	×	74.85±10.97	331.78±49.88	50.63±4.73	18.62±1.95	15.96±2.45	25.00±9.93
image, text	✓	105.25±14.03	349.52±49.75	59.76± 5.04	25.02±2.57	19.93±2.11	30.00±10.51

garden is a celebration of life", "A good book can be a loyal companion on lonely nights", "The embrace of nature can heal and rejuvenate the spirit", "A treasure hunt ignites the spirit of adventure", "The warmth of homemade cookies fills the home with love", "A playful kitten brings joy and mischief to our lives", "The scent of pine trees evokes memories of the forest",]}. Additionally, we generated a dataset of over 500 question-answer pairs using GPT to serve as noise for the QA content. A random subset of these is shown below: { "question": "What is the national flower of Egypt?", "answer": "The national flower of Egypt is the lotus." , "question": "How do I make a vegetable soup?", "answer": "To make a vegetable soup, chop vegetables, simmer them in broth, and season with herbs and spices." , "question": "What is the largest gulf in the world?", "answer": "The largest gulf in the world is the Gulf of Mexico." , "question": "How do I clean my humidifier?", "answer": "To clean your humidifier, empty it, wash the tank with soap and water, and rinse thoroughly." , "question": "What is the national animal of Egypt?", "answer": "The national animal of Egypt is the steppe eagle." , "question": "How do I make a cup of lavender tea?", "answer": "To make a cup of lavender tea, steep dried lavender flowers in hot water for 5-7 minutes." , "question": "What is the largest palace in the world?", "answer": "The largest palace in the world is the Palace of the Parliament in Romania." , "question": "How do I check my computer's hard drive health?", "answer": "To check your computer's hard drive health, use a diagnostic tool like CrystalDiskInfo or HD Tune." , "question": "What is the national flower of Greece?", "answer": "The national flower of Greece is the violet." , "question": "How do I make a chicken soup?", "answer": "To make a chicken soup, sim-



Other Sensors Input_0: You can see that there is a car. It is speed, straight-line distance from you, and angle in the direction your heading are respectively 0.0 m/s, 12.4 m, 180.0 °.You are now -0.000m laterally away from your driving route.
Question_0: What should be your next steering action?
VLA4CD_pred_0: Your next steering action should be to turn the steering wheel to the left.
Openvla_pred_0**:** your steering action, your steering action, your steering action, your steering action, your steering action, your steering action, your steering action, your steering action, your steering action, your steering action, your steering
DriverGPT4_pred_0: 000000000000000000000000000000000
VLA4CD(no-image)_pred_0: right
VLA4CD(no-language)_pred_0: -- ,- ,- ,- ,- ,- ,- ,- ,- ,- ,- ,- ,- ,- ,-

---

Other Sensors Input_1: You can see that there are 2 cars. Their speed, straight-line distance from you, and angle in the direction your heading are respectively 5.07 2.4 m/s, 14.89 9.64 m, 14.18 179.72 °.You are now -0.029m laterally away from your driving route.
Question_1: How does the detected car's speed impact your driving action?
VLA4CD_pred_1: The detected car's speed impacts your driving action because if the detected car's speed is 0.000
Openvla_pred_1: driving speed and your speed and your speed and your speed and your speed and your speed and
DriverGPT4_pred_1: 00000000000000000000000000000000
VLA4CD(no-image)_pred_1: The car's speed does not impact my driving action.
VLA4CD(no-language)_pred_1: -- ,- ,- ,- ,- ,- ,- ,- ,- ,- ,- ,- ,- ,- ,-

Figure 7: An example show on how VLASCD smoothly engages in conversation with a human while simultaneously making real-time action decisions during the driving process

mer chicken, vegetables, and broth together, and season with herbs and spices." , "question": "What is the largest strait in the world?", "answer": "The largest strait in the world is the Strait of Malacca." , "question": "How do I clean my dehumidifier?", "answer": "To clean your dehumidifier, empty it, wash the tank with soap and water, and rinse thoroughly." , "question": "What is the national animal of Greece?", "answer": "The national animal of Greece is the dolphin." , "question": "How do I make a cup of rose tea?", "answer": "To make a cup of rose tea, steep dried rose petals in hot water for 5-7 minutes." , "question": "What is the largest castle in the world?", "answer": "The largest castle in the world is Prague Castle in the Czech Republic." , "question": "How do I check my computer's network speed?", "answer": "To check your computer's network speed, use an online speed test tool and follow the instructions." , "question": "What is the national flower of Turkey?", "answer": "The national flower of Turkey is the tulip." , "question": "How do I make a beef stew?", "answer": "To make a beef stew, simmer beef, vegetables, and broth together, and season with herbs and spices." , "question": "What is the largest canal in the world?", "answer": "The largest canal in the world is the Grand Canal in China." , "question": "How do I clean my air purifier?", "answer": "To clean your air purifier, remove and wash the filters, and wipe the exterior with a damp cloth." , "question": "What is the national animal of Turkey?", "answer": "The national animal of Turkey is the gray wolf." , "question": "How do I make a cup of turmeric tea?", "answer": "To make a cup of turmeric tea, steep fresh turmeric slices in hot water for 5-7 minutes." }

A.11 Our defined set of problems

Randomly selected a question from the set of questions. = {"What are you seeing/observing?", "What are you paying attention to and why?", "Are there any traffic lights? What's the color of the traffic light?", "What's your current speed and steering angle?", "What is your action and why?", "Summarize the current driving scenario at a high level.", "How are you going to drive in this situation and why?", "What's the straight-line distance to the nearest car?", "What is the angle of the nearest car relative to your heading?", "Is there any lateral deviation from your driving route?", "What should be your next steering action?", "What should be your next acceleration command?", "Is there any moving object around you?", "Describe the position of the car relative to your heading.", "What is your current lateral position relative to your route?", "What would be a safe driving action given the detected car's details?", "What is the speed of the detected car?", "How far is the detected car from you?", "What angle should you adjust your steering to avoid collision?", "Why is it important to note the angle of the detected car?", "Is the detected car in motion?", "What should you be cautious of given the car's position?", "What action should be taken to maintain alignment on your driving route?", What should you avoid in this situation to prevent collision?, "What considerations are necessary for the detected car's speed?", "What's the importance of your current lateral position in planning the next action?", Why did you brake just now?, "What factors are influencing your next driving decision?", "Is there any obstacle directly ahead?", "How should you interpret the car's angle for your steering decision?", "What immediate adjustments are necessary for safe driving?", "How does the detected car's speed impact your driving action?", "What should be your focus given the detected car's proximity and angle?", "What safe action is suggested based on the current scenario?", "What should you avoid in this situation to prevent collision?", "Is there a need for a speed adjustment?", "How will your steering angle change based on the detected car's angle?", "What should you consider for maintaining a safe path?", "How would you describe the current traffic conditions?", "What immediate action is necessary given your current lateral position?", "What factors need to be monitored to ensure safe navigation?", "Is the detected car influencing your path directly?", "What is the priority in adjusting your speed and direction?", " "}

A.12 Loss weights

Our loss weighting analysis provided several key insights: First, increasing the image reconstruction weight (from 0.5 to 2) was found to degrade decision-making performance while maintaining language generation capabilities. Second, elevating the language loss weight (from 0.1 to 10) significantly improved conversational quality but substantially compromised driving performance. Third, boosting the action prediction weight (from 0.1 to 10) led to markedly better driving performance with only negligible effects on language outputs. Based on these findings, we determined the optimal weighting scheme for autonomous driving applications to be: language loss weight = 0.1, action loss weight = 10, and image reconstruction weight = 0.5. This configuration achieved the best balance between driving performance and auxiliary task capabilities.

A.13 Further results on VLASCD in answering questions

Other Sensors Input_33: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you're heading are respectively 2.45 m/s, 9.73 m, 179.82 °.You are now -0.009m laterally away from your driving route.
Question_33: What factors need to be monitored to ensure safe navigation?
VLASCD_pred_33: You need to monitor the speed, straight-line distance from you, and angle in the direction you're heading are respectively 2.45 m/s, 9.73 m, 179.82 °.
Openvla_pred_33: monitored to safe to ensure safe and to ensure safe monitored to safe
DriverGPT4_pred_33: 00000000000000000000000000000000000
VLASCD_image_pred_33: speed, straight-line distance from you, and angle in the direction you're heading are respectively 2.45 m/s, 9.73 m, 179.82 °.You are now -0.009m laterally away from your driving route.
VLASCD_language_pred_33: -

Other Sensors Input_34: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you're heading are respectively 2.49 m/s, 9.69 m, 179.81 °.You are now -0.011m laterally away from your driving route.
Question_34: What should be your next steering action?
VLASCD_pred_34: Your next steering action should be to turn the steering wheel to the left.
Openvla_pred_34: your steering action and your next steering action and your steering action and your steering action and your steering action and
DriverGPT4_pred_34: 0000000000000000000000000000000000
VLASCD_image_pred_34: right
VLASCD_language_pred_34: -

Other Sensors Input_35: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you're heading are respectively 2.47 m/s, 9.66 m, 179.79 °.You are now -0.012m laterally away from your driving route.
Question_35: What's the importance of your current lateral position in planning the next action?
VLASCD_pred_35: The current lateral position is important in planning the next action because if you are too far away from your driving route.
Openvla_pred_35: your current position in planning the next action
DriverGPT4_pred_35: 000000000000000000000000000000000
VLASCD_image_pred_35: Your current lateral

position is important in planning the next action because if you don’t plan the next action, you will not be able to reach your destination.
VLASCD_language_pred_35: -

Other Sensors Input_36: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you’re heading are respectively 2.41 m/s, 9.64 m, 179.78 °.You are now -0.014m laterally away from your driving route.
Question_36: What safe action is suggested based on the current scenario?
VLASCD_pred_36: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you’re heading are respectively.
Openvla_pred_36: suggested based on the current safe action on the current
DriverGPT4_pred_36:
0000000000000000000000000000000000000
VLASCD_image_pred_36: The car is speed, straight-line distance from you, and angle in the direction you’re heading are respectively 2.41 m/s, 9.64 m, 179.78 °.You are now -0.014m laterally away from your driving route.
VLASCD_language_pred_36: - -

Other Sensors Input_37: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you’re heading are respectively 2.37 m/s, 9.63 m, 179.77 °.You are now -0.016m laterally away from your driving route.
Question_37: Is there any lateral deviation from your driving route?
VLASCD_pred_37: There is no lateral deviation from your driving route.
Openvla_pred_37: driving route from your route and your route from your driving route from your route from your
DriverGPT4_pred_37:
0000000000000000000000000000000000000
VLASCD_image_pred_37: Yes, there is lateral deviation from your driving route.
VLASCD_language_pred_37: -

Other Sensors Input_38: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you’re heading are respectively 2.37 m/s, 9.63 m, 179.76 °.You are now -0.018m laterally away from your driving route.
Question_38: How will your steering angle change based on the detected car’s angle?
VLASCD_pred_38: Your steering angle will change based on the detected car’s angle.
Openvla_pred_38: your steering on the detected car’s steering
DriverGPT4_pred_38:
0000000000000000000000000000000000
VLASCD_image_pred_38: The steering angle will change based on the detected car’s angle.

Other Sensors Input_39: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you’re heading are respectively 2.41 m/s, 9.63 m, 179.75 °.You are now -0.020m laterally away from your driving route.
Question_39: Is there any moving object around you?
VLASCD_pred_39: Yes, there is a car. Its speed, straight-line distance from you, and angle in the direction you’re heading are respectively 2.
Openvla_pred_39: moving object of you and your work with your work?
DriverGPT4_pred_39:
0000000000000000000000000000000000000
VLASCD_image_pred_39: Yes, there is a car.
VLASCD_language_pred_39: - -

Other Sensors Input_40: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you’re heading are respectively 2.45 m/s, 9.63 m, 179.74 °.You are now -0.022m laterally away from your driving route.
Question_40: Is there any moving object around you?
VLASCD_pred_40: Yes, there is a car. Its speed, straight-line distance from you, and angle in the direction you’re heading are respectively 2.
Openvla_pred_40: moving object of you and your work with your work?
DriverGPT4_pred_40:
00000000000000000000000000000000
VLASCD_image_pred_40: Yes, there is a car.
VLASCD_language_pred_40: - -

Other Sensors Input_41: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you’re heading are respectively 2.46 m/s, 9.63 m, 179.73 °.You are now -0.024m laterally away from your driving route.

Question_41: How does the detected car's speed impact your driving action?

VLASCD_pred_41: The detected car's speed impacts your driving action because if the detected car's speed is faster than your driving speed

Openvla_pred_41: speed with your speed and your speed with your speed speed with your speed with your speed with your speed with your speed with

DriverGPT4_pred_41:
00000000000000000000000000000000000

VLASCD_image_pred_41: The car's speed impacts my driving action because the car's speed is increasing, which means that the car's speed is getting faster

VLASCD_language_pred_41: - -

Other Sensors Input_42: You can see that there is a car. Its speed, straight-line distance from you, and angle in the direction you're heading are respectively 2.43 m/s, 9.63 m, 179.73 °.You are now -0.027m laterally away from your driving route.

Question_42: What is your action and why?

VLASCD_pred_42: My action is to turn left. I am turning left because I want to turn left.

Openvla_pred_42: " your action" and your action and your " your action" and your action and your

DriverGPT4_pred_42:
00000000000000000000000000000000000

VLASCD_image_pred_42: My action is straight-line distance from you, and why is because I am going straight.
My action is straight-line distance from you, and why is because I am going straight. My action is straight-line distance from you, and why is because I am going straight.

VLASCD_language_pred_42: - -

Other Sensors Input_43: You can see that there are 2 cars. Their speed, straight-line distance from you, and angle in the direction you're heading are respectively 5.07 2.4 m/s, 14.89 9.64 m, 14.18 179.72 °.You are now -0.029m laterally away from your driving route.

Question_43: What safe action is suggested based on the current scenario?

VLASCD_pred_43: What safe action is suggested based on the current scenario? What safe action is suggested based on the current scenario? What safe action is suggested based

Openvla_pred_43: What safe action is based on the current? What safe action is based on the current

DriverGPT4_pred_43:
00000000000000000000000000000000000

VLASCD_image_pred_43: straight-line distance from you, and angle in the direction you're heading are respectively 5.07 2.4 m/s, 14.89 9.64 m, 14.18 179.72 °.

VLASCD_language_pred_43: - - - ,- - ,-

Other Sensors Input_44: You can see that there are 2 cars. Their speed, straight-line distance from you, and angle in the direction you're heading are respectively 4.98 2.38 m/s, 14.15 9.66 m, 14.95 179.71 °.You are now -0.032m laterally away from your driving route.

Question_44: Is the detected car in motion?

VLASCD_pred_44: Yes answer: No answer: Yes answer: No answer

Openvla_pred_44: motion in the detected car in motion in the detected car in motion in the detected car in motion in the detected car in motion in the detected car in motion in the detected car in motion in the detected car in motion in the detected car in motion in the detected car in motion in the detected car in motion in the

DriverGPT4_pred_44:
000000000000000000000000000000000000

VLASCD_image_pred_44: Yes, the detected car is in motion. ""

VLASCD_language_pred_44: - -
