Generative Diffusion Language Model for End-to-End Autonomous Driving: Parallel Planning and Explainable Action

Jiaru Zhang
Postdoc researcher
Institute of Physical AI, Purdue University

02/03/2026
IPAI Seminar Series

PURDUE UNIVERSITY | Purdue Digital Twin Lab

About Jiaru Zhang

Sep. 2015 – Jun. 2019
Shanghai, China
Bachelor's Degree in Computer Science

Shanghai, China
PhD's Degree in Computer Science
Sep. 2019 – Jun. 2024

Nov. 2022 – Aug. 2023
Beijing, China
Research Intern

Jan. 2025 – Current
West Lafayette, IN
Postdoc Researcher

Supervisors

Ziran Wang (CCE)
*Autonomous vehicles

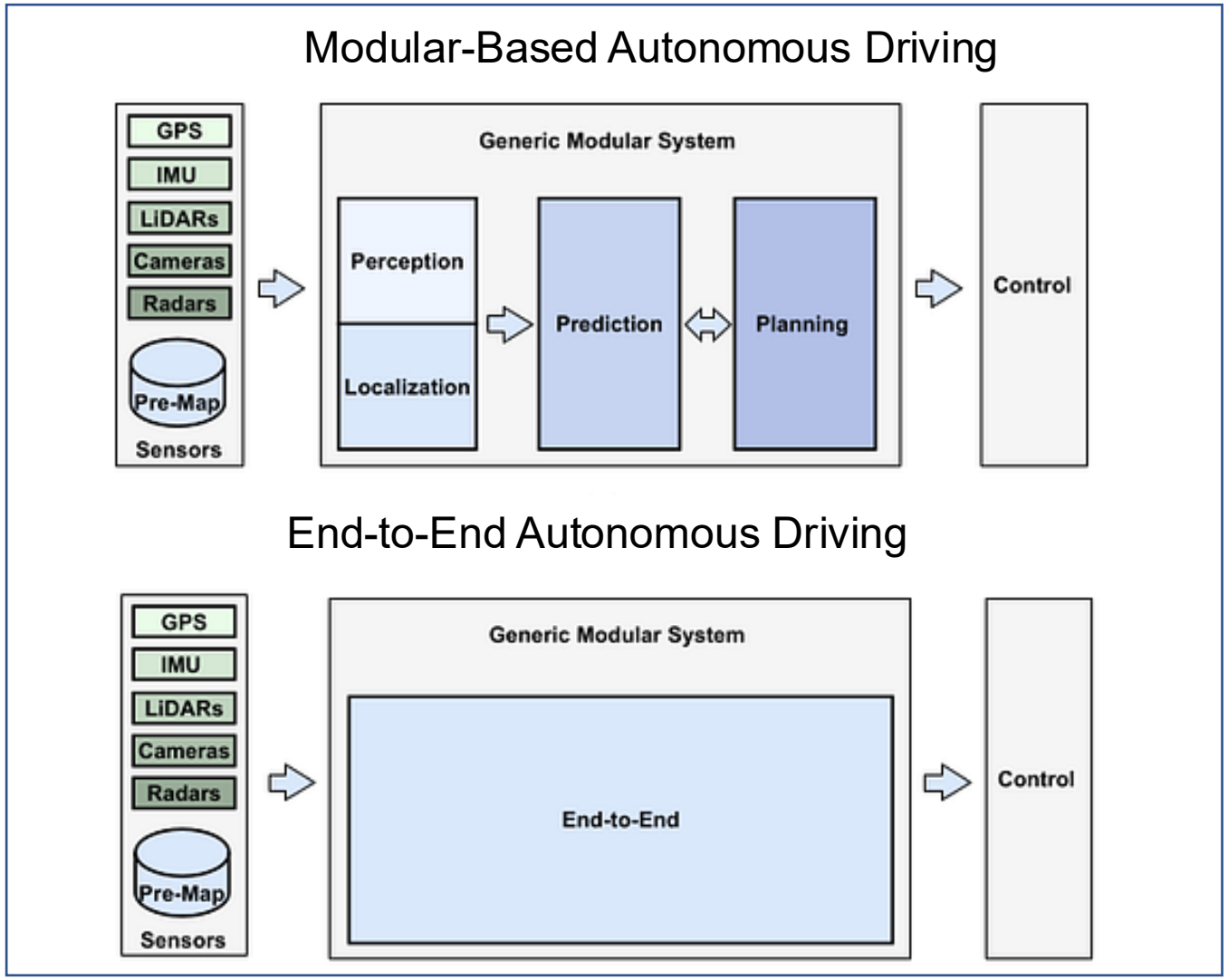
Ruqi Zhang (CS)
*Machine learning

Research Interests:

- Generative AI
- Bayesian Deep Learning
- Applications in Autonomous Driving

End-to-End Autonomous Driving

[Definition] End-to-end autonomous driving systems are fully differentiable programs that take raw sensor data as input and produce a plan and/or low-level control actions as output.



Traditional Modular Pipelines

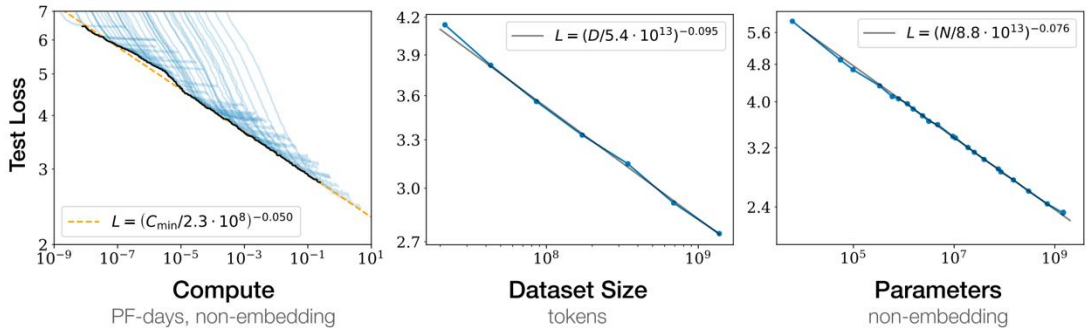
- **Decompose** into perception → prediction → planning → control
- **Error accumulation** across modules

End-to-End Autonomous Driving

- **Directly** map raw sensor data → driving decisions
- Enables **global optimization**
- Reduce **redundant computational costs.**

LLMs/VLMs in End-to-End Autonomous Driving

Instruction-following and scalability explains the success of autoregressive LLMs/VLMs.



Instruction Following

- Conditional Generation: $p(response|prompt)$.
- AR generate responses or actions that **align with human intent.**
- The human can **interact** with the AI systems.

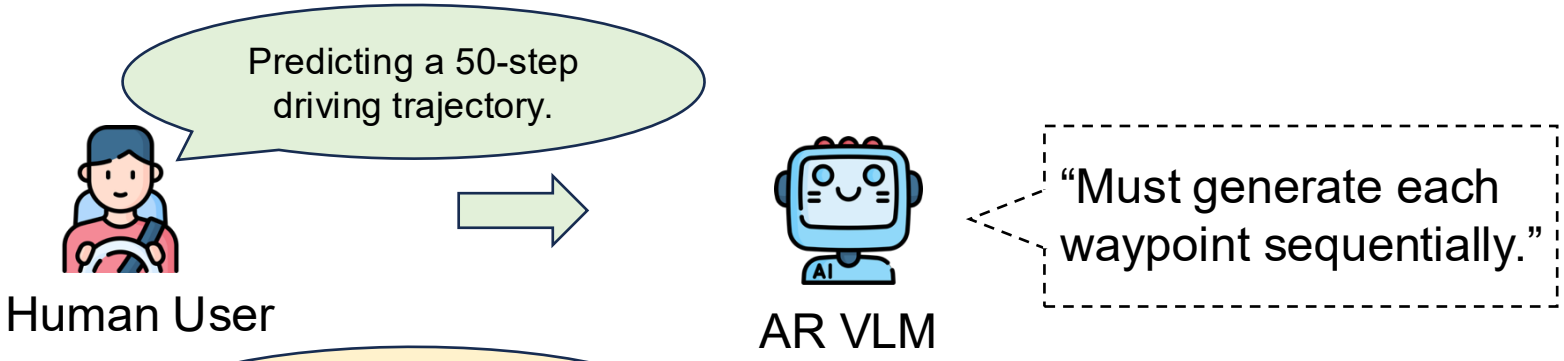
Scalability

- Performance **improves predictably** with more data, parameters, and compute.
- Larger models → better at reasoning, multimodal fusion, and instruction-following.
- Autoregressive **training scales stably.**

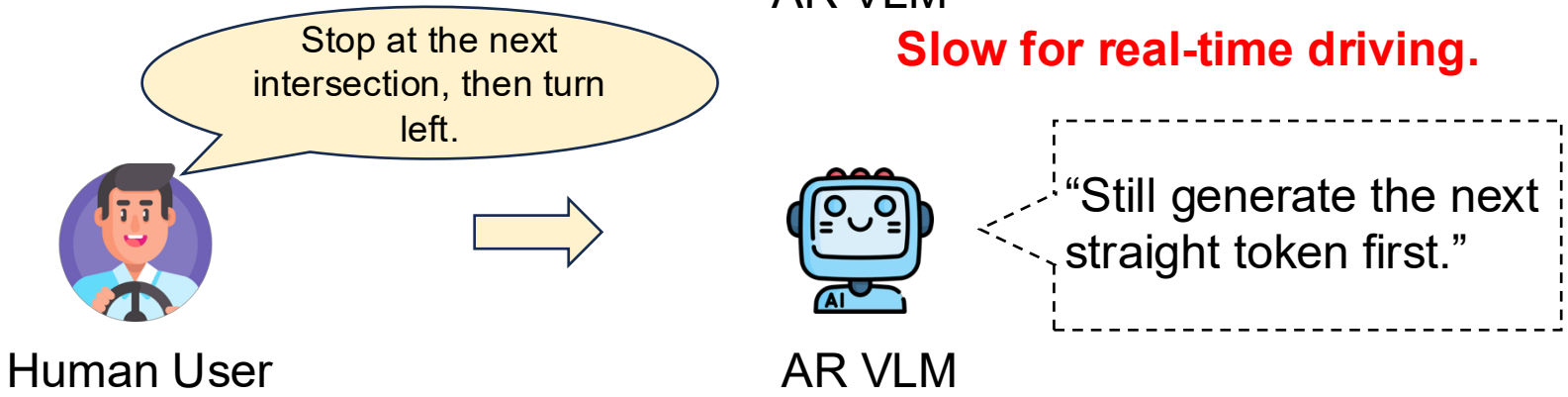
Limitations of Autoregressive VLMs

AR models suffer from high latency and reversal curse from the left-to-right generation paradigm.

a) Generation speed is linear to generation length.



a) Lack of bidirectional reasoning.



A diffusion-based generation method different from the traditional left-to-right approach

The diffusion generating method is integrated into the VLM.

Prompt: Explain what artificial intelligence is

Core Idea

- **Forward process**: Progressively corrupt structured data into noise.
- **Reverse process:** Learn to iteratively denoising and reconstructing data from noise.

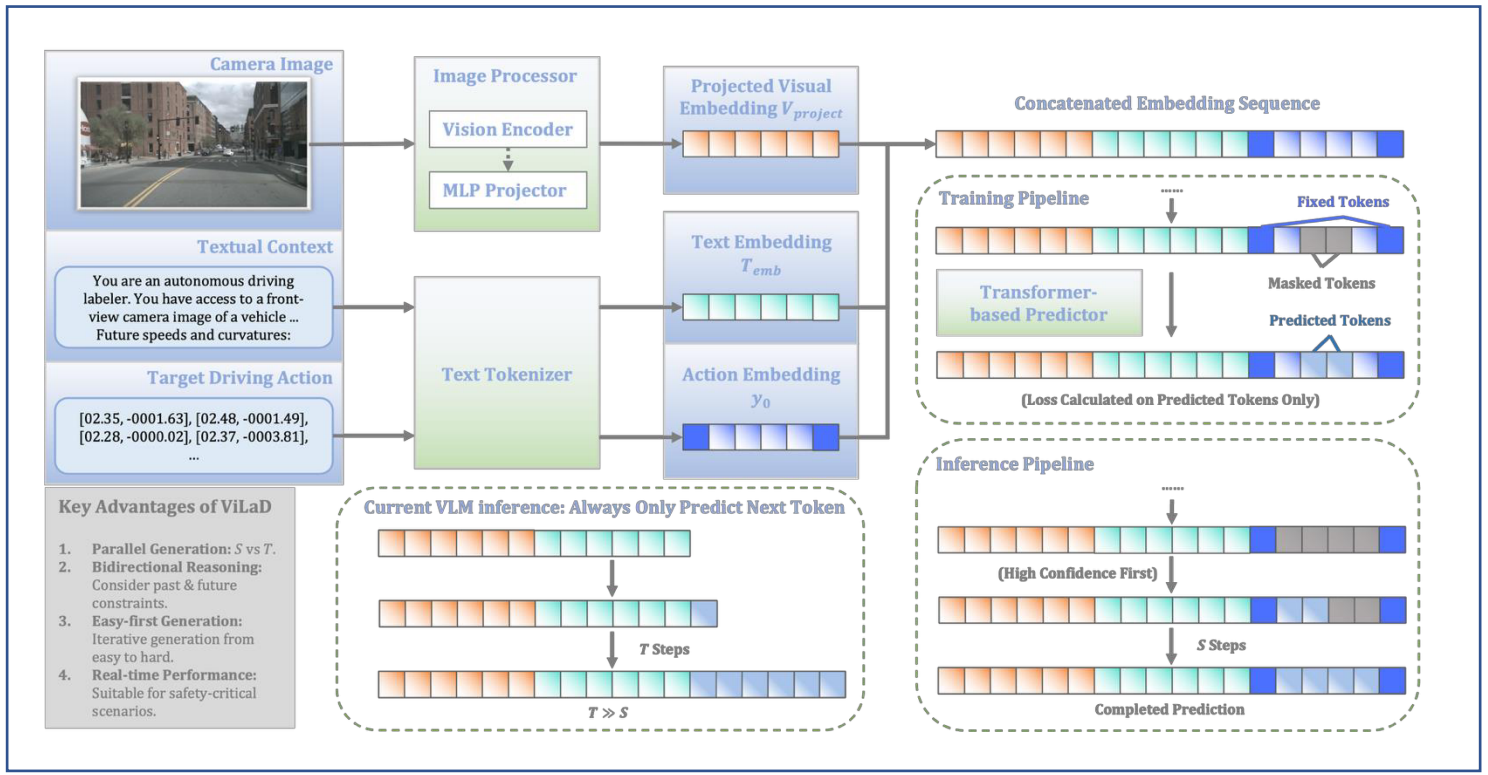
Key Properties

- **Parallel generation**: Tokens/decisions refined in groups, not strictly sequential.
- **Bidirectional reasoning**: Considers both past and future context.
- **Easy-to-hard refinement**: Confident parts generated first, uncertain parts refined later.

Vision Language Diffusion Models (ViLaD) is what we want to leverage in the autonomous driving task.

ViLaD for End-to-End Autonomous Driving

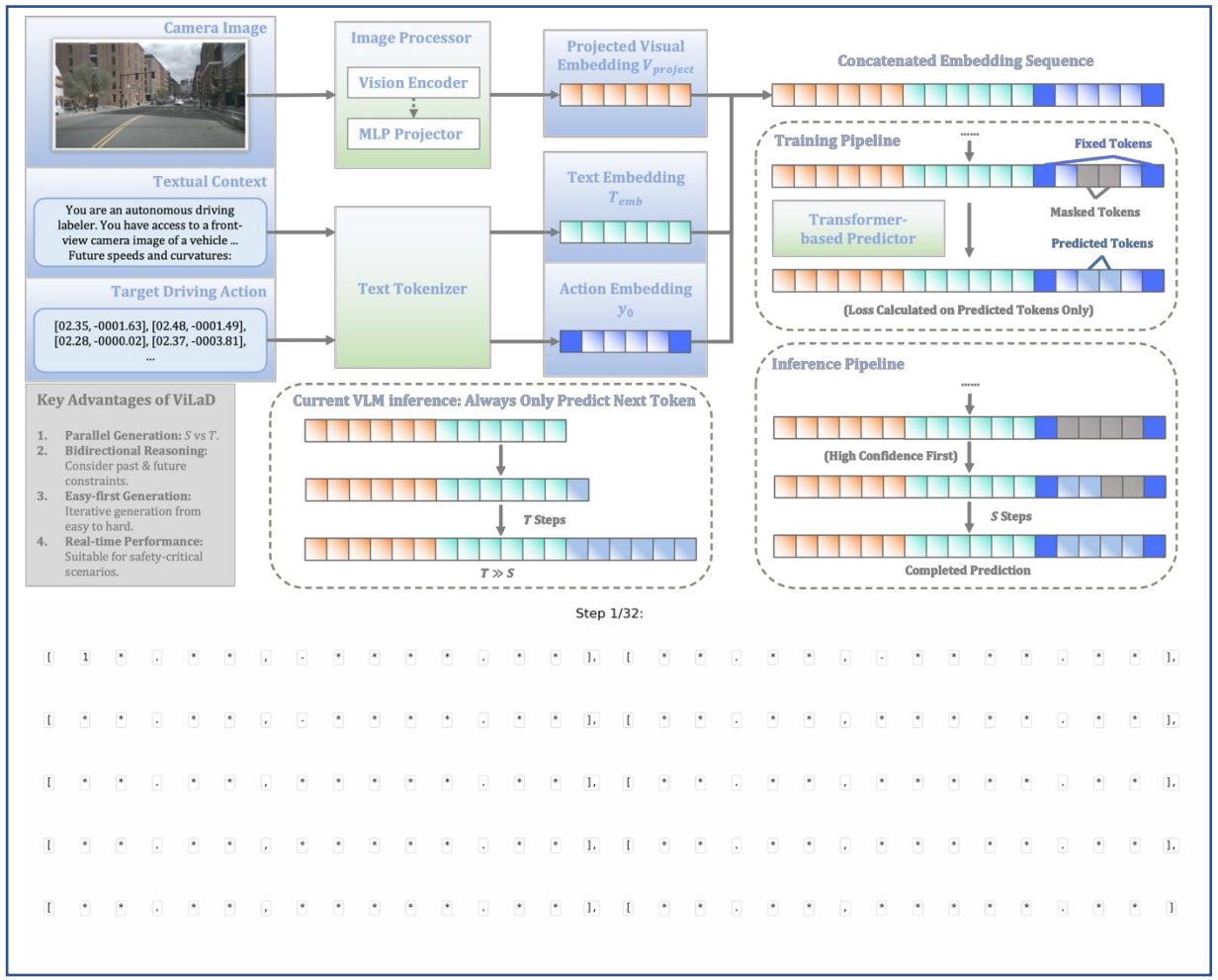
[Motivation] Moving from autoregressive Vision Language Models to Large Vision Language Diffusion Models for autonomous driving.



- **Vision Encoder:** Processes camera images to extract key visual features.
- **MLP Connector**: Aligns the visual features with the language model's understanding.
- **Diffusion Language Backbone** (LLaVA-based Structure): Generates driving decisions using a **masked diffusion technique.**
- **Tokenizer:** Generate textual tokens.

We want to increase both the performance and accuracy of the current models.

ViLaD for End-to-End Autonomous Driving



Training

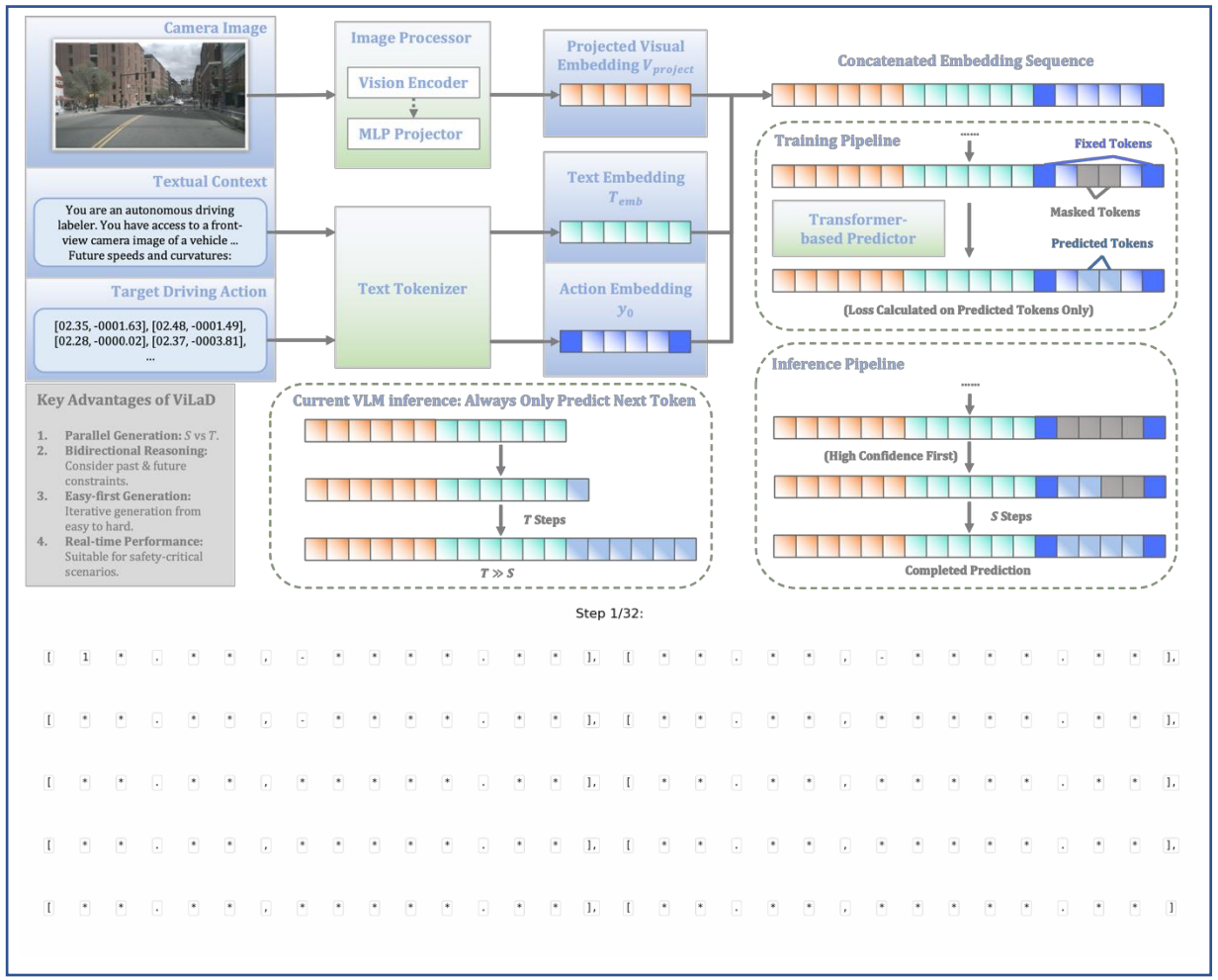
- Parts of the output are **randomly masked**, and the model learns to predict these masked portions
- Objective: Accurately **predict the masked** driving decision tokens

Masked Language Modeling

the chef cooked the meal

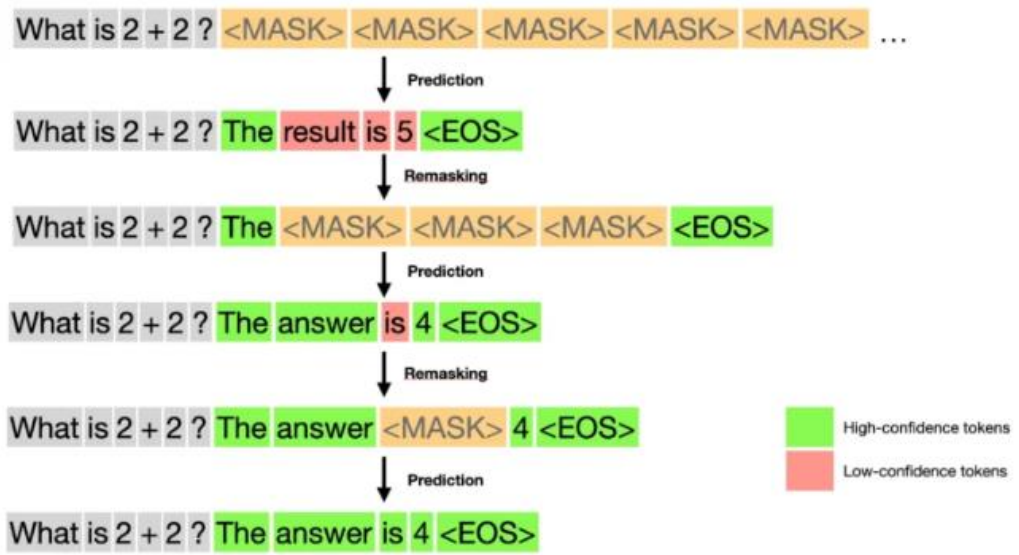
Train to predict masked tokens.

ViLaD for End-to-End Autonomous Driving



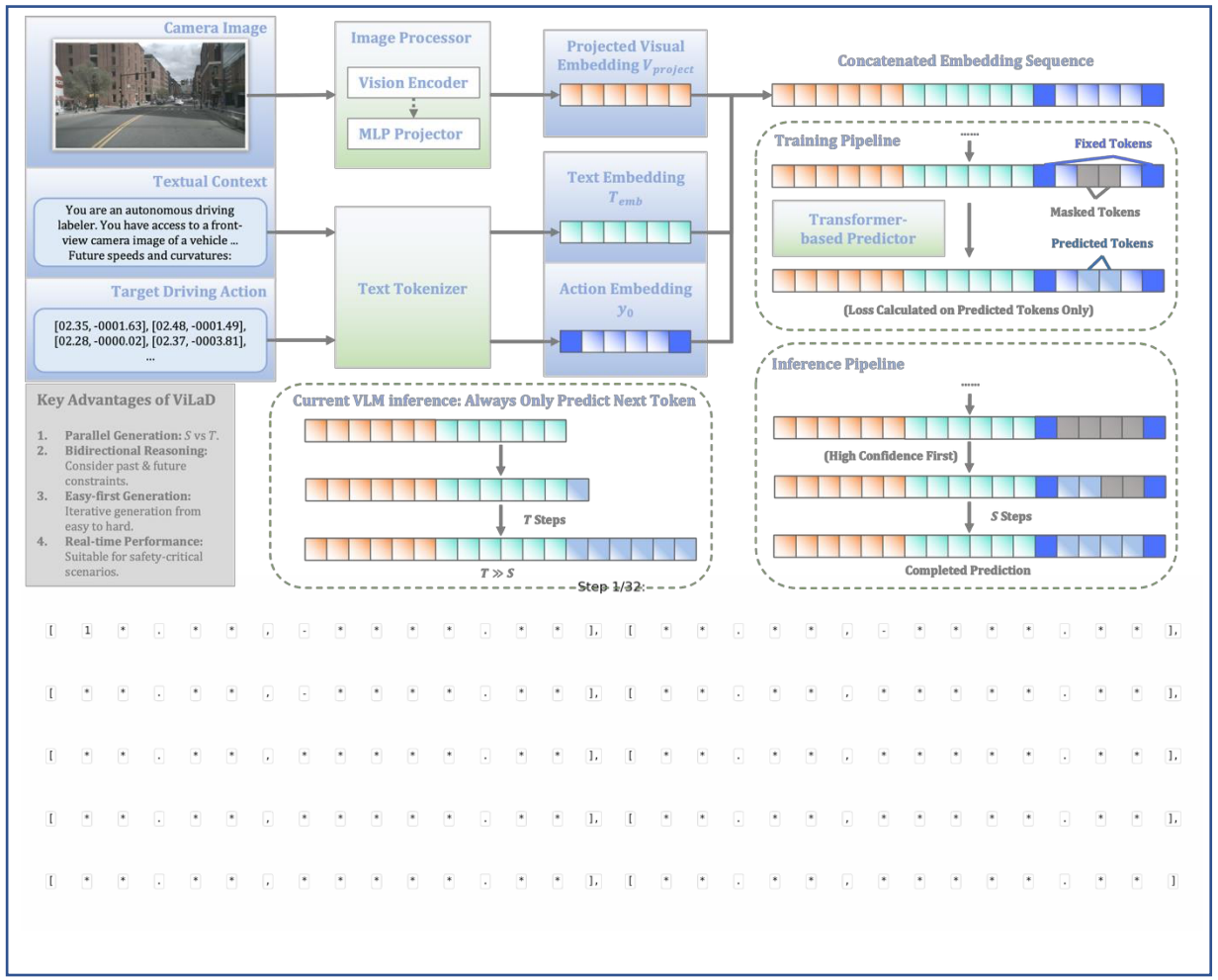
Inference

- The process starts with **a fully masked** driving decision (except fixed tokens).
- The model then progressively **demasks** the decision over several steps.



Inference from fully masked tokens.

ViLaD for End-to-End Autonomous Driving



Important Contributions

- The **first** using Vision Language Diffusion Models in autonomous driving.
- To improve the quality and the speed of parallel token generation, **all tokens that exceed a certain confidence threshold** will be unmasked.
- During both training and inference, **certain tokens are kept fixed** and are never masked.

Implementation Details

- Steps: 32
- Confidence Threshold: 0.5
- dLLM Cache is utilized
- Base checkpoint: LLaDA-V
- LoRA Fine-tuning

Experiment Setup

Our experiment contains both open benchmarks and real-world case studies.

Dataset

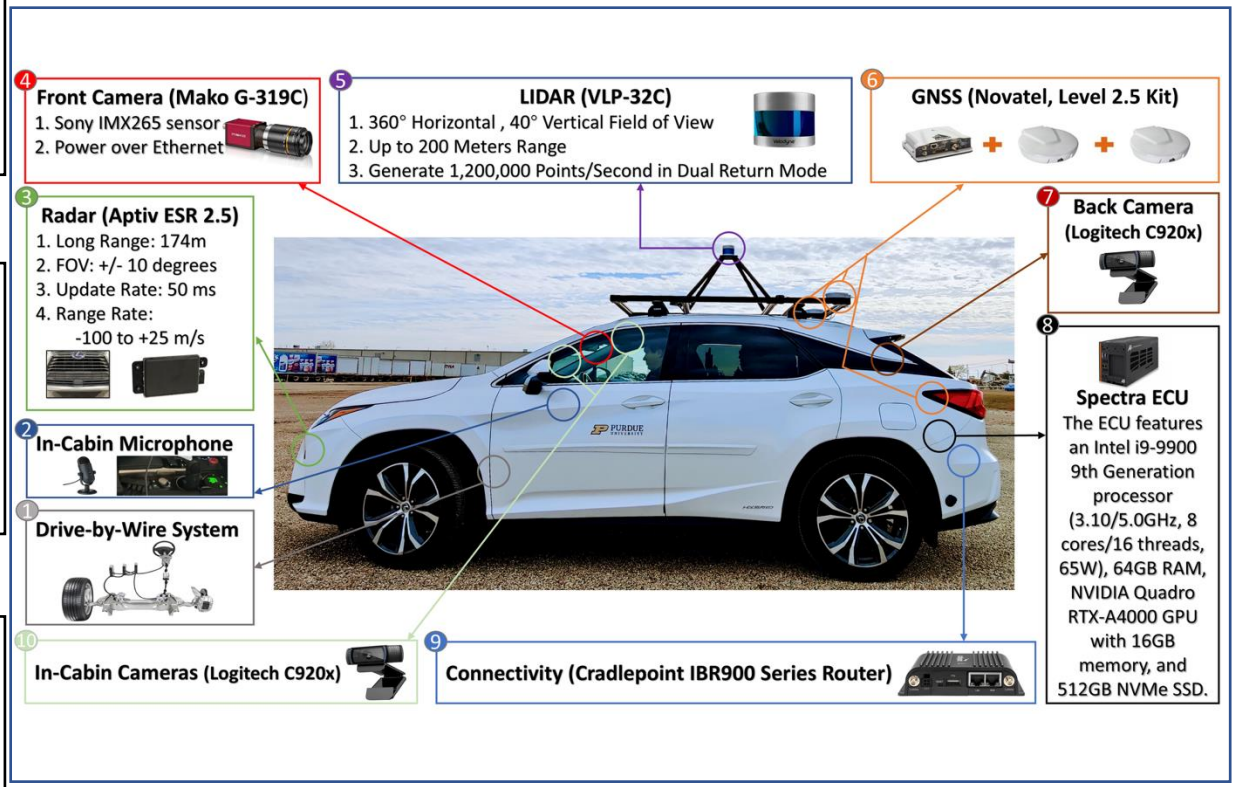
- **NuScenes** benchmark.
- Input: front camera images + textual driving context
- Output: structured driving decisions.

Baselines

- **Autoregressive VLMs**:LLaVA-1.6-7B; Llama-3.2-11B; Qwen2-VL-7B.
- **Evaluated under**: zero-shot, chain-of-thought prompting, supervised fine-tuning (SFT).

Evaluation Metrics

- **Accuracy**: L2 error of predicted trajectories.
- **Safety**: Failure rate (≥10m error at 1s horizon).
- **Efficiency**: Inference latency.



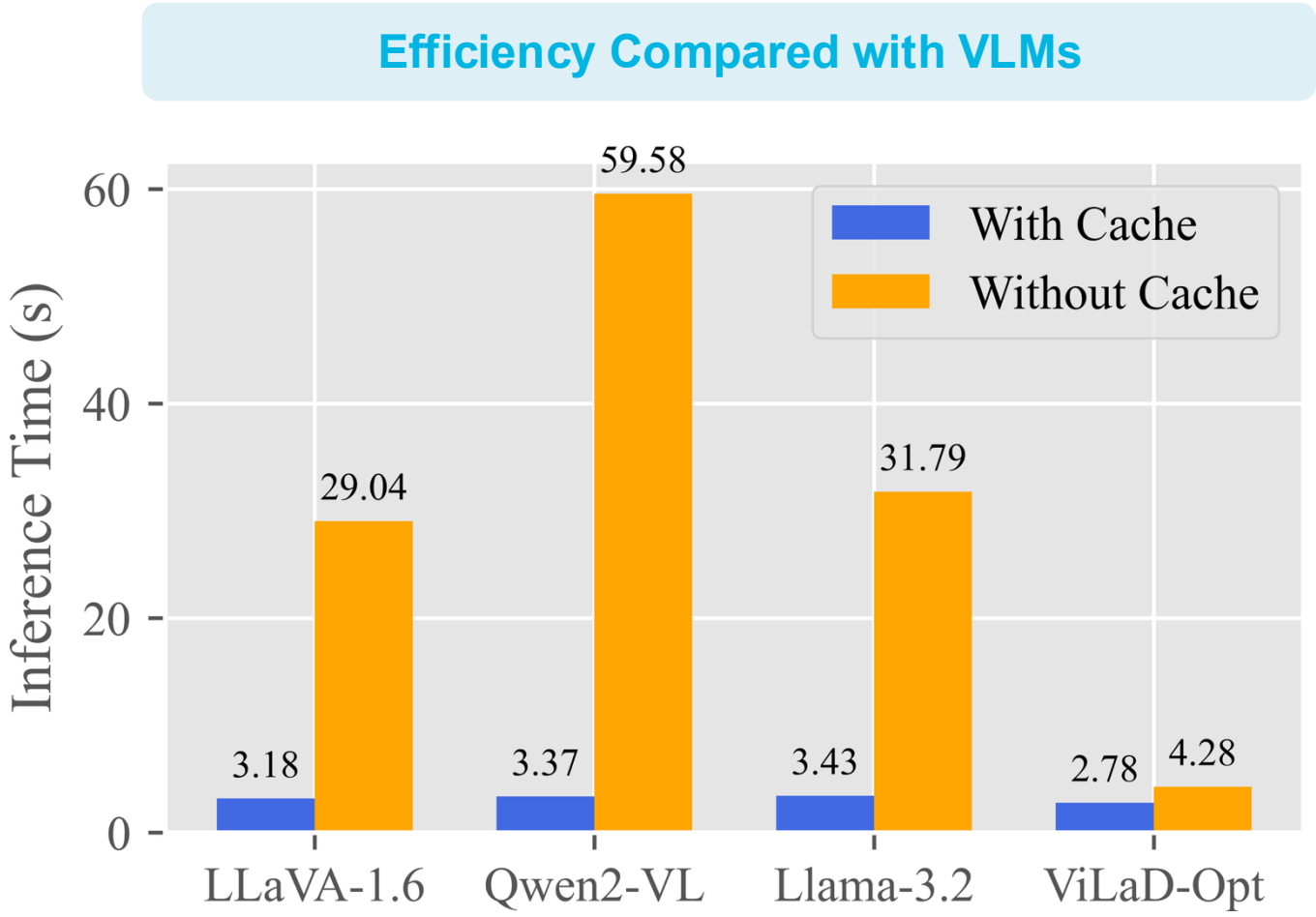
Planning Accuracy Results

Planning Performance Compared with VLMs

Method	Learning	L2 (m) 1 s (↓)	L2 (m) 2 s (↓)	L2 (m) 3 s (↓)	L2 (m) Avg (↓)	Failure Rate (%) (↓)
LLaVA-1.6-7B*	Zero-Shot	1.66	3.54	4.54	3.24	4.06
Llama-3.2-11B*		1.50	3.44	4.04	3.00	23.92
Qwen2-VL-7B*		1.22	2.94	3.21	2.46	24.00
ViLaD-Zero		1.09	2.51	3.74	2.45	0.03
LLaVA-1.6-7B*	Open-EMMA	1.49	3.38	4.09	2.98	6.12
Llama-3.2-11B*		1.54	3.31	3.91	2.92	22.00
Qwen2-VL-7B*		1.45	3.21	3.76	2.81	16.11
ViLaD-CoT		1.24	2.74	4.13	2.71	0.17
LLaVA-1.6-7B	SFT	0.91	2.50	3.44	2.28	55.25
Llama-3.2-11B		0.80	2.31	3.10	2.07	0.06
Qwen2-VL-7B		1.32	2.94	3.98	2.74	0.03
ViLaD-SFT		**0.79**	**1.92**	2.83	1.85	**0.00**
ViLaD-Opt	Efficient Optimized SFT	0.81	1.93	**2.69**	**1.81**	**0.00**

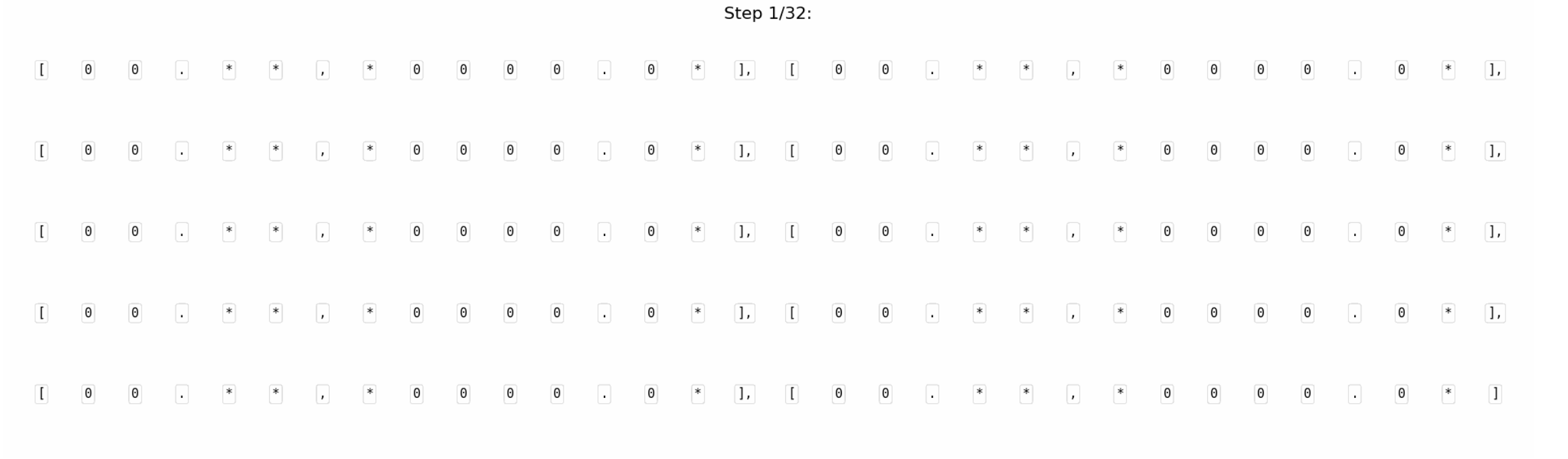
ViLaD even outperforms strongest autoregressive models.

Efficiency Results



Our ViLaD can provide quicker response than traditional LLMs/VLMs

Visualization



Generate the output within 11 steps (suppose to be 32 steps).

Visualization

Easy-first, multi-tokens, and bi-directional generation.

Case Study: Real-World Deployment

Real-World Case Study

Smart Parking Scenarios

In this scenario, the ego vehicle searches for the most suitable spot based on the human's requirement and its awareness of the driving environment. Our goal is to assess the framework's planning capability and how effectively it responds to human instructions.

Scenario

- **Parking Selection:** Choosing safe parking spots from commands.

Case Study: Real-World Deployment

Real-World Case Study

End-to-End Planning Scenarios

In this scenario, our model produces planning trajectories conditioned on the human's instruction Our goal is to assess the framework's planning ability, instruction-following capability, generation latency, and the smoothness of the vehicle's transition across different trajectories.

Scenario

- **Parking Selection:** Choosing safe parking spots from commands.
- **E2E Instruction Following:** Executing navigation commands (e.g., "turn left at the next intersection").

Case Study: Real-World Deployment

Real-World Case Study

Scenario	Model Performance		Safety Metrics		Comfort Metrics	
	Latency (s)	Success Rate (%)	Long. Vel. Var.(m^2/s^2)	Lat. Vel. Var.(10^{-4} m^2/s^2)	Max Long. Accel.(m/s^2)	Max Lat. Accel.(m/s^2)
Parking Selection	0.18	95.18	1.07	2.66	1.43	1.12
E2E Instruction Following	0.32	85.91	0.62	1.53	1.39	0.68

Model Performance

- Low latency: 0.18 – 0.32s.
- High success rate: 95% (parking), 86% (instruction following).

ViLaD achieves real-time (<0.5 s latency), safe, and comfortable performance in real-world vehicle tests.

Summary of for ViLaD

- New Paradigm
  - **First** to adapt **Large Vision–Language Diffusion** Models (LVLD) for end-to-end autonomous driving.
  - Overcomes **autoregressive bottlenecks** (latency, unidirectionality).

- Technical Innovations
  - **Fixed Pattern Training** → stable, efficient learning.
  - **Confidence-Aware Decoding** → parallel, faster, and accurate generation.
  - Integration with **dLLM Cache** for inference acceleration.

- Real-World Impact
  - **State-of-the-art results** on nuScenes (1.81 m avg L2, ~0% failure).
  - **Real-time efficiency** (Around 2 s for 8B model; <1 s for 1B model).
  - On-vehicle deployment: interactive parking with 95% success rate and safe maneuvers.

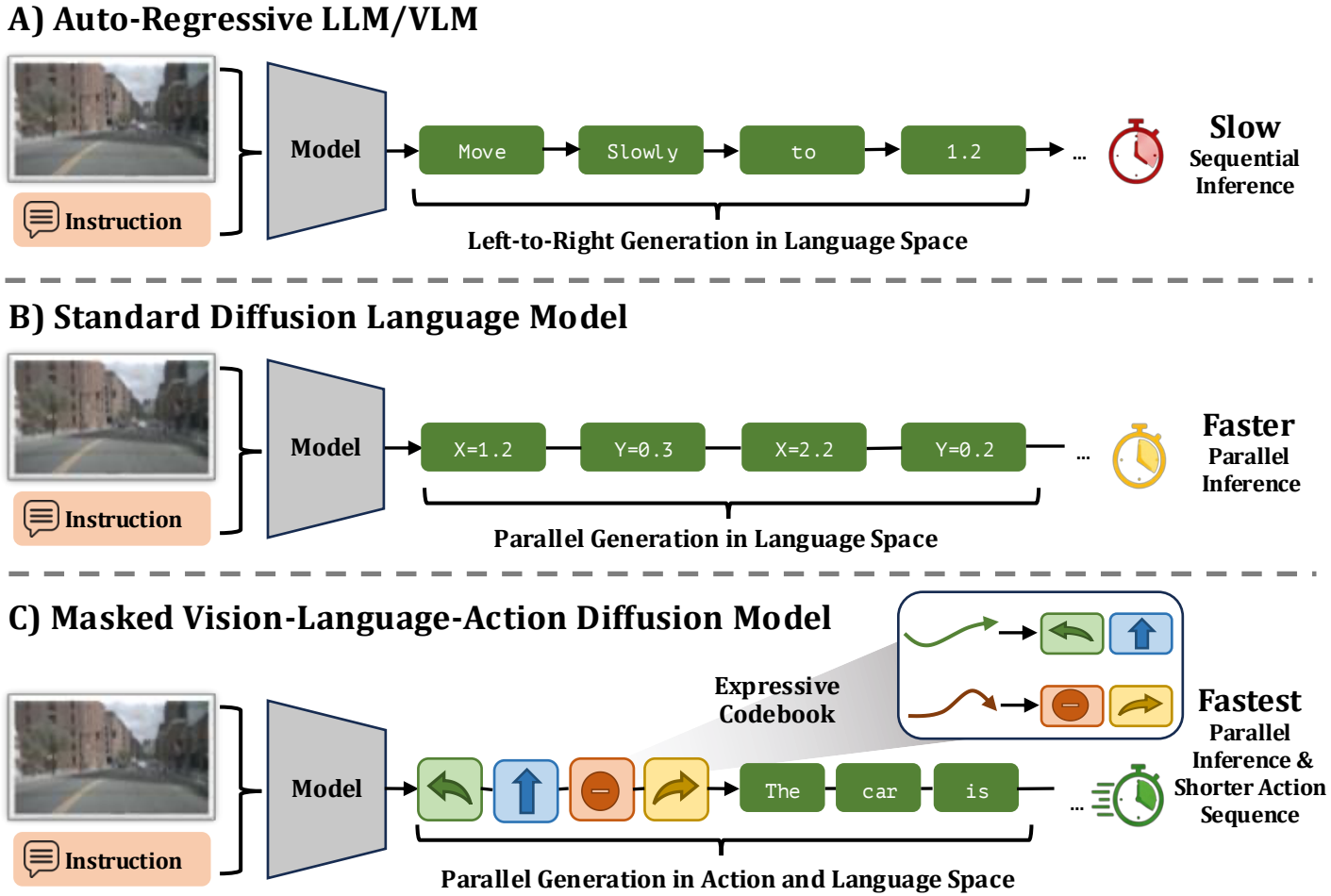
Next Step: Masked Vision-Language Action Diffusion (MVLAD)

Problem

- Language cannot efficiently represent driving actions
- ViLaD does not provide driving explanations

Solution

- **VLA modeling**: Efficient modeling of driving actions
- **Explanation**: Explainable driving decisions



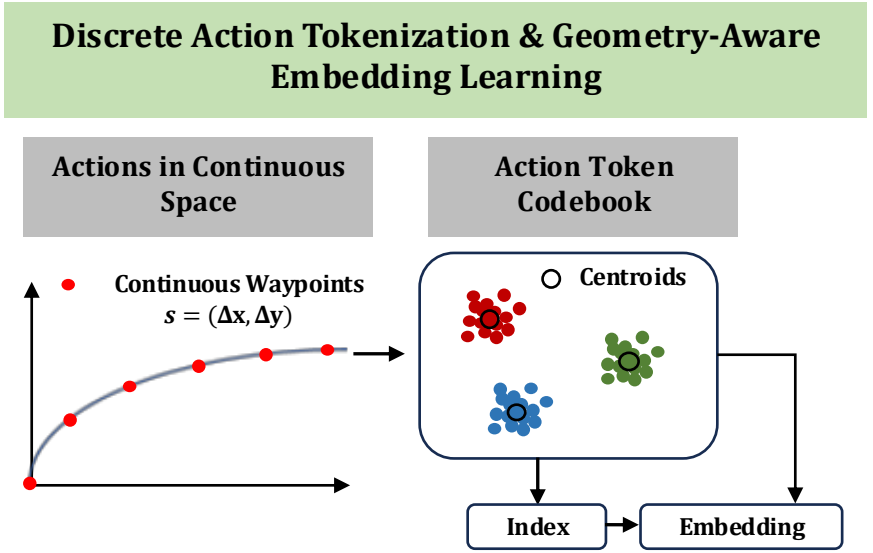
Action Token Modeling

Problem

- How to process the continuous driving action space with a finite vocabulary?

Solution

- Optimization target: Construct a representative centroid set C to approximate all waypoints $\boldsymbol{w}$:
  - $\min \sum_{i=1}^{M} \min_{\{c \in C\}} |\boldsymbol{w_i} - c|$
- Solving this optimization target with **k-means.**



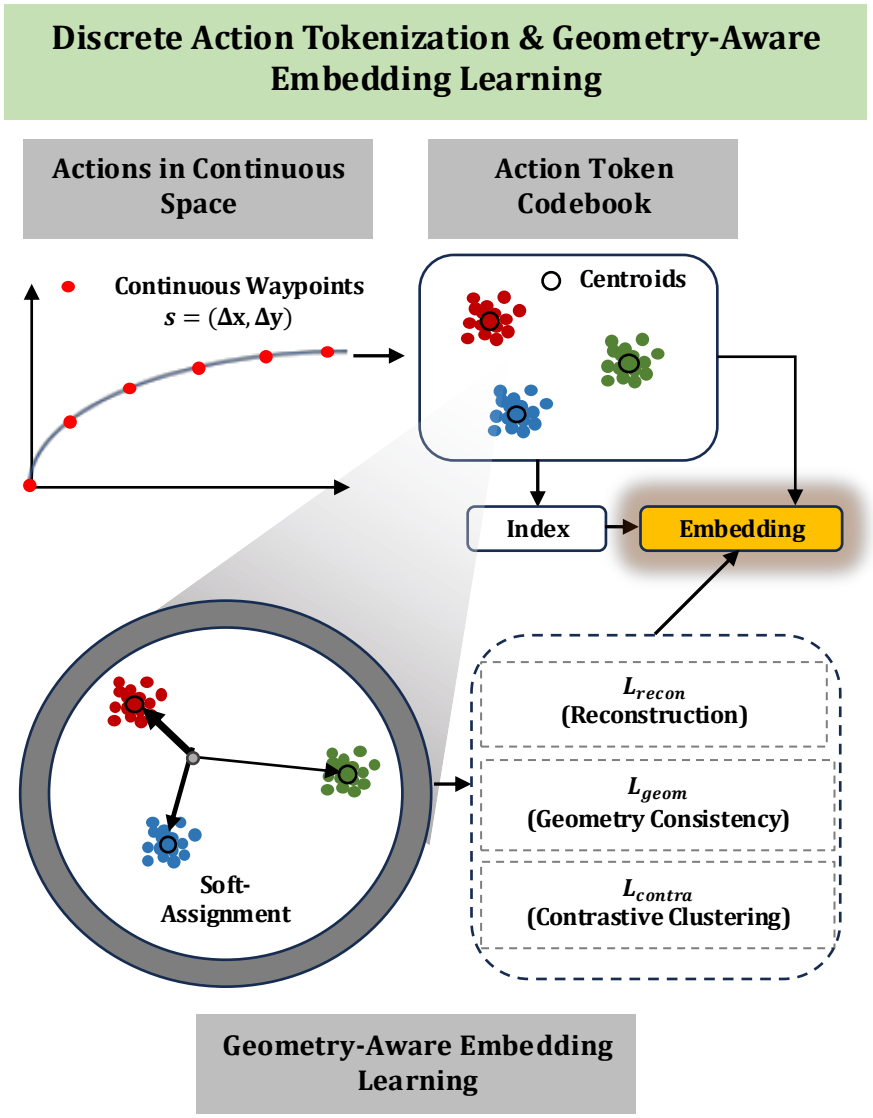
Action Token Embedding

Problem

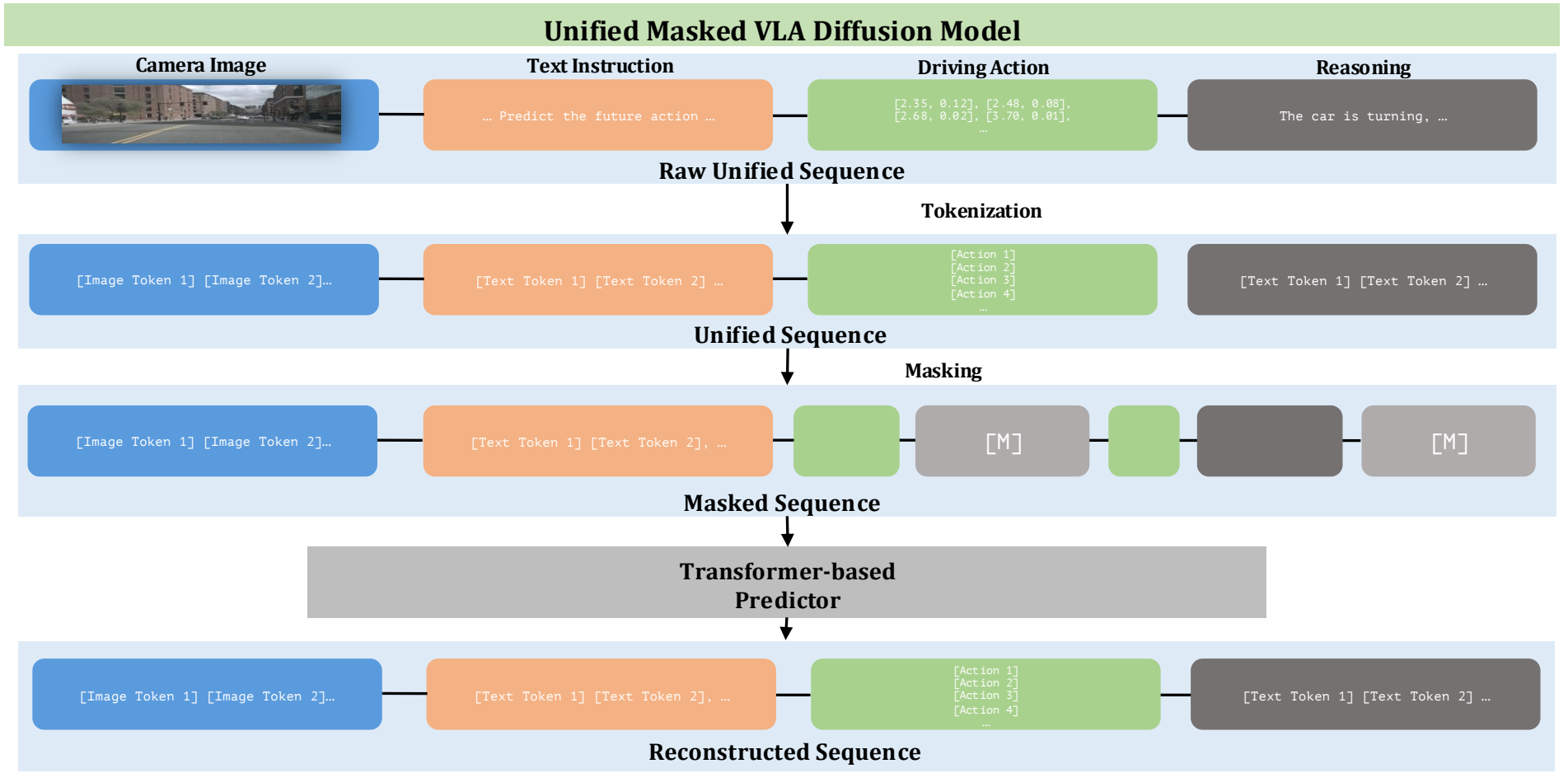
- How to set the embeddings of these new action tokens?

Solution

- **Motivation:** Physically neighbors should have similar embeddings
- **Embedding learning:**
  - Soft-assignment
  - Reconstruction loss
  - Geometry consistency loss
  - Contrastive clustering loss



Unified Modeling and Training



Unified modeling of perception, action, and explanation. The model learns both ”how to drive” and “why drive like this” in a unified sequence.

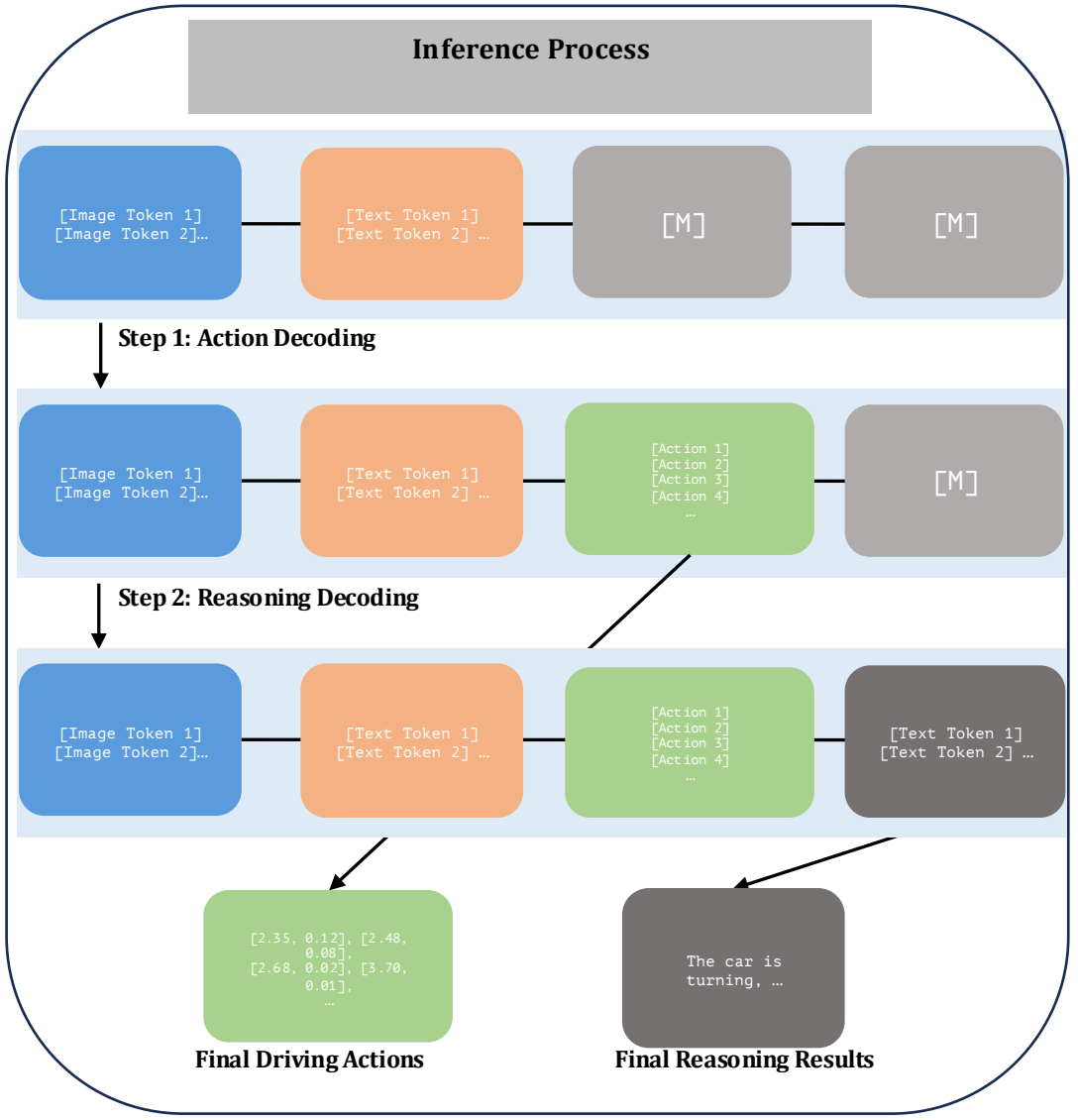
Inference Speed

Problem

- The explanation sequence is much longer than actions.
- Unified sequence results in slow planning speed.

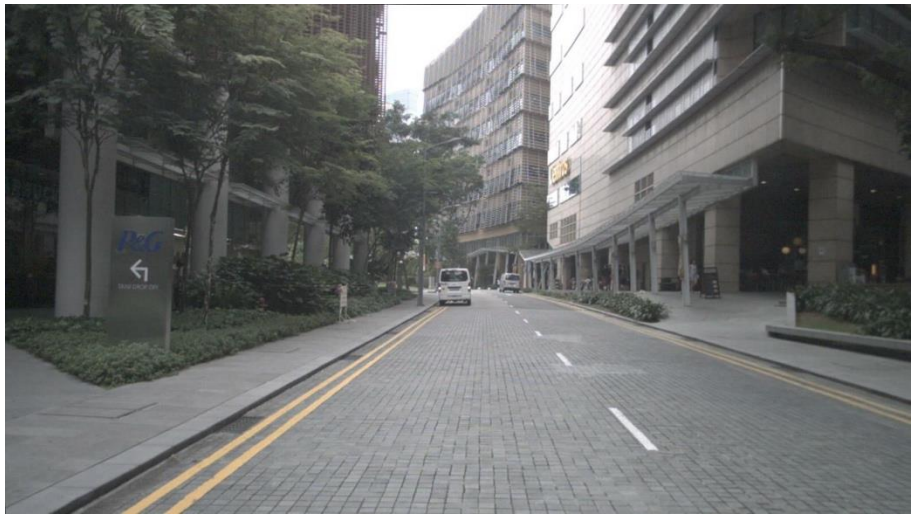
Solution

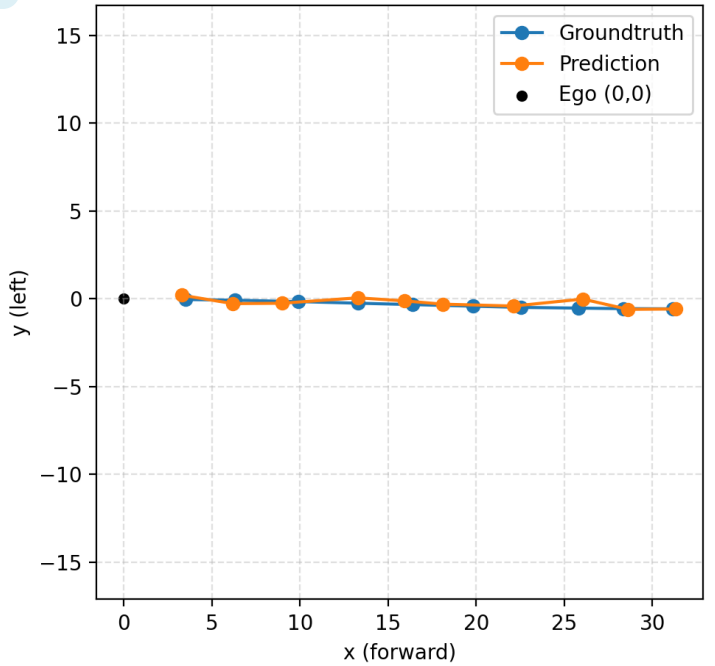
- Step 1: Action decoding
- Step 2: Reasoning decoding
- Results: Fast driving decision and consistent driving explanation.



Results: Examples

Examples

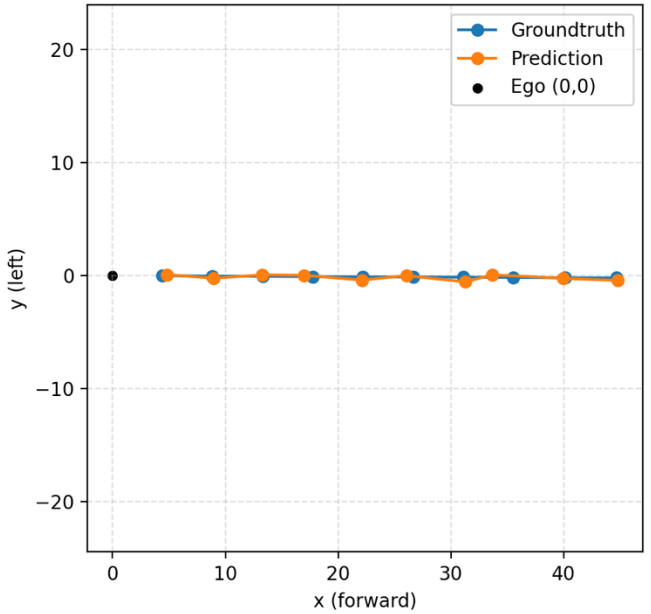




- **Groundtruth Explanation:**
  - Narration: the car is **cruising steadily down the intersection**
  - Reasoning: because the **traffic light is green** and the **path forward is clear** of any vehicles or obstacles
- **Predicted Explanation**
  - Narration: the car **proceeds straight through the intersection**
  - Reasoning: because the **traffic lights are green**, **allowing are passage**

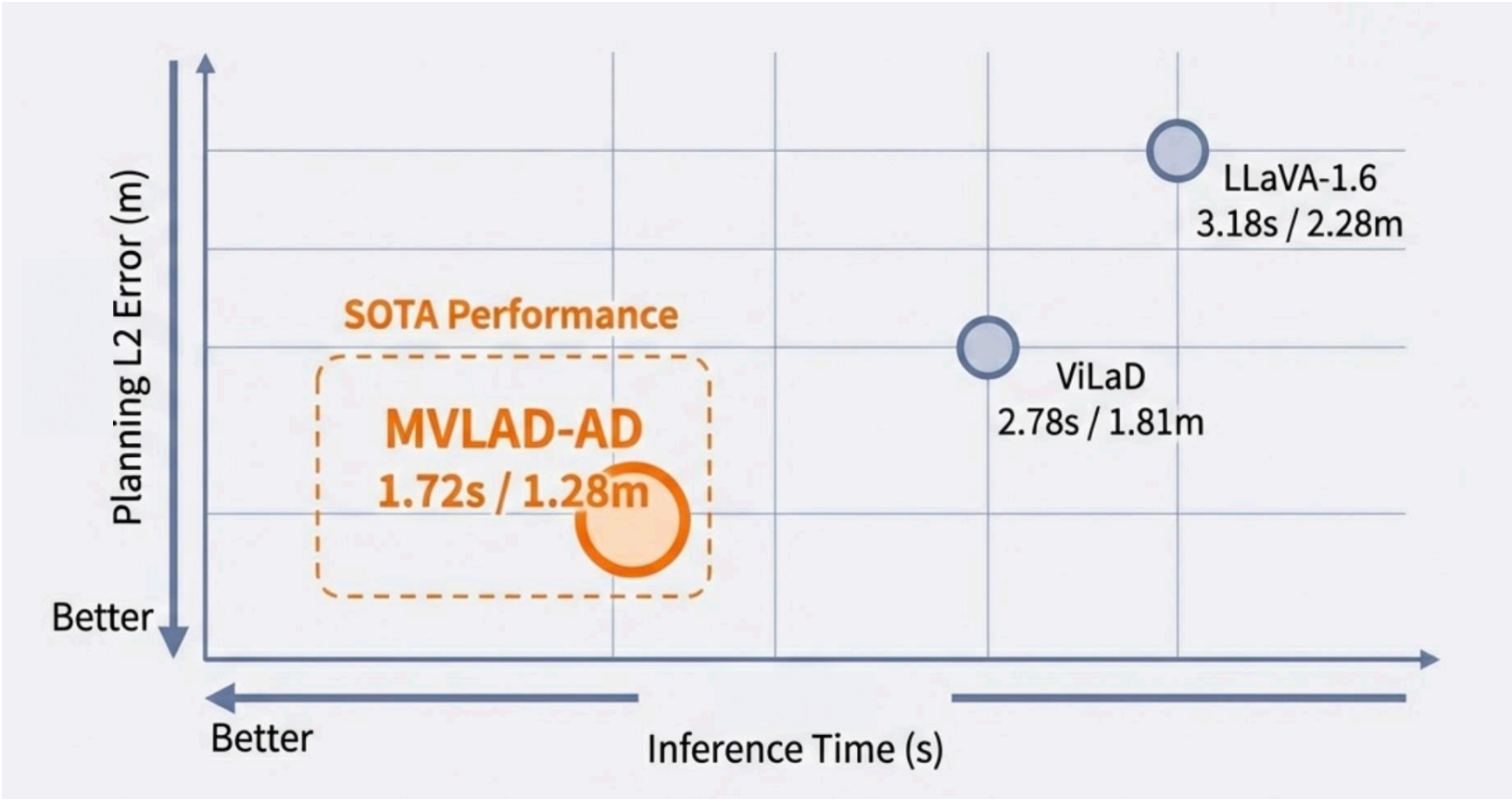
Results: Examples

Examples



- **Groundtruth Explanation:**
  - Narration: the car proceeds straight on the **rain-soaked road**
  - Reasoning: to stay aligned with the lane markings, avoiding any **drifting** due to the slippery conditions
- **Predicted Explanation**
  - Narration: the car moves straight along **the wet lane**
  - Reasoning: because the traffic flowing is clear and any **obstructions**

Results: Better Planning Performance



Compared with ViLaD and other LLM-based end-to-end autonomous driving methods, MVLAD achieves better planning precision and speed.

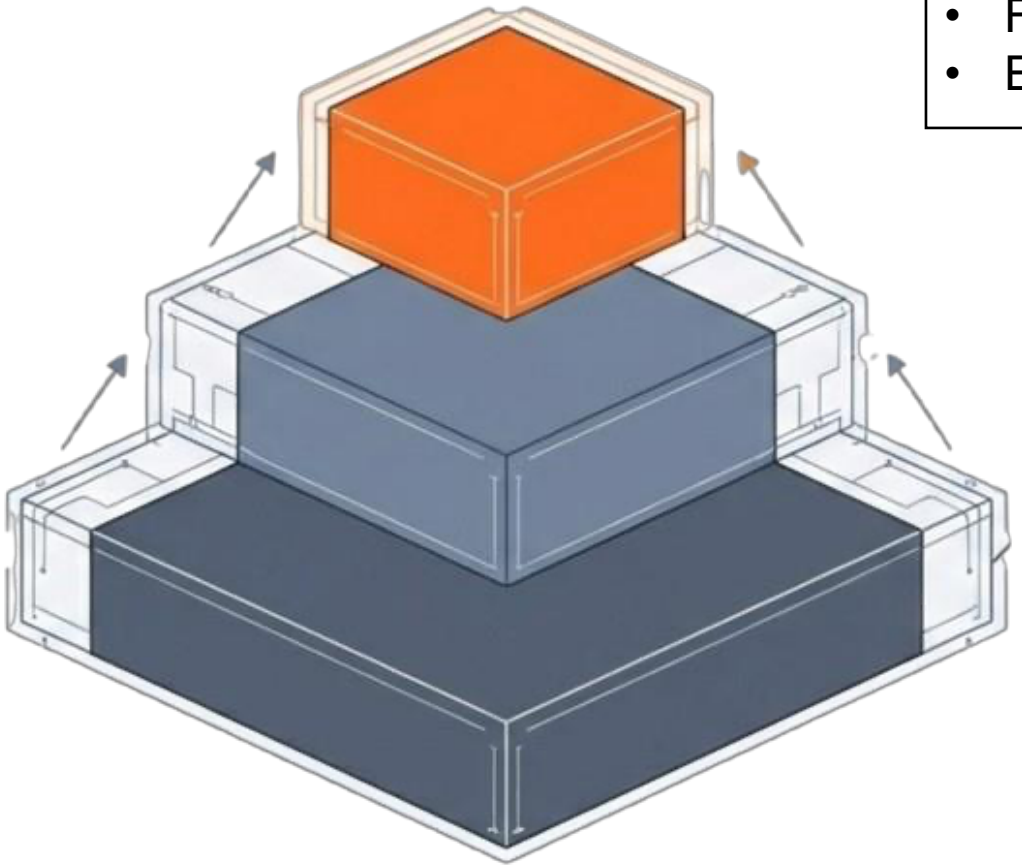
Results: Reasoning Ability

Quantitative Comparison

Dataset	Nu-X				NuScenes-QA		
Metrics	CIDEr	BLEU-4	METEOR	ROGUE-L	H0	H1	ALL
GPT-4o	19.0	3.95	10.3	24.9	42.0	34.7	37.1
Gemini-1.5	17.6	3.43	9.3	23.4	40.5	32.9	35.4
TOD3Cap	14.5	2.45	10.5	23.0	53.0	45,1	49.0
Hint-AD	22.4	4.18	13.2	27.6	55.4	48.0	50.5
ALN-P3	**28.6**	5.59	14.7	35.2	57.1	50.9	52.9
MVLAD	19.5	**13.0**	**36.8**	**37.3**	**57.7**	**55.1**	**55.9**

MVLAD achieves better reasoning performance than all baselines on two representative benchmarks.

Conclusion



Level 3: MVLAD

- Faster inference on short action sequence
- Explainable driving decisions

Level 2: ViLaD

- Fast parallel inference
- Bidirectional reasoning

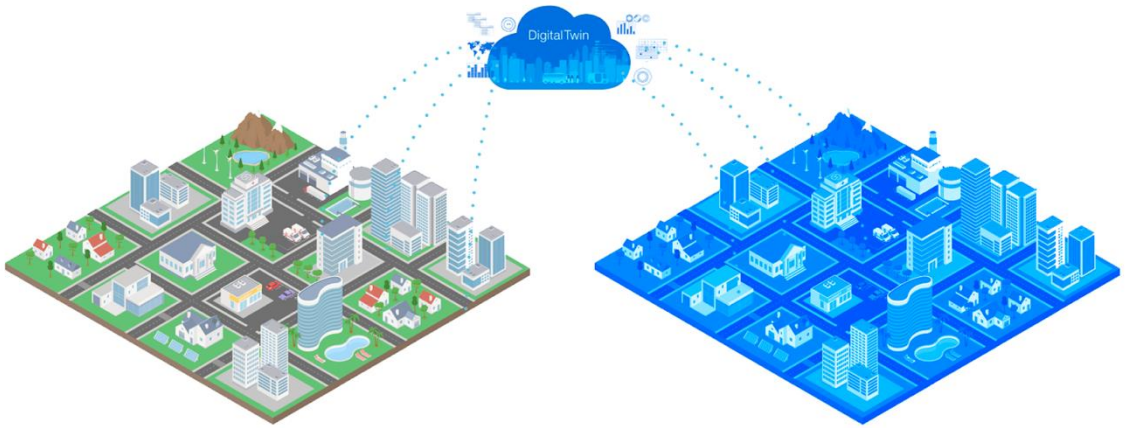
Level 1: Autoregressive VLM

- Low left-to-right inference
- Reversal curse

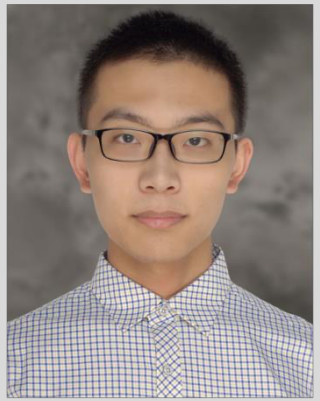
Generative Diffusion Language Model for End-to-End Autonomous Driving: Parallel Planning and Explainable Action

THANK YOU EVERYONE for attending this presentation! Any questions?

PURDUE UNIVERSITY | Purdue Digital Twin Lab



Contact Information



Jiaru Zhang

Postdoc Researcher

Supervisors: Prof. Ziran Wang, Prof. Ruqi Zhang

Purdue University

- Email: jiaru@purdue.edu
- Website: http://jiaruzhang.github.io