EMMA: End-to-End Multimodal Model for Autonomous Driving

Robert Azarcon, Vedika Agarwal, Tanmay Chavan

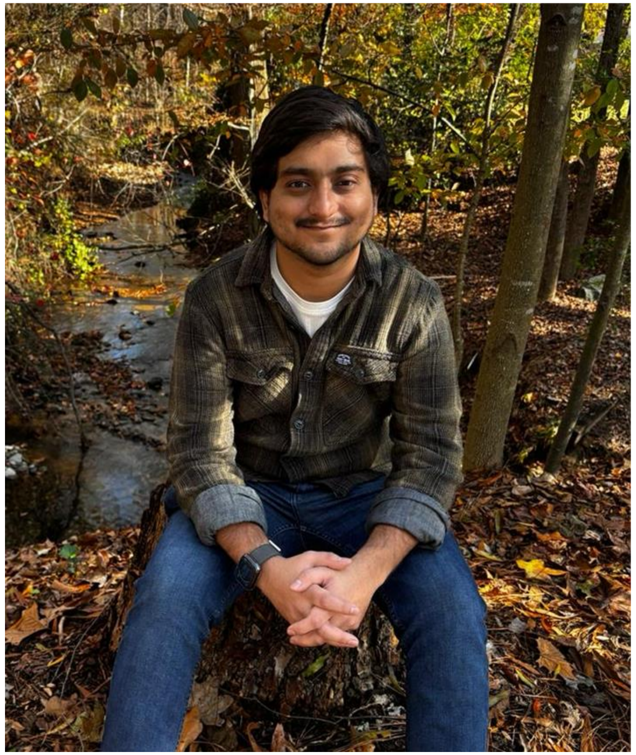
Nov 5, 2025
CS 8803 VLM Presentation

Georgia Tech

Presenters

Robert Azarcon
MSCS ‘27

Vedika Agarwal,
2nd year MSCS
Machine learning specialization



Tanmay Chavan
MSCS ‘26

Introduction

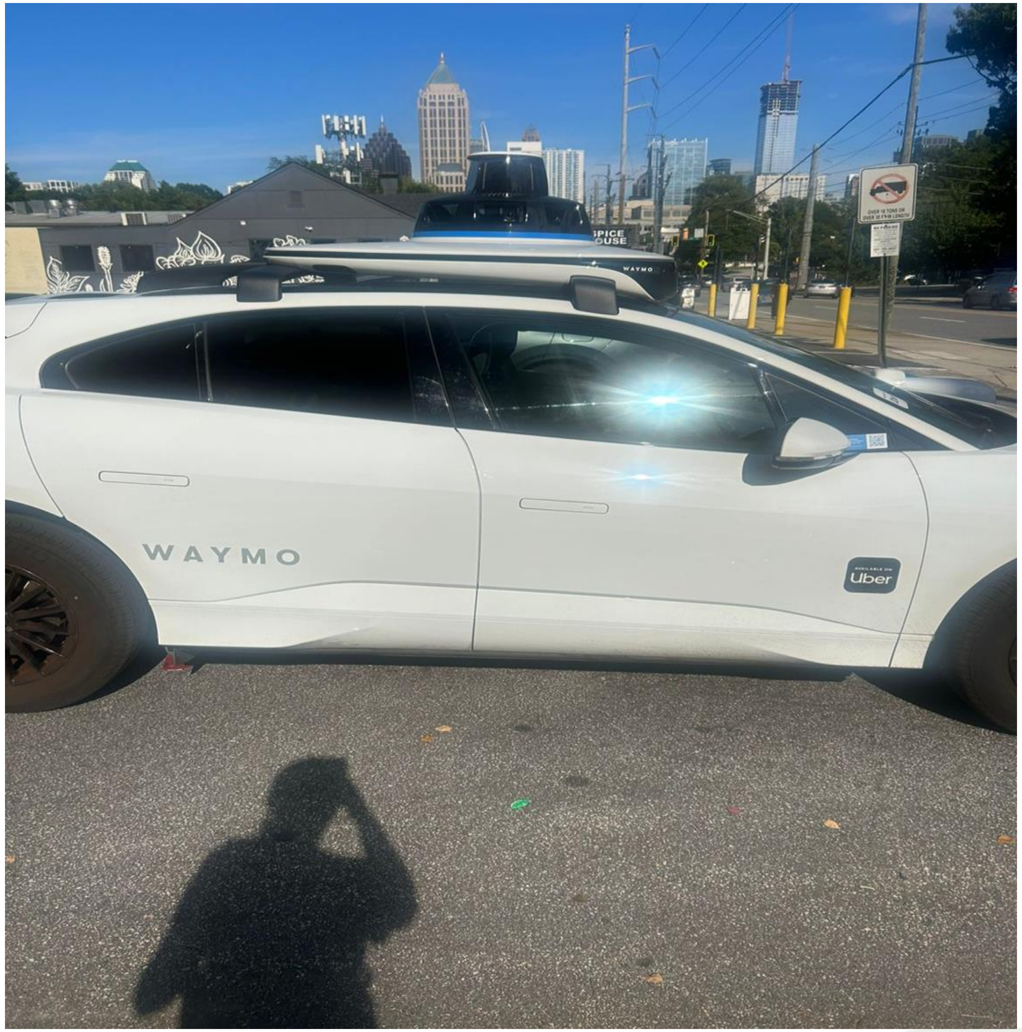
What is autonomous driving?

<u>3 areas of cognition + example subtasks:</u>

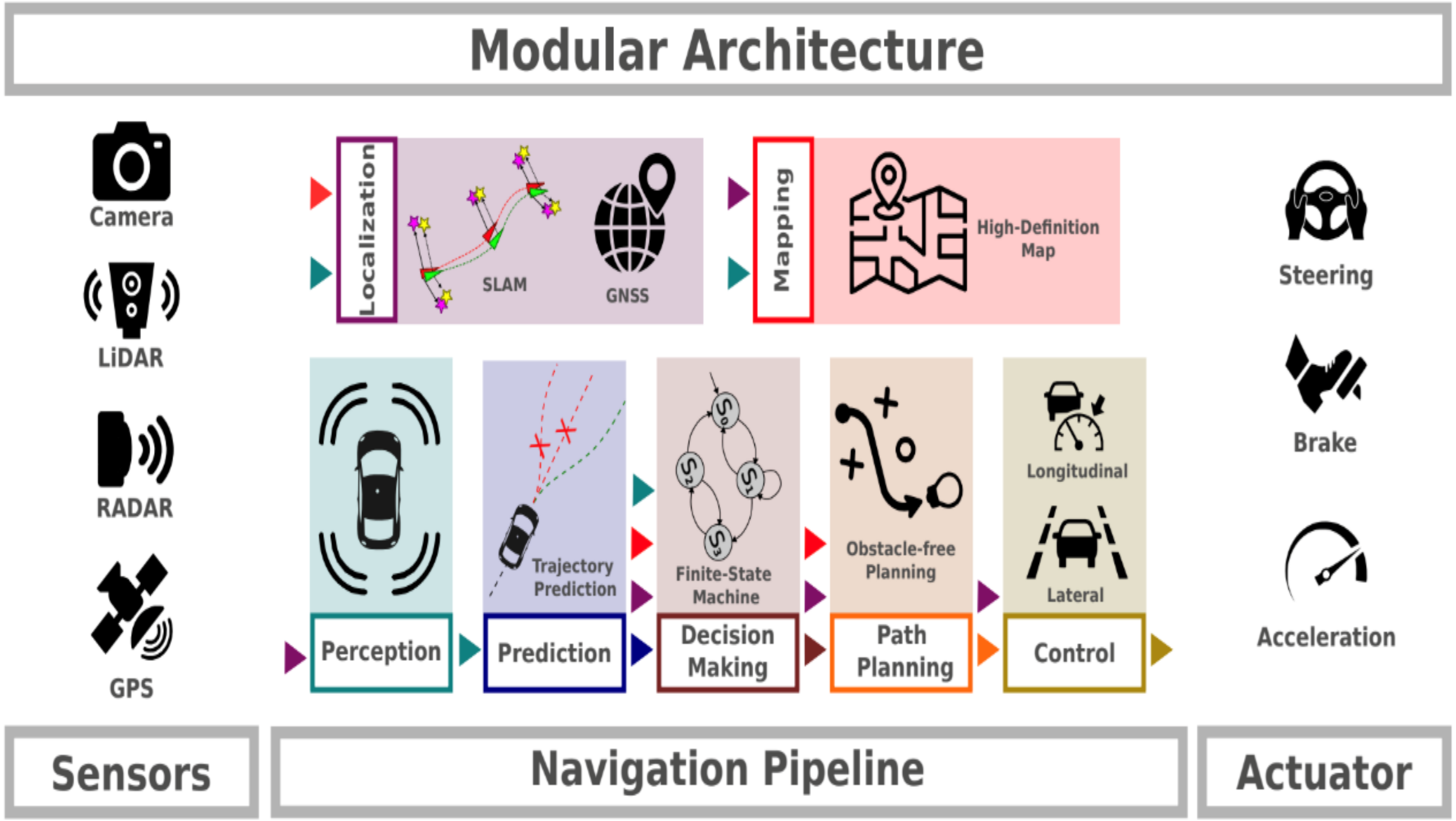
Perception
- Object detection + tracking
- Localization
- Scene understanding

Prediction
- Trajectory prediction
- Interaction modeling
- Intent prediction

Planning
- Route planning
- Behavior planning
- Trajectory/motion planning



Traditional Modular Approaches



Some intermediate representations:

- Bounding boxes
- Semantic map
- List of waypoints

Specialized modules for tasks + intermediate representations between modules

Traditional Modular Approaches

Advantages

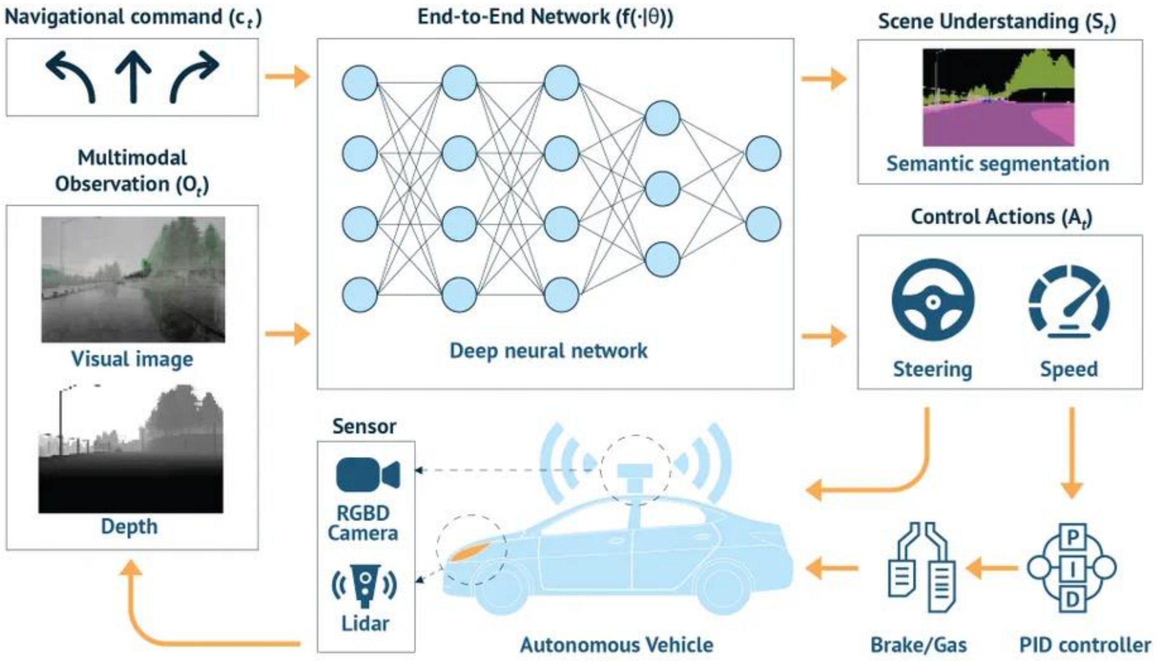
- **Responsibility Separation:** Easier development, debugging, and validation per module
- **Flexibility:** Modules can be individually upgraded, replaced, or modified
- **Interpretability:** Transparent and easier to explain or audit for safety and regulations.

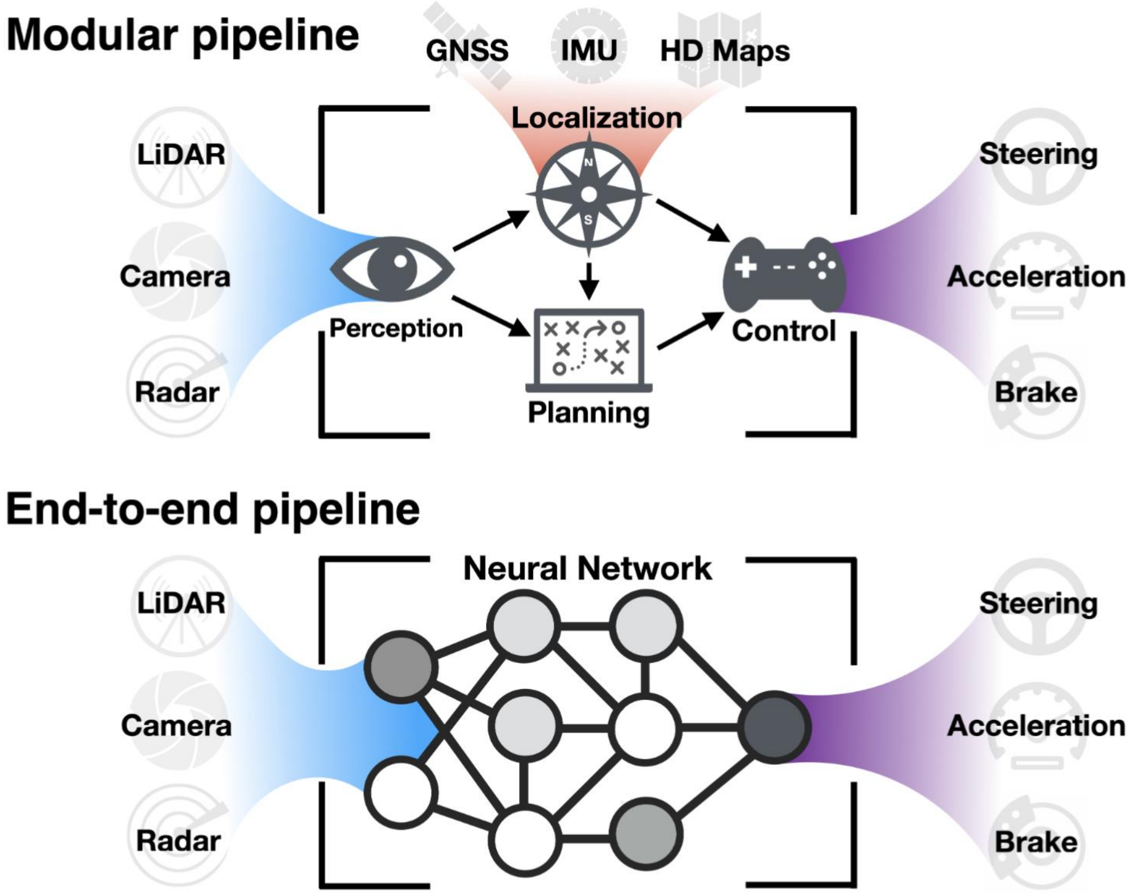
Challenges

- **Error Propagation:** Mistakes in perception could cascade through the rest of the pipeline.
- **Integration Complexity:** Ensuring fast and consistent communication between modules is difficult.
- **Pre-defined interfaces:** Need to engineer specific, symbolic interfaces between modules

Currently deployed paradigm in AVs is modular approach

End-to-End Approach





Idea: Directly map sensor inputs to actions with neural network

End-to-End Approach

- Replaces the traditional modular pipeline with a single neural network that directly maps sensor inputs → driving actions

- Learns all intermediate representations (perception, prediction, planning) jointly and implictly, rather than as separate, explicitly defined interfaces

- Avoids accumulating errors as seen in modular approach

- Enables better reasoning across environments and driving conditions

- **Challenges:**
  - Harder to interpret/debug for safety
  - High computational cost/inference latency
  - More complex training/evaluation
  - Still requires extensive closed loop testing (open-loop dataset evaluation may not correlate with real-world driving performance

Georgia Tech

EMMA and the E2E Generalist Model

EMMA: *End-to-End Multimodal Model for Autonomous Driving* (Waymo, 2024).

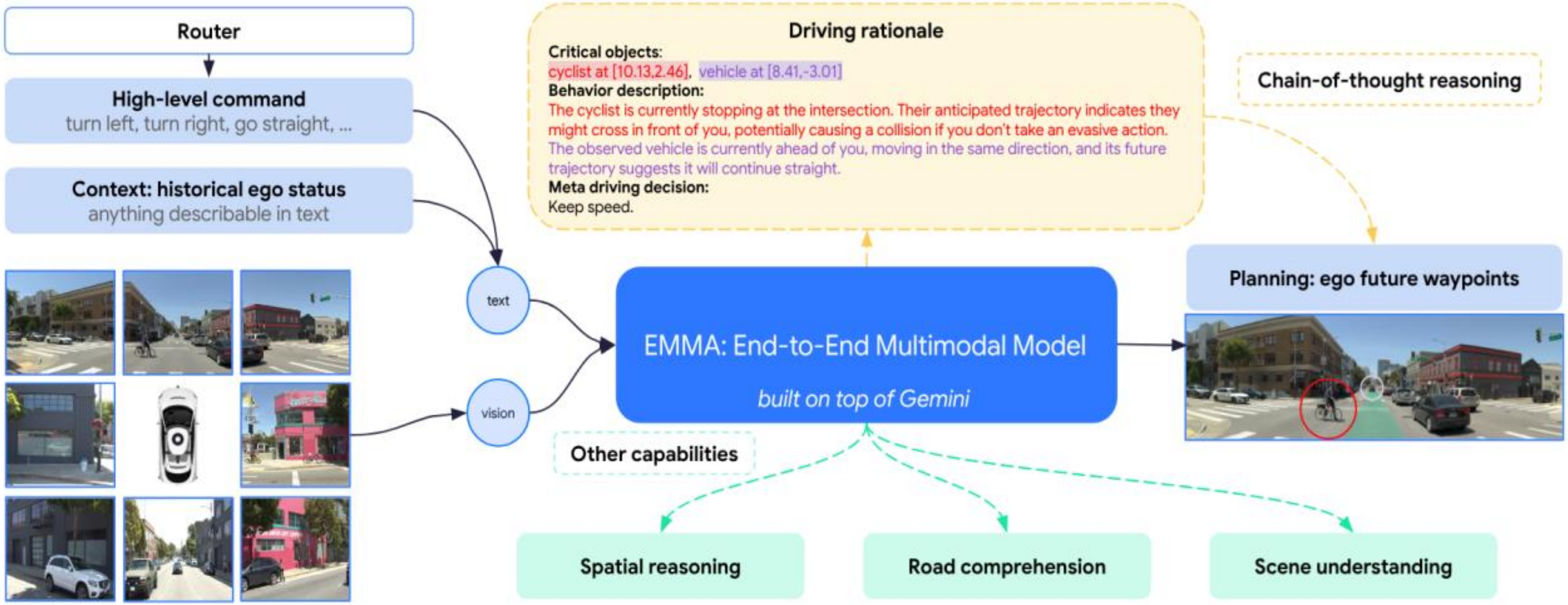
- Able to jointly perform range of tasks across perception/prediction/planning
- Leverages Gemini, a large-scale VLM (harness pre-trained extensive world knowledge)

How does language help?
- Allows for task prompting via textual inputs
- Enables complex reasoning via chain-of-thought
- Utilize semantic relationships between objects

Authors claim co-training on upstream tasks will boost trajectory planning performance

Methods

EMMA Architecture overview



Receive text/vision then output future waypoints

EMMA Architecture Overview

$$\mathbf{O}_{\text{trajectory}} = \mathcal{G}(\mathbf{T}_{\text{intent}}, \mathbf{T}_{\text{ego}}, \mathbf{V})$$

EMMA takes three main inputs in text/vision domain:

1. High-level driving command (derived from Google Maps)
   a. “turn right”, “go straight”, “turn left”, etc
2. Ego-vehicle history
   a. Set of plain text representing waypoint coordinates (x, y)
3. Surround-view camera videos capturing the driving scene
   a. Series of RGB images from each camera

Primary task (Planning) output:

- Set of BEV (Birds Eye View) 2D waypoints in textual representation (x,y)

Text to Float Conversion and Specialized Tokens

EMMA explores two key methods to represent continuous motion outputs:

1. **Text-to-Float Conversion:** Converts text-based predictions (like “move 5 meters ahead”) into numeric trajectory coordinates.

2. **Specialized Tokens:** Uses dedicated tokens to directly represent control or position values within the model vocabulary. Location space discretized via learned or manually defined scheme.

Why prefer text over special tokens (or vice versa)?

$$\texttt{text}(\{(x_i, y_i)\}) \qquad \texttt{tokenize}(\{(x_i, y_i)\})$$

Text to Float Conversion and Specialized Tokens

Pros/Cons of each method?

Textual Representation Pros

- Already leverages Gemini's pre-training, no need to change vocabulary
- All tasks share same unified language space
- Special tokens for spatial locations would require some discretization which introduces complexity

Textual Representation Cons

- Large numbers or many decimal places will require more tokens
- Potential for invalid outputs (e.g., "9.0a" or "1.i3")

Author Findings: Text-to-float conversion also allows smoother integration with Gemini's language backbone while maintaining numeric precision.

Planning with Chain of Thought Reasoning

Model is asked in this order:

1) R1, Scene description: Describe weather, traffic situation, road conditions, etc
   a) Example: *It is currently raining and the road is very wet*

1) R2, Critical objects: Identify and locate important objects (vehicle, human)
   a) Example: *Pedestrian at [9.01, 3.22], vehicle at [11.58, 0.35]*

1) R3, Behavior description of critical objects
   a) Example: *The pedestrian is preparing to cross street.*

1) R4, Meta driving decisions: A summary of driving plan given previous observations.
   a) Example: *I should keep my current low speed.*

Would these be sufficient for good driving rationale?

Georgia Tech

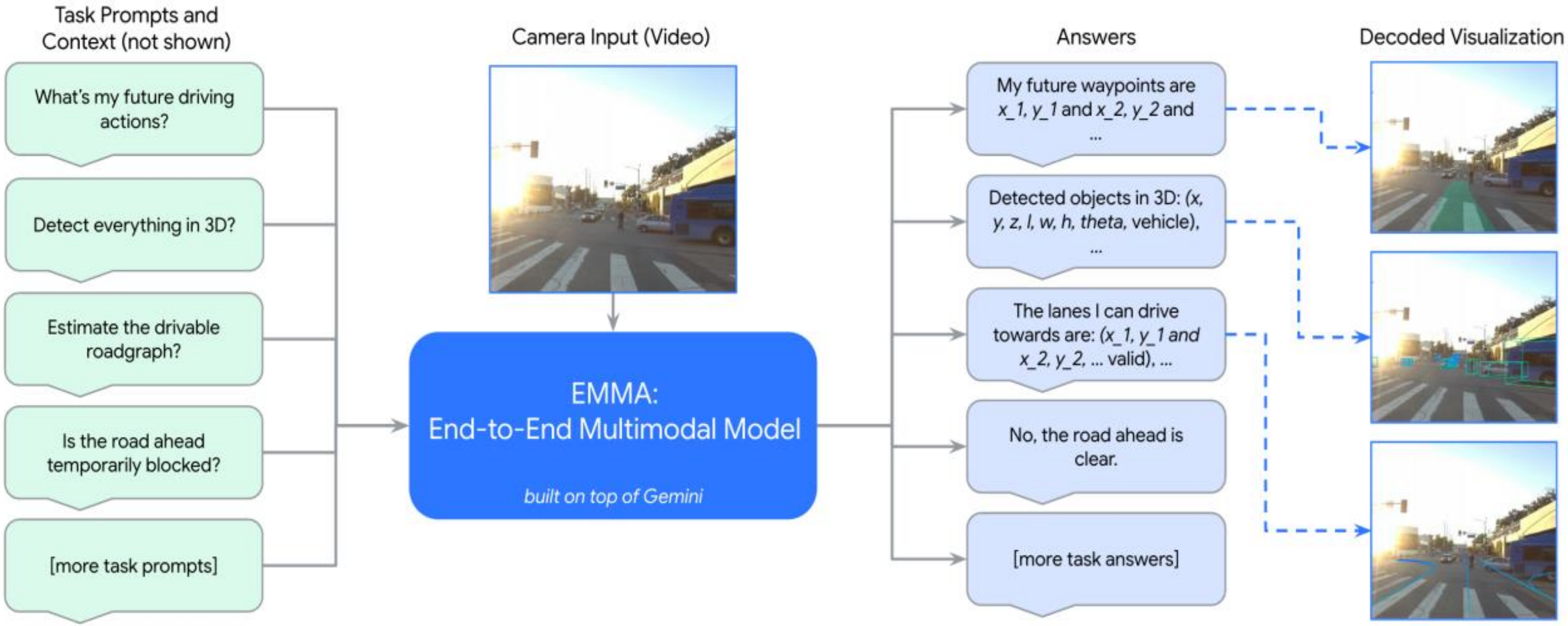
How are the COT captions created?

- Automated tool w/o human labels for scalability
- Leverage pre-existing expert prediction/perception models
- Meta driving decisions computed via some heuristic based on ego vehicle GT trajectory

Speed at 0s	Speed at 1s	Speed at 3s	Meta Decision Description
stationary	stationary	stationary	"Stay stationary."
stationary	moving	-	"Start moving soon."
stationary	stationary	moving	"Stay stationary for now, then start moving soon."
moving	constant	constant	"Keep speed."
moving	constant	increase	"Keep speed, then accelerate."
moving	constant	decrease	"Keep speed, then brake."
moving	increase	increase	"Accelerate."
moving	increase	constant	"Accelerate, then keep high speed."
moving	increase	decrease	"Accelerate, then brake."
moving	decrease	decrease	"Brake."
moving	decrease	constant	"Brake, then keep low speed."
moving	decrease	increase	"Brake, then accelerate."

12 categories of meta driving decisions

EMMA - Generalist Training and Multi-Task Learning



Size Weighted Dataset Sampling: $|\mathbf{D}_{\mathrm{task}}| / \sum_t |\mathbf{D}_{\mathrm{t}}|$

EMMA trains on 3 other tasks

EMMA Generalist Training (Spatial Reasoning)

3D object detection:

$$\mathbf{O}_{\text{boxes}} = \texttt{set}\{\texttt{text}(x, y, z, l, w, h, \theta, cls)\}$$

Use a fixed text prompt ("detect every object in 3D") to get the boxes:

$$\mathbf{O}_{\text{boxes}} = \mathcal{G}(\mathbf{T}_{\text{detect\_3D}}, \mathbf{V})$$

Interesting finding: sorting 3D boxes by depth improves detection quality

Georgia Tech

EMMA Generalist Training (Road Graph Estimation)

Predicting Drivable Lanes:

Outputs an ordered set of polyline waypoints

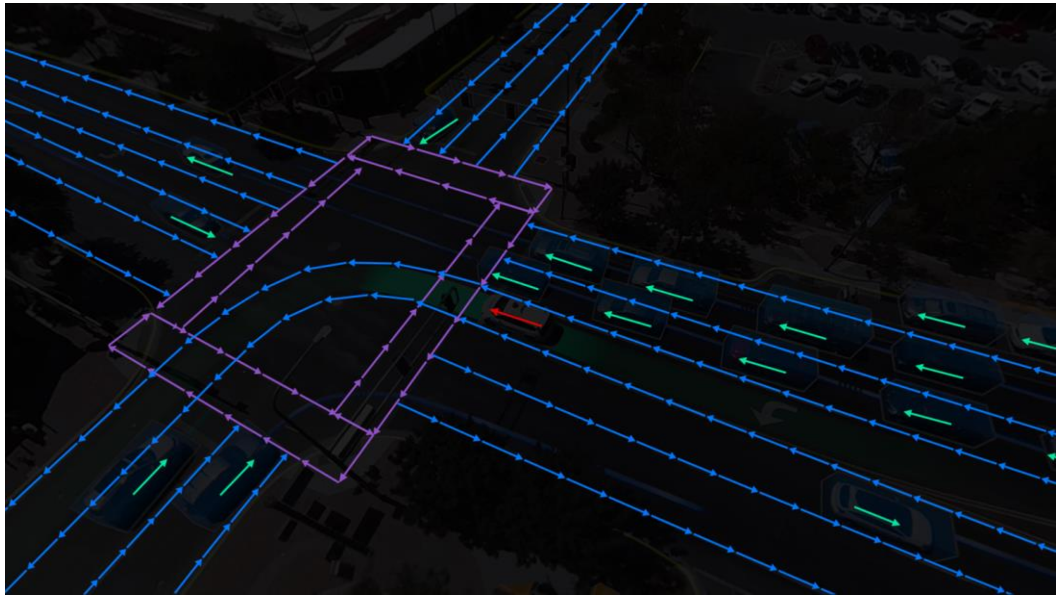
`"(x1,y1 and...  and xn,yn);..."`

where (x,y) are floats converted to text

$$\mathbf{O}_{\text{roadgraph}} = \mathcal{G}(\mathbf{T}_{\text{estimate\_roadgraph}}, \mathbf{V})$$

What is a road graph?

Collection of road elements where the edges are relationships between each element



Polyline Ground Truth Label Generation

Fixed sampling vs Dynamic Sampling

Given a set of points how to choose which points to represent as the polyline?

Fixed sampling

- Set number of points to sample from a curve

Dynamic Sampling

- Variable # of points based on lane shape

Global vs Ego-Origin Frame

What coordinate frame to represent the polyline points in?

Global frame

- Lane points derived from fixed HD map's global coords

Ego-Origin Frame

- Points are sampled relative to the ego-origin frame each timestep

EMMA Generalist Training (Scene Understanding)

Identify Temporary Blockages:

Model is asked “Is the road head temporarily blocked?”

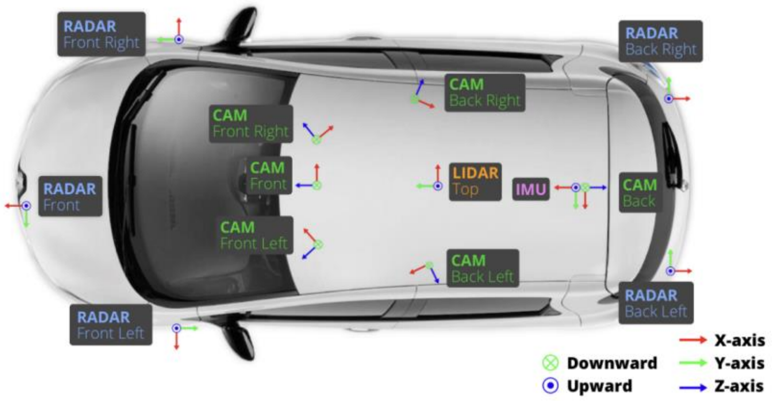
$$\mathbf{O}_{\text{temporary\_blockage}} = \mathcal{G}(\mathbf{T}_{\text{temporary\_blockage}}, \mathbf{T}_{\text{road\_user}}, \mathbf{V})$$

Text denoting other objects on road, T_{road_user} is given as another training input.

Experiments

Datasets - nuScenes

1. Contains 1000 scenes in diverse settings
2. Each scene spans over 20 seconds
3. Dataset provides full 360-degree view
4. Contains data from 6 cameras, 5 radars, and 1 LiDAR.



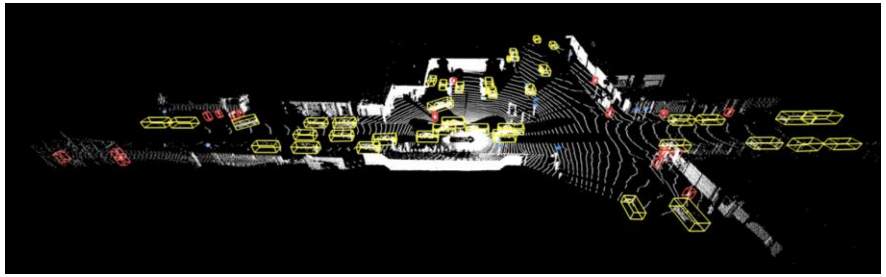
Caesar et. al., “nuScenes: A multimodal dataset for autonomous driving”

Datasets - WOMD and WOD

1. The Waymo Open Motion Dataset (WOMD) is primarily made for object trajectories
2. Contains 103,000 driving scenarios, further split into 1.1M examples.
3. 1 second for input context, 8 seconds for evaluation
4. Contains map features such as traffic signal states and lane characteristics
5. The authors also validate 3D object detection task on Waymo Open Dataset (WOD) benchmark for object detection, containing camera, LiDAR, and 3D box data.

	KITTI	NuScenes	Argo	Ours
Scenes	22	1000	113	1150
Ann. Lidar Fr.	15K	40K	22K	230K
Hours	1.5	5.5	1	6.4
3D Boxes	80K	1.4M	993k	12M
2D Boxes	80K	–	–	9.9M
Lidars	1	1	2	5
Cameras	4	6	9	5
Avg Points/Frame	120K	34K	107K	177K
LiDAR Features	1	1	1	2
Maps	No	Yes	Yes	No
Visited Area (km^2)	–	5	1.6	76

Figure 6. Parallelogram cover of all level 13 S2 cells touched by all ego poses in San Francisco, Mountain View, and Phoenix.



LiDAR labels

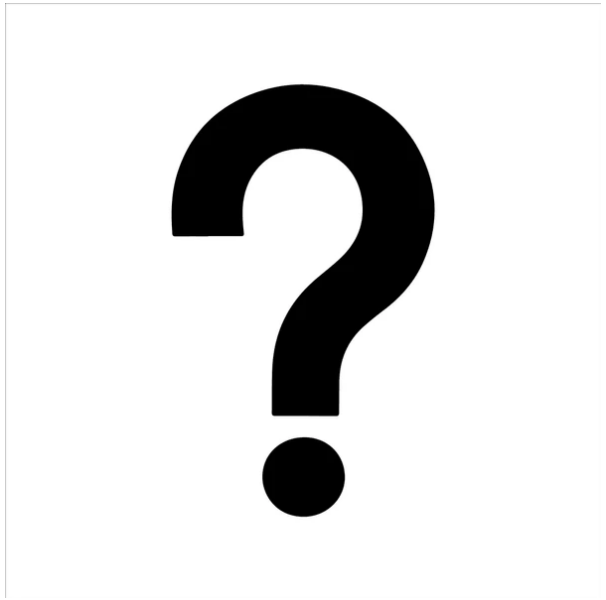
Sun et. al., "Scalability in Perception for Autonomous Driving: Waymo Open Dataset"

Datasets - Internal datasets

1. Internal motion planning dataset
    a. **24 million scenes**, each 30 seconds long
    b. Sample one frame per scenario for training

1. Internal dataset for detection
    a. 12 million examples

1. Internal road graph dataset
    a. 8 million examples
    b. Sample one example every 30 seconds



Dataset Name	Total Hours of Driving	Number of Training Examples
nuScenes (Caesar et al., 2020)	6	18,686
WOMD (Chen et al., 2024a)	572	487,061
Internal Motion Planning Dataset	203,117 (355x)	24,374,046 (50x)
WOD (Sun et al., 2020)	6	158,081
Internal Detection Dataset	6250	11,765,140
Internal Roadgraph Dataset	64,135	8,304,671

End-to-End Motion Planning experiments

1. Simple training strategy - given camera images, ego vehicle history, and driving intent, predict the future ego waypoints.
2. MotionLM and Wayformer used as baselines.
3. MotionLM requires detailed input (agent location history, road graphs) while EMMA is relatively simple.
4. MotionLM & Wayformer sample trajectories to report final guess

1. Results on WOMD:
    a. Models primarily trained on WOMD
    b. EMMA+ represents WOMD + internal dataset
    c. EMMA† represents PaLI-X instead of Gemini

Method	L2 (m) 1s	L2 (m) 3s	L2 (m) 5s
MotionLM* (Seff et al., 2023)	0.045	0.266	0.696
Wayformer* (Nayakanti et al., 2023)	0.046	0.252	0.628
EMMA† (based on PaLI)	0.034	0.274	0.797
EMMA+† (based on PaLI)	0.031	0.239	0.680
EMMA	0.032	0.248	0.681
EMMA (w/ CoT)	0.030	0.241	0.664
EMMA+	0.030	0.225	0.610
EMMA+ (w/ CoT)	**0.027**	**0.203**	**0.543**

Table 2: End-to-end motion planning experiments on an internal planning benchmark. CoT denotes equipping with chain-of-thought reasoning (Eq. 3). EMMA+ achieves the best quality across different prediction time horizons. EMMA† and EMMA+† denotes using PaLI-X (Chen et al., 2024d) as our base model, while the default EMMA and EMMA+ use Gemini as the base model. *Enhanced, reproduced baselines.

$$\mathbf{O}_{\text{trajectory}} = \mathcal{G}(\mathbf{T}_{\text{intent}}, \mathbf{T}_{\text{ego}}, \mathbf{V}).$$

EMMA trained on WOMD beats MotionLM. EMMA w/ CoT doesn't beat Wayformer for 5s prediction time. EMMA+ has best performance.

End-to-End Motion Planning experiments

1. Authors compare performances on nuScenes

1. Predict over next 3 seconds based on 2 seconds of historical data, evaluate using L2 norm (lower the better)

1. EMMA outperforms pre-existing supervised and self-supervised approaches, without using EMMA+

Method	self-supervised?	L2 (m) 1s	L2 (m) 2s	L2 (m) 3s	Avg L2 (m)
UniAD (Hu et al., 2023)	✗	0.42	0.64	0.91	0.66
DriveVLM (Tian et al., 2024)	✗	0.18	0.34	0.68	0.40
VAD (Jiang et al., 2023)	✗	0.17	0.34	0.60	0.37
OmniDrive (Wang et al., 2024a)	✗	0.14	0.29	0.55	0.33
Ego-MLP* (Zhai et al., 2023)	✓	0.15	0.32	0.59	0.35
BEV-Planner (Li et al., 2024)	✓	0.16	0.32	0.57	0.35
EMMA (random init)	✓	0.15	0.33	0.63	0.37
EMMA	✓	0.14	0.29	0.54	0.32
EMMA+	✓	**0.13**	**0.27**	**0.48**	**0.29**

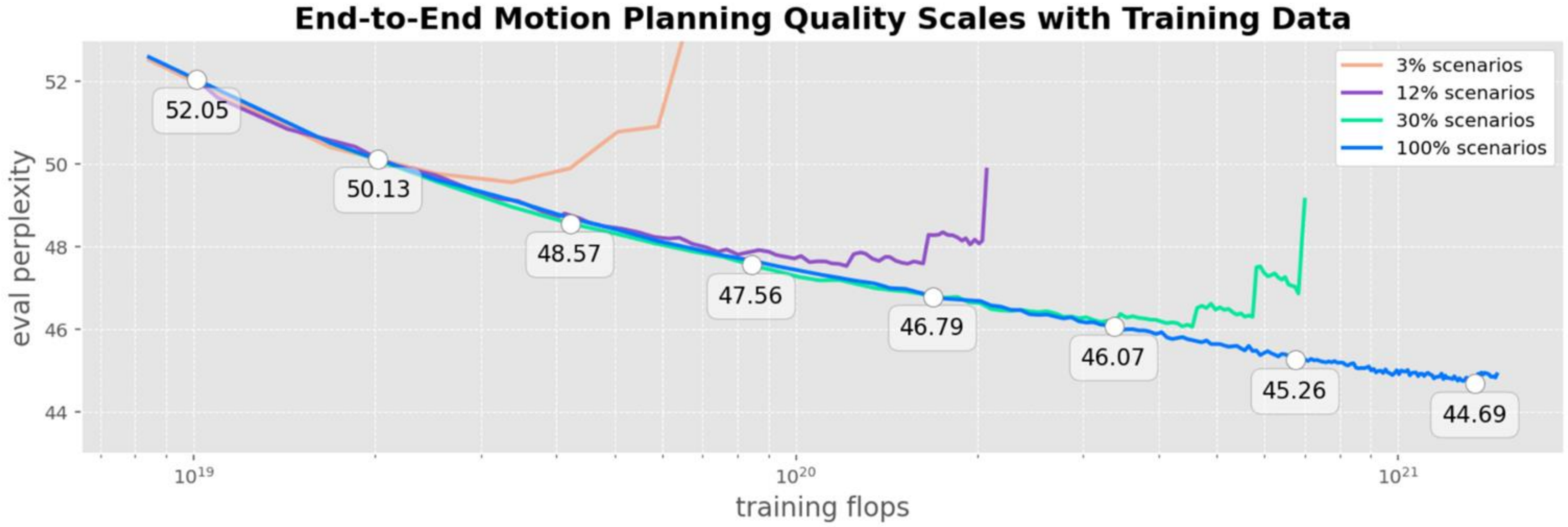
Table 3: End-to-end motion planning experiments on nuScenes (Caesar et al., 2020). EMMA (random init) denotes models are randomly initialized; EMMA denotes models are initialized from Gemini; EMMA+ denotes models that are pre-trained on our mega-scale internal data. EMMA achieves state-of-the-art performance on the nuScenes planning benchmark, outperforming the supervised (with perception and/or human labels) prior art by 6.4% and self-supervised (no extra labels) prior art by 17.1%. *Ego-MLP results are taken from a reproduced version in BEV-Planner.

Chain-of-Thought reasoning experiments

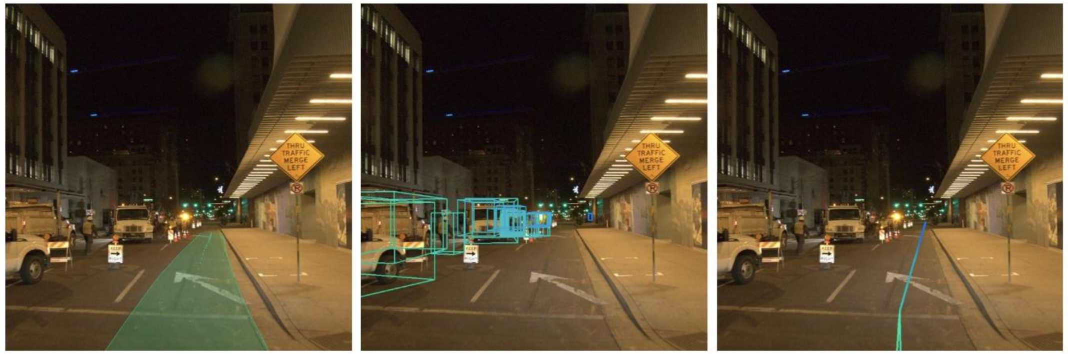
1. Experiments conducted on internal datasets
2. Task: use 2 seconds of history to predict 5 seconds into the future
3. CoT provides a 6.7% overall improvements (l2 norm) over standard end-to-end planning
4. Improves explainability of the model
5. The approach scales well with increasing data

Scene description	Critical object	Meta decision	Behavior description	Relative improvements over baseline e2e planning
✓	✗	✗	✗	+ 0.0%
✗	✓	✗	✗	+ 1.5%
✗	✗	✓	✗	+ 3.0%
✗	✓	✓	✗	+ 5.7%
✗	✓	✓	✓	+ 6.7%

Table 4: Ablation study on chain-of-thought reasoning components. It improves end-to-end planning quality by up to 6.7% by combining all elements. In particular, driving meta-decision and critical objects contribute the improvements of 3.0% and 1.5%, respectively. The details of each component is described in Section 2.2.

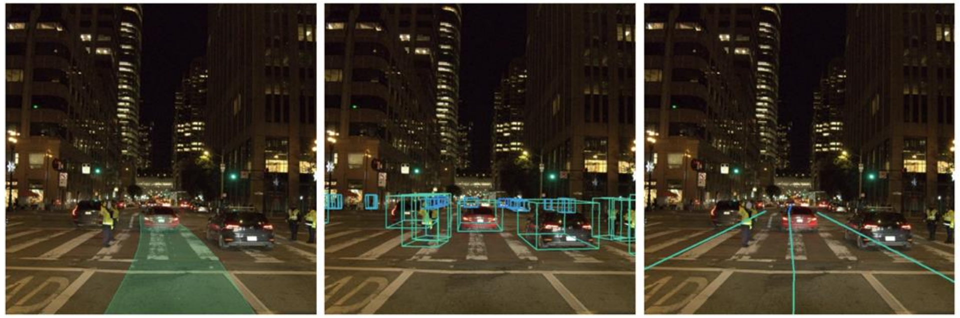


Visualizations



(e) As a construction zone blocks the left lanes, our predicted trajectory suggests passing through on the right, while the road graph estimation correctly identifies the blocked area.

(f) Our lane is blocked by construction cones, so our predicted trajectory suggests to move into the left lane, even though it's in the opposite direction. EMMA captured the blockage and performed a detour.



(g) A traffic controller signals to proceed through the intersection, and our predicted trajectory aligns with the instruction.

(h) Our predicted trajectory suggests to stop as we approach an intersection with a yellow light, demonstrating cautious and safe behavior.

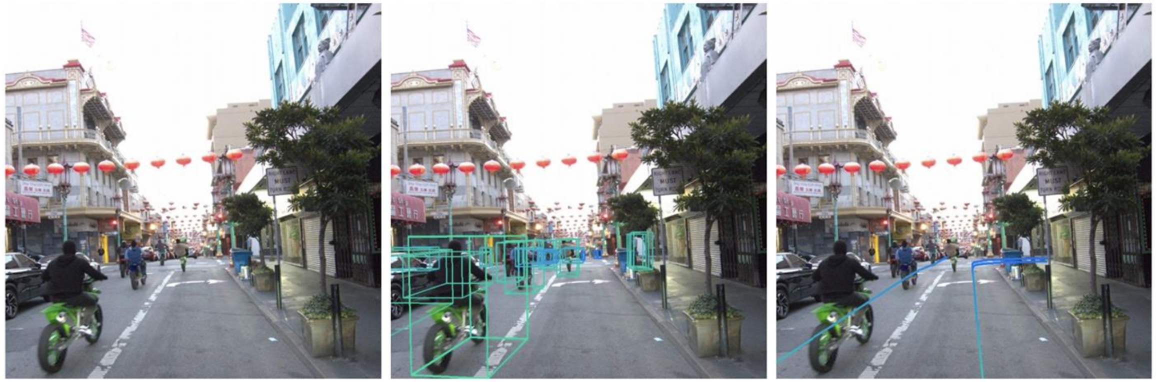
Figure 9: EMMA prediction visualization. Each row contains a scenario with our model's predictions: end-to-end planning trajectory (left), 3D object detection (middle), and road graph estimation (right).

Georgia Tech

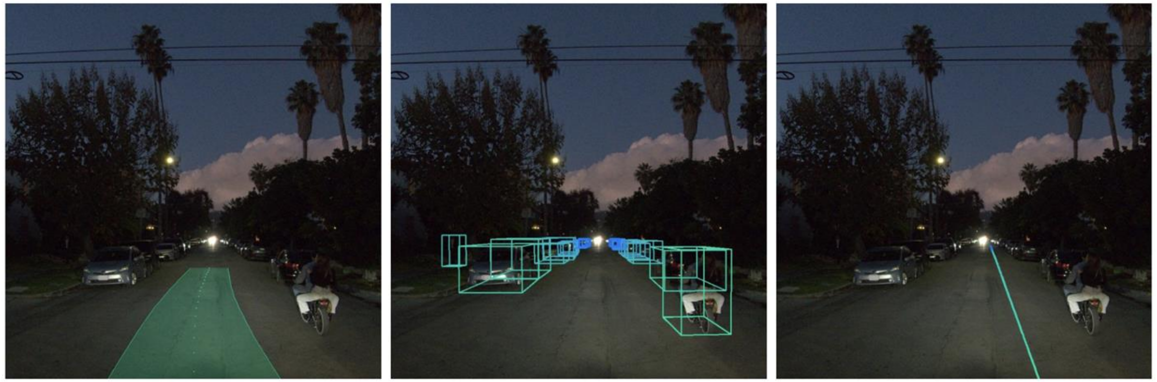
Visualizations

(i) While crossing an intersection, our predicted trajectory nudges slightly to the left due to nearby cars and a bicyclist partially occupying our lane.

(j) Our model predicts a driving trajectory to patiently wait at a red light (left). The model also accurately predicts surrounding 3D objects (middle) and road graph with lane centers (right).



(k) A fleet of fast-moving motorcyclists pass by. The predicted trajectory suggests pausing to allow them to pass safely. Notably, motorcyclists are accurately identified by our model (middle).



(l) A motorbike is moving on a narrow lane at night, and yields to the right. Our predicted trajectory adjusts, guiding us to pass safely by nudging slightly to the left.

Georgia Tech

3D Object Detection

1. 3D object detection evaluated on the WOD benchmark

1. Since EMMA cannot generate confidence scores, its F1 scores are compared with the precision-recall curves of other models

1. EMMA achieves mediocre performance, but EMMA+ performs competitively

“With sufficient data and a large enough model, a multimodal approach can surpass specialized expert models in 3D detection quality”

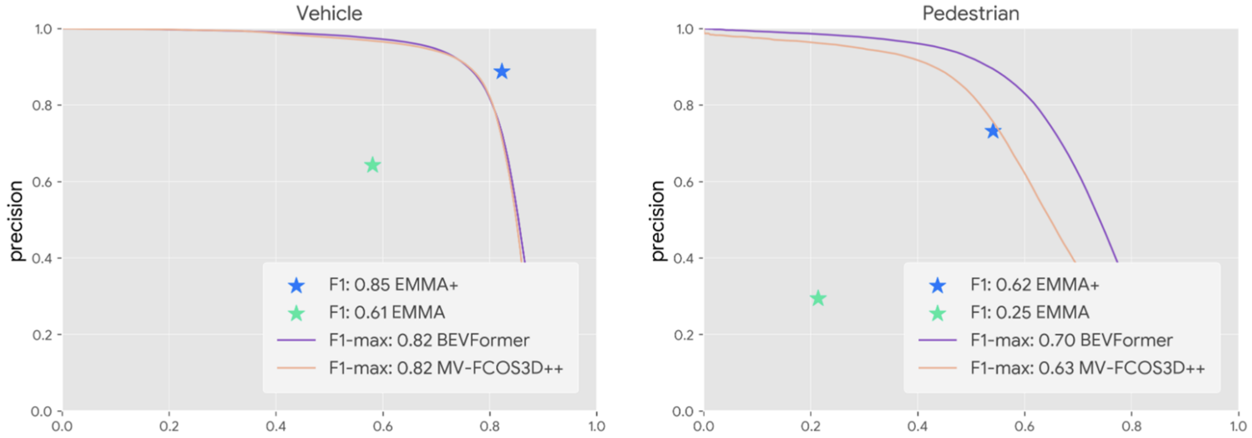


Figure 5: Camera-primary 3D object detection experiments on WOD (Sun et al., 2020) using the standard LET matching (Hung et al., 2024). EMMA+ achieves competitive performance on the detection benchmark in both precision/recall and F1-score metrics. Compared to state-of-the-art methods, it achieves 16.3% relative improvements in vehicle precision at the same recall or 5.5% recall improvement at the same precision.

Road Graph Estimation

1. Creation of graph-based map of the road network by predicting group of polylines

1. Precision/recall measured by:

   a. comparing lane polyline (ground-truth) with predicted polylines
   b. rasterizing polylines into a BEV grid with 1 meter resolution

1. Road graph polylines defined by start and end points of each lane, with intermediate points for curvature

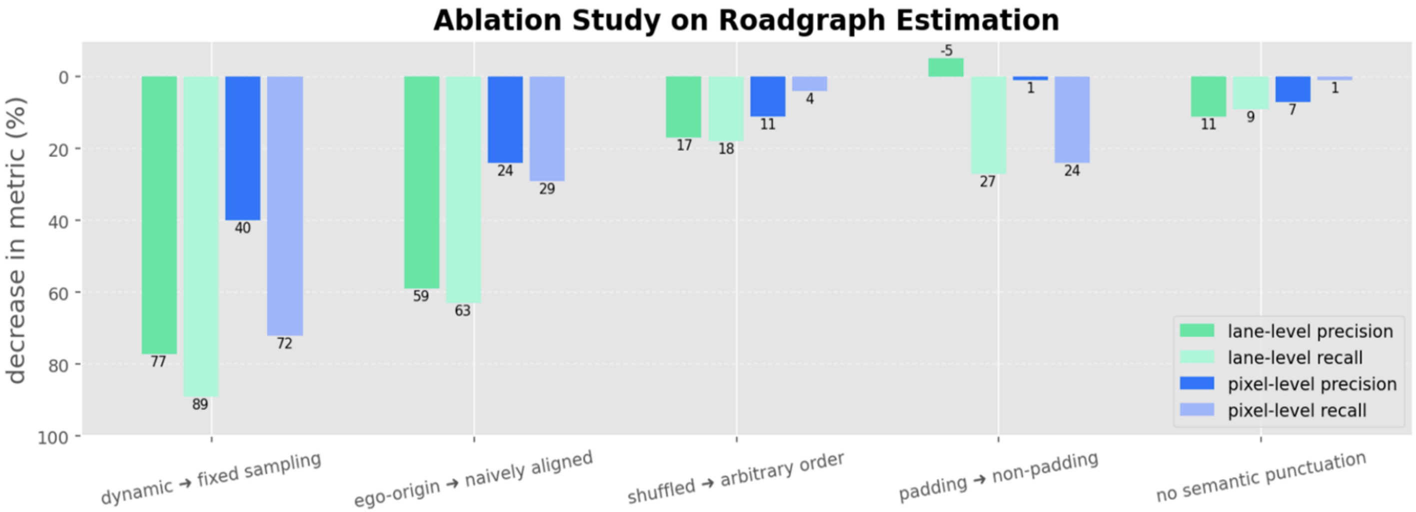


Figure 6: Ablation study on road graph estimation. To evaluate the influence of different components in our road graph estimation model, we ablate each configuration and measure the corresponding impact on quality. Dynamic sampling (**leftmost**) of road graph polylines based on lane curvature and length proves to be the most significant factor, leading to a substantial 70% to 90% change in lane-level precision and recall. In contrast, aligning the model with a language-like representation, *i.e.*, semantic punctuation (**rightmost**), has a more modest effect, contributing to only <10% change in precision and recall of any metric.

Road Graph Estimation

1. Findings:
   a. Dynamic point sampling is better than sampling fixed number of points -
      i. dynamically adjust the number of points per polyline according to the curvature and length of the lane
   b. Ego-origin aligned sample intervals are better than naively aligned sample intervals
      i. instead of global coordinate frame, start from ego vehicle coordinate frame origin.
   c. Padding improves performance:
      i. padding targets to prevent early termination is highly effective

   a. Punctuations improve quality:
      i. e.g., "(x,y and x,y);..." instead of "xy xy;..."

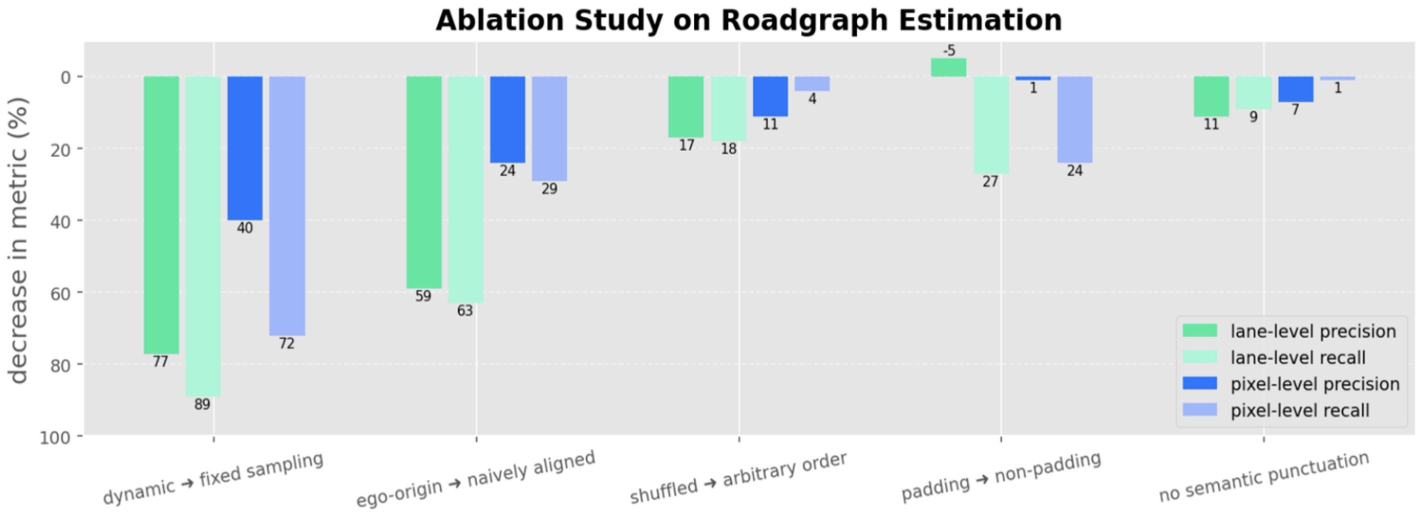


Figure 6: Ablation study on road graph estimation. To evaluate the influence of different components in our road graph estimation model, we ablate each configuration and measure the corresponding impact on quality. Dynamic sampling (**leftmost**) of road graph polylines based on lane curvature and length proves to be the most significant factor, leading to a substantial 70% to 90% change in lane-level precision and recall. In contrast, aligning the model with a language-like representation, *i.e.*, semantic punctuation (**rightmost**), has a more modest effect, contributing to only <10% change in precision and recall of any metric.

Scene Understanding

1. Task: to detect temporary blockages on roads

1. Baseline obtained by human annotation ('filtering' removes all ambiguous human data points)

1. Pre-training model on road graph estimation followed by fine-tuning enhances performance.

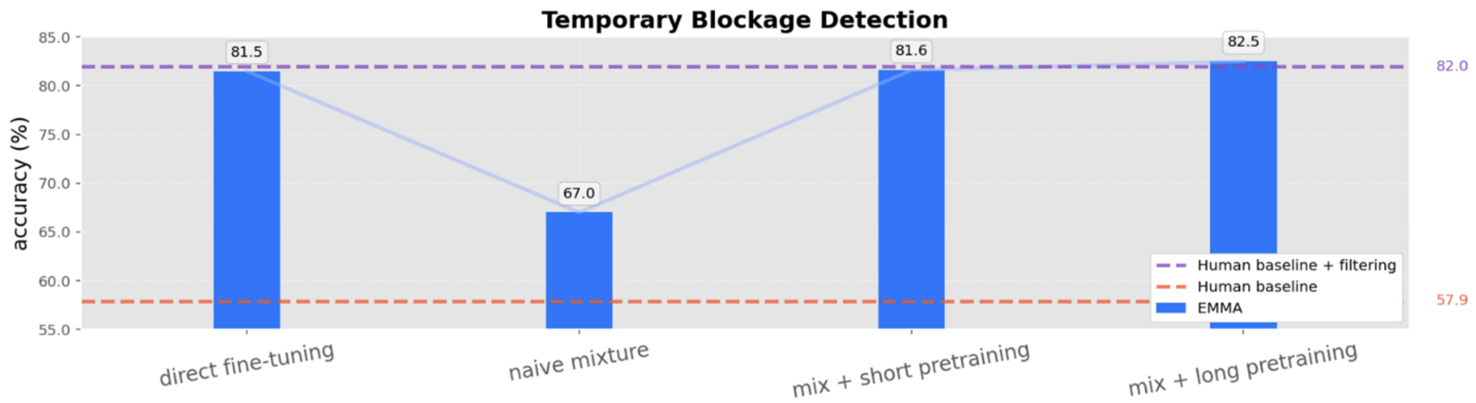


Figure 7: Scene understanding experiments. `direct fine-tuning` denotes solely using the temporal blockage data during fine-tuning; `naive mixture` denotes co-training this scene task with road graph estimation; `mix + short pretraining` denotes pre-training on road graph esitmation first, and then fine-tune on the mixture of both tasks; `mix + long pretraining` denotes a longer pre-training before fine-tuning. The naive fine-tuning is already close to strong human baseline, but long-pretraining with training mixture can further boost the quality.

Generalist Training

1. EMMA Generalist is co-trained on primarily three tasks:
   a. End-to-end planning
   b. 3D object detection
   c. Road graph estimation
2. Co-training on all three tasks provides a boost of 5.5% on detection.
3. Complementary tasks provide better performance after co-training.

			Relative improvement over single task		
e2e planning	3D detection	road graph	e2e planning	detection	road graph
	✓	✓	-	+1.6% (±1.0%)	+2.4% (±0.8%)
✓	✓		+1.4% (±2.8%)	+5.6% (±1.1%)	-
✓		✓	−1.4% (±2.9%)	-	+3.5% (±0.9%)
✓	✓	✓	+1.4% (±2.8%)	+5.5% (±1.1%)	+2.4% (±0.8%)

Table 5: Generalist co-training experiments. (±∗) indicates standard deviation. By co-training on multiple tasks, EMMA gains a broader understanding of driving scenes, enabling it to handle various tasks at inference time, while enhancing individual task performance. Notably, certain task pairings yield greater benefits than others, suggesting these tasks are complementary. Co-training all three tasks together yields the best quality.

Georgia Tech

Related Works

Evolution of End-to-End Driving Models

Early foundations: *ALVINN (1988)* pioneered end-to-end driving using shallow neural networks.

Deep learning era: *DAVE-2 (2016)* and *ChauffeurNet (2019)* leveraged deep architectures with perception + motion planning modules.

Multimodal & multi-task advances: Integrated multimodal inputs and learning from *Codevilla et al. (2018)*, *Prakash et al. (2021)*, *Chitta et al. (2022)*.

Reinforcement learning approaches: Explored adaptive control via *Chekroun et al. (2023)*, *Chen et al. (2021)*, *Kendall et al. (2019)*.

Unified planning frameworks: *VAD*, *UniAD*, *PARA-Drive*, and *GenAD* integrate perception, prediction, and planning in open-loop setups.

Key challenge:

- Many methods *overfit to ego-vehicle status* despite strong benchmark performance (*AD-MLP, BEV-Planner*).

EMMA direction: Revisits simplicity of early E2E models, augmented with modern MLLMs for generalizable

Vision Language Model in Driving

Explainable and generalizable driving: Recent works combine VLMs and planning for transparent reasoning.

DriveGPT4 & LMDrive: Use LLMs for *Q&A-style reasoning* and control signal prediction.

Drive Anywhere: Adds *patch-aligned feature extraction* for text-based decision making.

OmniDrive: Employs a *3D vision-language model* for spatial reasoning and motion planning.

Graph-based & CoT reasoning: Approaches like *Sima et al. (2024)*, *Tian et al. (2024)*, and *Wang et al. (2024)* use VQA and chain-of-thought for multi-task learning.

Modular architectures: *LLM-Drive (2024)* uses object-level vector inputs for planning.

Lightweight VLMs: EM-VLM4AD (2024) uses the T5 transformer + gated pooling attention

EMMA's contribution: End-to-end fine-tuning of an MLLM for open-world, generalist driving

Multimodal Large Language Models

MLLMs extend LLMs to handle multiple modalities: integrating *vision, text, and context* reasoning.

Early vision-language works: *Donahue (2015)*, *Vinyals (2015)*, *Chen (2022)* addressed image captioning and detection.

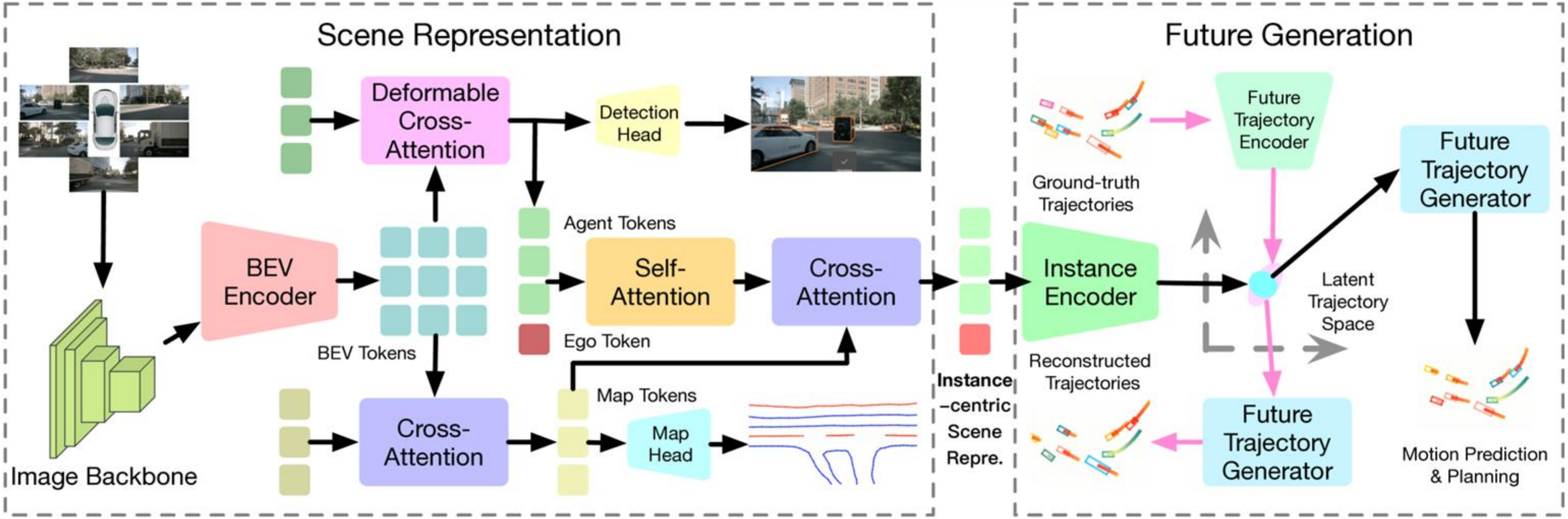
Scaling for generalization: *Flamingo*, *CoCa*, *PaLI* show strong few-shot and zero-shot performance across visual-language tasks.

Recent multimodal LLMs: *Gemini*, *GPT-4o*, and *Llama3-V* natively integrate vision + language.

Applications beyond driving: Used in *robotic navigation* (Zhang et al., 2024) and *manipulation* (Brohan et al., 2023).

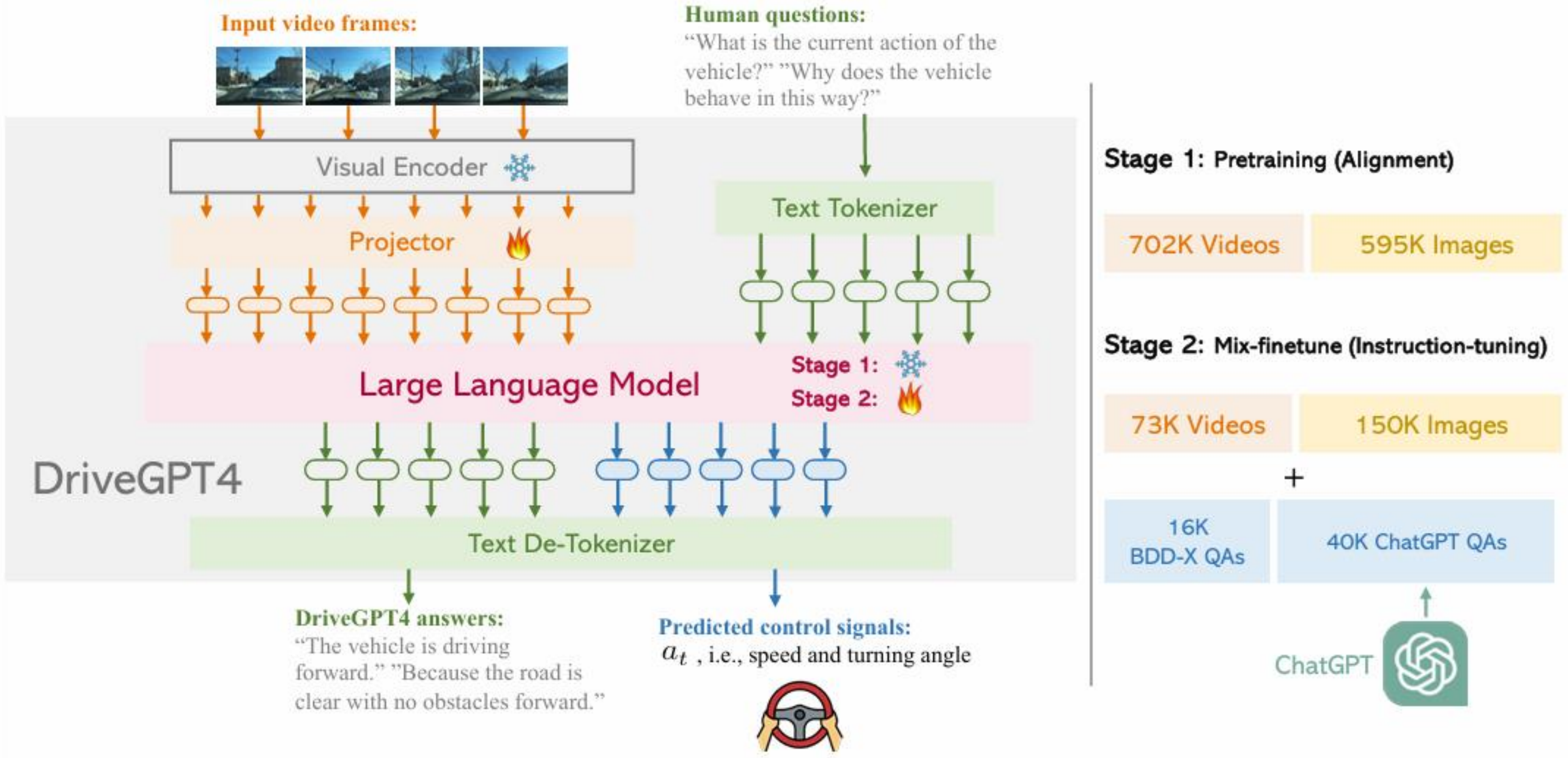
EMMA's focus: Applying MLLMs for autonomous driving, enhancing *reasoning, explainability,* and *generalization* in a generalist E2E framework.

GenAD: Generative End-to-End Autonomous Driving



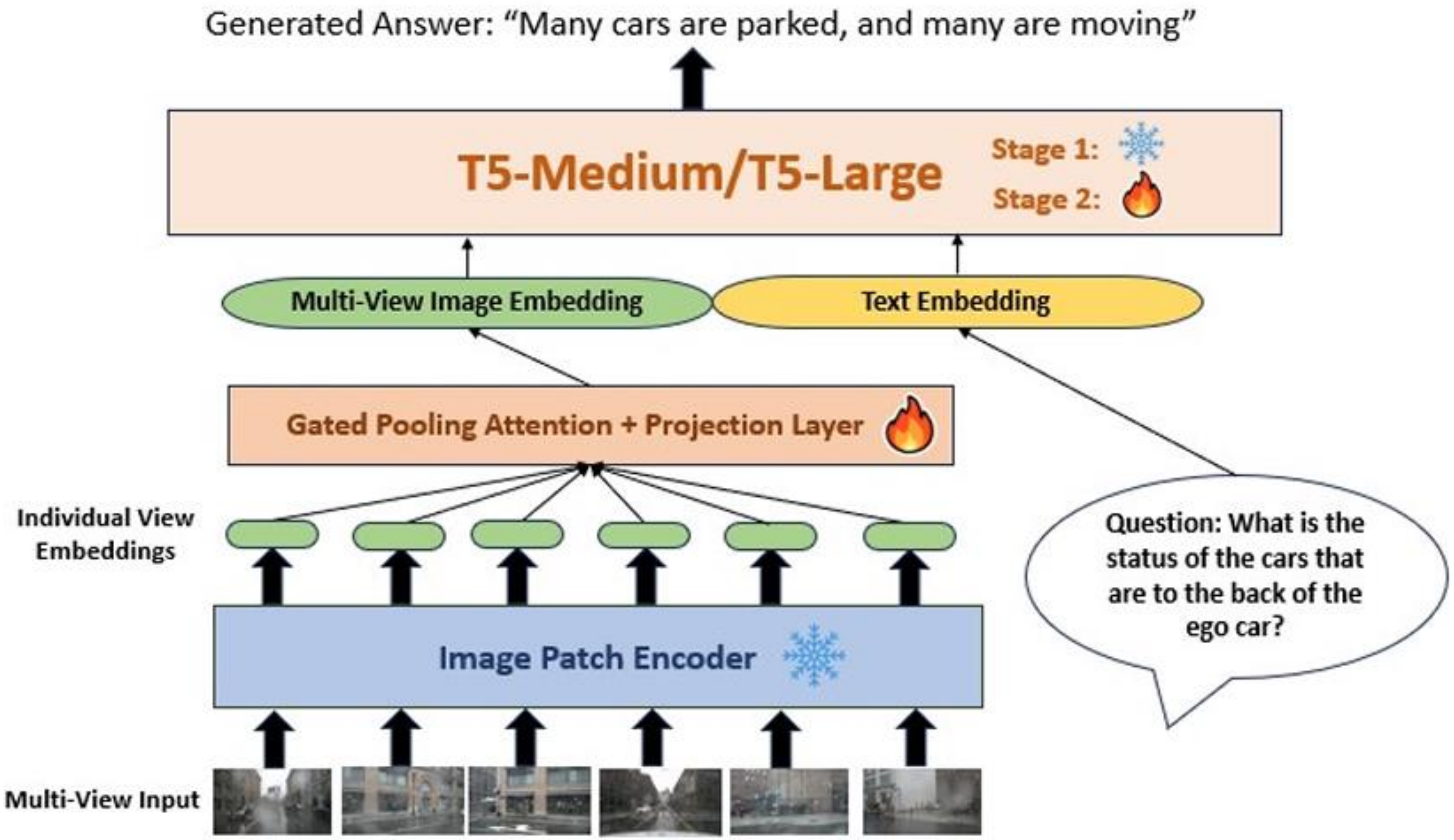
- Uses specialized scene tokens
- VAE approach instead of an MLLM
- Simultaneous trajectory generation in latent space

DriveGPT4



- Outputs lower-level control signals instead of trajectory waypoints

EM-VLM4AD

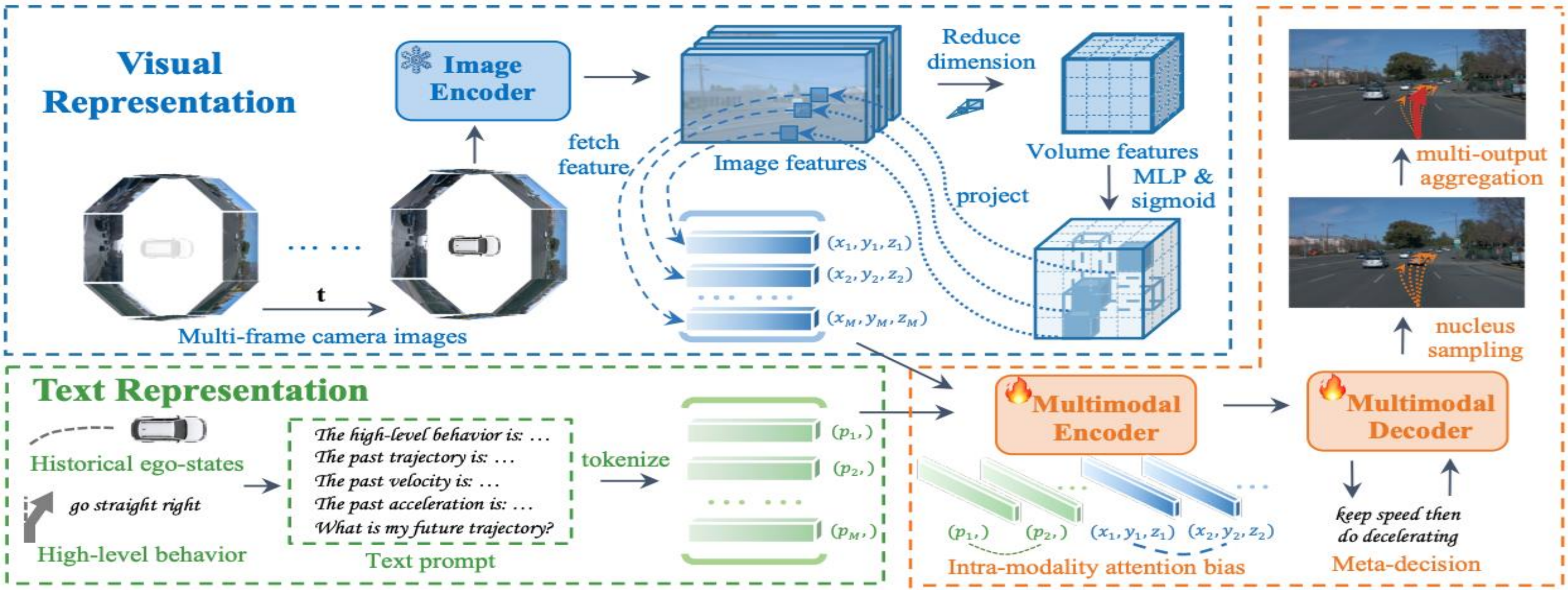


- Much smaller T5 transformer used
- Aggregates camera views into a single embedding

Extensions beyond

S4-Driver: Scalable Self-Supervised Driving Multimodal Large Language Model with Spatio-Temporal Visual Representation

Yichen Xie[1,†,*] Runsheng Xu[2,*] Tong He[2] Jyh-Jing Hwang[2] Katie Luo[3,†]
Jingwei Ji[2] Hubert Lin[2] Letian Chen[4,†] Yiren Lu[2] Zhaoqi Leng[2]
Dragomir Anguelov[2] Mingxing Tan[2]
[1] UC Berkeley, [2] Waymo LLC, [3] Cornell University, [4] Georgia Institute of Technology



Key Innovations

Self-supervised training: Learns directly from large-scale driving logs *without human labels*

Spatio-temporal representation: Encodes multi-view video into a unified 3D + time volume

Scalable learning: Performance improves naturally with more unlabeled data

3D motion planning: Operates directly in vehicle coordinate space for accurate trajectory prediction

Multi-hypothesis decoding: Aggregates multiple predicted trajectories for stable planning

Hierarchical reasoning: Combines high-level decision making with low-level control

Generalization power: Outperforms supervised models across multiple driving benchmarks

Why it Matters ?

- Reduces reliance on manual annotation and dataset curation
- Enables end-to-end learning that scales like LLMs
- Bridges perception and planning with rich 3D temporal reasoning
- Opens the door for truly generalist driving models across domains
- Improves adaptability to unseen environments and real-world variability
- Paves the way for safer, data-driven self-improving systems

EMMA v/s S4 driver comparative outlook ?

Training: EMMA → supervised multi-task; S4-Driver → self-supervised from raw driving logs

Inputs: EMMA → cameras, LiDAR, maps, text; S4-Driver → multi-view video with 3D + temporal encoding

Planning & Reasoning: EMMA → Chain-of-Thought explanations; S4-Driver → hierarchical planning with trajectory aggregation

Outputs: EMMA → textual rationale + predicted trajectories; S4-Driver → 3D vehicle trajectories

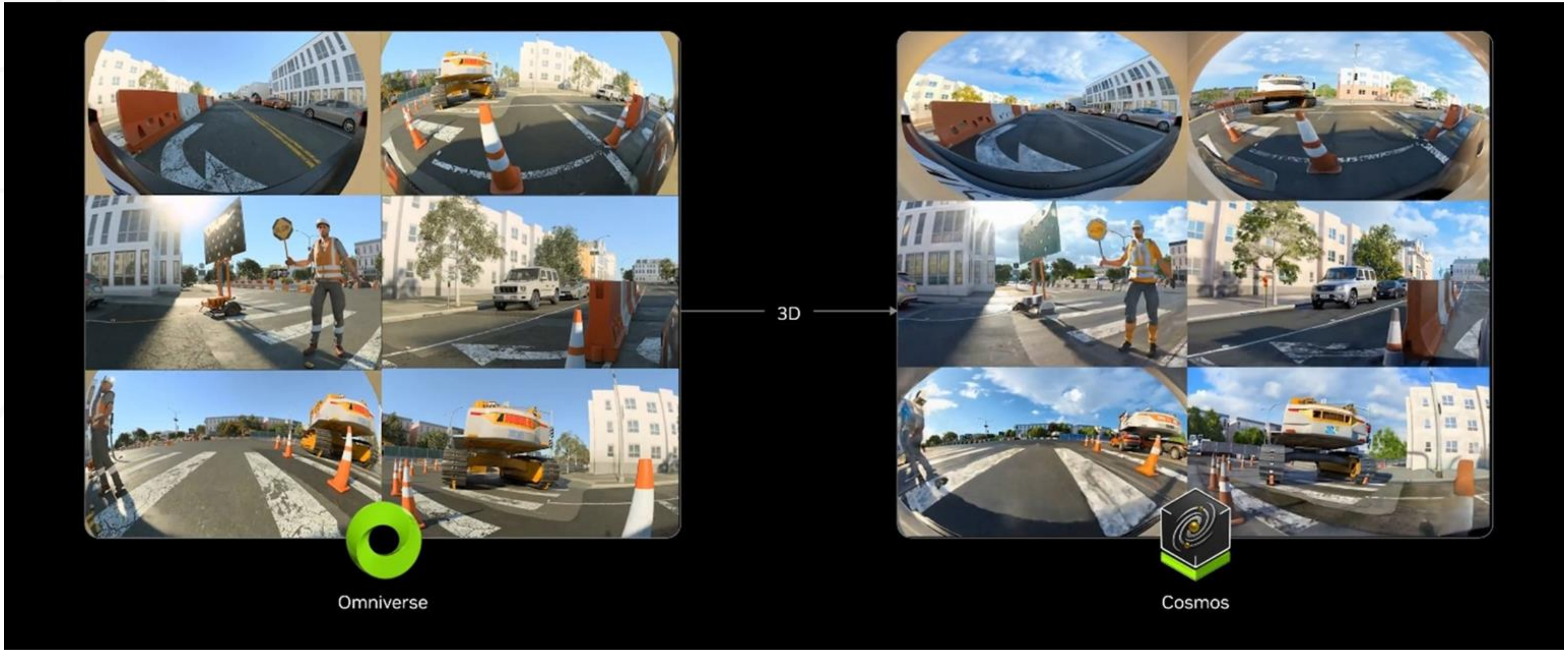
Scalability: EMMA depends on labeled datasets; S4-Driver scales naturally with more unlabeled data

Generalization: EMMA → multi-task generalist; S4-Driver → strong zero-shot generalization with rich 3D/temporal representation

Georgia Tech

Strengths: EMMA → explainable and versatile; S4-Driver → data-efficient and highly scalable

NVIDIA Cosmos (for closed-loop eval)



Leverage world foundation models for realistic synthetic data generation
Omniverse for virtual world building and SITL simulation

Conclusion/Analysis

Conclusion/Analysis

- EMMA demonstrates a generalist end-to-end driving model integrating perception, prediction, and planning.
- Multimodal inputs + language reasoning enable the model to understand complex driving scenes.
- Chain-of-Thought (CoT) improves interpretability of decision-making.
- Multi-task training enhances robustness across tasks and outperforms specialist models.
- Shows promise for scalable, generalist autonomous driving

Strengths and Weakness

Strengths:

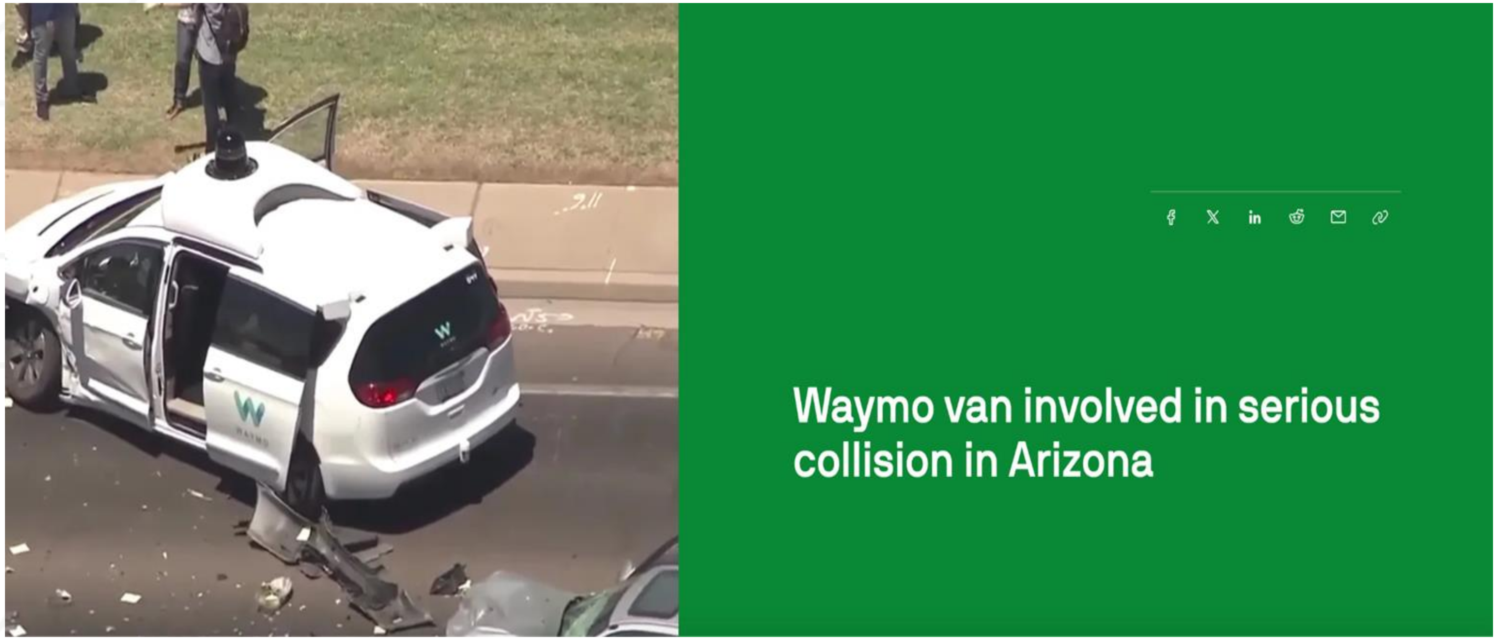
- Handles multiple driving tasks in one framework.
- Singular, streamlined and fully differentiable system
- Textual rationales increase explainability
- Chain of thought reasoning
- No reliance on HD maps
- Motion planning is self-supervised
- Strong benchmark performance

Weakness:

- Only can process 4 frames
- Could be more complex to train
- Cannot use LiDAR and radar input
- Verification of the predicted driving signals
- Model only evaluated on open-loop scenarios
- Expensive sensor simulation for closed-loop evaluation
- Challenges of onboard deployment

Open Discussions

Are these methods safe ? Can we roll it out 100 %



Outrage as Google-run driverless car flattens beloved tabby cat without stopping | Daily Mail Online

Visit ›

Baidu's Apollo Go Hits 250,000 Weekly Robotaxi Rides, Keeps Pace with Alphabet's Waymo

Are these methods safe ? Can we roll it out 100 %

LOCAL NEWS

Waymo crash under investigation in Phoenix

According to Waymo, its vehicle was hit by the second vehicle involved in the crash.

Newsweek

Elon Musk Reacts to San Francisco Cat Being Killed by Waymo Driverless Car

Waymo's robotaxis are coming to three new cities

Image: Cath Virginia / The Verge, Getty Images

/ San Diego, Las Vegas, Detroit

by Andrew J. Hawkins
Nov 3, 2025, 10:18 AM EST

20 Comments (All New)

Thank You

References

Hwang, J.-J., Xu, R., Lin, H., Hung, W.-C., Ji, J., Choi, K., Huang, D., He, T., Covington, P., Sapp, B., Zhou, Y., Guo, J., Anguelov, D., & Tan, M. (2024, October 30). *EMMA: End-to-end Multimodal Model for Autonomous Driving*. arXiv. https://arxiv.org/abs/2410.23262

Xie, Y., Xu, R., He, T., Hwang, J.-J., Luo, K., Ji, J., Lin, H., Chen, L., Lu, Y., Leng, Z., Anguelov, D., & Tan, M. (2025, May 30). *S4-Driver: Scalable self-supervised driving multimodal large language model with spatio-temporal visual representation*. arXiv. https://arxiv.org/abs/2505.24139

Georgia Tech