Carnegie Mellon University
Robotics Institute

Carnegie Mellon University
School of Computer Science

Vision-Language-Augmented Multi-Modal Multi-Agent Motion Prediction

Presenter: Patrick Chen
Supervised by Prof. Katia Sycara, Yaqi Xie
12/12/2025

Outline

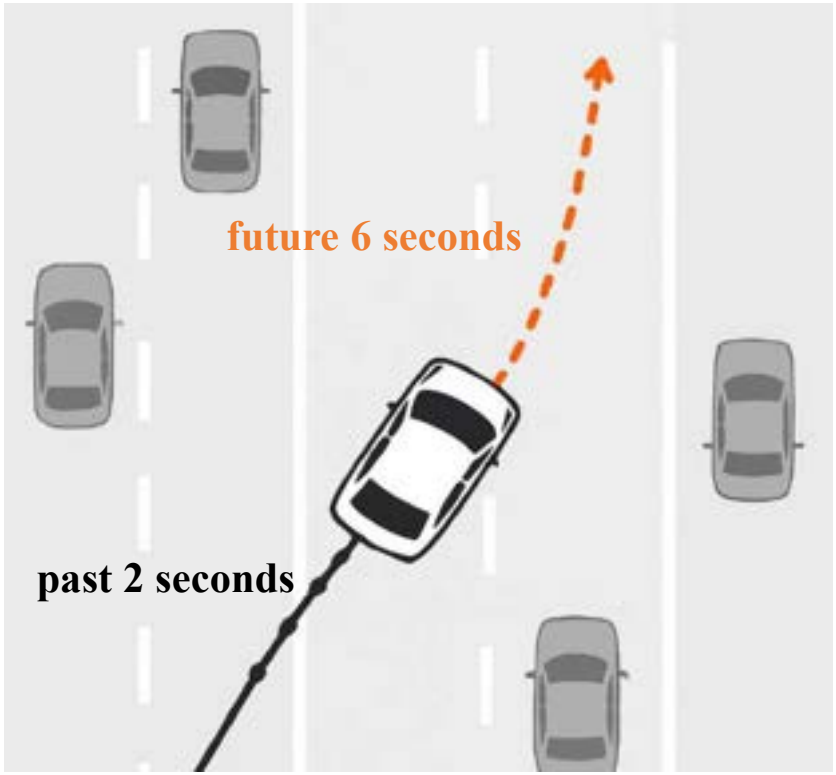
- Introduction & Motivation
- Related Work
- Proposed Method
- Results
- Conclusion & Summary

Outline

- Introduction & Motivation
- Related Work
- Proposed Method
- Results
- Conclusion & Summary

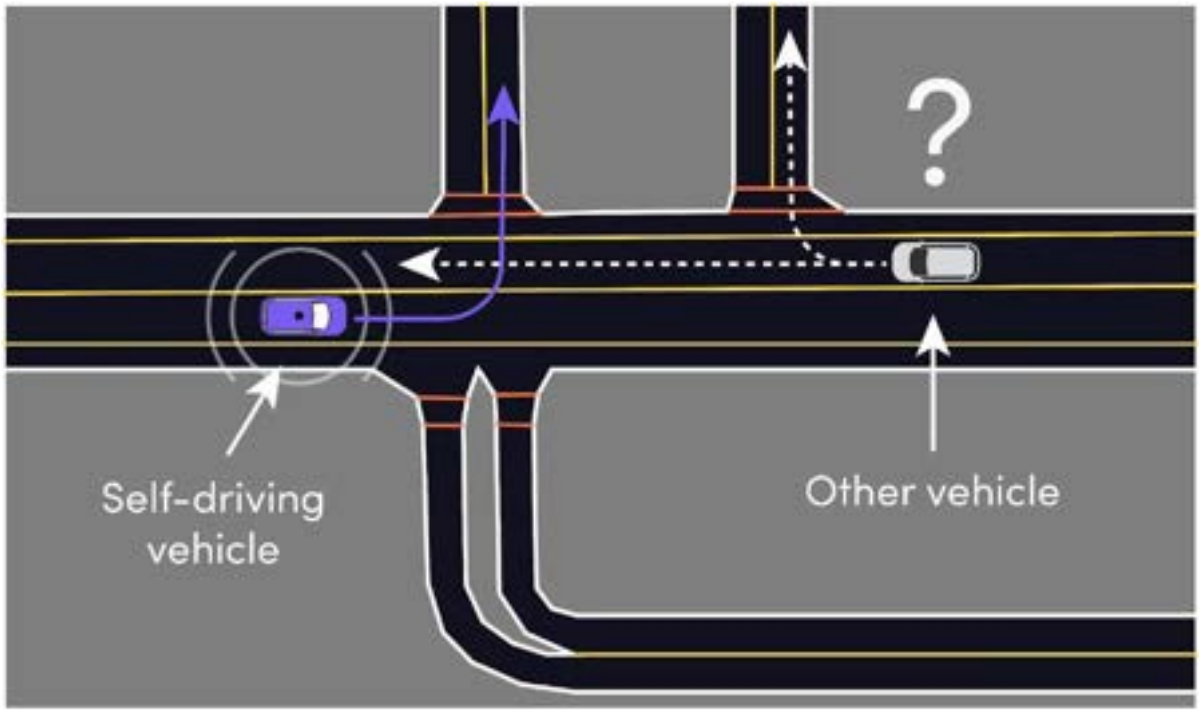
What is Motion Prediction?

- Goal: Given past trajectories, predict future motion of road agents.
  - Typical setting:
    - Past: Ns history (e.g. N=2, given 2s history)
    - Future: Xs prediction horizon (e.g. X=6, predict future 6s)
    - Each agent history is a sequence of (x, y) positions in bird-eye-view (BEV).



Multi-Agent Motion Prediction

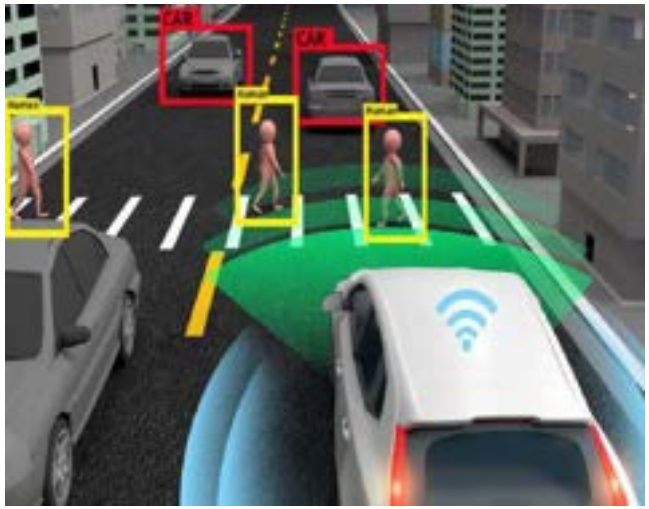
- Real scenes have multiple interacting agents (cars, trucks, pedestrians)
- Multi-agent motion prediction: predict all agents' future trajectories jointly
- Model must:
  - Capture inter-agent interactions (following, yielding, merging)
  - Understand the scene context (lanes, intersections, barriers)

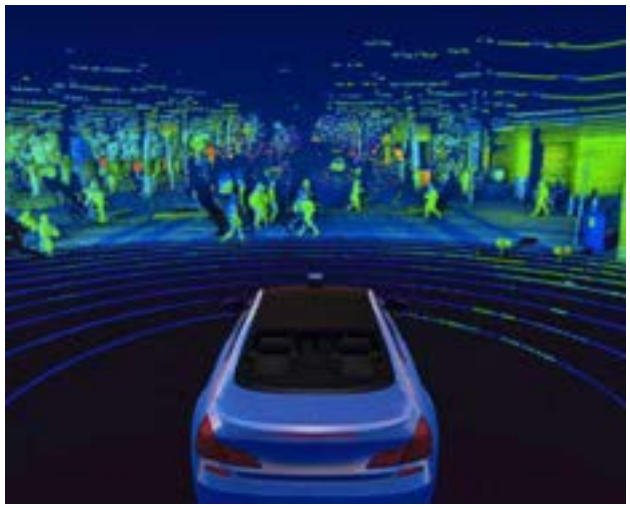


[1]. https://medium.com/wovenplanetlevel5/how-to-build-a-motion-prediction-model-for-autonomous-vehicles-29f7f81f1580

Motivation: Why Multi-Modal + Multi-Agent?

- Multi-modal perception is critical
  - Individual trajectory history of each agent is important.
  - Multi-agent interactions matter.
  - Camera gives rich semantics (lanes, traffic lights, signs, road markings).
  - LiDAR provides accurate geometry and distance, robust under lighting changes.
  - Knowledge to the scene context is another vital information.
- Our goal: learn a multi-modal model that jointly reasons about all agents' future trajectories





[2]. https://www.rsipvision.com/adas-sensors-lidars/

Outline

- Introduction & Motivation
- Related Work
- Proposed Method
- Results
- Conclusion & Summary

Related Work

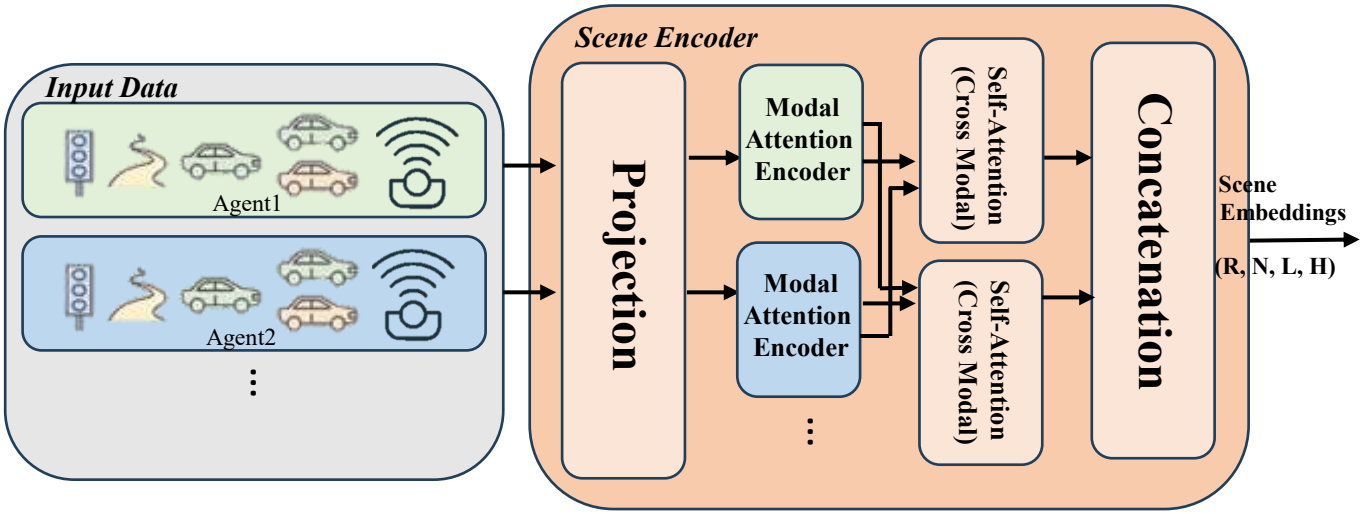
- This project builds on the following related works:
  - LLM-Augmented MTR [6] – Leverages GPT-4V with TC-Map BEV renderings and prompts to inject traffic knowledge into motion forecasting.
  - DGCN_ST_LANE [4] – Lane-based multi-agent trajectory prediction that uses a dynamic graph convolutional network over lane graphs to model agent history and inter-agent interactions.
  - THOMAS [2] – ICLR 2022 multi-agent predictor that outputs future trajectories as heatmaps and learns are combination module to sample scene-consistent, collision-free joint trajectories.
  - CASPFormer [5] – BEV-image transformer with deformable attention that performs multi-modal motion prediction from rasterized BEV context, achieving state-of-the-art results on nuScenes.

Outline

- Introduction & Motivation
- Related Work
- Proposed Method
- Results
- Conclusion & Summary

Proposed Method (1/3)

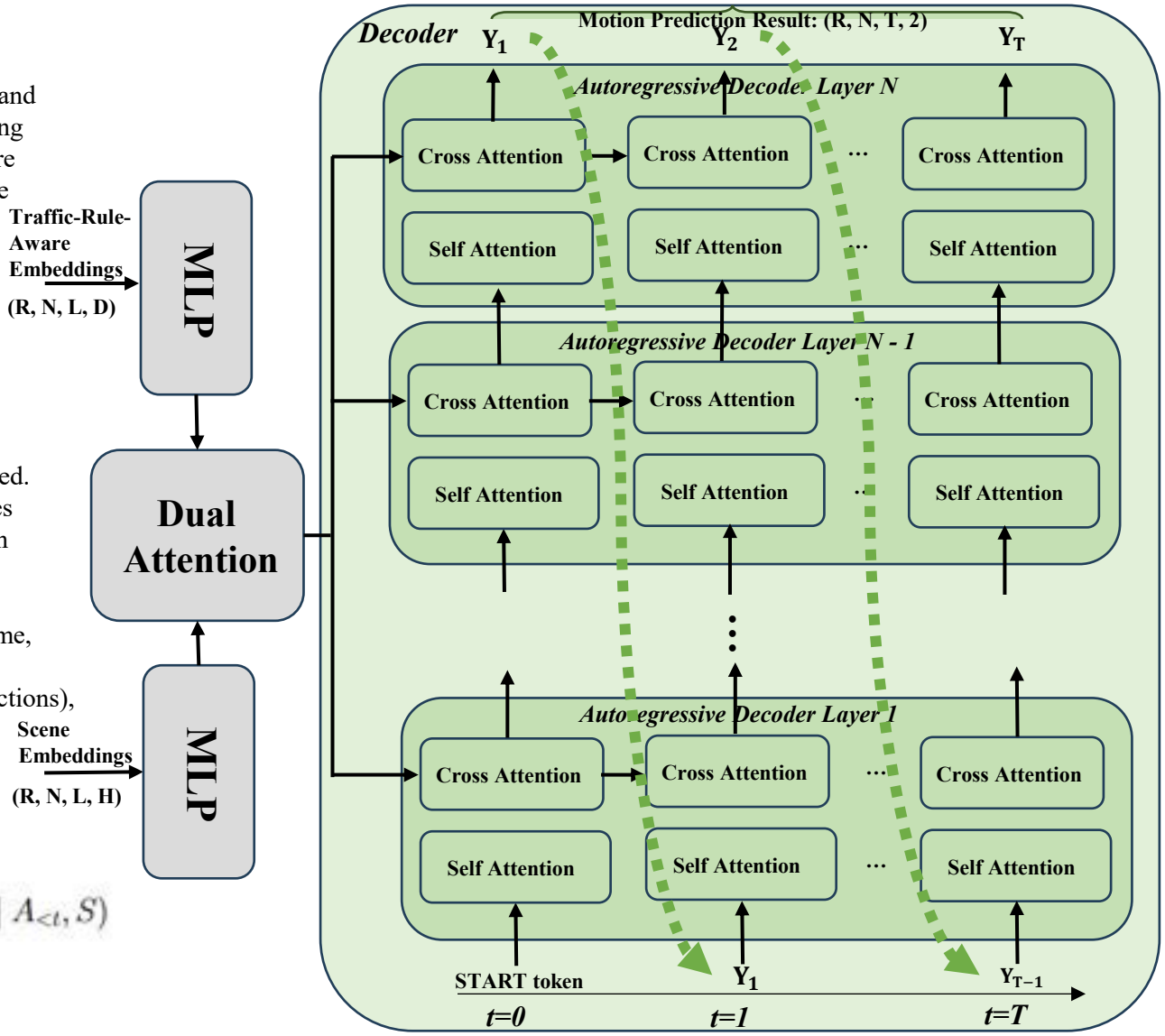
- Input Data - all inputs are projected into a unified embedding space.
  - Agent State History - past positions and headings
  - Lane Centerlines - lane center positions from HD maps
  - Traffic Light Signals - current state and spatial position
  - Agent Interactions – relative position vectors to nearby agents
  - Sensor Data – ego camera images and LiDAR point clouds
- Scene Encoder
  - Each modality above is encoded individually, and then fused via cross attention among modals to produce a unified scene embedding capturing spatial context for all agents with shape (R, N, L, H). Here, R refers to the number of rollouts, N refers to the number of agents, L is the length of the sequence, and H is the hidden dimension.



Proposed Method (2/3)

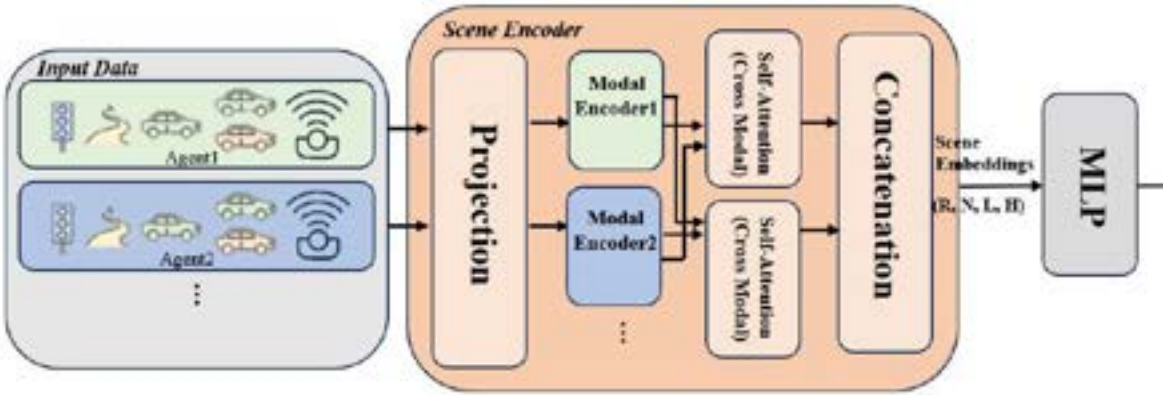
- Decoder
  - **Dual Attention Fusion**: Scene and traffic-rule embeddings are fused using bi-directional cross-attention to ensure both modalities are integrated into the prediction process.
  - **Autoregressive Generation**: The decoder generates predictions autoregressively, attending to previous tokens with self-attention and integrating scene and traffic-rule information through cross-attention.
  - **Training & Inference**: During training, teacher forcing is used. During inference, the model generates predictions step-by-step using its own previous outputs.
  - **Output:** The decoder produces motion predictions for agents over time, with output shape (R,N,T,2):
    R: Number of rollouts (distinct predictions),
    N: Number of agents,
    T: Number of time steps,
    2: Predicted x and y coordinates for each agent.

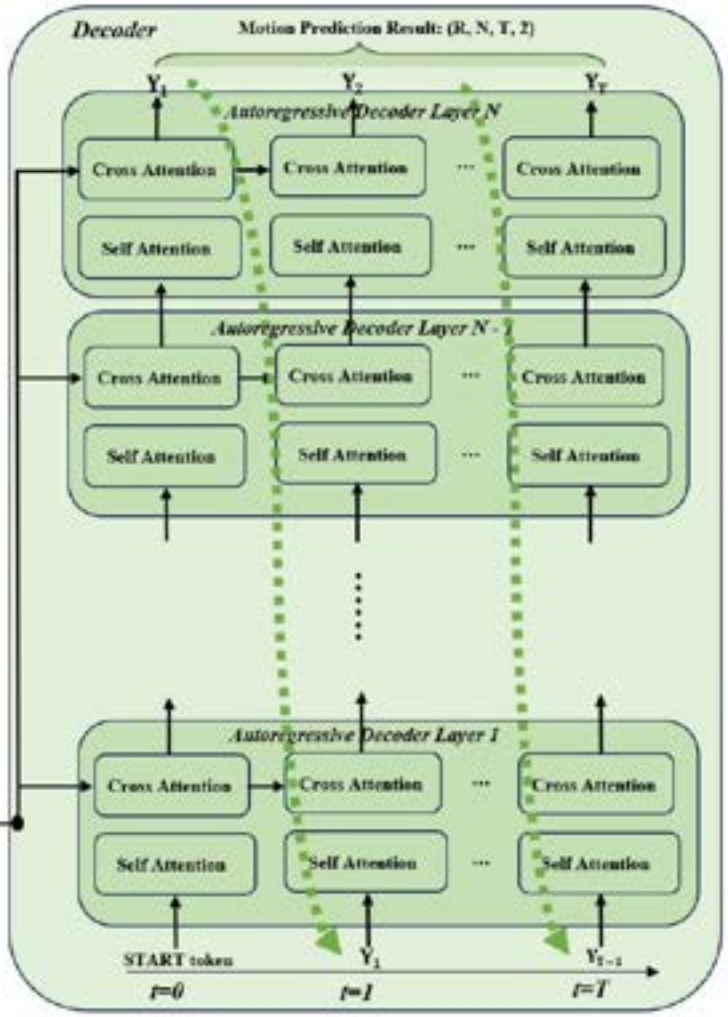
$$\mathcal{L}_{\text{motion}} = -\sum_{t=1}^{T}\sum_{n=1}^{N} \log p_\theta(a_t^n \mid A_{<t}, S)$$



Proposed Method (3/3)

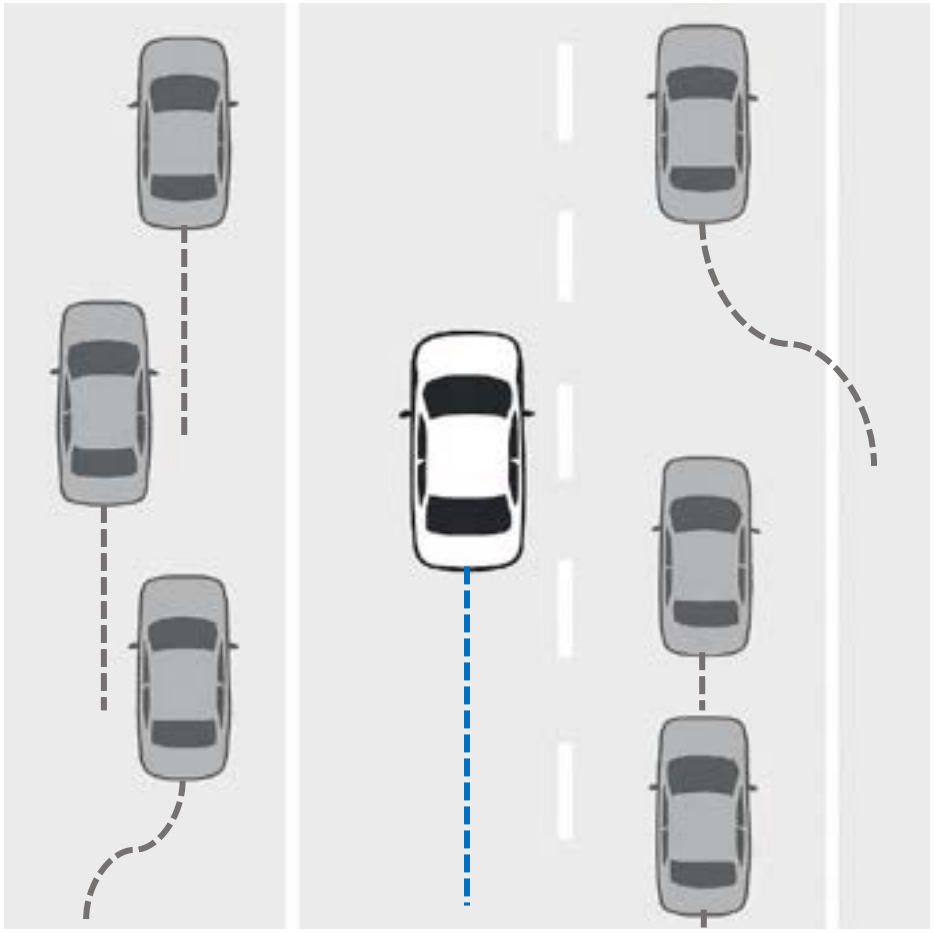
- Overall Model Architecture
  - Input Data
    - Agent State History - past positions and headings
    - Lane Centerlines - lane center positions from HD maps
    - Traffic Light Signals - current state and spatial position
    - Agent Interactions – relative position vectors to nearby agents
    - Sensor Data – ego camera images and LiDAR point clouds
  - Scene Encoder
    - Each modality is encoded individually, and then fused via cross attention to a unified embedding.
  - Decoder
    - **Autoregressive Generation:** The decoder generates predictions autoregressively, attending to previous tokens with self-attention and integrating scene information through cross-attention.
    - **Training & Inference:** During training, teacher forcing is used with MSE Loss. During inference, the model generates predictions step-by-step using its own previous outputs.





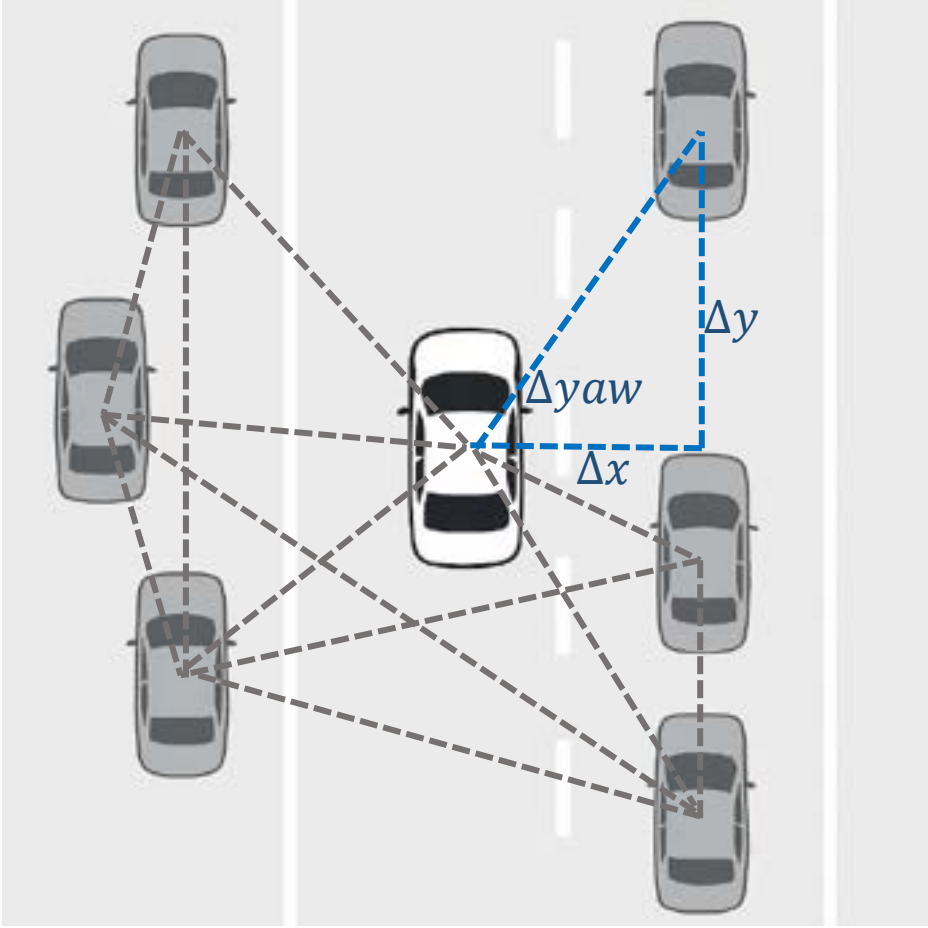
What We Have Done So Far (1/5)

- Baseline 1: Uni-Modal (History only)
  - Use each agent's trajectory history
  - Use GRU network to encode each agent's trajectory history



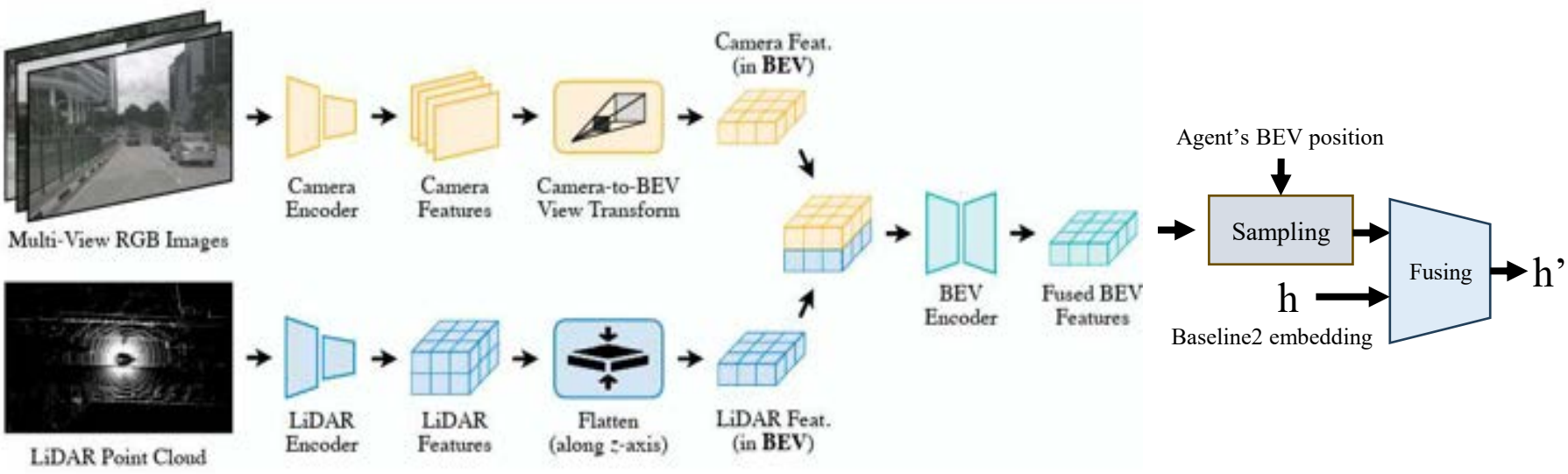
What We Have Done So Far (2/5)

- Baseline 2: History + Multi-Agent Relationships
  - Use each agent's geometry relationship: [Δx, Δy, Δyaw, Δv_x, Δv_y, Δa_x, Δa_y]
  - Use attention to encode each agent's relationship embedding



What We Have Done So Far (3/5)

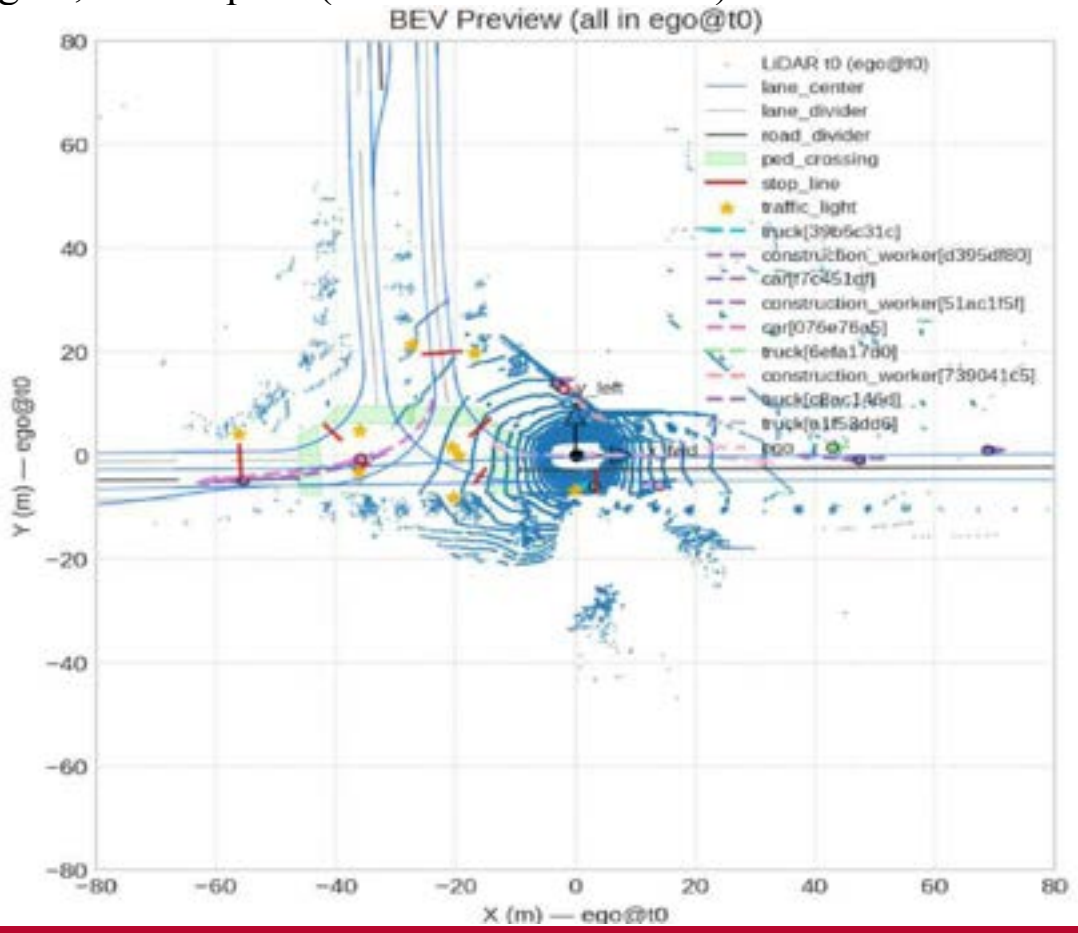
- Baseline 3: History + Multi-Agent Relationship + BEV Fusion
  - Use ego car's camera image + LiDAR sensors
  - Use BEV Fusion framework
  - Extract each agent's BEV feature through back projection and sampling and then fuse together with previous embeddings.



[3]. Z. Liu et al., *"BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation"*, NeurIPS 2022.

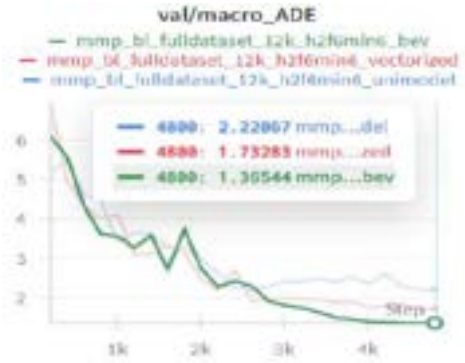
What We Have Done So Far (4/5)

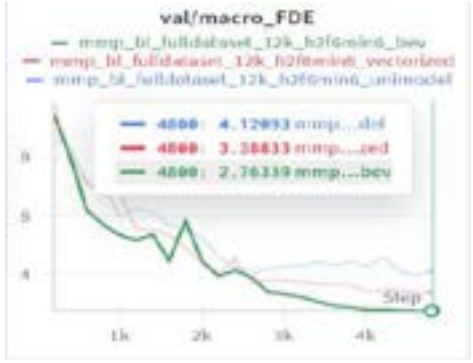
- Dataset Generation: nuScenes
  - Sweep the whole nuScenes dataset with past 2s and future 6s context window
  - Define BEV region to be x: [-80m, 80m], y: [-80m, 80m]
  - Generating 12,782 samples. (8-1-1 for train-val-test)



What We Have Done So Far (5/5)

- Quantitative Results





Method	ADE	FDE
nuScenes Leaderboard		
DGCN_ST_LANE (Agent History + Agent Relationship)	**1.092**	3.624
DSS	1.192	6.640
CASPFormer (Agent History + Agent Relationship + Image)	1.148	6.702
Ours Method		
Baseline1: Uni-model (Agent History)	2.221	4.121
Baseline2: Agent History + Agent Relationship	1.733	3.388
Baseline3: Agent History + Agent Relationship + BEV	1.365	**2.763**

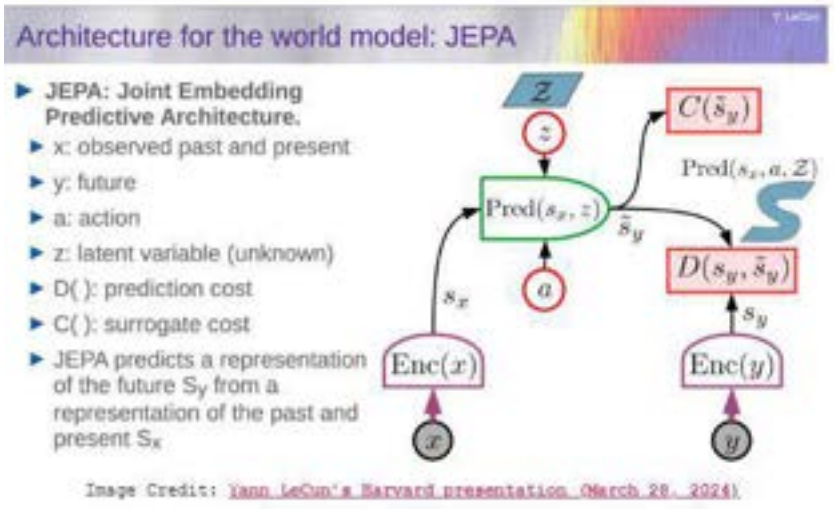
What We Did Next

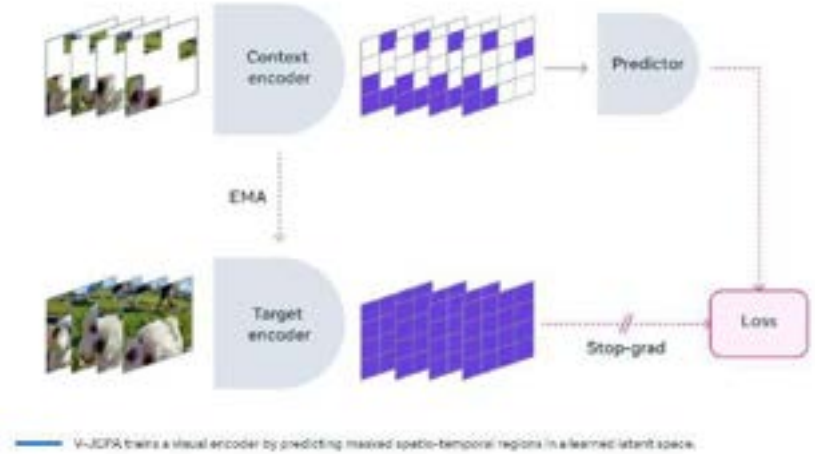
- Upgrade the encoder:
  - Make the model learn the world dynamics.
  - Make the model learn the traffic rules.

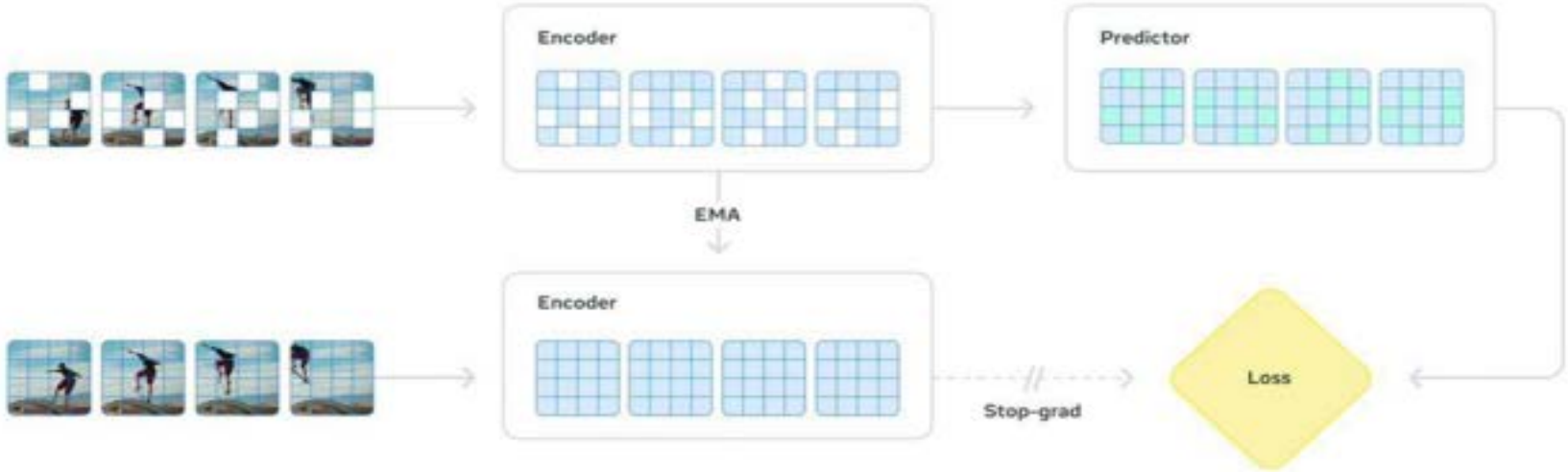
What We Did Next

- Upgrade the encoder:
  - Make the model learn the world dynamics.
  - Make the model learn the traffic rules.

JEPA SSL Pretraining

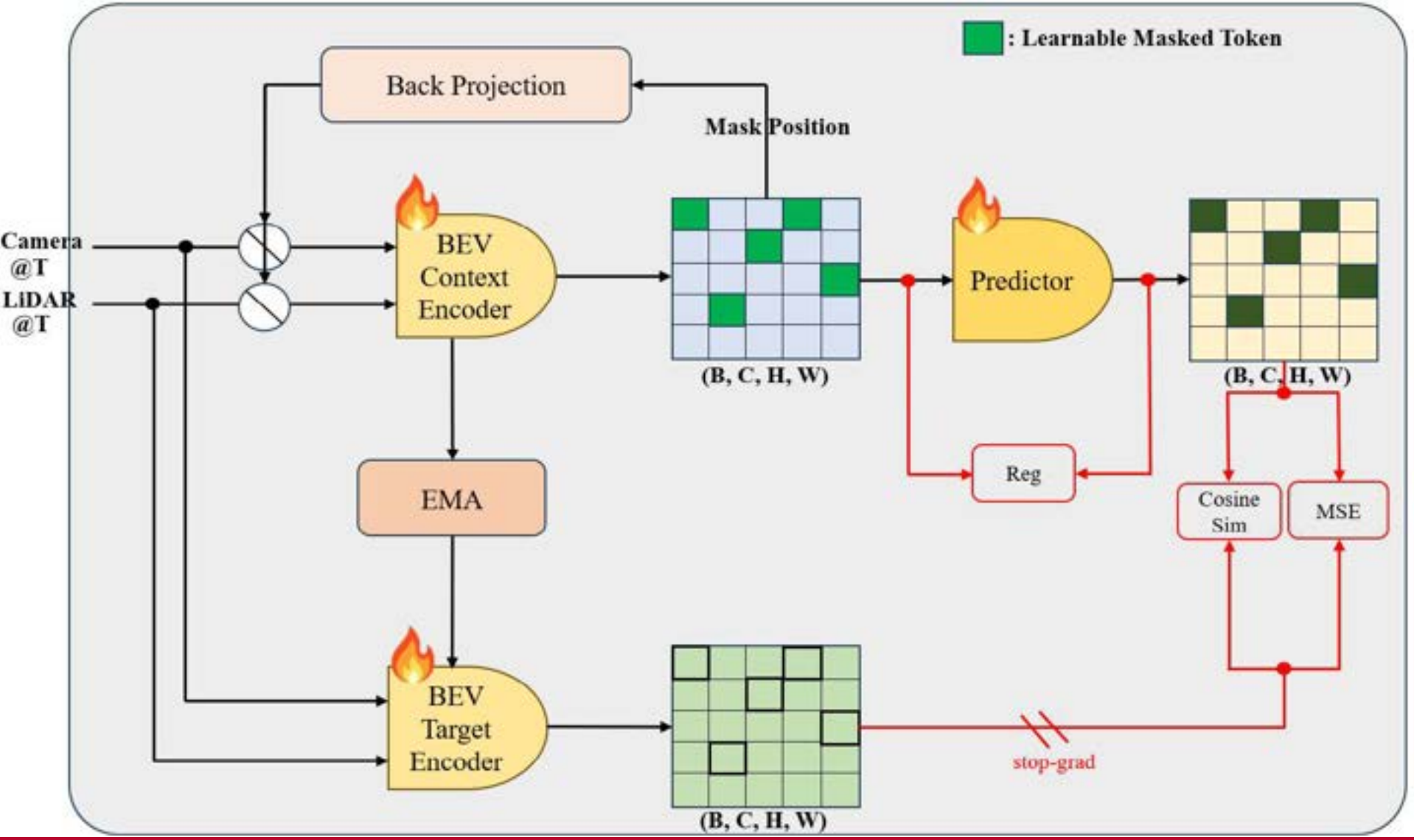




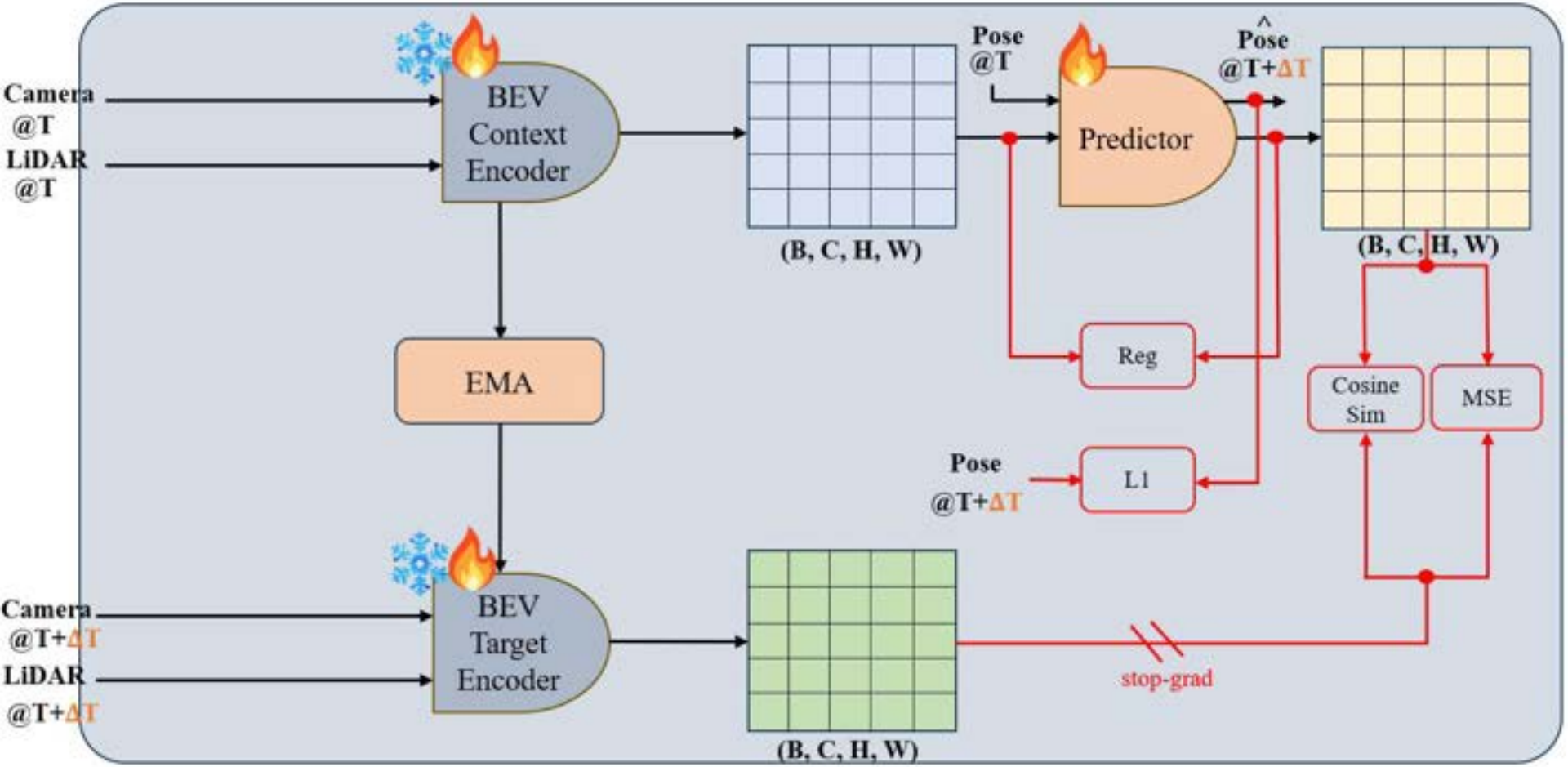


[4]. https://ai.meta.com/research/publications/revisiting-feature-prediction-for-learning-visual-representations-from-video/
[5]. https://ai.meta.com/blog/v-jepa-2-world-model-benchmarks/

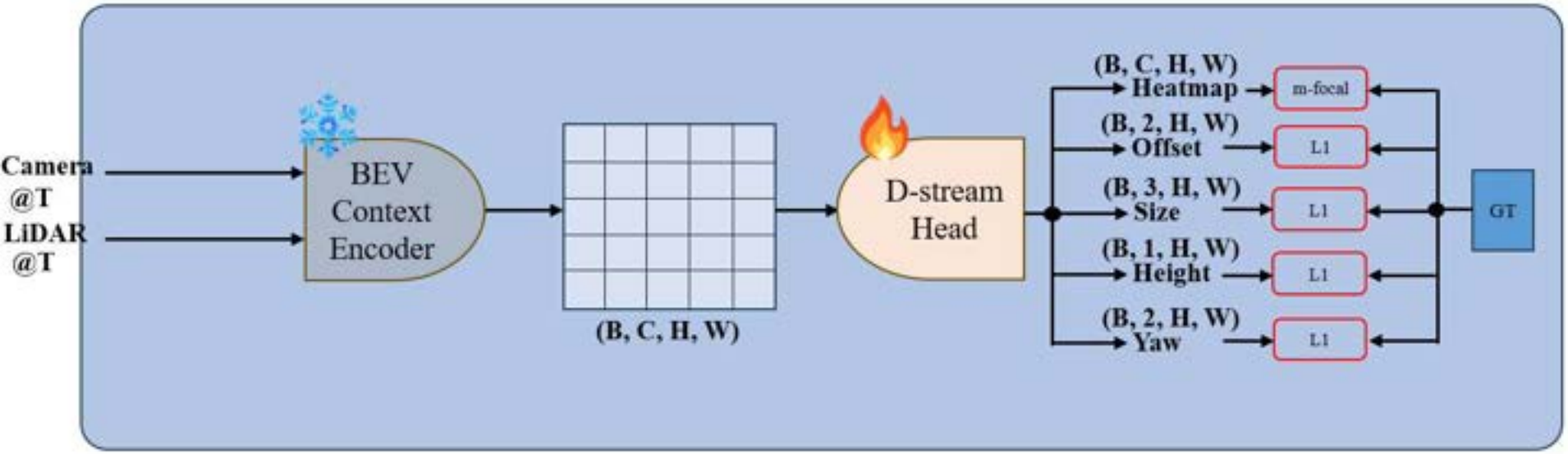
JEPA SSL Pretraining – Stage1 Spatial Pretraining



JEPA SSL Pretraining – Stage2 Temporal Pretraining



Fine-Tuning Stage



Fine-Tuning Stage

- JEPA [9] shows good visual recognition results on various downstream tasks.

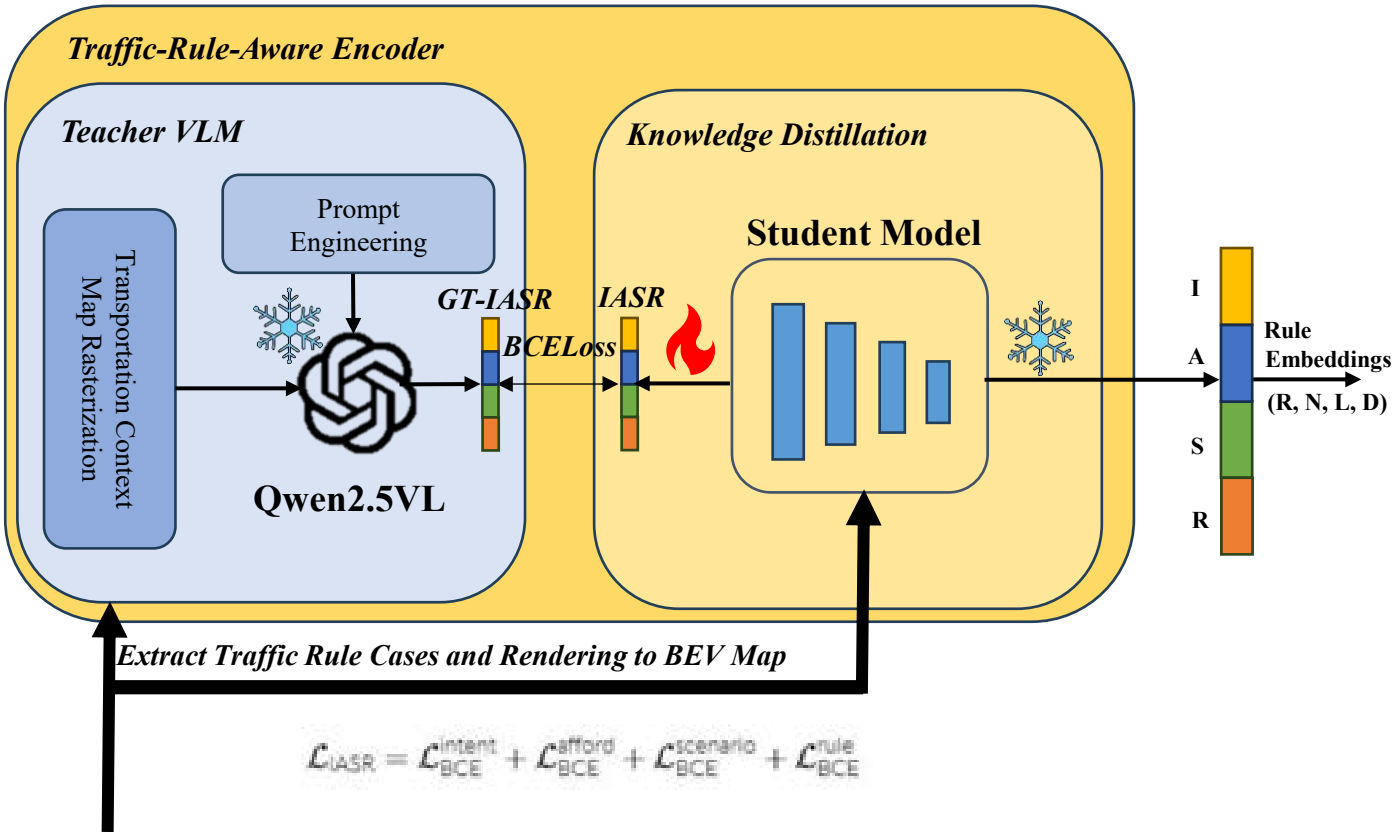
Method	Arch.	Params.	Data	*Video Tasks* K400 (16×8×3)	SSv2 (16×2×3)	AVA	*Image Tasks* IN1K	Places205	iNat21
Methods pretrained on Images									
I-JEPA	ViT-H/16_{512}	630M	IN22K	79.7	50.0	19.8	84.4	66.5	85.7
OpenCLIP	ViT-G/14	1800M	LAION	81.8	34.8	23.2	85.3	**70.2**	83.6
DINOv2	ViT-g/14	1100M	LVD-142M	**83.4**	50.6	24.3	**86.2**	68.4	**88.8**
Methods pretrained on Videos									
MVD	ViT-L/16	200M	IN1K+K400	79.4	66.5	19.7	73.3	59.4	65.7
OmniMAE	ViT-H/16	630M	IN1K+SSv2	71.4	65.4	16.0	76.3	60.6	72.4
VideoMAE	ViT-H/16	630M	K400	79.8	66.2	20.7	72.3	59.1	65.5
VideoMAEv2	ViT-g/14	1100M	Un.Hybrid	71.2	61.2	12.9	71.4	60.6	68.3
Hiera	Hiera-H	670M	K400	77.0	64.7	17.5	71.4	59.5	61.7
V-JEPA	ViT-L/16	200M	VideoMix2M	80.8	69.5	25.6	74.8	60.3	67.8
	ViT-H/16	630M		**82.0**	71.4	**25.8**	75.9	61.7	67.9
	ViT-H/16_{384}	630M		81.9	**72.2**	25.0	**77.4**	**62.8**	**72.6**

What We Did Next

- Upgrade the encoder:
  - Make the model learn the world dynamics.
  - Make the model learn the traffic rules.

VLM Knowledge Distillation (1/4)

- We rendered BEV map from input data and craft prompt together and sent to Qwen2.5VL model to extract traffic rule embedding, and treat it as a teacher to distill the information through supervised training a student model. During inference, we directly use the rendered BEV map and the frozen student model to output the calculated traffic rule embedding.



VLM Knowledge Distillation (2/4)

- Proposed IAS Embedding:
  {intention_type, affordance_type, scenario_type} $\in R^{27}$

```
intention_type = {
   0: "KEEP_LANE_OR_GO_STRAIGHT",
   1: "PREPARE_OR_DO_LEFT_TURN",
   2: "PREPARE_OR_DO_RIGHT_TURN",
   3: "STOP_FOR_TRAFFIC_CONTROL",
   4: "STOP_FOR_PEDESTRIAN_OR_OBJECT",
   5: "PULL_OVER_OR_PARK_LEGAL",
   6: "STATIONARY_IN_LANE_ABNORMAL",
   7: "U_TURN_OR_LANE_CHANGE"          # U-turn
}

affordance_type = {
   0: "CAN_GO_STRAIGHT",
   1: "CAN_TURN_LEFT",
   2: "CAN_TURN_RIGHT",
   3: "FRONT_BLOCKED_BY_OBSTACLE",
   4: "FRONT_BLOCKED_BY_RULE",
   5: "MUST_STOP_AT_STOP_LINE",
   6: "MUST_YIELD_TO_PEDESTRIAN",
   7: "MUST_YIELD_TO_ONCOMING",
   8: "LANE_CHANGE_ALLOWED",
   9: "ROADSIDE_STOP_ALLOWED"
}
```

```
scenario_type = {
   0: "STRAIGHT_FREE_FLOW_ROAD",
   1: "INTERSECTION_SIGNALIZED",
   2: "INTERSECTION_STOP_OR_YIELD",
   3: "PEDESTRIAN_CROSSING_ZONE",
   4: "MERGE_OR_ONRAMP_OR_RAMP",
   5: "PARKING_OR_DRIVEWAY_AREA",
   6: "ROADSIDE_OR_SHOULDER",
   7: "COMPLEX_MULTI_AGENT_NEGOTIATION",
   8: "UNSURE"
}
```

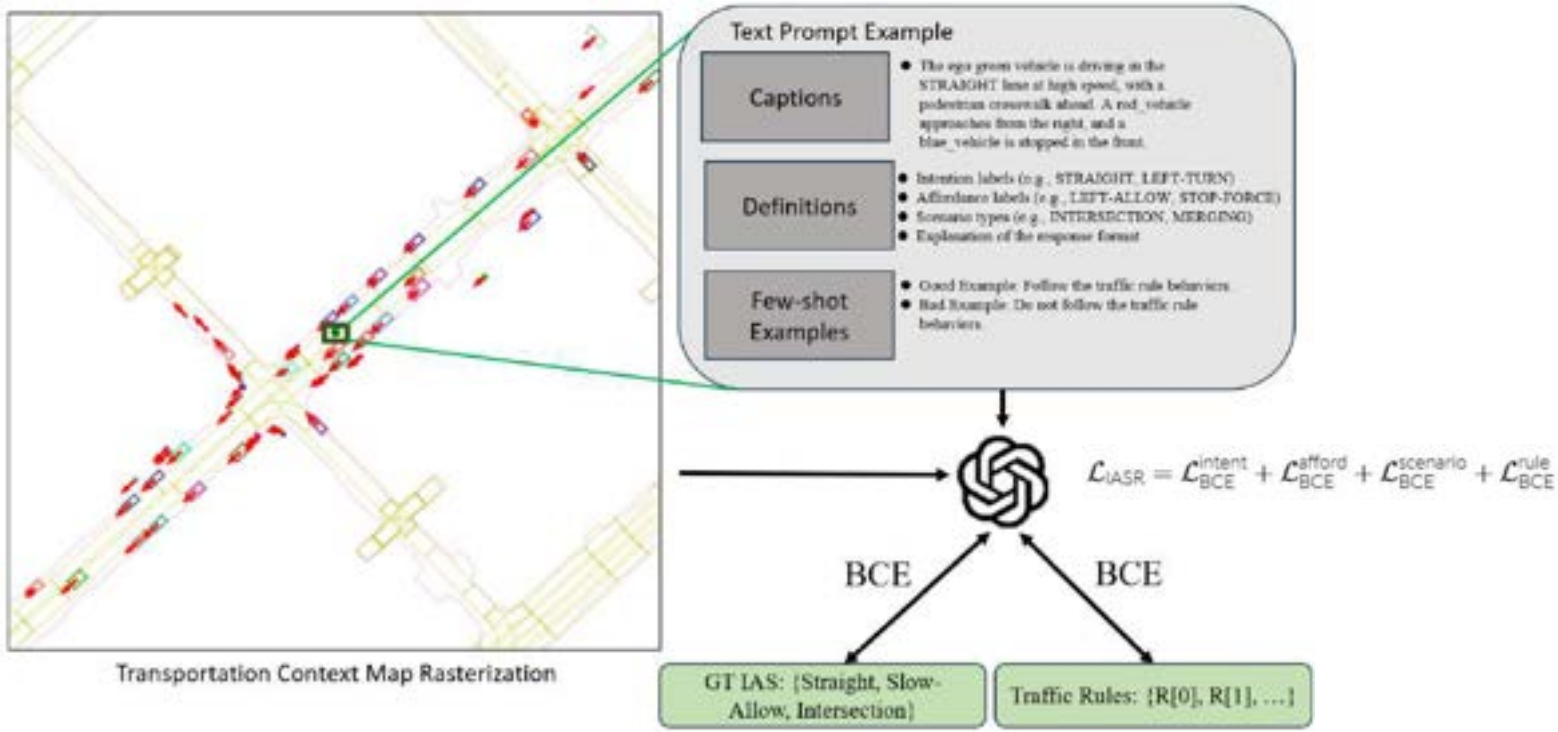
VLM Knowledge Distillation (3/4)

- Proposed Traffic Rules Embedding:
  $\{\text{traffic\_rules}\} \in R^9$

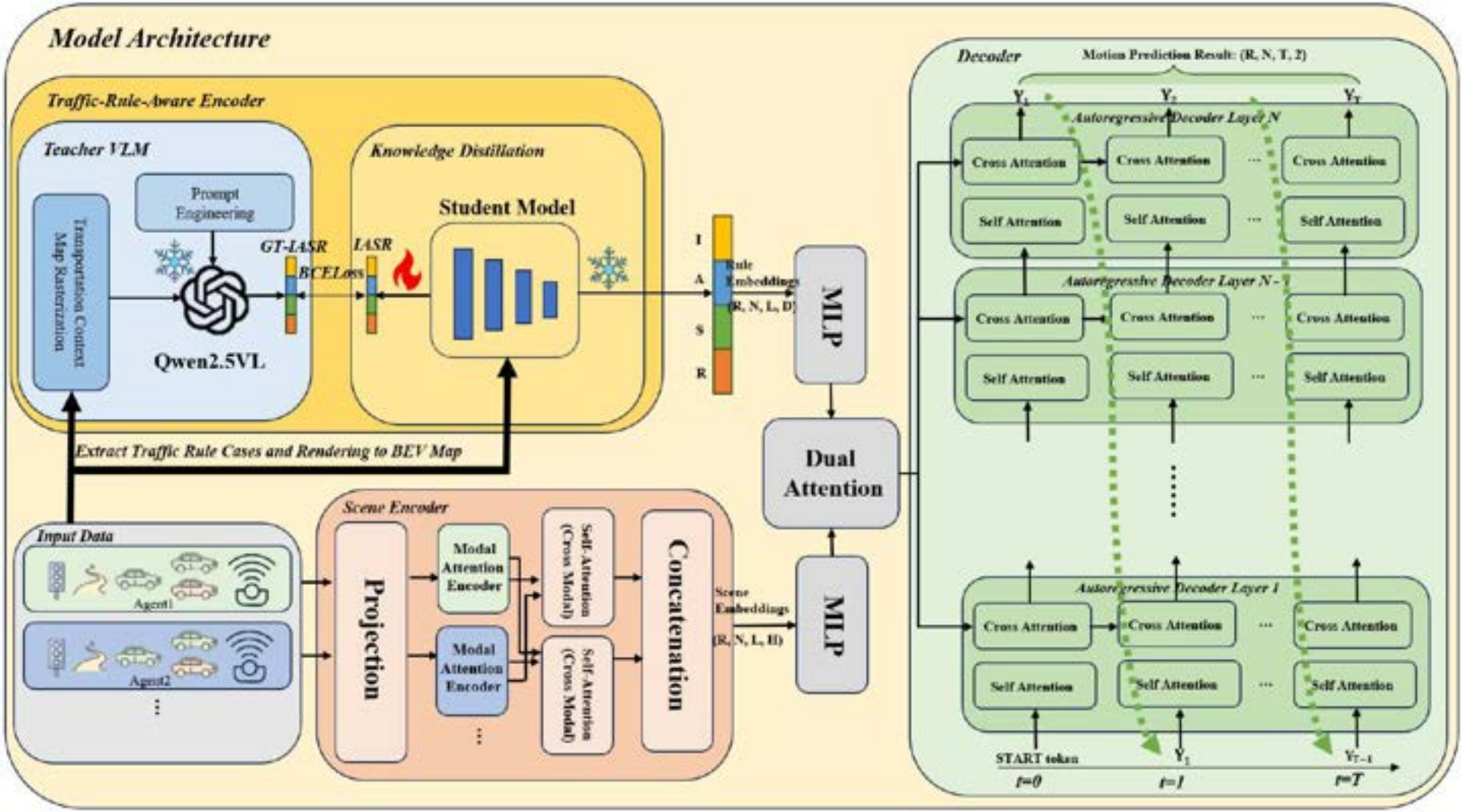
```
TRAFFIC_RULES = {
   0: "RULE_STOP_LINE_AHEAD",
   # A visible STOP LINE lies ahead along the current lane direction.
   1: "RULE_MUST_STOP_BEFORE_STOP_LINE",
   # The agent is approaching an intersection with a STOP LINE and
   # should come to a complete stop before crossing that line.
   2: "RULE_TRAFFIC_LIGHT_AHEAD",
   # A TRAFFIC LIGHT icon is located ahead at the intersection that
   # controls this approach (state unknown; we only know it exists).
   3: "RULE_CROSSWALK_AHEAD",
   # A CROSSWALK (green shaded area) lies ahead on the agent's lane.
   4: "RULE_AGENT_ON_OR_BLOCKING_CROSSWALK",
   # Some traffic agent (colored dot) is currently inside, entering,
   # or stopped on the CROSSWALK area.
   5: "RULE_INTERSECTION_CONFLICT_ZONE_AHEAD",
   # The agent's lane is entering an intersection region where lane
   # centers from other directions cross, creating potential conflicts.
   6: "RULE_TURN_MUST_YIELD_TO_THRU_TRAFFIC",
   # The agent is making a turning movement across another stream
   # of traffic that continues straight; the straight-moving traffic
   # has priority.
   7: "RULE_THRU_LANE_HAS_PRIORITY",
   # The agent is on a main through lane (straight lane center passing
   # through the junction), and crossing/merging side approaches should yield.
   8: "RULE_NO_STOPPING_IN_THRU_LANE",
   # The agent is currently in the middle of a through lane, away from
   # STOP LINE / CROSSWALK / intersection entries; stopping here is not allowed.
}
```

VLM Knowledge Distillation (4/4)

- The prompts are crafted with an auto-generated scene caption, predefined label categories, traffic rules, and a strict response format [7]. Few-shot examples are included to guide consistent outputs.
- Definition of output traffic-rule-aware embedding:
  - Intentions (I): Agent's intended motion (e.g., STRAIGHT, LEFT-TURN, STOP) encoded as a weighted one-hot vector R^8.
  - Affordances (A): Legality/feasibility of actions (e.g., LEFT-ALLOW, STOP-FORCE) encoded as a binary vector R^{10}.
  - Scenario Types (S): High-level context (e.g., INTERSECTION, MERGING) encoded as a binary vector R^9.
  - Rules(R): Traffic rules to drive IAS decisions (e.g., STOP-LINE-AHEAD, NO-STOPPING), encoded as a binary vector R^9.



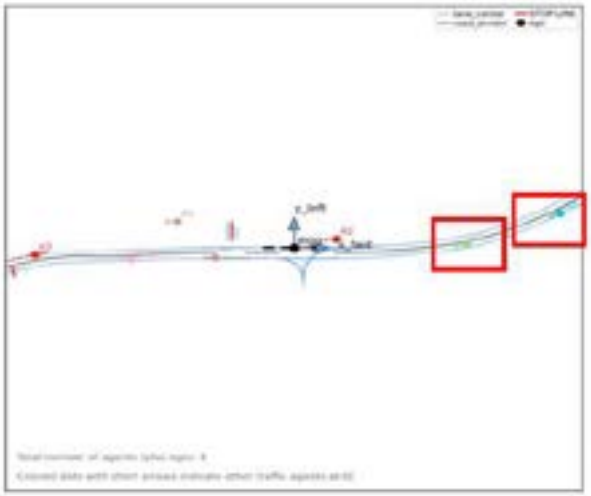
Put It Together: The Whole Proposed Model



Outline

- Introduction & Motivation
- Related Work
- Proposed Method
- Results
- Conclusion & Summary

Results: Knowledge Distillation



A4:
I=[0.9, 0.1, 0, 0, 0, 0, 0, 0] → GO STRAIGHT
A=[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] → CAN GO STRAIGHT
S=[1, 0, 0, 0, 0, 0, 0, 0, 0] → STRAIGHT ROAD
R=[0, 0, 0, 0, 0, 0, 0, 1, 1] → THRU LANE PRIORITY, NO STOP THRU LANE

A0:
I=[0.9, 0.1, 0, 0, 0, 0, 0, 0] → GO STRAIGHT
A=[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] → CAN GO STRAIGHT
S=[1, 0, 0, 0, 0, 0, 0, 0, 0] → STRAIGHT ROAD
R=[0, 0, 0, 0, 0, 0, 0, 1, 1] → THRU LANE PRIORITY, NO STOP THRU LANE

Knowledge Distillation Result from Qwen2.5VL-32B-Instruct

A4:
I=[0, 0, 0, 0, 0, 0, 0, 1.0] → U-TURN OR LANE CHANGE
A=[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] → CAN GO STRAIGHT
S=[0, 0, 0, 0, 0, 0, 0, 0, 0] → NO OUTPUT
R=[1, 0, 0, 0, 0, 0, 0, 0, 0] → RULE STOP LINE AHEAD

A0:
I=[0, 0, 0, 0, 0, 0, 0, 1.0] → U-TURN OR LANE CHANGE
A=[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] → CAN GO STRAIGHT
S=[0, 0, 0, 0, 0, 0, 0, 0, 0] → NO OUTPUT
R=[1, 0, 0, 0, 0, 0, 0, 0, 0] → RULE STOP LINE AHEAD

Knowledge Distillation Result from Qwen2.5VL-7B-Instruct

Results: Motion Prediction

- Metrics
  - We evaluate with three standard metrics: minADE, minFDE, and MR, averaged over all agents. Let $p_t^k = (x_t^k, y_t^k)$ be the predicted position at time t for rollout k, and p_t^* the ground-truth.
    - minADE: Best average L2 distance over T steps across k rollouts:

$$\min_k \frac{1}{T} \sum_{t=1}^{T} \|\mathbf{p}_t^k - \mathbf{p}_t^*\|_2$$

    - minFDE: Best final-step L2 distance across k rollouts:

$$\min_k \|\mathbf{p}_T^k - \mathbf{p}_T^*\|_2$$

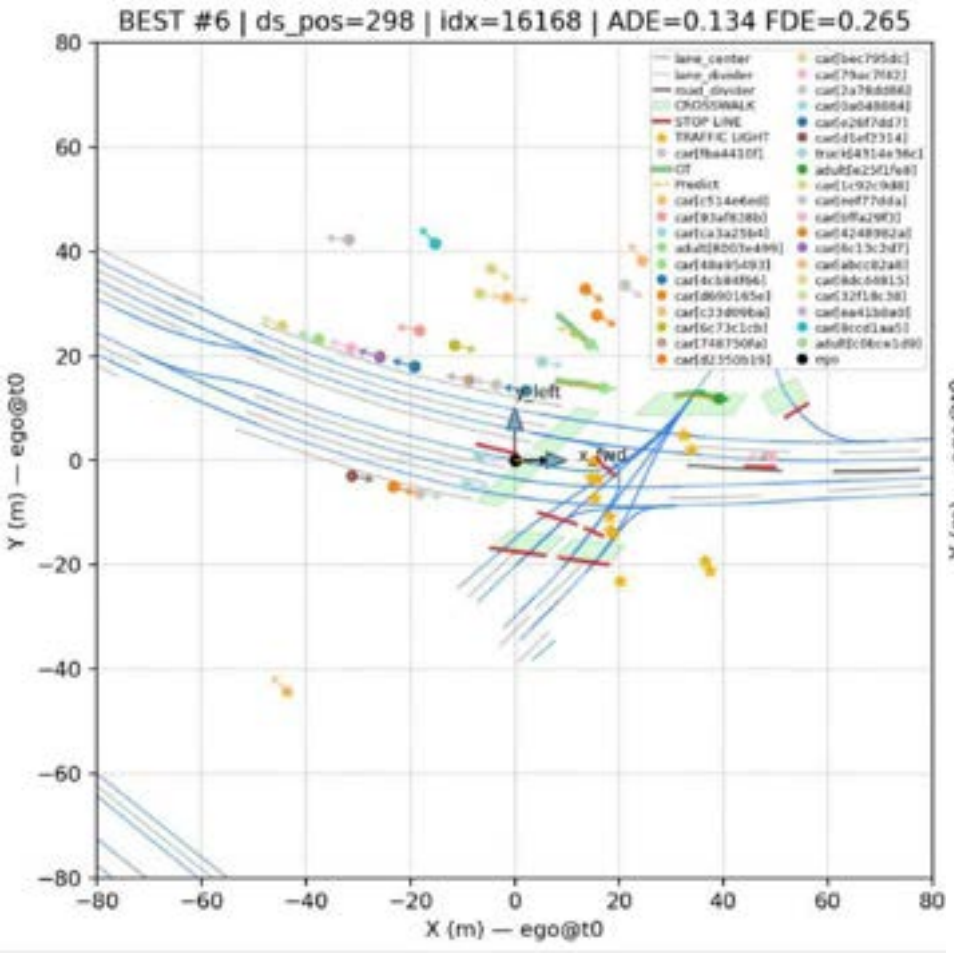
Results: Motion Prediction Quantitative Results

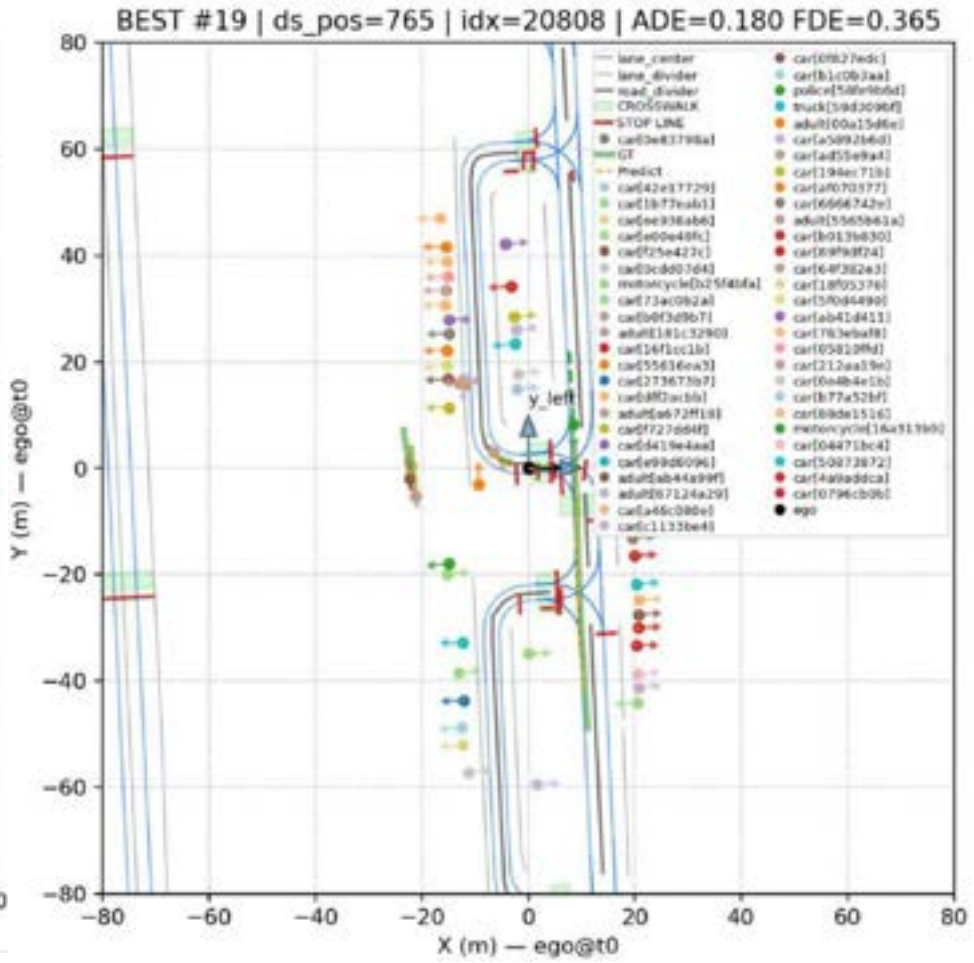
- Given past 2 seconds, the goal is to predict all agents' future trajectories in 5 seconds.
- We evaluate with standard minADE and minFDE metrices, averaged over all agents.
- Evaluated on nuScenes Dataset[1] and the leaderboard.

Method	minADE↓	minFDE↓
DGCN_ST_LANE [4]	**1.092**	3.624
CASPFormer [5]	1.148	6.702
THOMAS [2]	1.325	6.712
Ours	1.121	**2.346**

Results: Motion Prediction Qualitative Results (1/2)

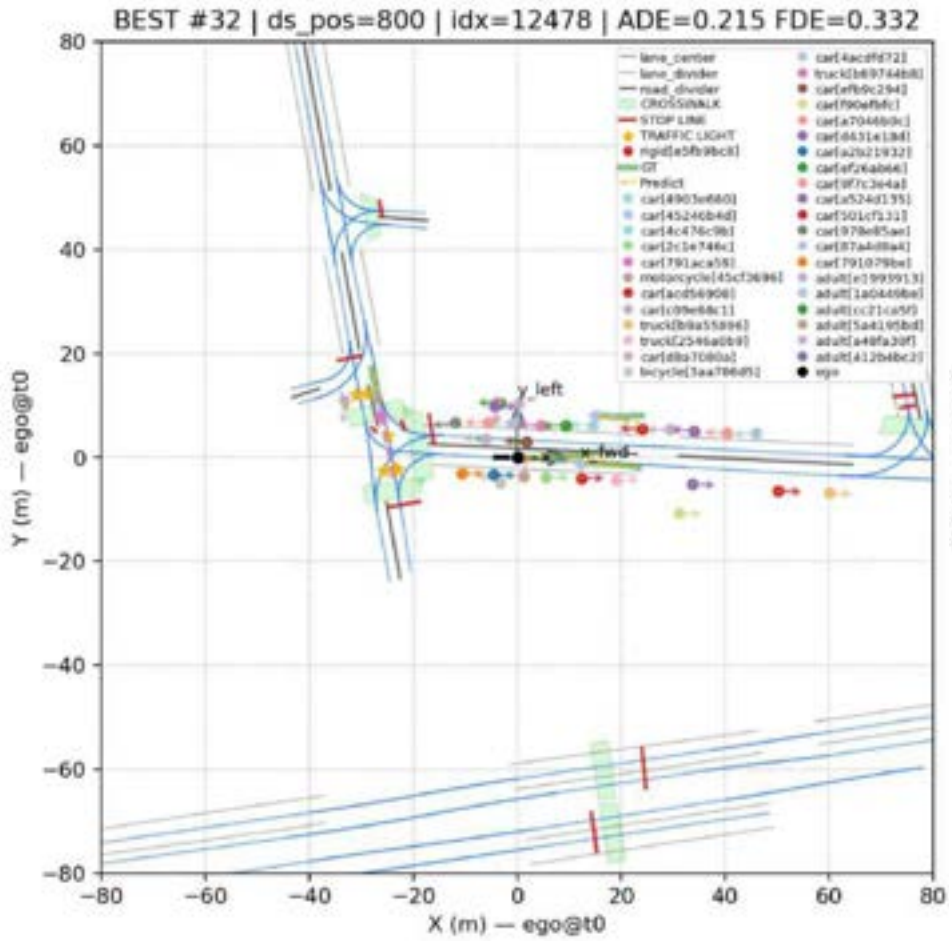
- Visualization of the multi-agent motion prediction results.

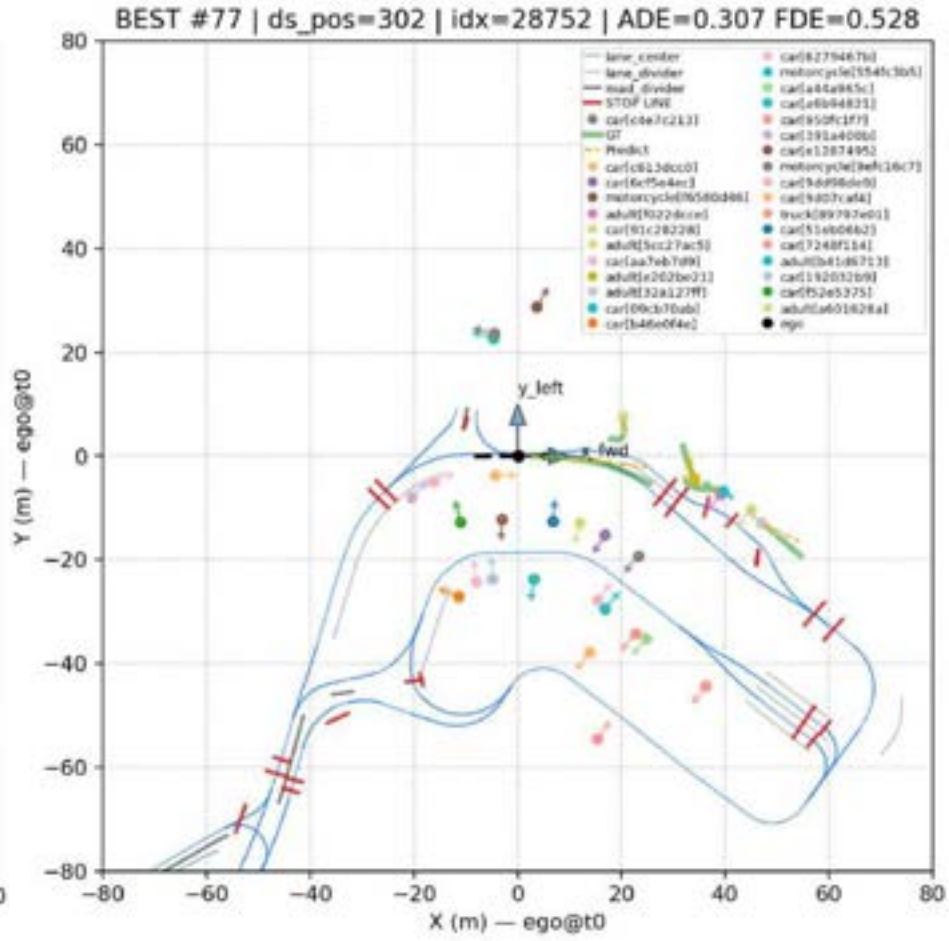




Results: Motion Prediction Qualitative Results (2/2)

- Visualization of the multi-agent motion prediction results.





Outline

- Introduction & Motivation
- Related Work
- Proposed Method
- Results
- Conclusion & Summary

Conclusion & Summary

- Our current results rely on distillation from Qwen2.5-VL-7B-Instruct, which may limit accuracy. Distilling from Qwen2.5-VL-32B-Instruct could further improve performance.
- nuScenes is relatively simple: in many scenes, agents mostly drive straight with limited turning or complex interactions. As a result, our model may not be evaluated at its full capacity.
- Because JEPA self-supervised learning and traffic-rule distillation already enable the model to capture and predict scene dynamics, it is worth testing whether agent-history and relationship modalities can be removed without degrading performance.
- Since most motion prediction methods still depend on agent history, predicting all agents' future trajectories using only ego-camera inputs would be a valuable and impactful direction.

References (1/2)

- [1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan,Yu Pan, Giancarlo Baldan, and Oscar Beijbom.nuscenes: A multimodal dataset for autonomous driving.arXiv preprint arXiv:1903.11027, 2019.
- [2] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde.THOMAS: Trajectory heatmap output with learned multi-agent sampling.In International Conference on Learning Representations (ICLR), 2022.
- [3] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han.Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation, 2024.
- [4] Kailu Wu, Xing Liu, Feiyu Bian, Yizhai Zhang, and Panfeng Huang.An integrating comprehensive trajectory prediction with risk potential field method for autonomousdriving, 2024.
- [5] Harsh Yadav, Maximilian Schaefer, Kun Zhao, and Tobias Meisen.CASPFormer: Trajectory prediction from BEV images with deformable attention.In Pattern Recognition: 27th International Conference, ICPR 2024, Kolkata, India, Proceedings, Part XVII,volume 15317 of Lecture Notes in Computer Science. Springer, 2025.
- [6] Xiaoji Zheng, Lixiu Wu, Zhijie Yan, Yuanrong Tang, Hao Zhao, Chen Zhong, Bokui Chen, and JiangtaoGong.Large language models powered context-aware motion prediction, 2024.
- [7] Gemma Roig, Jonathan L. McCormack, and others. AudioSet: An ontology and large-scale dataset for audio event recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2017. https://doi.org/10.1109/ICASSP.2017.7952561

References (2/2)

- [7] Xiaoji Zheng, Liwu Xu, Zhijie Yan, Yuanrong Tang, Hao Zhao, Chen Zhong, Bokui Chen, and Jiangtao Gong. Large language models powered context-aware motion prediction, 2024.
- [8] Zhai, Y., Zhou, H., Xu, X., Ma, Z., Liu, Z., & Qiao, Y. (2021). Self-Supervised Audio-Visual Learning Using Cross-Modal Contrastive Learning and Sound Source Localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 1559–1568.
- [9] Bardes, A., Garrido, Q., Ponce, J., Chen, X., Rabbat, M., LeCun, Y., ... & Ballas, N. (2024). Revisiting feature prediction for learning visual representations from video. arXiv preprint arXiv:2404.08471.
- [10] Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., & Song, S. (2023). Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. Int. J. Robotics Res., 44, 1684-1704.
- [11] Choi, Dooseop & Min, KyoungWook. (2022). Hierarchical Latent Structure for Multi-Modal Vehicle Trajectory Forecasting. 10.48550/arXiv.2207.04624.
- [12] Liao, B., Chen, S., Yin, H., Jiang, B., Wang, C., Yan, S., Zhang, X., Li, X., Zhang, Y., Zhang, Q., & Wang, X. (2024). DiffusionDrive: Truncated Diffusion Model for End-to-End Autonomous Driving. 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12037-12047.

Appendix