# MoLan: Motion–Language Understanding and Generation for Traffic Simulation

**Anonymous authors**
Paper under double-blind review

## Abstract

Realistic, reactive, and controllable multi-agent traffic simulation is essential for the scalable development and evaluation of autonomous driving systems. Natural language provides an intuitive interface both for controlling how traffic scenes unfold and for querying the behaviors and interactions represented in them. Existing approaches, however, typically specialize in either language-conditioned motion generation or motion-grounded scene understanding. We introduce MoLan, a Motion–Language model that connects pretrained traffic and language towers through lightweight bidirectional bridge blocks. The resulting architecture supports native motion simulation and text generation, language-controlled closed-loop multi-agent rollouts, and motion-grounded traffic-scene understanding within a single model. MoLan outperforms task-specific baselines on both language-controlled simulation and motion-grounded question answering, while joint training across both directions further improves closed-loop realism, indicating that motion-understanding supervision can strengthen language-controlled generation.

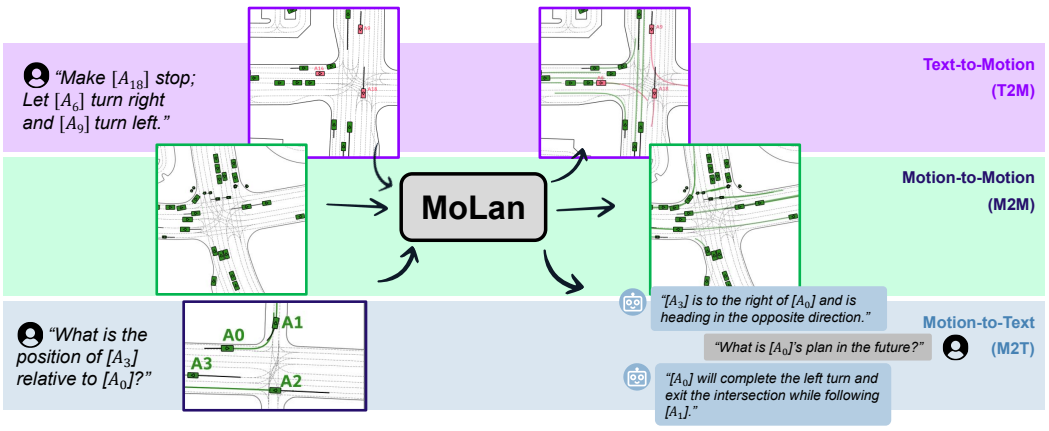



Figure 1: We introduce MoLan, a motion–language model supporting unified motion understanding and generation without compromising the proficiency of the underlying language and traffic models. By connecting frozen language and motion models through lightweight bridges, MoLan supports four tasks: Motion-to-Motion (M2M), Text-to-Motion (T2M), Motion-to-Text (M2T), and Text-to-Text (T2T), enabling language-controlled scene generation and motion-grounded understanding.

## 1 Introduction

Developing safe and reliable autonomous vehicles requires extensive testing in diverse, rare, and dangerous scenarios that are difficult to encounter or reproduce on real roads. Closed-loop traffic simulation provides a scalable and risk-free environment for developing and evaluating autonomous driving systems before real-world deployment (Montali et al., 2023, Gulino et al., 2023, Dosovitskiy et al., 2017, Caesar et al., 2021, Hallgarten et al., 2024). For a simulator to be truly useful, it must generate *realistic* multi-agent behavior while being *reactive*—preserving the reactions that arise as agents interact with one another and with the tested system. Realism and reactivity alone, however, are not sufficient. Naturally occurring driving data is dominated by routine behavior and cannot



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

expose every interaction or failure mode that a system may encounter. The simulator must therefore also be *controllable*, allowing users to deliberately construct targeted and counterfactual scenarios that probe specific system responses. Natural language is a natural medium for this control: it provides an intuitive, high-level interface through which users can specify complex spatial configurations and agent behaviors without requiring specialized engineering expertise or low-level parameter tuning. For example, a user could request, *"make* $[\mathtt{A}_1]$ *overtake* $[\mathtt{A}_2]$ *from the left lane,"* specifying both the behavior of each agent and the interaction between them within a single instruction.

Beyond control, language provides a common medium for describing and reasoning about traffic scenes. Recent language-conditioned simulators use textual instructions to generate targeted agent behaviors (Tan et al., 2024, Chang et al., 2025, Chen et al., 2026), while driving-oriented language models interpret traffic scenes and explain agent behavior, intentions, and interactions (Chen et al., 2024, Li et al., 2025, Ghosh et al., 2026). As summarized in Table 1, existing methods either support language-conditioned motion generation (*text-to-motion*, T2M) or provide motion-grounded scene understanding (*motion-to-text*, M2T), but not both within a single framework.

Table 1: **Comparison of recent state-of-the-art methods on diverse motion-relevant tasks.** "Reactive" denotes closed-loop multi-agent simulation; $*$ means the modality is partially supported (iMotion controls a selected focal-agent).

| | Understanding | Generation | | |
|---|---|---|---|---|
| | M2T | M2M | T2M | Reactive |
| CAT-$k$ (Zhang et al., 2025) | | ✓ | | ✓ |
| ProSim (Tan et al., 2024) | | ✓ | ✓ | ✓ |
| ECoSim (Chen et al., 2026) | | ✓ | ✓ | ✓ |
| LangTraj (Chang et al., 2025) | | ✓ | ✓ | ✓ |
| iMotion (Felemban et al., 2025) | $*$ | $*$ | $*$ | |
| Motion-LLaVA (Li et al., 2025) | ✓ | | | |
| MoLan (Ours) | ✓ | ✓ | ✓ | ✓ |

Training a unified motion–language model without leveraging a pretrained traffic simulator requires learning realistic closed-loop dynamics in addition to cross-modal grounding. Existing language-conditioned simulators such as ProSim and LangTraj first learn a task-specific traffic-generation model and subsequently introduce language conditioning through additional training stages (Tan et al., 2024, Chang et al., 2025). This staged training couples cross-modal learning with the difficult problem of learning a high-quality simulator, and does not extend to motion-grounded understanding. Since strong pretrained specialists already exist for traffic motion and language, we instead ask: *Can their complementary capabilities be connected—enabling a single model to both control traffic scenes through natural-language instructions and explain their multi-agent interactions through language—while preserving each specialist's native computation pathways?*

We introduce MoLan (Motion–Language) to answer this question. MoLan connects frozen traffic and language towers through lightweight dual-stream bridge blocks. Dual-stream cross-attention exchanges only cross-modal information, agent-grounded representations associate linguistic references with the corresponding traffic agents, and zero-initialized gates leave the pretrained computation intact at initialization. Native motion and language forward passes bypass the bridges, preserving the original pathways by construction. Beyond combining the two capabilities, this design lets us investigate whether they provide complementary supervision. Our experiments show that motion understanding improves language-controlled simulation. MoLan also outperforms task-specific baselines for both language-controlled simulation and motion-grounded question answering.

Our **contributions** are summarized as follows:

1. We introduce MoLan, to our knowledge the first traffic motion–language model that can both generate interactive multi-agent traffic from natural-language instructions and answer questions about traffic scenes. MoLan provides a single natural-language interface for creating targeted test scenarios and understanding the resulting agent interactions, while preserving the native generation capabilities of the underlying models—and, because the towers stay frozen and only the bridges are trained, MoLan generalizes as a plug-and-play interface across different motion and language model families and architectures.
2. We show that a unified model can outperform task-specific baselines on both language-controlled traffic simulation and motion-grounded question answering , demonstrating that generation and understanding benefit from joint training.
3. Finally, we demonstrate that training MoLan on M2T and T2M tasks shows motion-understanding supervision transfers to language-controlled generation, improving overall closed-loop simulation quality and, most notably, the realism of interactions among agents.

## 2 Related Work

**Data-driven multi-agent traffic simulation.** Learning-based traffic simulators model the joint behavior of multiple agents from structured scene representations such as agent histories, road maps, and traffic signals. TrafficGen (Feng et al., 2023) jointly models scene layouts and agent dynamics, while MotionLM (Seff et al., 2023), Trajeglish (Philion et al., 2024), and SMART (Wu et al., 2024) formulate traffic motion as autoregressive token prediction. Building on this formulation, TrajTok (Zhang et al., 2026) studies the effect of trajectory tokenization; CAT-$k$ (Zhang et al., 2025) and SMART-R1 (Pei et al., 2026) improve closed-loop rollout quality through supervised and reinforcement fine-tuning, respectively; and InfGen (Yang et al., 2025b) and RosettaSim (Xiao et al., 2026) extend autoregressive simulation to long horizons in which agents may enter and leave the scene. LLM2AD (Wang et al., 2025) investigates which architectural components of language models transfer to tokenized motion generation, whereas UniMotion (Song et al., 2025) unifies simulation, prediction, and planning within a motion-only framework. Although several of these methods adopt language-model architectures or next-token objectives, their tokens represent traffic states rather than natural-language instructions or questions. They therefore provide strong motion-to-motion models but no natural-language interface for controlling and understanding traffic scenes.

**Language-controllable traffic simulation.** A separate line of work uses language to specify the traffic behaviors to be generated. CTG++ and other LLM-guided simulators (Zhong et al., 2023, Liu et al., 2025) translate language requests into objectives or constraints that guide trajectory generation. LCTGen (Tan et al., 2023) generates traffic scenes from textual descriptions, while InteractTraj (Xia et al., 2024) generates trajectories for specified agent interactions. Trajectory-LLM (Yang et al., 2025a) instead uses language models to construct trajectory-prediction data from textual descriptions. More closely related to our setting, ProSim (Tan et al., 2024), LangTraj (Chang et al., 2025), and ECoSim (Chen et al., 2026) directly condition multi-agent motion generators on language or multimodal prompts, enabling users to control the behavior of selected agents during simulation. These approaches establish language as an effective interface for motion generation, but do not aim to provide general motion-grounded question answering about the road context, agent behaviors, and possible interactions in a traffic scene.

**Language understanding of structured traffic motion.** Language has also been used to describe and reason about driving scenes. Early datasets and models study behavior explanation and risk description (Kim et al., 2018, Malla et al., 2023), while nuScenes-QA, DriveLM, and Reason2Drive formulate driving-scene understanding through visual question answering or structured reasoning (Qian et al., 2024, Sima et al., 2024, Nie et al., 2024). Closer to our structured-motion setting is Driving with LLMs (Chen et al., 2024), which grounds a language model in object-level numerical representations, while WOMD-Reasoning and Motion-LLaVA (Li et al., 2025) support question answering over trajectories, road geometry, and traffic signals. Most closely, RAD-LAD (Ghosh et al., 2026) jointly learns scene question answering and ego-trajectory planning, showing that language supervision can improve closed-loop planning. These works establish the value of language for interpreting structured traffic scenes, but none combines motion-grounded question answering with reactive, language-controlled generation of the complete multi-agent scene.

**Unified motion–language modeling.** Unified motion–language modeling has been explored primarily for human motion. MotionGPT and MotionGPT-2 (Jiang et al., 2023, Wang et al., 2026b) represent human motion within language-oriented generative frameworks, while MoLAM and MotionGPT3 (Park et al., 2025, Zhu et al., 2026) extend unified modeling to motion reasoning and human interactions. LLaMo (Li et al., 2026) retains modality-specific computation streams and operates on continuous motion representations rather than converting motion entirely into text tokens. In driving, iMotion-LLM (Felemban et al., 2025) moves toward a shared motion–language interface, but focuses on the ego agent's trajectories rather than reactive closed-loop simulation of the complete multi-agent scene. Transferring this idea to traffic, however, introduces additional requirements: (i.) the model must preserve structured map context, (ii.) maintain the identities of a variable set of agents, and (iii.) capture their causal interactions throughout a closed-loop rollout. MoLan retains separate traffic-motion and language pathways and learns only their bidirectional interaction, preserving the native motion and text generation pathways of both towers.

## 3 Problem Formulation

We consider a traffic scene containing $N$ interacting agents and contextual information $\mathcal{C}$, including the road map and traffic-signal states. Our goal is to support both motion and language generation from this structured scene through four tasks: *motion-to-motion* (M2M), *text-to-motion* (T2M), *motion-to-text* (M2T), and *text-to-text* (T2T).

**Multi-agent traffic simulation.** Let $\mathbf{s}_t = \{s_t^i\}_{i=1}^N$ denote the joint state of all agents $\mathbf{a} = \{a_i\}_{i=1}^N$ at timestep $t$. We define $\mathbf{s}_{0:t_0}$ as the state history available to the simulator up to timestep $t_0$. Multi-agent traffic simulation models the future evolution of the joint agent states from the observed history and scene context. The task is to predict future agent states over a fixed horizon $T$, formulated as estimating the conditional distribution $p_{\text{motion}}(\mathbf{s}_{t_0+1:t_0+T} \mid \mathbf{s}_{0:t_0}, \mathcal{C})$, which factorizes over time into an autoregressive prediction problem:

$$p_{\text{motion}}(\mathbf{s}_{t_0+1:t_0+T} \mid \mathbf{s}_{0:t_0}, \mathcal{C}) = \prod_{t=t_0+1}^{t_0+T} p_{\text{motion}}(\mathbf{s}_t \mid \mathbf{s}_{<t}, \mathcal{C}). \tag{1}$$

We refer to this native traffic-simulation pathway as the M2M pathway.

**Language modeling.** Given a prompt $\mathbf{x}$ and an output sequence $\mathbf{y} = (y_1, \ldots, y_L)$, an autoregressive language model defines

$$p_{\text{text}}(\mathbf{y} \mid \mathbf{x}) = \prod_{j=1}^{L} p_{\text{text}}(y_j \mid y_{<j}, \mathbf{x}). \tag{2}$$

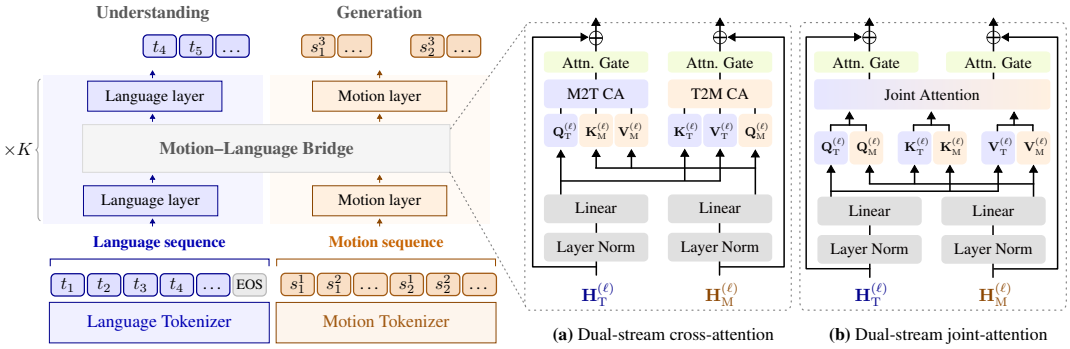
This corresponds to the language model's native T2T pathway.

**Unified motion–language modeling.** To enable the model to generate language-grounded simulations and answer questions about driving scenes, we adopt a dual-tower paradigm that leverages pretrained motion and language models. The pretrained motion and language models natively support M2M and T2T, respectively. We extend the motion policy to $p_{\text{motion}}(\mathbf{s}_{t_0+1:t_0+T} \mid \mathbf{s}_{0:t_0}, \mathcal{C}, \mathbf{x})$ for T2M, where $\mathbf{x}$ is a language instruction, and extend the language model to $p_{\text{text}}(\mathbf{y} \mid \mathbf{x}, \mathbf{s}_{0:t_0}, \mathcal{C})$ for M2T, where $\mathbf{x}$ is a question and $\mathbf{s}_{0:t_0}$ is a supplied traffic scene. The towers therefore retain separate computation streams and communicate through dedicated trainable motion–language bridges.

## 4 Motion–Language Understanding and Generation

MoLan combines pretrained traffic–motion and language models, both of which remain frozen to preserve their original capabilities while enabling cross-modal learning. Freezing the towers keeps each backbone's full proficiency intact throughout training, allows the interface to be trained efficiently with only a small fraction of parameters, and makes the architecture plug-and-play: either pretrained expert can be swapped for a stronger one without retraining from scratch. Dual-stream cross-modal blocks connect selected intermediate layers during training and inference, extending the native motion and language pathways to support language-grounded motion generation and motion-grounded language understanding. As shown in Figure 2, we insert lightweight bridge blocks between selected layers of the motion and language towers (*e.g.*, at early, middle, or late depths), allowing the two modalities to interact without requiring them to have the same number of layers. Each block allows the two pathways to perform cross-modal information exchange and returns updated features to both modalities.

**Input representation.** The motion stream represents each agent with a sequence of motion tokens over time. At a given layer, these tokens are encoded as agent-wise motion features $\mathbf{H}_{\text{M}} \in \mathbb{R}^{N \times T \times D_{\text{M}}}$. Before reaching a motion–language bridge, these features aggregate road and neighboring-agent context through SMART's native map–agent cross-attention and agent-interaction layers. Each scene may be paired with user-provided text prompts that contain multiple sentences or questions describing or asking about agent behaviors, interactions, and scenario properties. Following ProSim (Tan et al., 2024), we use this format $[\texttt{A}_i]$ when mentioning the $i$-th agent. For example, to instruct agent $a_3$ to slow down, the user would say *"let* $[\texttt{A}_3]$ *slow down,"* or if the user wants to know why $a_5$ yielded to $a_7$, the user would say *"why did* $[\texttt{A}_5]$ *yield to* $[\texttt{A}_7]$*?"* To make sure the language model parses them correctly, we add the agent-reference tokens $[\texttt{A}_i]$ as new tokens to the text tokenizer. Natural-language prompts are then tokenized as $\mathbf{x} := \{x_i\}_{i=1}^L$, and the language backbone maps them to hidden states $\mathbf{H}_{\text{T}} \in \mathbb{R}^{L \times D_{\text{T}}}$ at a given layer.




**(a)** Dual-stream cross-attention **(b)** Dual-stream joint-attention

Figure 2: **Overview of the Motion–Language Bridge Architecture of MoLan. Left:** The language (T) and traffic-motion (M) towers retain their respective tokenizers and computation streams. Motion–language bridge blocks connect selected intermediate layers, allowing motion features to condition language generation for scene understanding and language features to condition motion generation for controllable simulation. The two backbones remain frozen, and only the added bridge modules are trained. **Right:** We consider two bidirectional fusion mechanisms. **(a)** In dual-stream cross-attention, text queries attend to motion keys and values for M2T, while motion queries attend to text keys and values for T2M. **(b)** In dual-stream joint attention, the projected queries, keys, and values from both modalities are concatenated and processed through a joint-attention operation.

### 4.1 Dual-Stream Fusion

The motion and language towers are independently pretrained on different modalities and may have different hidden dimensions and numbers of layers. As shown in Figure 2, we insert $K$ lightweight blocks between the motion and language layers and connect their intermediate hidden states. Each block enables information exchange between the motion and language streams while keeping their original towers separate.

Specifically, let $\mathbf{H}_{\mathrm{M}}^{(\ell)}$ and $\mathbf{H}_{\mathrm{T}}^{(\ell)}$ denote the motion and text hidden states entering the $\ell$-th bridge block. We normalize and project them into query, key, and value representations:

$$\mathbf{Q}_{\bullet}^{(\ell)} = \mathrm{Linear}_{\mathrm{Q}_\bullet}(\mathrm{LN}(\mathbf{H}_{\bullet}^{(\ell)})), \quad \mathbf{K}_{\bullet}^{(\ell)} = \mathrm{Linear}_{\mathrm{K}_\bullet}(\mathrm{LN}(\mathbf{H}_{\bullet}^{(\ell)})), \quad \mathbf{V}_{\bullet}^{(\ell)} = \mathrm{Linear}_{\mathrm{V}_\bullet}(\mathrm{LN}(\mathbf{H}_{\bullet}^{(\ell)})), \tag{3}$$

where $\bullet \in \{\mathrm{M}, \mathrm{T}\}$ and $\mathrm{LN}(\cdot)$ denotes layer normalization (Ba et al., 2016). Each modality's $N \times T$ motion grid (respectively, $L$-token text sequence) is treated as a sequence of $NT$ motion tokens (respectively, $L$ text tokens), and the projections map both streams to a shared attention dimension $D_A$, yielding $\mathbf{Q}_{\bullet}^{(\ell)}, \mathbf{K}_{\bullet}^{(\ell)}, \mathbf{V}_{\bullet}^{(\ell)} \in \mathbb{R}^{S_\bullet \times D_A}$ with $S_{\mathrm{M}} = NT$ and $S_{\mathrm{T}} = L$. Attention outputs are then projected back to the hidden dimension of the corresponding tower before being added to its residual stream, so that motion and language features of their original shapes ($N \times T \times D_{\mathrm{M}}$ and $L \times D_{\mathrm{T}}$) are returned to the towers.

**Dual-stream cross-attention.** As shown in Figure 2a, dual-stream cross-attention exchanges information between the motion and language tokens through two symmetric streams. In the M2T stream, the text tokens provide the queries, while the motion tokens provide the keys and values:

$$\mathbf{O}_{\mathrm{T}}^{(\ell)} = \mathrm{Attn}_{\mathrm{M2T}}^{(\ell)}\left(\mathbf{Q}_{\mathrm{T}}^{(\ell)}, \mathbf{K}_{\mathrm{M}}^{(\ell)}, \mathbf{V}_{\mathrm{M}}^{(\ell)}\right), \tag{4}$$

where $\mathrm{Attn}(\cdot)$ denotes the multi-head attention (Vaswani et al., 2017). Conversely, in the T2M stream, the motion tokens provide the queries, while the text tokens provide the keys and values, yielding $\mathbf{O}_{\mathrm{M}}^{(\ell)}$. Cross-attention is a natural choice for our dual-tower model because each pretrained tower already models within-modality dependencies. The bridge blocks therefore learn only to exchange information between the two pretrained representation spaces, providing a simpler optimization target than jointly modeling intra- and inter-modal interactions with limited trainable parameters. It also requires at most half the attention-map memory of joint attention, which is especially important in multi-agent traffic modeling, where the motion sequence grows with both the number of agents and modeled timesteps.

**Dual-stream joint-attention.** We also consider the dual-stream joint-attention design for comparison. As shown in Figure 2b, after the motion and text hidden states are projected, their query, key, and value representations are concatenated along the sequence dimension:

$$[\mathbf{O}_{\mathrm{T}}^{(\ell)}; \mathbf{O}_{\mathrm{M}}^{(\ell)}] = \mathrm{Attn}_{\mathrm{Joint}}^{(\ell)}\left([\mathbf{Q}_{\mathrm{T}}^{(\ell)}; \mathbf{Q}_{\mathrm{M}}^{(\ell)}], [\mathbf{K}_{\mathrm{T}}^{(\ell)}; \mathbf{K}_{\mathrm{M}}^{(\ell)}], [\mathbf{V}_{\mathrm{T}}^{(\ell)}; \mathbf{V}_{\mathrm{M}}^{(\ell)}]\right), \tag{5}$$

where $[\cdot\,;\cdot]$ denotes concatenation along the sequence dimension. The joint-attention output is then split into the corresponding text and motion outputs, $\mathbf{O}_{\mathrm{T}}^{(\ell)}$ and $\mathbf{O}_{\mathrm{M}}^{(\ell)}$. Joint attention therefore allows all motion and text tokens to interact within a single attention operation. Note that within-modality attention remains causal: each text token attends only to earlier text tokens in the prompt, and each motion token attends only to motion tokens from the current or earlier timesteps of the rollout. We compare the two fusion mechanisms in Appendix B.

### 4.2 Agent-Grounded Attention

Dual-stream fusion alone does not establish correspondence between an agent-reference token in language and the temporal motion features of the referenced agent. To establish this correspondence, we define a learnable embedding $\mathbf{E}(i)$ for each agent; we cap scenes at 128 agents, matching the maximum number of agents per scenario in WOMD. In the M2T stream, when the text sequence refers to agent $i$ using $[\mathrm{A}_i]$ at position $j$, we add $\mathbf{E}(i)$ to the corresponding text query $\mathbf{Q}_{\mathrm{T}}(j)$ and to the motion keys $\mathbf{K}_{\mathrm{M}}$ of every temporal feature belonging to $i$. Conversely, in the T2M stream, we add the same embedding to the motion queries $\mathbf{Q}_{\mathrm{M}}$ of every temporal feature belonging to $i$ and to the corresponding text key $\mathbf{K}_{\mathrm{T}}(j)$. For joint attention, we apply the same association to the corresponding text and motion representations before concatenation. Thus, language references and motion features belonging to the same agent share a common identity across both fusion directions.

### 4.3 Adaptive Gating

The bridge layers are learned from scratch, and directly adding their outputs to the pretrained hidden states may disturb the computation already learned by either tower, particularly at the beginning of training. To ensure stable learning and to guarantee that the conditioned model produces the same outputs as the original towers at initialization, we use a $\tanh$ gating mechanism (Alayrac et al., 2022) to control the contribution of each bridge output through a zero-initialized adaptive gate:

$$\widetilde{\mathbf{H}}_{\bullet}^{(\ell)} = \mathbf{H}_{\bullet}^{(\ell)} + \tanh(g_{\bullet}^{(\ell)}) \cdot \mathbf{O}_{\bullet}^{(\ell)}, \tag{6}$$

where $\bullet \in \{\mathrm{M}, \mathrm{T}\}$, $\mathbf{O}_{\bullet}^{(\ell)}$ denotes the corresponding bridge output, and $g_{\bullet}^{(\ell)}$ is a layer-specific learnable scalar initialized to 0. When both $\tanh(g_{\mathrm{M}}^{(\ell)})$ and $\tanh(g_{\mathrm{T}}^{(\ell)})$ are close to zero, the model primarily relies on the original pretrained representations of the language and motion models, respectively. During training, the gates learn how strongly each modality should incorporate information from the other modality. We track the evolution of the learned gates throughout training, which progressively modulate cross modal information exchange; further analysis is provided in Figure 6 in Appendix B.

## 5 Experiments and Results

**Datasets and tasks.** We train and evaluate traffic-motion on the Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021). M2M and T2M follow the Waymo Open Sim Agents Challenge (WOSAC) protocol (Montali et al., 2023): the model observes one second of scene history and generates an eight-second closed-loop rollout. For T2M, we use ProSim-Instruct-520k dataset (Tan et al., 2024), which provides language annotations for all scenarios in the official WOMD training and validation sets, comprising over 520k prompt–scenario pairs and approximately 10M text descriptions. For M2T, WOMD-Reasoning (Li et al., 2025) provides 2.94M question–answer pairs covering 63k WOMD scenarios. We extend its automatic labeling pipeline to the remaining scenarios in the WOMD training and validation sets. Specifically, we use its rule-based program to convert the map and agent motion into textual scene descriptions, and then apply the original prompt templates to Qwen2.5-72B-Instruct (Qwen et al., 2025) to generate questions and answers about the map, surrounding agents, interactions, and intentions.

**Baselines.** For closed-loop motion we compare with SMART (Wu et al., 2024), CAT-$k$ (Zhang et al., 2025), MotionLM (Seff et al., 2023), and TrajTok (Zhang et al., 2026). For T2M we compare with ProSim (Tan et al., 2024), LangTraj (Chang et al., 2025), and ECoSim (Chen et al., 2026)[1]. For M2T

[1]We note that ECoSim reports results obtained using only 1% of the available training data; for a fair comparison, we train ECoSim using 100% of the available training data.

Table 2: **Language-controlled closed-loop simulation on WOMD.** The best and second-best results are highlighted in bold and underline, respectively. ECoSim and MoLan both use CAT-$k$ as the motion model.

| | RMM ↑ | Kinematic metrics ↑ | Interaction metrics ↑ | Map-based metrics ↑ | min ADE ↓ |
|---|---|---|---|---|---|
| *Unconditional: Motion-to-Motion (M2M)* | | | | | |
| CAT-$k$ (Zhang et al., 2025) | 0.773 | 0.467 | 0.806 | 0.907 | 1.501 |
| ProSim (Tan et al., 2024) | 0.703 | 0.481 | 0.702 | 0.831 | 2.641 |
| LangTraj (Chang et al., 2025) | 0.716 | 0.420 | 0.790 | 0.790 | 2.890 |
| ECoSim (Chen et al., 2026) | 0.773 | 0.467 | 0.806 | 0.907 | 1.501 |
| MoLan (Ours) | 0.773 | 0.467 | 0.806 | 0.907 | 1.501 |
| *Language controlled: Text-to-Motion (T2M)* | | | | | |
| ProSim (Tan et al., 2024) | 0.710 (+0.949%) | 0.432 | 0.746 | 0.822 | 2.320 (+13.83%) |
| LangTraj (Chang et al., 2025) | 0.719 (+0.419%) | 0.430 | 0.800 | 0.780 | 2.520 (−14.68%) |
| ECoSim (Chen et al., 2026) | <u>0.776</u> (+0.362%) | <u>0.481</u> | <u>0.803</u> | <u>0.910</u> | <u>1.255</u> (+16.39%) |
| MoLan (Ours) | **0.786** (**+1.631%**) | **0.490** | **0.819** | **0.912** | **1.183** (**+21.22%**) |

generation, we compare against Motion-LLaVA (Li et al., 2025). Because no official implementation or pretrained checkpoint was publicly available, we reimplemented Motion-LLaVA following its published description and trained it on the same M2T dataset used for MoLan. Inspired by Wang et al. (2026a), we also construct a vision-language-model (VLM) baseline that represents each traffic scene as a bird's-eye-view image, adapting their prompting scheme to WOMD. Finally, we report the performance of single-task variants of MoLan trained exclusively on either T2M or M2T.

**Implementation details.** Unless mentioned otherwise, we use SMART (Wu et al., 2024) as the motion backbone, trained in closed-loop using CAT-$k$ (Zhang et al., 2025). For the language backbone, we use LLaMA3.2-1B (Grattafiori et al., 2024). MoLan uses six bridge blocks uniformly placed between the layers of the two towers. To demonstrate that MoLan is model agnostic and assess its generality across different motion and language model families and architectures, we evaluate several alternative backbones and report the results in Table 5. **Evaluation metrics.** For motion generation with and without language conditioning, we report the WOSAC realism meta-metric (RMM) and its kinematic, interaction, and map-based components. We also quantify the relative improvement (%) in RMM and min ADE of the model with and without language-grounding. For question-answering, we report BLEU-$k$ ($k = 1, 4$), ROUGE-L, CIDEr, and BERTScore (Zhang et al., 2020). For factual questions, which ask about directly observable properties of the traffic scene, such as an agent's position, speed, or motion state, we follow WOMD-Reasoning (Li et al., 2025) and report the accuracy of numerical answers within a 1.0m tolerance, and the root median squared error between predicted and ground truth values.

**Training protocol.** We train MoLan in a single stage using the motion and language next-token objectives defined in Section 3. For T2M, we follow CAT-$k$ closed-loop training (Zhang et al., 2025) and apply the motion next-token loss to all modeled agents. For M2T, we apply the language cross-entropy loss only to the answer tokens. We observe that motion understanding (M2T) converges much faster and can dominate the optimization of the motion generation objective. To mitigate this imbalance, we therefore sample T2M and M2T batches in an 80:20 ratio. Additional details for training hyperparameters and evaluation metrics are provided in Appendix A.

### 5.1 Main Results

**Language-controlled closed-loop traffic simulation.** We first evaluate whether MoLan can introduce language control without compromising the quality of the underlying closed loop motion model. As shown in Table 2, **MoLan preserves the performance of the underlying CAT-$k$ model in the absence of language conditioning**, achieving the same RMM of 0.773 and min ADE of 1.501, whereas ProSim and LangTraj both degrade relative to CAT-$k$. With language conditioning, **MoLan achieves the best performance among language controlled simulators**, with an RMM of 0.786 and a min ADE of 1.183. Compared with the unconditional MoLan, this corresponds to a relative improvement of 1.63% in RMM and 21.22% in min ADE, while improving over ECoSim by 1.29% in RMM and 5.74% in min ADE. These results show that MoLan preserves the capabilities of the underlying motion model while introducing language control that further improves closed loop motion quality. The qualitative examples in Figure 3 further illustrate the MoLan's ability to follow instructions across diverse WOMD scenarios with and with language conditioning.

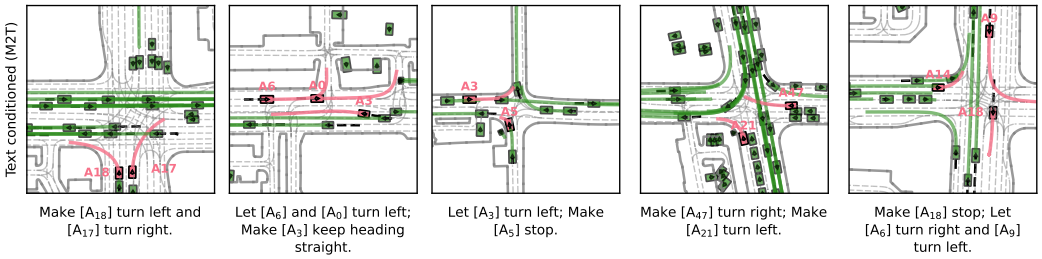



Figure 3: **Qualitative comparison of language-controlled rollouts across WOMD scenarios.**

Table 3: **Motion-grounded language generation on WOMD-Reasoning.** We use Qwen2.5-VL-3B-Instruct as the VLM. The best and second-best results are highlighted in bold and underline, respectively.

| | All questions | | | | | Factual questions | |
|---|---|---|---|---|---|---|---|
| | BLEU-1 ↑ | BLEU-4 ↑ | ROUGE ↑ | CIDEr ↑ | BERTScore ↑ | Accuracy ↑ | Median Error (m) ↓ |
| *Motion understanding: Motion-to-Text (M2T)* | | | | | | | |
| Non-finetuned VLM w. BEV input | 0.234 | 0.168 | 0.498 | 2.782 | 0.234 | 0.037 | 20.223 |
| Finetuned VLM w. BEV input | 0.618 | 0.276 | 0.553 | 3.821 | 0.287 | 0.134 | 10.641 |
| Motion-LLaVA (Li et al., 2025) | <u>0.670</u> | <u>0.339</u> | <u>0.745</u> | <u>5.158</u> | <u>0.521</u> | <u>0.554</u> | <u>2.238</u> |
| MoLan (Ours) | **0.787** | **0.491** | **0.826** | **5.742** | **0.668** | **0.681** | **0.513** |

**Motion-grounded traffic-scene understanding.** We evaluate MoLan on WOMD-Reasoning and compare it with Motion-LLaVA (Li et al., 2025) and Qwen2.5-VL-3B-Instruct. For the latter, we follow Wang et al. (2026a) and use a single-frame BEV map as input for the VLM to understand the current autonomous driving scene. We evaluate both the off-the-shelf model and a variant fine-tuned on the BEV-based question-answering data. As shown in Table 3, the BEV-based models perform substantially worse than models that directly process motion trajectories. This is consistent with the findings of Li et al. (2025) and Wang et al. (2026a), and highlights the difficulty of recovering precise agent states and interactions from rendered BEV images. **MoLan achieves the strongest overall performance, outperforming Motion-LLaVA across all metrics, including factual questions**, where it achieves 68.1% accuracy and a median error of 0.513 m compared with 55.4% and 2.238m, respectively. These results show that MoLan effectively grounds language generation in its motion representations, as further illustrated by the qualitative examples in Figure 4, where MoLan answer questions about scenes involving multiple agents.

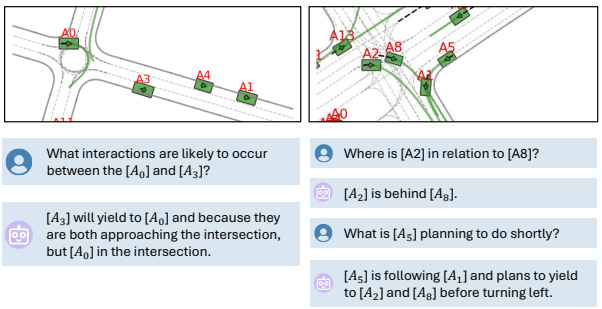



Figure 4: Qualitative comparison of motion-grounded question answering of MoLan across WOMD scenarios.

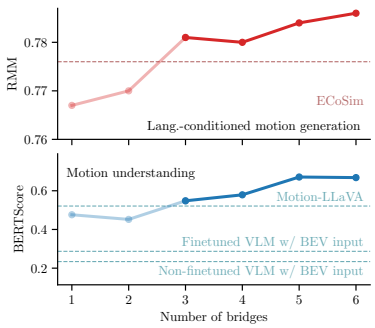



Figure 5: Effect of number of motion–language bridge blocks.

**Does Motion Understanding Improve Language-Controlled Generation?** To isolate whether motion understanding supervision benefits language controlled simulation, we compare the T2M only and jointly trained variants in Table 4; both variants are trained for the same number of steps on the same scenarios, differing only in whether motion

Table 4: **Effect of motion understanding supervision on language controlled generation.**

| | T2M | M2T | RMM ↑ | BERTScore ↑ |
|---|---|---|---|---|
| M2T-only | | ✓ | 0.712 | 0.623 |
| T2M-only | ✓ | | 0.777 | 0.605 |
| Joint training | ✓ | ✓ | **0.786** | **0.668** |

understanding supervision is included. **Joint training improves RMM from 0.777 to 0.786**, with the gain in interaction realism being particularly notable given that the motion understanding objective explicitly captures agent behaviors, intentions, and interactions. Since both pretrained towers remain frozen, these gains indicate that motion grounded language supervision provides a useful learning signal for the cross modal interface and language controlled multi agent generation.

### 5.2 Ablation Studies

**Generalization across different language and motion models.** To assess the generalization capabilities of MoLan, we replace each frozen tower with language models from different families and scales and with alternative next-token-prediction-based (NTP) motion models, while keeping the bridge architecture, objectives, and evaluation protocol fixed. As shown in Table 5, **regardless of the language or motion backbone, language grounding consistently improves motion generation over the corresponding unconditional model**, improving RMM by 1.49% to 1.65% and reducing min ADE by 12.39% to 21.22%. This robustness across language model families, trajectory tokenizations, and motion architectures extends our previous finding that motion understanding provides a beneficial learning signal for language-controlled motion generation. Among language backbones, performance remains comparable, with RMMs of 0.781 to 0.786 and BERTScores of 0.668 to 0.691, while TrajTok achieves the highest language conditioned RMM (0.788) with a BERTScore comparable to CAT-$k$ (0.666 vs. 0.668). Overall, the consistent gains across different backbone choices suggest that the benefit of language grounding is not tied to a particular architecture, supporting **MoLan as a plug-and-play model for pretrained motion and language models.**

Table 5: **Generalization across language and motion backbones.** MoLan consistently improves language-conditioned motion generation over the corresponding unconditional models across different language model families, scales, trajectory tokenizations, and motion architectures, while maintaining comparable motion-grounded language understanding.

| | Motion backbone | Language backbone | RMM ↑ | BERTScore ↑ | min ADE ↑ |
|---|---|---|---|---|---|
| | *Unconditional: Motion-to-Motion (M2M)* | | | | |
| | CAT-$k$ (Zhang et al., 2025) | – | 0.773 | – | 1.501 |
| | MotionLM (Seff et al., 2023) | – | 0.756 | – | 1.702 |
| | TrajTok (Zhang et al., 2026) | – | 0.776 | – | 1.439 |
| MoLan (Ours) | CAT-$k$ (Zhang et al., 2025) | Qwen2.5-0.5B (Qwen et al., 2025) | 0.781 (+0.972%) | 0.687 | **1.171** (+22.000%) |
| | | LLaMA3.2-1B (Grattafiori et al., 2024) | **0.786** (+1.631%) | 0.668 | 1.183 (+21.220%) |
| | | LLaMA3.2-3B (Grattafiori et al., 2024) | 0.783 (+1.560%) | **0.691** | 1.181 (+21.340%) |
| | CAT-$k$ (Zhang et al., 2025) | LLaMA3.2-1B (Grattafiori et al., 2024) | 0.786 (+1.631%) | **0.668** | 1.183 (+21.220%) |
| | MotionLM (Seff et al., 2023) | | 0.769 (+1.653%) | 0.659 | 1.491 (+12.385%) |
| | TrajTok (Zhang et al., 2026) | | **0.788** (+1.491%) | 0.666 | **1.169** (+18.775%) |

**Optimal number of motion–language bridge blocks.** We study the effect of the number bridge blocks. As shown in Figure 5, performance improves consistently as the number of bridges increases, peaking at six—the maximum supported by SMART's six-layer architecture. Notably, with as few as three bridge blocks, MoLan already outperforms all baselines on both language-controlled simulation and motion-grounded question answering, surpassing ECoSim in RMM and exceeding Motion-LLaVA and both BEV-based VLM baselines in BERTScore. This indicates that a small number of bridges is sufficient to establish an effective cross-modal interface between the frozen towers, with additional bridges yielding further, comparatively incremental gains. We therefore adopt six uniformly distributed bridge blocks for our main results, as this configuration maximizes performance.

## 6 Conclusion

We introduced MoLan, a modular architecture that connects frozen pretrained traffic and language towers through lightweight bidirectional motion–language bridges. This design enables MoLan to generate interactive traffic scenes from language instructions and answer questions about agent behaviors and interactions, while retaining the native performance of both towers. MoLan outperforms task-specific baselines in both closed-loop traffic realism and motion-grounded question answering. Joint training on both tasks further improves the RMM over T2M-only training, showing that motion-understanding supervision is beneficial for language-controlled simulation. The instruction distribution of current benchmarks supports relatively simple agent directives; extending MoLan to richer instruction and question distributions is a natural direction for future work.

### AI USE STATEMENT

In this work, we used generative AI tools for two purposes. First, we used large language models to augment the WOMD-Reasoning (Li et al., 2025) dataset by generating question and answer pairs for the remaining scenarios in WOMD (Ettinger et al., 2021). Second, we used generative AI assistants and coding agents to edit author-written text for clarity and readability and to assist with code development. All AI assisted outputs were reviewed by the authors. The authors take full responsibility for the final content of this work, including all text, claims, code, and data produced with the aid of generative AI.

## REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bišković, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: A Visual Language Model for Few-Shot Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. In *Neural Information Processing Systems (NeurIPS) – Deep Learning Symposium*, 2016.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Wei-Jer Chang, Wei Zhan, Masayoshi Tomizuka, Manmohan Chandraker, and Francesco Pittaluga. LANGTRAJ: Diffusion Model and Dataset for Language-Conditioned Trajectory Simulation. In *International Conference on Computer Vision (ICCV)*, 2025.

Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with LLMs: Fusing Object-Level Vector Modality for Explainable Autonomous Driving. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Yu-Hsiang Chen, Wei-Jer Chang, Yi-Ting Chen, and Masayoshi Tomizuka. ECoSim: Data Efficient Fine-Tuning for Controllable Traffic Simulation. In *European Conference on Computer Vision (ECCV)*, 2026.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning (CoRL)*, 2017.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *International Conference on Computer Vision (ICCV)*, 2021.

Abdulwahab Felemban, Nussair Hroub, Jian Ding, Eslam Abdelrahman, Xiaoqian Shen, Abdu-allah Mohamed, and Mohamed Elhoseiny. iMotion-LLM: Instruction-Conditioned Trajectory Generation. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

Anurag Ghosh, Srinivasa Narasimhan, Manmohan Chandraker, and Francesco Pittaluga. RAD-LAD: Rule and Language Grounded Autonomous Driving in Real-Time. *arXiv preprint arXiv:2603.28522*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024. Technical report.

Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, John D. Co-Reyes, Rishabh Agarwal, Rebecca Roelofs, Yao Lu, Nico Montali, Paul Mougin, Zoey Yang, Brandyn White, Aleksandra Faust, Rowan McAllister, Dragomir Anguelov, and Benjamin Sapp. Waymax: An Accelerated, Data-Driven Simulator for Large-Scale Autonomous Driving Research. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Marcel Hallgarten, Julian Zapata, Martin Stoll, Katrin Renz, and Andreas Zell. Can vehicle motion planning generalize to realistic long-tail scenarios? In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. MotionGPT: Human Motion as a Foreign Language. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual Explanations for Self-Driving Vehicles. In *European Conference on Computer Vision (ECCV)*, 2018.

Yiheng Li, Cunxin Fan, Chongjian Ge, Zhihao Zhao, Chenran Li, Chenfeng Xu, Huaxiu Yao, Masayoshi Tomizuka, Bolei Zhou, Chen Tang, et al. Womd-reasoning: A large-scale dataset for interaction reasoning in driving. In *International Conference on Machine Learning (ICML)*, 2025.

Zekun Li, Sizhe An, Chengcheng Tang, Chuan Guo, Ivan Shugurov, Linguang Zhang, Amy Zhao, Srinath Sridhar, Lingling Tao, and Abhay Mittal. Llamo: Scaling pretrained language models for mfied motion understanding and generation with continuous autoregressive tokens. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zhiyuan Liu, Leheng Li, Yuning Wang, Haotian Lin, Hao Cheng, Zhizhe Liu, Lei He, and Jianqiang Wang. Controllable Traffic Simulation through LLM-Guided Hierarchical Reasoning and Refinement. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. DRAMA: Joint Risk Localization and Captioning in Driving. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.

Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nick Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, Brandyn White, and Dragomir Anguelov. The waymo open sim agents challenge. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2Drive: Towards Interpretable and Chain-based Reasoning for Autonomous Driving. In *European Conference on Computer Vision (ECCV)*, 2024.

Jeongeun Park, Sungjoon Choi, and Sangdoo Yun. A Unified Framework for Motion Reasoning and Generation in Human Interaction. In *International Conference on Computer Vision (ICCV)*, 2025.

Muleilan Pei, Shaoshuai Shi, and Shaojie Shen. Advancing Multi-Agent Traffic Simulation via R1-Style Reinforcement Fine-Tuning. In *International Conference on Learning Representations (ICLR)*, 2026.

Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. In *International Conference on Learning Representations (ICLR)*, 2024.

Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multimodal visual question answering benchmark for autonomous driving scenario. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *International Conference on Computer Vision (ICCV)*, 2023.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European Conference on Computer Vision (ECCV)*, 2024.

Nan Song, Junzhe Jiang, Jingyu Li, Xiatian Zhu, and Li Zhang. UniMotion: A Unified Motion Framework for Simulation, Prediction and Planning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Shuhan Tan, Boris Ivanovic, Xinshuo Weng, Marco Pavone, and Philipp Kraehenbuehl. Language conditioned traffic generation. In *Conference on Robot Learning (CoRL)*, 2023.

Shuhan Tan, Boris Ivanovic, Yuxiao Chen, Boyi Li, Xinshuo Weng, Yulong Cao, Philipp Krähenbühl, and Marco Pavone. Promptable closed-loop traffic simulation. In *Conference on Robot Learning (CoRL)*, 2024.

Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S. Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, and Benjamin Sapp. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Mingyi Wang, Jingke Wang, Tengju Ye, Junbo Chen, and Kaicheng Yu. Do LLM Modules Generalize? A Study on Motion Generation for Autonomous Driving. In *Conference on Robot Learning (CoRL)*, 2025.

Yinuo Wang, Mining Tan, Yuanxin Zhong, Siyuan Cheng, et al. Bird's-eye-view informed reasoning driver. In *International Conference on Learning Representations (ICLR)*, 2026a.

Yuan Wang, Di Huang, Yaqi Zhang, Wanli Ouyang, Jile Jiao, Xuetao Feng, Yan Zhou, Pengfei Wan, Shixiang Tang, and Dan Xu. MotionGPT-2: A General-Purpose Motion-Language Model for Motion Generation and Understanding. In *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2026b.

Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. SMART: Scalable multi-agent real-time motion generation via next-token prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Junkai Xia, Chenxin Xu, Qingyao Xu, Yanfeng Wang, and Siheng Chen. Language-Driven Interactive Traffic Trajectory Generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Lingyu Xiao, Zexin Feng, and Xintao Yan. Long-term Traffic Simulation via Structured Autoregressive Modeling. In *European Conference on Computer Vision (ECCV)*, 2026.

Kairui Yang, Zihao Guo, Gengjie Lin, Haotian Dong, Zhao Huang, Yipeng Wu, Die Zuo, Jibin Peng, Ziyuan Zhong, Xin Wang, Qiang Guo, Xiaosong Jia, Junchi Yan, and Di Lin. Trajectory-LLM: A Language-Based Data Generator for Trajectory Prediction in Autonomous Driving. In *International Conference on Learning Representations (ICLR)*, 2025a.

Xiuyu Yang, Shuhan Tan, and Philipp Krähenbühl. Long-term Traffic Simulation with Interleaved Autoregressive Motion and Scenario Generation. In *International Conference on Computer Vision (ICCV)*, 2025b.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*, 2020.

Zhejun Zhang, Peter Karkus, Maximilian Igl, Wenhao Ding, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Closed-loop supervised fine-tuning of tokenized traffic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Zhiyuan Zhang, Xiaosong Jia, Guanyu Chen, Qifeng Li, Zuxuan Wu, Yu-Gang Jiang, and Junchi Yan. Trajtok: What makes for a good trajectory tokenizer in behavior generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. Language-Guided Traffic Simulation via Scene-Level Diffusion. In *Conference on Robot Learning (CoRL)*, 2023.

Bingfan Zhu, Biao Jiang, Sunyi Wang, Shixiang Tang, Tao Chen, Linjie Luo, Youyi Zheng, and Xin Chen. MotionGPT3: Human Motion as a Second Modality. In *International Conference on Learning Representations (ICLR)*, 2026.

# A Additional Experimental Setup Details

### A.0.1 Training Details

We train MoLan in a single stage using the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of $5 \times 10^{-5}$, a minimum learning rate ratio of $0.05$, and gradient clipping with a maximum norm of $0.5$. The models reported in the main tables were trained for 6 epochs on 8 H100 GPUs, utilizing LLaMA3.2-1B as the language model and CAT-$k$ as the motion model. The batch size was set to 4 per GPU. Training is performed in BF16 precision. The motion and language backbones remain frozen throughout training, and only the motion language bridge parameters are optimized. We jointly train the T2M and M2T objectives, sampling T2M and M2T batches with an 80:20 ratio. For T2M, we use closed-loop training and compute the motion next-token prediction loss over all modeled agents. For M2T, the language modeling loss is computed only over the answer tokens. The MoLan architecture uses 6 bridge blocks, with a hidden dimension of 256 for the query, key, and value projections, and 8 attention heads per bridge block.

We reproduce the Motion-LLaVA architecture and fine-tuning procedure introduced in WOMD-Reasoning. Specifically, we use LLaVA-v1.5-7B (Liu et al., 2023) as the pretrained language model, together with the MultiPath++ Varadarajan et al. (2022) motion vector encoder[2]. The encoded motion features are projected into the language model embedding space and jointly processed with the textual input. Following the original training strategy, we train all model parameters on the complete training set in a single stage for one epoch, using the same model configuration and optimization hyperparameters as the original implementation. The only difference is the training data: rather than restricting training to the original WOMD-Reasoning dataset, we train on our full augmented dataset.

During inference, for motion generation, and to ensure fairness, we do not use any heuristic strategies during inference when sampling future rollouts. We generate 32 joint future samples per scenario.

## A.1 WOSAC Challenge Metrics

The Waymo Open Sim Agents Challenge (WOSAC) (Montali et al., 2023) evaluates the distributional realism of closed-loop traffic simulation using nine statistical features grouped into kinematic, interaction, and map-based metrics. Following the standard protocol, a model generates $R = 32$ joint eight-second rollouts containing up to 128 agents for each scenario. For scenario $i$, agent $a$, timestep $t$, and feature $F_j$, WOSAC constructs an empirical feature distribution $p_{i,j,a}$ from the generated rollouts and evaluates the negative log-likelihood (NLL) of the logged trajectory $\mathbf{x}^{\star}_{i,a,t}$:

$$\mathrm{NLL}(i, a, t, j) = -\log p_{i,j,a}\big(F_j\big(\mathbf{x}^{\star}_{i,a,t}\big)\big)\,. \tag{7}$$

Let $v_{i,a,t} \in \{0, 1\}$ denote whether agent $a$ is valid at timestep $t$, and let $N_{i,a} = \sum_t v_{i,a,t}$. The feature likelihood for an agent is computed as the exponential of its mean log-likelihood over valid timesteps:

$$m(i, a, j) = \exp\left(-\frac{1}{N_{i,a}} \sum_t v_{i,a,t}\, \mathrm{NLL}(i, a, t, j)\right). \tag{8}$$

The evaluator then averages over the set of evaluated agents $\mathcal{A}_i$ in each scenario,

$$m(i, j) = \frac{1}{|\mathcal{A}_i|} \sum_{a \in \mathcal{A}_i} m(i, a, j), \tag{9}$$

and reports the WOSAC realism meta-metric (RMM) as a weighted average of the nine feature likelihoods over the $S$ evaluation scenarios:

$$M = \frac{1}{S} \sum_{i=1}^{S} \sum_{j=1}^{9} w_j m(i, j), \qquad \sum_{j=1}^{9} w_j = 1. \tag{10}$$

Higher component and RMM values indicate closer agreement between the generated and logged traffic distributions.

**Kinematic metrics.** The kinematic score aggregates four features describing the motion dynamics of individual agents: linear speed, linear acceleration magnitude, angular speed, and angular acceleration

[2]Following Li et al. (2025), we use the MultiPath++ implementation provided by: https://github.com/stepankonev/waymo-motion-prediction-challenge-2022-multipath-plus-plus.

magnitude. Given position $\mathbf{p}_t$, heading $\theta_t$, and sampling interval $\Delta t$, the corresponding quantities are

$$\mathbf{v}_t = \frac{\mathbf{p}_{t+1} - \mathbf{p}_t}{\Delta t}, \qquad \mathbf{a}_t = \frac{\mathbf{v}_{t+1} - \mathbf{v}_t}{\Delta t}, \tag{11}$$

and

$$\omega_t = \frac{d(\theta_{t+1}, \theta_t)}{\Delta t}, \qquad \alpha_t = \frac{\omega_{t+1} - \omega_t}{\Delta t}, \tag{12}$$

where $d(\cdot, \cdot)$ denotes the minimum angular difference on the unit circle. WOSAC evaluates the empirical likelihoods of $\|\mathbf{v}_t\|_2$, $\|\mathbf{a}_t\|_2$, $|\omega_t|$, and $|\alpha_t|$, thereby measuring whether the generated agents exhibit realistic changes in position and heading.

**Interaction metrics.** The interaction score aggregates the signed distance to the nearest object, collision occurrence, and time-to-collision (TTC). The nearest-object feature is computed using the signed Gilbert–Johnson–Keerthi (GJK) distance between oriented agent bounding boxes. A negative signed distance indicates an overlap and activates the binary collision feature. TTC estimates the remaining time before two agents collide under a constant-velocity extrapolation. Together, these features measure whether the joint rollouts reproduce realistic inter-agent spacing, collision statistics, and anticipatory interactions.

**Map-based metrics.** The map score contains two features: the signed distance to the nearest road edge and a binary road-departure indicator. The signed distance measures an agent's position relative to the drivable-area boundary, while road departure records whether the agent leaves the road at any point during the rollout. Their likelihoods quantify whether the generated trajectories remain consistent with the scene geometry and reproduce the map-adherence statistics of logged traffic.

**Displacement metrics.** WOSAC metrics evaluate distributional realism but do not directly measure whether a language-conditioned rollout follows the behavior specified by its instruction. We therefore additionally report minimum average displacement error (minADE). For rollout $r$ and agent $a$, the average displacement error over its valid timesteps $\mathcal{T}_{i,a}$ is

$$\mathrm{ADE}_{i,r,a} = \frac{1}{|\mathcal{T}_{i,a}|} \sum_{t \in \mathcal{T}_{i,a}} \left\| \mathbf{p}^{(r)}_{i,a,t} - \mathbf{p}^{\star}_{i,a,t} \right\|_2 . \tag{13}$$

We average this error over the evaluated agents within each joint rollout and select the best of the $R$ generated rollouts:

$$\mathrm{minADE}_i = \min_{r \in \{1,\ldots,R\}} \frac{1}{|\mathcal{A}_i|} \sum_{a \in \mathcal{A}_i} \mathrm{ADE}_{i,r,a}. \tag{14}$$

Lower minADE indicates stronger spatial agreement between the generated trajectories and the logged future associated with the language input.

## B Additional Experimental Results

**How does cross-modal conditioning emerge during training?** We track the absolute $\tanh$ gate values of the T2M and M2T streams throughout training in Figure 7B. Since we initialize the bridge gates to zero, MoLan initially follows the computation of the frozen motion and language towers. During training, the gates quickly grow across all six bridge blocks and in both directions, showing that each modality progressively learns to leverage information from the other. Gates also grow with depth, with deeper blocks converging to larger values than earlier ones—suggesting that cross-modal grounding matters most where representations are more semantic (agent intentions, instruction meaning), while early, largely modality-specific features rely less on cross-modal exchange.

**Cross-attention outperforms joint-attention.** Table 6 compares dual-stream cross-attention and joint-attention, both with and without $\tanh$ gating. Cross-attention consistently outperforms joint attention for both language-controlled simulation and motion-grounded understanding under both configurations, with the gated variant performing best overall. We attribute this difference to the frozen dual-tower setting. The motion and language backbones already model within-modality dependencies through their native layers. Joint attention mixes intra- and cross-modal interactions, forcing the interface to relearn existing dependencies. In contrast, dual-stream cross-attention preserves each stream and focuses learning on cross-modal information exchange. This improves cross-modal performance while reducing attention memory, especially as motion sequences grow with the number of agents and timesteps.

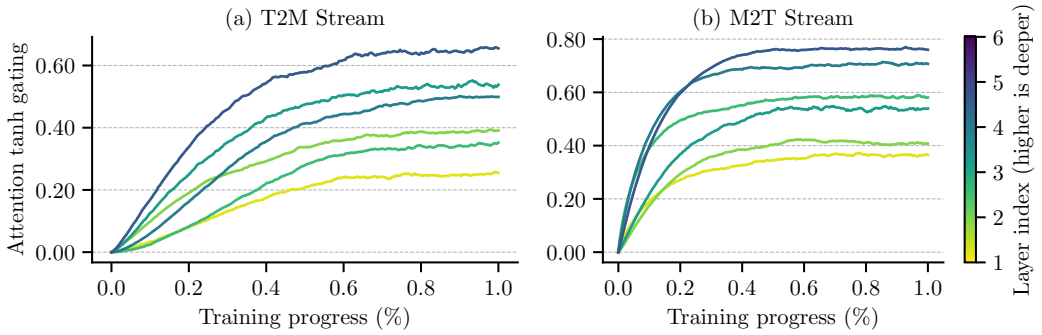



Figure 6: **Evolution of absolute** tanh **gate values across the layers of MoLan during training.**

Table 6: **Ablation of cross-modal fusion.** We compare joint-attention with our dual-stream cross-attention, with and without tanh gating, under the same frozen backbones and training setup.

| | tanh gating | RMM ↑ | BERTScore ↑ |
|---|---|---|---|
| Joint-Attention | × | 0.774 | 0.602 |
| | ✓ | 0.781 | 0.642 |
| Cross-Attention | × | 0.778 | 0.613 |
| | ✓ | **0.786** | **0.668** |

**Bridge block placement.** We evaluate four placement strategies while keeping the number of bridge blocks fixed: concentrating them in the early, middle, or late layers, or distributing them uniformly across both towers. As shown in Table 7, uniform placement achieves the highest Composite Score, whereas concentrating the blocks in the late layers performs worst. Early and intermediate representations retain detailed agent-motion and linguistic information that can be progressively integrated by the subsequent frozen layers. In contrast, features near the output layers are already specialized toward motion- and language-token prediction, leaving limited depth for cross-modal information to influence generation. Distributing the bridges throughout the towers therefore enables motion and language to interact across multiple levels of representation while preserving the modality-specific computation performed by each backbone.

Table 7: **Ablation of motion–language bridge block placement.** We use CAT-$k$ with 6 motion layers and LLaMA3.2-1B with 16 language layers. We fix the number of bridge blocks to 6 and vary their placement across the two towers.

| Strategy | RMM ↑ | BERTScore ↑ |
|---|---|---|
| Early layers | 0.784 | **0.671** |
| Middle layers | 0.782 | 0.653 |
| Late layers | 0.781 | 0.633 |
| Uniform | **0.786** | 0.669 |