Unifying Language-Action Understanding and Generation for Autonomous Driving

Supplementary Material

Appendix

A. Action Tokenization

A.1. Log-Coordinate Transformation

We visualize the log-transformed coordinate space to facilitate intuitive comparison. We devise a non-uniform quantization scheme that prioritizes precision near the ego-vehicle by first applying a non-linear transformation to the waypoint coordinates (x, y) along each axis independently.

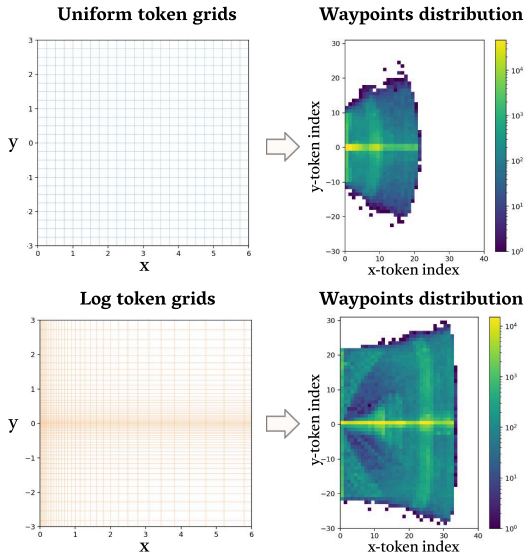

Figure S1. Comparison of uniform and log token grids, with the corresponding waypoint distributions under each grid.

A.2. Number of Action Tokens

We evaluate the effect of the number of action tokens on driving performance on the Bench2Drive [2] benchmark. To this end, we adopt a non-uniform quantization scheme that prioritizes precision near the ego-vehicle by first applying a non-linear transformation to the waypoint coordinates (x, y) along each axis independently. Specifically, each coordinate $z \in \{x, y\}$ is transformed using a signed logarithmic function:

$$z' = \text{sign}(z) \cdot \log\big(1 + k \cdot |z|\big). \tag{S1}$$

We then vary the symmetric logarithmic (symlog) scaling factor k from 5.0 to 10.0, which increases the number of action tokens from 5,656 to 7,245. The parameter k controls the mapping from physical coordinates (both x and y, in meters) to the transformed space, determining the degree of compression prior to binning and, in turn, the effective bin widths in the original coordinate space.

Table S1. Effect of the number of action tokens (controlled by k) in closed-loop evaluation [1].

Method	Driving Score↑	Success Rate (%)↑
k = 5.0 (5,656 tokens)	91.01	74.55
k = 10.0 (7,245 tokens)	89.85	70.45

A.3. Spatial Soft-Labeling

We evaluate the spread scale in the spatial soft-labeling on driving performance on the Bench2Drive [2] benchmark by varying the spread parameter σ from 1.2 to 3.0. Larger σ yields softer targets by broadening the Gaussian smoothing and distributing probability over a wider neighborhood. Specifically, for a ground-truth token a_{gt}, we construct the target distribution $q(a)$ over all action tokens $a \in \mathcal{C}_{\text{action}}$ as a normalized 2D Gaussian centered at the coordinates of a_{gt}:

$$q(a) = \frac{1}{Z} \exp\left(-\frac{\|\text{pos}(a) - \text{pos}(a_{gt})\|_2^2}{2\sigma^2}\right), \tag{S2}$$

where $\text{pos}(a)$ maps an action token to its 2D coordinates in the spatial grid, $\|\cdot\|_2^2$ denotes the squared Euclidean distance, σ is a hyperparameter controlling the spread of the distribution, and Z is a normalization constant ensuring $\sum_{a \in \mathcal{C}_{\text{action}}} q(a) = 1$.

Table S2. Effect of the spread scale parameter σ in the spatial soft-labeling in closed-loop evaluation [1].

Method	Driving Score↑	Success Rate (%)↑
$\sigma = 1.2$	91.01	74.55
$\sigma = 3.0$	89.73	69.55

B. Dataset

B.1. Action Dreaming

We conduct experiments on the *Action Dreaming* [3] dataset and its offline, nonreactive simulator, which generates alternative ego-vehicle trajectories and assesses their feasibility with respect to collision avoidance and traffic-rule compliance. Ego-trajectory prediction is implemented with a kinematic bicycle model controlled by PID controllers (PDM-lite [5]), driven either by perturbed ground-truth actions or by PID commands computed from predefined path waypoints and target speeds. The dataset provides simulator states and short-horizon forecasts for dynamic objects to enable collision checking. The simulator supports several modes (Objects/Collision, Faster, Slower, Target Speed, Lane Changes, Stop) to induce diverse behaviors and trajectories. we use Success Rate as the metric. Each category has its own definition of success, which we detail in the following:

- Objects (Collision): This describes the task of driving towards or crashing into specific objects. The path is evaluated first. If the path of the expert trajectory and the ground truth dreamer trajectory is different (Average Displacement Error $ADE > 1.0$) it is counted as success if the predicted path is closer to the ground truth dreamer path than to the expert path ($ADE_{pred2expert} > ADE_{pred2dreamer}$). If the dreamer path is nearly identical to the expert path ($ADE < 1.0$) the instruction is about correct speed predictions (*e.g.*, if the instruction is "drive towards a dynamic object" it is important to get the speed right and not just the path). The success is then defined as $ADE_{pred2dreamer} < 1.0$ and the average predicted speed is within 30% of the ground truth dreamer speed.
- Faster (Speed up): For each predicted speed waypoint, derive target speeds for future timesteps and fit a linear regression to obtain the slope s. Let v denote the current ego speed at the start of the sequence. Success is defined as $s > 0.05\,v$.
- Slower (Slow down): Same computation as Faster. Success is defined as $s < -0.05\,v$.
- Target Speed: Since the target speed may not be reached in the prediction horizon of the waypoints, predictions are compared with the ground truth actions instead of directly comparing to the target speed. Two rules define success: First, if the predicted target speed inferred from the last two waypoints is in a 20% range of the instructed target speed. Second, if the predicted target speed inferred from the last two waypoints is in the 20% range of the speed of the last two waypoints of the ground truth speed waypoints. This can be different from the instructed target speed due to limitations in the acceleration rates of the vehicle.
- Lane Changes: Compare the final waypoint of the predicted path with the final waypoints of the ground-truth dreamer path and the ground-truth expert path. Success is defined when the predicted final location is closer to the dreamer's final location than to the expert's final location.
- Stop: Success is defined when the minimum predicted speed over the sequence is below $0.1\,\mathrm{m/s}$.

B.2. VQA-DriveLM

The VQA data from SimLingo [3] are sourced from the DriveLM-Carla [4] dataset and generated using its data-creation pipeline. Question–answer pairs are extracted from the adopted dataset rather than the original DriveLM release; the training split contains 28M QA pairs over 1M frames in Town 12. Evaluation follows DriveLM keyframe selection to focus on informative frames, and the validation split is balanced across answer types. Labels are heuristically auto-generated, and the dataset includes GPT-4–based paraphrase augmentation (up to 20 variants per QA) to mitigate phrase-level overfitting, with variants sampled at load time.

B.3. Commentary

The commentary labels in SimLingo [3] are automatically generated from a subset of saved simulator state using heuristic, template-based rules. Each label comprises: (1) a route action with justification—default "Follow the route," replaced only for scenarios requiring lane deviation (*e.g.*, obstacle, encroaching vehicle), with phase-specific templates for before/during/after the deviation; (2) a speed action categorized as remain stopped, stop now, maintain (or maintain reduced) speed, increase speed, or slow down, with a special "Wait for a gap before changing lanes" case when stationary prior to a deviation; and (3) a speed reason derived from IDM [6] features that identify the leading object (vehicle/pedestrian/static control) and its attributes/state, from which concise rationales are composed (*e.g.*, due to pedestrian crossing, behind a red SUV, because the light is red/green). When near a junction, an additional notice summarizes other vehicles' positions and motions (*e.g.*, junction clear, vehicle moving away, oncoming traffic).

C. Qualitative Results

Figure S2 provides further qualitative results from the closed-loop evaluation, illustrating our model's performance in a variety of driving scenarios.

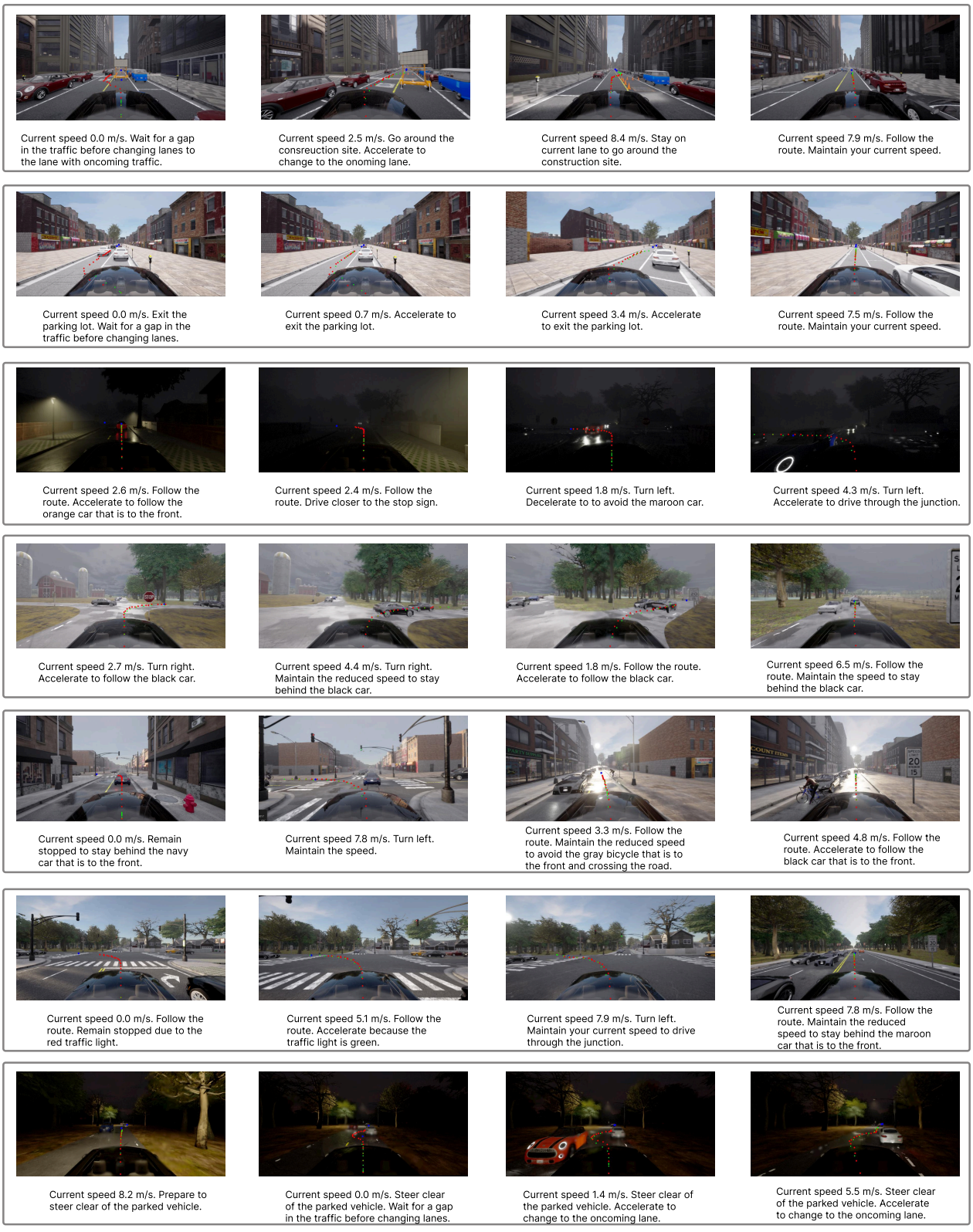

Figure S2. Qualitative results of our proposed model during closed-loop evaluation in the CARLA simulator. The figure showcases representative driving scenarios, such as navigating intersections and avoiding obstacles.

References

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

[2] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *NeurIPS 2024 Datasets and Benchmarks Track*, 2024. 1

[3] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11993–12003, 2025. 2

[4] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023. 2

[5] 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*, pages 256–274. Springer, 2024. 2

[6] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000. 2