AutoTraces: Autoregressive Trajectory Forecasting via Multimodal Large Language Models

Supplementary Material

6. Point Encoder and Decoder Architecture

To encode 2D waypoint coordinates into LLM-compatible representations, we employ an encoder-decoder architecture. As shown in Fig. 7(a), the encoder processes (x, y) coordinates using sinusoidal positional encoding with geometrically decreasing wavelengths:

$$\begin{aligned} \mathrm{PE}(p_d, 2i) &= \sin\left(p_d \cdot \lambda^{-2i/D}\right), \\ \mathrm{PE}(p_d, 2i+1) &= \cos\left(p_d \cdot \lambda^{-2i/D}\right), \end{aligned} \tag{5}$$

where p_d denotes coordinate values, $i \in [0, D/2-1]$ indexes frequency bands, D is the embedding dimension, and $\lambda = 1000$. This multi-frequency encoding captures both fine-grained and coarse spatial relationships. The decoder (Fig. 7(b)) reconstructs waypoints through linear layers with GELU activations. All linear layers use Xavier initialization ($\sigma = 0.1$) and zero biases. This design provides two advantages: (1) effective generalization to unseen coordinates, and (2) seamless spatial-linguistic fusion for multimodal reasoning.

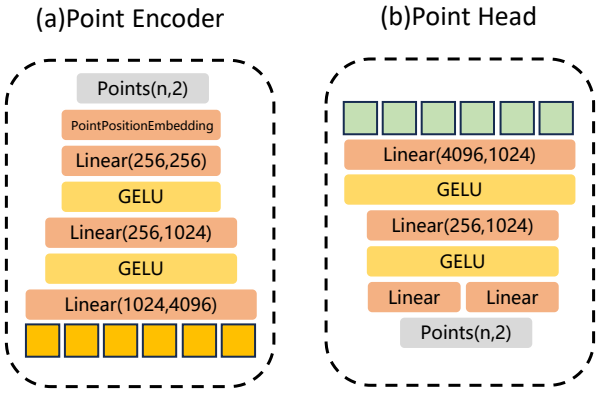

Figure 7. Architecture of the point encoder and decoder. The encoder transforms the input into a higher-dimensional space, and the decoder projects it back to the original dimensionality.

7. Comparison on computational cost

The table below compares the computational cost of AutoTraces and CityWalker. All experiments in the manuscript fine-tune a 7B base model. To further test scalability, we additionally fine-tune a smaller 0.5B base model. Notably, even with the smaller 0.5B model, performance improves, partly due to its stronger pretraining on larger-scale data. In this 0.5B setting, our method significantly outperforms CityWalker (1.266 vs. 1.806 for L1) with about 4× higher computation. These results not only further validate the effectiveness of the point embedding which does not solely rely on the large parameter scale of foundation models, but also demonstrate that our embedding paradigm is compatible with recent VLMs and can benefit from advances in VLMs, with the potential for lower computational cost as more efficient models emerge.

Method	Base Model	GFLOPS	GMACs	L1/L2 (T=10)
Autotraces	LLaVA-NeXT-Video-7B	2.125e4	1.063e4	1.384/1.089
Autotraces*	LLaVA-OneVision-0.5B	6.420e3	3.210e3	1.266/1.002
CityWalker	N/A	1.541e3	0.771e3	1.806/1.407

8. Comparison on Instruction Following

Under identical few-shot fine-tuning conditions, our method achieves significantly higher Instruction Execution Accuracy compared to the pure text-based LLaVA-Video (Tab. 3). As shown in Tab. 4 & Tab. 5, pure text-based numerical prediction suffers from frequent instruction-following errors and instability, including abnormal repetition that prolongs generation. In contrast, our approach reduces errors to just 2 in 2503 samples, with consistent output formats and minimal stability impact. By decoupling trajectory prediction from the LLM's reasoning core, our method enables more stable inference, reduces training costs, and decreases token usage, offering a viable pathway for LLM integration in numerical tasks.

9. Prompt Instruction

In Fig. 2 (d), we summarize the key elements covered in our main prompt. In Tab. 6 we provide complete prompt inputs and corresponding outputs , including both our AutoTraces and LLaVA-Video. In the prompt instruction, we encode both the historical positions and the target position using a unified `<point>` modality. This ensures the alignment of input and output waypoint information, enabling the model to couple and comprehend information within the same token space. Furthermore, the prompt instruction includes an interpretation of the video information, encompassing video length, frame rate, a description of the task, and an explanation of the `<point>` token. This comprehensive prompt instruction helps the LLM better understand the task content and generate a corresponding number of trajectory waypoints as required.

10. Future Work

Built on a modular and extensible framework, our approach enables flexible adaptation to diverse embodied tasks while maintaining scalability. The data pipeline supports arbitrary robot navigation datasets and variable-length inputs/outputs without architectural changes, enabling seamless prediction horizon switching via minimal fine-tuning. The decoupled `<point>` modality design prevents catastrophic forgetting in the LLM and allows independent upgrades of vision-language backbones or point processors. These capabilities facilitate cross-domain transfer and future integration of advanced models. To facilitate reproducibility and future research, we will open-source our codebase, filling a gap in available implementations for autoregressive LLM-token-based trajectory prediction. We hope to extend this numerical encoding approach beyond trajectory prediction to arbitrary spatio-temporal sequence data, enabling LLMs to handle diverse numerical reasoning tasks more efficiently and opening new avenues for combining LLMs with numerical reasoning in future work.

Table 4. Analysis of common success and failure modes in LLaVA-Video

Success Case
Expected: 20 trajectory points
Model Output (Correct): ASSISTANT: Based on the video and given information, the predicted points of the trajectory over the next 20 seconds are [[-1.627, -7.708], [-1.817, -8.678], [-2.003, -9.648], [-2.203, -10.633], [-2.402, -11.618], [-2.601, -12.583], [-2.801, -13.573], [-3.003, -14.548], [-3.203, -15.528], [-3.403, -16.492], [-3.603, -17.473], [-3.803, -18.443], [-4.003, -19.423], [-4.203, -20.393], [-4.403, -21.363], [-4.603, -22.333], [-4.803, -23.303], [-5.003, -24.273], [-5.203, -25.233], [-5.403, -26.193]].$< /s >$

Failure Case 1: Violation of Length Specifications
Expected: 20 trajectory points
Model Output (Incorrect): ASSISTANT: Based on the video and given information, the predicted points of the trajectory over the next 20 seconds are [[-3.003, -7.733], [-3.438, -8.708], [-3.8430, -9.6730], [-4.2170, -10.6330], [-4.5940, -11.5730], [-4.9540, -12.5230], [-5.3090, -13.4730], [-5.6730, -14.4330], [-6.0370, -15.3830], [-6.4020, -16.3330], [-6.7730, -17.2830], [-7.1370, -18.2330], [-7.5090, -19.1930], [-7.8740, -20.1430], [-8.2370, -21.0930], [-8.6070, -22.0430], [-8.9730, -22.9930], [-9.3430, -23.9380], [-9.7170, -24.9010], [-10.0830, -25.8530], [-10.4530, -26.7930], [-10.8270, -27.7380], [-11.5350, -27.6060]].$< /s >$
Issue: 23 points generated

Failure Case 2: Failure to Terminate Generation
Expected output ASSISTANT: Based on the video and given information ... [-10.4530, -26.7930], [-10.8270, -27.7380], [-11.5350, -27.6060]].$< /s >$
Model Output (Incorrect): ASSISTANT: Based on the video and given information, the predicted points of the trajectory over the next 20 seconds are [[0.5170, -8.178], [0.4780, -9.233], [0.4340, -10.273099999999999], ... , [0.1000, -26.653000000000003]]:[[0.3670, -28.617000000000003], ... , [0.3670, -29.653000000000003], [0.3670, -29.653000000000003], [0.3670, -29.65300000
Issue: Abnormal repetitive generation, preventing EOS output and resulting in truncation

Failure Case 3: Abnormal Format
Model Output (Incorrect): ASSISTANT: Based on the video and given information, ... ,[-18.547, -29.263], [-19.207, -30.493099999999998, -31.717099999999999]] ...
Issue: Introducing a z-coordinate in (x,y) trajectories

Table 5. Analysis of common success and failure modes in AutoTraces

Success Case
Expected: 20 trajectory points
Model Output (Correct): ASSISTANT: Based on the video and given information, the predicted tokens of the trajectory over the next 20 seconds are $< point\_start >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point\_end >$. $< /s >$

Failure Case: Violation of Length Specifications
Expected: 20 trajectory points
Model Output (Incorrect): ASSISTANT: Based on the video and given information, the predicted tokens of the trajectory over the next 20 seconds are $< point\_start >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point >< point\_end >$. $< /s >$
Issue: 19 points generated

Table 6. Prompts used in AutoTraces and LLaVA-Video

Prompt used for LLaVA-Video
USER: $< video >$ You are an expert in robot trajectory prediction. Analyze a video of 8 frames from the robot's first-person perspective, with each frame lasting 1 second. The robot's positions for each frame are [[0.0, 0.0], [-0.0534, -0.0703], [-0.0834, -0.1452], [-0.1419, -0.2216], [-0.2061, -0.3074], [-0.2892, -0.393], [-0.3879, -0.4566], [-0.4783, -0.5488]], and the target coordinate for the future twentieth second is [-1.6054, -2.6351]. Based on historical movement patterns and the final target, predict 6 socially compliant trajectory points for the next 6 seconds. Output only the list of coordinates in the format: [(x1, y1), (x2, y2), ...]. **ASSISTANT:** Based on the video and given information, the predicted tokens of the trajectory over the next 6 seconds are [[-0.562, -0.638], [-0.658, -0.718], [-0.75, -0.801], [-0.848, -0.888], [-0.942, -0.972], [-1.012, -1.04]].
Prompt used for AutoTraces
USER: $< video >$ You are an expert in robot trajectory prediction, and you need to analyze a video of 8 frames. The video is from the robot's first-person perspective, with each frame lasting 1 second. The robot's points for each frame are $< point\_start >< point >< point >< point >< point >< point >< point >< point >< point\_end >$. The coordinate for the future twentieth second is $< point\_start >< point >< point\_end >$, which will serve as the final target point.To improve prediction accuracy, tokens $< point\_start >< point >< point\_end >$ are used to encode the original coordinate data, where each token corresponds to a coordinate point. Please perform an analysis of the relationships between consecutive frames in the video, taking into account the historical coordinate information of the robot as well as the target point coordinate, and ensure that the predictions are socially compliant, adhering to common sense (such as yielding, etc.). Then, generate 6 predictive tokens for the trajectory over the next 6 seconds. **ASSISTANT:** Based on the video and given information, the predicted tokens of the trajectory over the next 6 seconds are $< point\_start >< point >< point >< point >< point >< point >< point >< point\_end >$.