ACT-BENCH: TOWARDS ACTION CONTROLLABLE WORLD MODELS FOR AUTONOMOUS DRIVING

Hidehisa Arai **Keishi Ishihara** **Tsubasa Takahashi** **Yu Yamaguchi**
Turing Inc.
{hidehisa.arai, keishi.ishihara}@turing-motors.com

ABSTRACT

World models have emerged as promising neural simulators for autonomous driving, with the potential to supplement scarce real-world data and enable closed-loop evaluations. However, current research primarily evaluates these models based on visual realism or downstream task performance, with limited focus on fidelity to specific action instructions. Although some studies address action fidelity, their evaluations rely on closed-source mechanisms, limiting reproducibility. To address this gap, we develop an open-access evaluation framework, ACT-BENCH, for quantifying action fidelity, along with a baseline world model, TERRA. Our framework includes a large-scale dataset pairing short context videos from nuScenes with corresponding future trajectories, which provide conditional inputs for generating future video frames and enable evaluation of action fidelity for executed motions. Leveraging this framework, we demonstrate that the state-of-the-art model does not fully adhere to given instructions, while TERRA demonstrates better action fidelity. **All components of our benchmark framework are publicly available at `https://turingmotors.github.io/actbench/` to support future research.**

1 INTRODUCTION

Autonomous driving has advanced rapidly in recent years (Yurtsever et al., 2020; Grigorescu et al., 2020; Chen et al., 2024a), aiming for safe navigation in complex and dynamic environments. However, achieving this goal still requires addressing significant challenges, such as collecting extensive real-world data especially for safety-critical scenarios and adapting to unpredictable road conditions.

One promising approach to addressing these challenges is through the use of world models (Ha & Schmidhuber, 2018; LeCun, 2022). In autonomous driving, world models are expected to serve as neural simulators (Zhu et al., 2024) that generate synthetic scenarios, supplementing real-world data, which can be difficult to collect, and enabling closed-loop evaluations of autonomous driving systems. In earlier research, world models for autonomous driving were developed and validated in simplified simulation environments to circumvent the complexity and risks involved in real-world data collection (Pan et al., 2022; Hu et al., 2022; Gao et al., 2024b). However, as the field matures, growing interest in accurately capturing real-world complexity has driven recent efforts to construct world models from actual driving (Wang et al., 2023; Hu et al., 2023a; Wang et al., 2024b; Lu et al., 2025; Wang et al., 2024b; Gao et al., 2024a).

Despite these advancements, practical applications still lack fidelity to action instructions—crucial for reliable, safety-critical simulations. Recently, Vista (Gao et al., 2024a) introduced an action fidelity-aware model that allows for a broad range of conditional inputs. However, its generated scenes do not fully follow the given instructions (Figure 1), suggesting insufficient fidelity.

To advance action fidelity-aware world models, open-access evaluation benchmarks are needed. DriveGAN (Kim et al., 2021) is the first work to measure fidelity by comparing ground-truth actions against those inferred from generated scenes. Later, GenAD (Yang et al., 2024a) introduced a trajectory-based metric. However, neither method has publicly released the evaluation models, limiting reproducibility. Similarly, DrivingDojo Dataset (Wang et al., 2024a) uses trajectory-based metric but does not disclose essential details—such as which (`scene`, `action`) pairs are used or

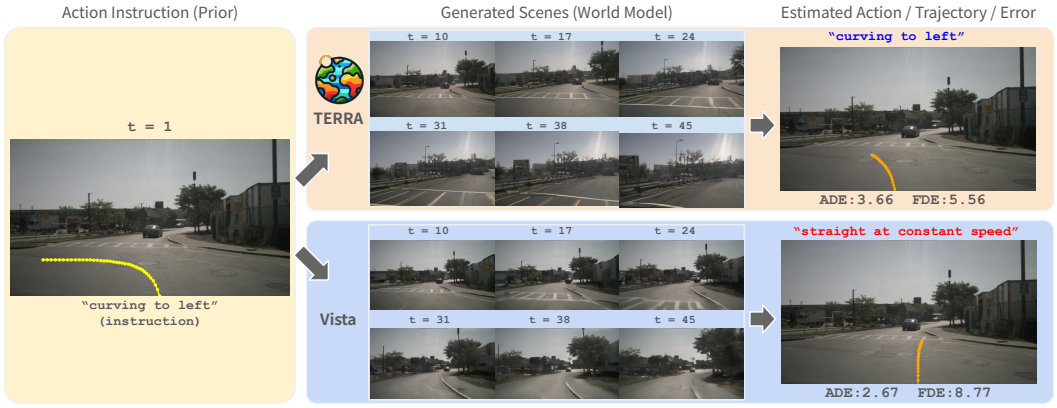

Figure 1: ACT-BENCH assesses the action controllability of world models by estimating actions, trajectories, and their deviations from the generated driving scenes using our motion estimator, ACT-ESTIMATOR. In the upper example, TERRA successfully follows the instruction to "curving to left." In contrast, the lower example illustrates that Vista fails to follow the instruction.

video length to be evaluated—and lacks comparisons with public models, further restricting reproducible benchmarking. Although Vista (Gao et al., 2024a) has made its world model public, it still depends on a closed benchmark. Beyond these concerns about non-open evaluation methods, there is also a critical lack of publicly available baseline models. While several promising approaches have been proposed, many are not publicly available or share only partial resources. This lack of open-access benchmarks and baseline models restricts broader research efforts in this domain.

To bridge these gaps, we introduce ACT-BENCH (**A**ction **C**ontrollability **T**est **Bench**mark) to evaluate action fidelity in driving world models. This benchmark is built on a nuScenes (Caesar et al., 2020)-based dataset, where each video clip is annotated with multiple trajectories and their corresponding high-level actions (e.g., "curving to right"). These annotations serve as ground truth for systematic evaluation. We also develop ACT-ESTIMATOR, a motion estimator model, to estimate action labels and reconstruct trajectories from generated scenes, which are subsequently compared to ground-truth ones to quantify how accurately a world model follows driving instructions.

Additionally, we introduce a baseline world model, TERRA. The model architecture of TERRA follows GAIA-1 (Hu et al., 2023a), but as GAIA-1 is not publicly accessible, TERRA represents the first open-access model sharing the GAIA-1's philosophy. To enhance action fidelity, TERRA is trained on three datasets: OpenDV-YouTube (Yang et al., 2024a), nuScenes (Caesar et al., 2020) and CoVLA (Arai et al., 2024). The use of a larger training dataset allows TERRA to better capture action fidelity, leveraging more annotated scenes than Vista, which was trained on two datasets, OpenDV-YouTube and nuScenes. Using the proposed benchmark and baseline model, we examine how well existing world models follow instructions. First, we confirm that our evaluator model achieves sufficient performance in assessment tasks. Our empirical studies show that TERRA outperforms Vista in action fidelity. Notably, in terms of the match rate between instructed high-level actions and their executions, Vista achieves a 30.72% match, while TERRA reaches 63.21%.

Our contributions include:

- A novel benchmark, ACT-BENCH, for evaluating action fidelity in driving world models.
- A baseline world model, TERRA, demonstrating state-of-the-art action fidelity.
- Empirical evidence that the model with state-of-the-art visual quality still falls short of faithfully following given instructions.

2 RELATED WORK

This section reviews the existing driving world models and evaluation metrics they utilized. The brief summary of these world models and metrics are summarized in Table 1.

2.1 DRIVING WORLD MODEL

World models offer agents a latent representation of the environment, enabling them to simulate potential futures and explore outcomes of various actions within this learned space (Ha & Schmidhuber, 2018; Hu et al., 2023b; LeCun, 2022; Zhu et al., 2024). The predictive capability of world models allows them to simulate and evaluate complex scenarios efficiently, in domains like representation learning (Wu et al., 2024; Bardes et al., 2024; Gupta et al., 2023; Schwarzer et al., 2021; Wu et al., 2023b), model-based reinforcement learning (Ha & Schmidhuber, 2018; Hafner et al., 2019a; 2021; 2023; Wu et al., 2023a), and model predictive control (Finn & Levine, 2017; Hafner et al., 2019b; Mendonca et al., 2023; Huang et al., 2024; Ebert et al., 2018), supporting advancements in areas such as game AI (Bruce et al., 2024; Łukasz Kaiser et al., 2020; Hafner et al., 2021; 2023; Micheli et al., 2023; Robine et al., 2023; Zhang et al., 2024; Alonso et al., 2024) and robotics (Hafner et al., 2019b;a; Huang et al., 2024; Wu et al., 2023a; Piergiovanni et al., 2019; Mendonca et al., 2023; Ma et al., 2024; Yang et al., 2024b).

Over the past two years, world models for autonomous driving have emerged and rapidly evolved, with recent advancements introducing models that emphasize different strengths: high-quality video generation (Hu et al., 2023a; Gao et al., 2024a; Jia et al., 2023; Yang et al., 2024a), consistent multi-view outputs (Wang et al., 2023; 2024b; Lu et al., 2025), and the ability to incorporate diverse conditional inputs (Hu et al., 2023a; Yang et al., 2024a; Lu et al., 2025; Wang et al., 2023; 2024b; Jia et al., 2023), such as text prompts, bounding boxes, and map information. Together, these developments enable visually realistic driving simulations across various scenarios, significantly expanding their utility for training and testing autonomous systems in more flexible and robust ways.

2.2 EVALUATION METRICS FOR DRIVING WORLD MODEL

Visual Quality Metric. A core task of driving world model is video generation. In recent years, evaluation of video generation models has commonly relied on metrics such as Fréchet Inception Distance (FID) (Heusel et al., 2017), Fréchet Video Distance (FVD) (Unterthiner et al., 2018; 2019), and CLIPSIM (Radford et al., 2021).

Action Fidelity. Evaluating how well generated driving videos adhere to action-based conditioning has been explored in prior works. DriveGAN (Kim et al., 2021) introduced a CNN-based action estimator to infer the driving action that caused a transition between frames, but its closed-source nature limits reproducibility. GenAD (Yang et al., 2024a) and Vista (Gao et al., 2024a) instead leverage monocular visual odometry (VO) to estimate the executed instruction and compare it with the intended input; however, their VO models and evaluation datasets remain unavailable, making standardized assessment difficult. Similarly, DrivingDojo (Wang et al., 2024a) employs a Structure-from-Motion (SfM) approach to reconstruct trajectories and compare them against conditioning trajectories, yet its evaluation data and conditions are not disclosed.

These methods rely solely on trajectory comparison, reducing diverse trajectory patterns to a single numerical metric. This oversimplification limits fidelity evaluation from multiple aspects. In contrast, our approach combines trajectory analysis with high-level action classification, enabling broader assessment and identifying performance differences across various driving behaviors, such as whether a model executes left turns more reliably than right turns.

3 ACT-BENCH

To establish an open benchmark for action fidelity in driving world models, we present **A**ction **C**ontrollability **T**est **Bench**mark (ACT-BENCH), a dataset with annotated actions and trajectories, along with an evaluation mechanism. ACT-BENCH also includes TERRA, a baseline model for performance comparison.

3.1 BENCHMARK DATASET

We construct a benchmark dataset designed specifically to assess how well generated frame sequences align with given trajectory instructions. This dataset leverages a subset of the validation

Table 1: **Comparison of Evaluation Metrics used for Autonomous Driving World Models.** "N/A" indicates that either the evaluation has not been conducted or that only qualitative assessment is available. Highlighted models and evaluation metrics are not publicly available.

Method	Evaluation Metrics	
	Visual Quality	Action Fidelity
DriveGAN (Kim et al., 2021)	FID, FVD	action estimator
GAIA-1 (Hu et al., 2023a)	N/A	N/A
DriveDreamer (Wang et al., 2023)	FID, FVD	N/A
WoVoGen (Lu et al., 2025)	FID, FVD	N/A
ADriver-I (Jia et al., 2023)	FID, FVD	N/A
DriveWM (Wang et al., 2024b)	FID, FVD	N/A
GenAD (Yang et al., 2024a)	FID, FVD, CLIPSIM	VO-based
Vista (Gao et al., 2024a)	FID, FVD	VO-based
DrivingDojo (Wang et al., 2024a)	FID, FVD	SfM-based

split from widely used nuScenes (Caesar et al., 2020) dataset, augmented with trajectory templates for precise conditional generation tasks.

Our dataset comprises short video segments captured from the `CAM_FRONT` camera in nuScenes. Although nuScenes contains various data modalities, such as multi-camera footage and LiDAR point clouds, we limit our focus to the front-facing camera view, as it captures the vehicle's immediate forward path—essential for action-following evaluation and sufficient for our purposes. Each video segment is paired with one or more trajectory templates, allowing a single context video to support multiple trajectory conditions. To obtain relevant video segments, we extract short intervals of 10 frames from 20-second nuScenes scenes, focusing on sequences where specific trajectory templates can be applied.

The trajectory templates span eight driving maneuver categories, each offering multiple curvature and speed options, ensuring broad coverage. Details of each category and the corresponding number of video-trajectory pairs are listed in Table 2. Each extracted interval consists of a 10-frame context video and the associated trajectory instructions.

Table 2: **The number of Video-trajectory pairs for each action category.** HS and LS represent high speed and low speed respectively.

Action	Curv L	Curv R	Starting	Stopping	Accel	Const@HS	Const@LS	Decel	**Total**
Number of Pairs	162	188	89	508	273	303	238	218	1979

We employ two filtering steps. First, we discard segment–trajectory pairs where the initial speed in the context video differs from the template's starting speed by over 10 km/h. Second, through visual inspection, we exclude any segment that involves object interactions likely to impact evaluation or generation. This process yields 1,979 carefully selected video–trajectory pairs. To support world models operating on a per-frame action-input basis, we provide trajectory instructions for each frame. We adjust each template for ideal vehicle orientation over the sequence to align naturally with each action. These multi-frame trajectories come from CoVLA dataset (Arai et al., 2024), which provides pre-processed per-frame trajectory annotations. See Appendix C for visualizations of the eight template trajectories.

3.2 Benchmark Metrics

To capture different aspects of action fidelity, we introduce two distinct metrics: *instruction-execution consistency* (IEC) and *trajectory alignment* (TA). IEC quantifies the degree of alignment between the given instructions and the executed high-level actions, while TA measures the distance between the estimated trajectory and its ground truth corresponding to the given instruction.

Instruction-Execution Consistency. Let a_j^{ins} represent the instruction provided as a prior for generating a scene with the world model to be evaluated, and a_j^{est} represent the estimated action derived from the generated scene. Both a_j^{ins} and a_j^{est} belong to the set of high-level actions, denoted as $\mathcal{A}$,

such that $a_j^{\text{ins}}, a_j^{\text{est}} \in \mathcal{A}$. For n samples, IEC can be assessed as follows:

$$\text{IEC} = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\{a_j^{\text{ins}} = a_j^{\text{est}}\} \tag{1}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, which returns 1 if a_j^{ins} and a_j^{est} match, and 0 otherwise.

Trajectory Alignment. Let $\tau^{\text{ins}} \in \mathbb{R}^{T \times d}$ denote the intended trajectory provided as a conditioning input, where T represents the number of points in the trajectory and d represents the dimensionality of each point. Similarly, let $\tau^{\text{est}} \in \mathbb{R}^{T \times d}$ denote the estimated trajectory derived from the generated video. The alignment between τ^{ins} and τ^{est} is quantified using a distance function $D : (\mathbb{R}^{T \times d}, \mathbb{R}^{T \times d}) \to \mathbb{R}$ that takes two trajectories as input and returns a scalar value, expressed as $D(\tau^{\text{ins}}, \tau^{\text{est}})$. Common choices for D include Average Displacement Error (ADE) and Final Displacement Error (FDE) (Phong et al., 2023), which provide meaningful evaluations of trajectory closeness. A smaller value indicates closer alignment.

$$\text{ADE} = \frac{1}{T} \sum_{t=1}^{T} \|\tau_t^{\text{ins}} - \tau_t^{\text{est}}\|_2 \tag{2}$$

$$\text{FDE} = \|\tau_T^{\text{ins}} - \tau_T^{\text{est}}\|_2 \tag{3}$$

3.3 ACT-ESTIMATOR

Our approach uses a model that performs high-level action classification and vehicle trajectory estimation on generated camera frames, forming the basis for our automated evaluation metric. We refer to trajectory estimation in this work as reconstruction of past vehicle positions from visual observations rather than prediction of future trajectories.

Dataset. The training dataset of ACT-ESTIMATOR is constructed independently of ACT-BENCH dataset. Each sample comprises a 4-second sequence of frames from nuScenes dataset, along with corresponding trajectory and a high-level action class label. The trajectories are derived from `ego_pose` information associated with `CAM_FRONT` sensor, and action labels are generated using a rule-based algorithm categorizing trajectories into nine classes. We also include a "Stopped" class to capture scenarios where predicted motion remains completely stationary, ensuring comprehensive coverage of potential behaviors and preserving diversity in the training data. Details on the labeling methodology are in the Appendix A.

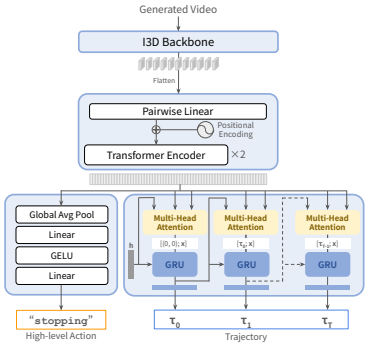

Figure 2: **Architecture of ACT-ESTIMATOR.** The model jointly performs high-level action classification and trajectory regression from an input video.

Joint Optimization. ACT-ESTIMATOR employs a multi-task framework that jointly classifies high-level vehicle movements (e.g., left or right turns) and predicts vehicle trajectories, leveraging shared representations from trajectory estimation to improve classification accuracy. The loss function is expressed as follows:

$$\mathcal{L}_{\text{total}} = \beta \cdot \mathcal{L}_{\text{classification}} + (1 - \beta) \cdot \mathcal{L}_{\text{trajectory}} \tag{4}$$

where β is a weighting factor that controls the relative importance of each task. We employ the cross-entropy loss as $\mathcal{L}_{\text{classification}}$ and smooth L1 loss (Girshick, 2015) as $\mathcal{L}_{\text{trajectory}}$.

Model Architecture. The architecture of ACT-ESTIMATOR is shown in Figure 2. ACT-ESTIMATOR utilizes proven I3D (Carreira & Zisserman, 2017) architecture as its backbone, extracting spatiotemporal features from input videos. These features are then flattened and passed through self-attention (Vaswani, 2017) layers to focus on critical parts within the video. Following this, the processed information is passed to dual-task heads, each tailored for one of the two tasks.

1. Classification Head: A multi-layer perceptron (MLP) predicts high-level action classes based on the features pooled through Global Average Pooling (Lin, 2013).

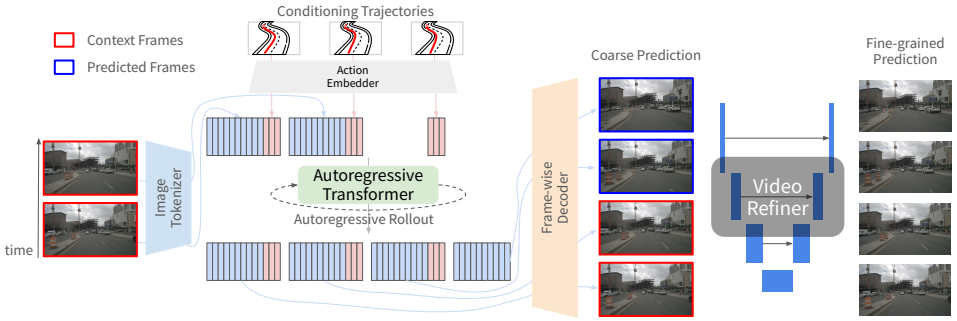

Figure 3: **TERRA's architecture overview.** TERRA follows the same design philosophy as GAIA-1 (Hu et al., 2023a) but omits text conditioning capability to maintain simplicity.

2. Trajectory Prediction Head: A GRU (Cho et al., 2014)-based unit with cross-attention (Bahdanau et al., 2015) that autoregressively predicts 2D trajectory coordinates (x, y) for each point in the path.

3.4 TERRA: A BASELINE WORLD MODEL

As a baseline world model, we introduce TERRA, an open-access world model designed for flexible trajectory control. TERRA shares the same design principles as GAIA-1 (Hu et al., 2023a), yet it is trained on three open datasets: OpenDV-YouTube (Yang et al., 2024a), nuScenes (Caesar et al., 2020) and CoVLA dataset (Arai et al., 2024). Notably, TERRA allows trajectory-based instructions to be input at each frame during conditioning, enabling precise control over the generated video. The brief architecture of TERRA is illustrated in Figure 3, and additional details are provided in the Appendix D. Table 3 reports the results of visual fidelity evaluation.

Table 3: **Comparison of visual fidelity metrics on nuScenes validation set.** For Vista, we reproduce the metric calculation and list the results in parentheses. For TERRA, we report the value computed using the same procedure.

Model	FID ↓	FVD ↓
DriveGAN (Kim et al., 2021)	73.4	502.3
DriveDreamer (Wang et al., 2023)	52.6	452.0
WoVoGen (Lu et al., 2025)	27.6	417.7
Drive-WM (Wang et al., 2024b)	15.8	122.7
GenAD (Yang et al., 2024a)	15.4	184.0
Vista (Gao et al., 2024a)	**6.9** (6.9)	**89.4** (162.1)
TERRA (Ours)	17.8	233.3

4 ESTIMATOR VALIDATION

We validate the performance of ACT-ESTIMATOR by assessing how reliably it performs against ground truth and related evaluation methods. The evaluations in the following subsections are conducted on a validation split consisting of 8,407 randomly selected samples from the dataset described in Section 3.3. This split is used exclusively for evaluation and is not included in the training process.

4.1 HIGH-LEVEL ACTION CLASSIFICATION

Figure 4 (Left) shows the results of high-level action classification, with accuracy exceeding **94%**. This highlights the model's strong capability to identify intended actions. Furthermore, ACT-ESTIMATOR maintains consistently high performance across all classes, demonstrating robust distinctions among various driving maneuvers and underscoring its reliability.

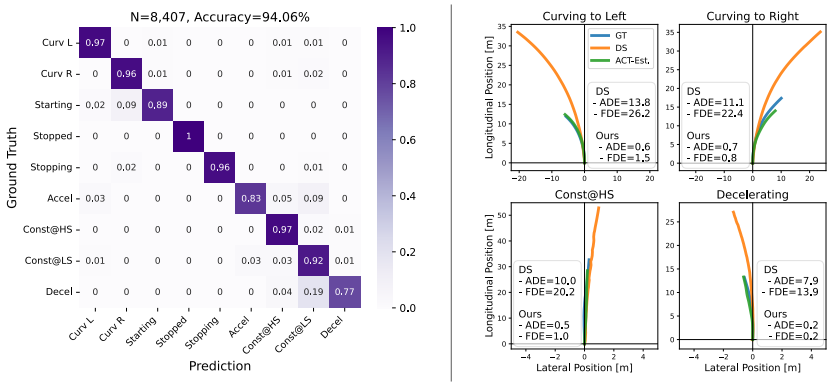

Figure 4: **Validation results of ACT-ESTIMATOR. (Left)** Confusion matrix for high-level action classification on the validation dataset. **(Right)** Examples of estimated vehicle trajectories. DROID-SLAM (DS) tends to overestimate the trajectory length while our model demonstrates higher alignment with the ground truth (GT) trajectories.

Table 4: **Fidelity of Estimated Vehicle Trajectories.** Comparison of ADE(↓) and FDE(↓) between our model and DROID-SLAM (DS), for estimated trajectories from video sequences. Our model shows consistently lower errors.

Action	Curv L	Curv R	Starting	Stopped	Stopping	Accel	Const@HS	Const@LS	Decel	**Avg.**
Ours (ADE)	**0.82**	**0.78**	**0.81**	0.76	**0.73**	**0.78**	**0.83**	**0.84**	**0.79**	**0.81**
DS (ADE)	9.91	9.51	3.25	**0.06**	7.15	7.67	6.92	8.39	9.03	7.52
Ours (FDE)	**1.61**	**1.54**	**1.61**	1.48	**1.41**	**1.52**	**1.64**	**1.64**	**1.58**	**1.59**
DS (FDE)	18.70	18.44	8.85	**0.09**	9.98	14.62	11.46	15.04	14.82	13.75

4.2 VEHICLE TRAJECTORY ESTIMATION

We measure ADE (Eq. 2) and FDE (Eq. 3) to quantify how the estimated trajectories align with their ground truths. As a baseline for comparison, we employ DROID-SLAM (Teed & Deng, 2021), a well-established SLAM (Durrant-Whyte & Bailey, 2006) approach. To adapt DROID-SLAM to our monocular setup, we combine it with Metric3D (Hu et al., 2024; Yin et al., 2023) for monocular depth estimation.

Vista (Gao et al., 2024a) and GenAD (Yang et al., 2024a) utilize an XVO (Lai et al., 2023)-based mechanism, namely inverse dynamics estimation, to infer trajectories from the generated scenes; however, the details of the mechanism are not disclosed yet (as discussed in Table 1). Due to this, we employ DROID-SLAM as the baseline even though there is a target domain gap.

Table 4 shows that our model outperforms this adapted DROID-SLAM setup on both ADE and FDE metrics, indicating superior precision in predicting vehicle trajectories. Figure 4 (Right) illustrates four estimated trajectories. DROID-SLAM tends to overestimate the trajectory length, especially in curving and high-speed scenarios. Our model demonstrates higher alignment with the ground truth trajectory, as reflected by lower ADE and FDE values.

5 ACTION CONTROLLABILITY EXPLORATION

This section explores action controllability for existing open-access world models: Vista and our model TERRA. Section 5.1 and 5.2 analyzes IEC performance and TA performance respectively.

We first summarize the process for generating action-conditioned scenes using Vista and TERRA. To effectively capture driving actions, a minimum duration of four seconds is considered essential. Vista is designed to generate sequences with a fixed input of three conditioning frames, producing 22 frames per round. Thus, we generate sequences over two rounds, resulting in a total of 44 frames (equivalent to 4.4 seconds of video). TERRA adopts the same setup as Vista for consistency.

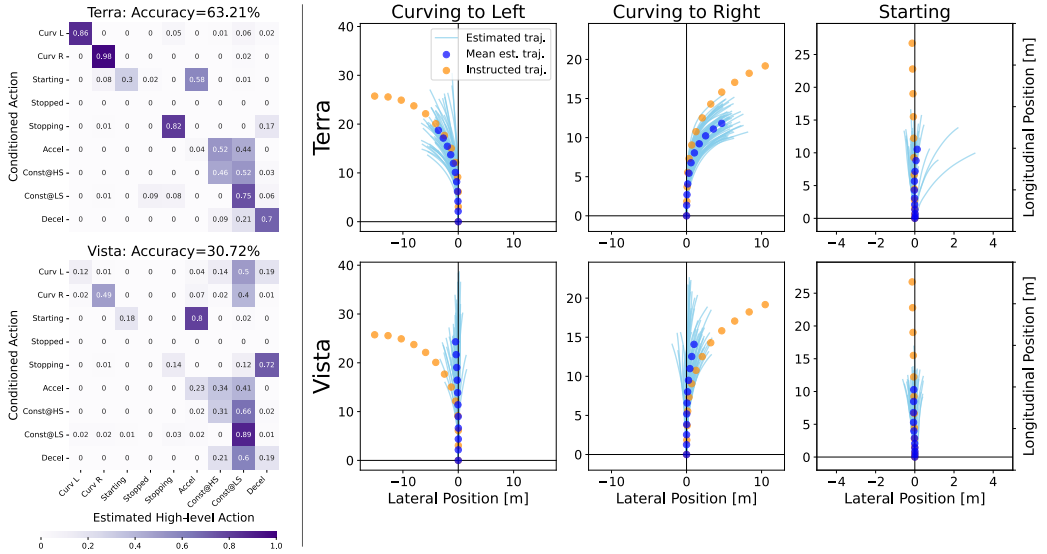

Figure 5: **Confusion Matrices and Visualized Trajectories for Vista and Terra.** (**Left**) These confusion matrices indicates that Vista struggles with curving actions, whereas TERRA achieves a higher match rate with the ground-truth actions. (**Right**) Visualized trajectories comparing estimated and instructed trajectories for different actions. Vista exhibits greater deviation from the intended trajectory, particularly in curving actions, while Terra more effectively follows the target trajectory's curvature.

Table 5: **Trajectory Alignment across Action Categories.** ADE(↓) and FDE(↓) are measured to evaluate how accurately each model generates motions that adheres to the conditioned trajectories across various high-level action categories. TERRA results in better alignment to the conditioned trajectories against Vista.

Action	Curv L	Curv R	Start	Stopping	Accel	Const@HS	Const@LS	Decel	**Avg.**
Vista (ADE)	**3.59**	3.73	3.23	3.50	6.46	5.72	2.79	6.37	4.50
TERRA (ADE)	3.67	**3.31**	**2.91**	**3.41**	**5.37**	**4.07**	**2.35**	**5.33**	**3.85**
Vista (FDE)	8.01	8.52	10.32	**3.97**	14.98	11.28	5.34	11.56	8.66
TERRA (FDE)	**6.58**	**7.05**	**9.99**	5.29	**13.53**	**8.86**	**5.23**	**10.75**	**8.05**

Action conditioning for Vista is restricted to a single action input per generation round, limiting its flexibility. In contrast, TERRA allows action conditioning on every frame, enabling trajectory inputs for each frame during the generation process. The frequency of instruction is determined by the specific constraints of each world model. As a result, each world model generates 1,979 videos using three conditioning frames and their corresponding trajectories from ACT-BENCH dataset.

5.1 INSTRUCTION-EXECUTION CONSISTENCY EVALUATION

Figure 5 (Left) illustrates the ratios of estimated action occurrences compared to their ground-truth actions for Vista and TERRA, respectively. Vista achieves a match rate of 30.72%, while TERRA achieves 63.21%. Although TERRA generally outperforms Vista across most action classes, particularly in turning maneuvers, it lags behind on "Accel" and "Const@LS," where Vista demonstrates relatively stronger performance. Nevertheless, Vista continues to struggle with other speed transitions and curved driving actions, resulting in a lower overall accuracy.

5.2 TRAJECTORY ALIGNMENT EVALUATION

Table 5 shows ADE and FDE, which reveal how closely each model's generated trajectories match the intended paths. TERRA outperforms Vista in both ADE and FDE across most high-level action

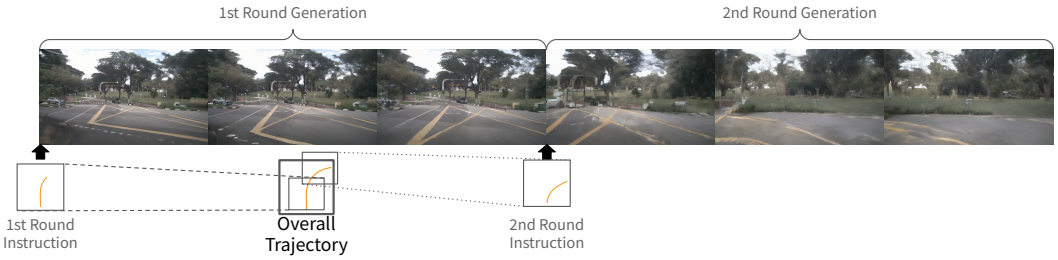

Figure 6: **An example of an abrupt, unnatural result in generation by Vista.** While the left three frames generated in the first round show nearly straight movement, the right three frames generated in the second round exhibit a significant shift to the right, resulting in a noticeably jerky motion when connected sequentially.

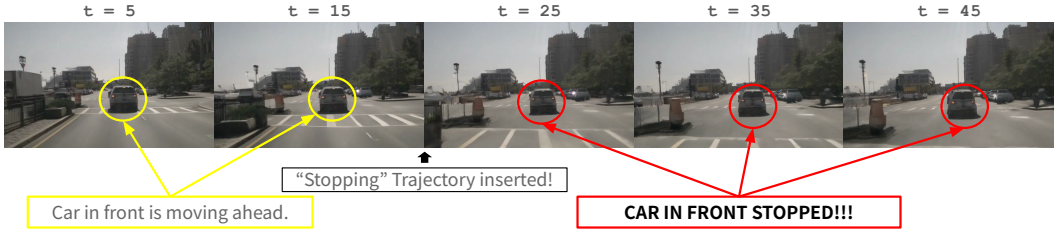

Figure 7: **An example of the *Causal Misalignment* in world models.** Leading car stops in response to the "`stopping`" instruction given to the ego vehicle, but it is expected to continue its motion independently of the ego vehicle's actions.

classes, reflecting its superiority in responding to provided trajectory. Figure 5 (Right) shows trajectory scatter plots for both models. Vista underperforms on curves, producing straighter paths, while Terra follows the instructed trajectory more closely, indicating greater proficiency at curving actions. These observations suggest that TERRA exhibits better action fidelity and controllability in response to given instructions. However, it still faces challenges in maintaining consistent travel distance.

5.3 NOTEWORTHY FINDINGS

We find two remarkable findings in the analysis of Vista and TERRA through our evaluation framework. Firstly, we observe that Vista, which allows action conditioning only at each generation round, exhibit abrupt and unnatural motion changes at round transitions (Figure 6). In contrast, no such phenomenon is observed in TERRA.

Secondly, we discover cases where actions directed at the ego-vehicle appear to inadvertently influence other agents visible in the context, even though these actions are not intended to affect them. For example, when given instructions for the ego-vehicle to gradually decelerate and stop while following a car ahead, we observe that the car in front also came to an unexpected stop (Figure 7). Such behavior, termed *Causal Misalignment*, which deviates from real-world dynamics, can pose a significant challenge when utilizing world models as simulators. Notably, we find that this phenomenon occurs in both models—highlighting the need for robust strategies to prevent ego-vehicle actions from triggering unintended effects on other agents.

6 CONCLUSION

We introduced ACT-BENCH, an open evaluation framework for action fidelity, comprising an annotated dataset, ACT-ESTIMATOR, and a baseline model, TERRA. Our findings show that while the state-of-the-art model struggled with adherence to instructions, TERRA demonstrated improved fidelity and the ability to generate diverse action-conditioned scenes. We hope ACT-BENCH fosters further research in driving world models.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=NadTwTODgC.

Hidehisa Arai, Keita Miwa, Kento Sasaki, Yu Yamaguchi, Kohei Watanabe, Shunsuke Aoki, and Issei Yamamoto. Covla: Comprehensive vision-language-action dataset for autonomous driving. *arXiv preprint arXiv:2408.10845*, 2024.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. URL http://arxiv.org/abs/1409.0473.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-JEPA: Latent video prediction for visual representation learning, 2024. URL https://openreview.net/forum?id=WFYbBOEOtv.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024a.

Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024b.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179.

Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *arXiv preprint arXiv:2405.17398*, 2024a.

Zeyu Gao, Yao Mu, Chen Chen, Jingliang Duan, Ping Luo, Yanfeng Lu, and Shengbo Eben Li. Enhance sample efficiency and robustness of end-to-end urban autonomous driving via semantic masked world model. *IEEE Transactions on Intelligent Transportation Systems*, 2024b.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020.

Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. MaskViT: Masked Visual Pre-Training for Video Prediction. 2023.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pp. 2451–2463. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution. https://worldmodels.github.io.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with Discrete World Models. 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering Diverse Domains through World Models. *arXiv preprint arXiv:2301.04104*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zachary Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. *Advances in Neural Information Processing Systems*, 35:20703–20716, 2022.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023a.

Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024.

Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Hao-Shu Fang, et al. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*, 2023b.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17853–17862, 2023c.

Weidong Huang, Jiaming Ji, Chunhe Xia, Borong Zhang, and Yaodong Yang. Safedreamer: Safe reinforcement learning with world models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tsE5HLYtYg.

Fan Jia, Weixin Mao, Yingfei Liu, Yucheng Zhao, Yuqing Wen, Chi Zhang, Xiangyu Zhang, and Tiancai Wang. ADriver-I: A General World Model for Autonomous Driving. *arXiv preprint arXiv:2311.13549*, 2023.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023.

Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. Drivegan: Towards a controllable high-quality neural simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5820–5829, 2021.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Lei Lai, Zhongkai Shangguan, Jimuyang Zhang, and Eshed Ohn-Bar. Xvo: Generalized visual odometry via cross-modal self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10094–10105, 2023.

Yann LeCun. A Path towards Autonomous Machine Intelligence. *Open Review*, 62, 2022.

M Lin. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jiachen Lu, Ze Huang, Zeyu Yang, Jiahui Zhang, and Li Zhang. Wovogen: World volume-aware diffusion for controllable multi-camera driving scene generation. In *European Conference on Computer Vision*, pp. 329–345. Springer, 2025.

Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024.

Haoyu Ma, Jialong Wu, Ningya Feng, Chenjun Xiao, Dong Li, HAO Jianye, Jianmin Wang, and Mingsheng Long. Harmonydream: Task harmonization inside world models. In *Forty-first International Conference on Machine Learning*, 2024.

Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured World Models from Human Videos. In *RSS*, 2023.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=vhFu1Acb0xb`.

Minting Pan, Xiangming Zhu, Yunbo Wang, and Xiaokang Yang. Iso-dream: Isolating and leveraging noncontrollable visual dynamics in world models. *Advances in neural information processing systems*, 35:23178–23191, 2022.

Tran Phong, Haoran Wu, Cunjun Yu, Panpan Cai, Sifa Zheng, and David Hsu. What truly matters in trajectory prediction for autonomous driving? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=nG35q8pNL9`.

AJ Piergiovanni, Alan Wu, and Michael S Ryoo. Learning real-world robot policies by dreaming. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7680–7687. IEEE, 2019.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TdBaDGCpjly.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-Efficient Reinforcement Learning with Self-Predictive Representations. 2021.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023.

Yuqi Wang, Ke Cheng, Jiawei He, Qitai Wang, Hengchen Dai, Yuntao Chen, Fei Xia, and Zhaoxiang Zhang. Drivingdojo dataset: Advancing interactive and knowledge-enriched driving world model. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14749–14759, 2024b.

Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15449–15458, 2024.

Jialong Wu, Haoyu Ma, Chaoyi Deng, and Mingsheng Long. Pre-training contextualized world models with in-the-wild videos for reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pp. 2226–2240. PMLR, 2023a.

Philipp Wu, Arjun Majumdar, Kevin Stone, Yixin Lin, Igor Mordatch, Pieter Abbeel, and Aravind Rajeswaran. Masked trajectory models for prediction, representation, and control. In *International Conference on Machine Learning*, pp. 37607–37623. PMLR, 2023b.

Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, Jun Zhang, Andreas Geiger, Yu Qiao, and Hongyang Li. Generalized predictive model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=sFyTZEqmUY.

Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9043–9053, 2023.

Lijun Yu, Jose Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *The Twelfth International Conference on Learning Representations*, 2024.

Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. Storm: Efficient stochastic transformer based world models for reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Zheng Zhu, Xiaofeng Wang, Wangbo Zhao, Chen Min, Nianchen Deng, Min Dou, Yuqi Wang, Botian Shi, Kai Wang, Chi Zhang, et al. Is sora a world simulator? a comprehensive survey on general world models and beyond. *arXiv preprint arXiv:2405.03520*, 2024.

Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB.

A DATASET CONSTRUCTION FOR ACT-ESTIMATOR

This section details the procedure for constructing the dataset used to train our ACT-ESTIMATOR. The dataset is derived from nuScenes dataset, specifically using sequences from the `CAM_FRONT` sensor. Each sequence consists of 44 frames, corresponding to approximately four seconds of video, and includes the associated trajectory data computed from `ego_pose` information. To maximize dataset size while maintaining temporal coherence, overlapping windows with a stride of one frame are applied to slice nuScenes frames into 44-frame segments. The trajectory data, representing the vehicle's position and orientation, is transformed into a local coordinate system centered on the initial frame of each segment for consistency.

High-level action labels are automatically assigned using a rule-based algorithm that categorizes trajectories into eleven predefined classes (see Table 6). The algorithm uses thresholds for various features, including changes in waypoint interval distances, trajectory curvature, and the angle between the trajectory tangent and the y-axis (0° representing straight ahead). These thresholds are empirically calibrated to balance class distribution across the dataset. This automated labeling process ensures accurate and consistent categorization of trajectories, making the dataset suitable for training and evaluating ACT-ESTIMATOR.

B ACT-ESTIMATOR

Architecture. The architecture of ACT-ESTIMATOR, as shown in Figure 2, is designed with simplicity and efficiency in mind. It combines I3D backbone, a Transformer Encoder to refine spatio-temporal features, and task-specific heads to handle high-level action classification and trajectory regression tasks effectively. This lightweight design strikes a balance between performance and computational cost, enabling robust classification capability of high-level actions and trajectory estimation while remaining computationally efficient for inference. The "**Pred class**" column in Table 6 indicates the action classes that are included in the classification task of ACT-ESTIMATOR. Notably, the classes `shifting towards left` and `shifting towards right` are excluded from the classification targets due to their very small sample sizes (each accounts for less than 1% of the dataset), which would lead to severe class imbalance and potentially degrade overall classification performance. However, these classes are still utilized in the trajectory reconstruction task.

Training procedure. ACT-ESTIMATOR is trained for 30,850 iterations on four H100 GPUs with a per-GPU batch size of 12 and gradient accumulation steps of 2, resulting in an effective batch size of 96. We use AdamW optimizer (Loshchilov, 2017) along with OneCycleLR scheduler, setting the maximum learning rate to 1.2×10^{-4}.

Table 6: Sample counts for each high-level action category in the dataset used to train the ACT-ESTIMATOR. The **Pred class** column indicates whether the category is included in the classification task, with excluded categories omitted due to class imbalance.

High-level Action Category	#Samples	Pred class
`Curving to Left (Curv L)`	7925	✓
`Curving to Right (Curv R)`	8264	✓
`Shifting towards Left`	285	
`Shifting towards Right`	353	
`Starting`	1952	✓
`Stopped`	3958	✓
`Stopping`	1809	✓
`Accelerating (Accel)`	1912	✓
`Decelerating (Decel)`	1903	✓
`Straight Const @ High Speed (Const@HS)`	9055	✓
`Straight Const @ Low Speed (Const@LS)`	8996	✓
Total	**46412**	

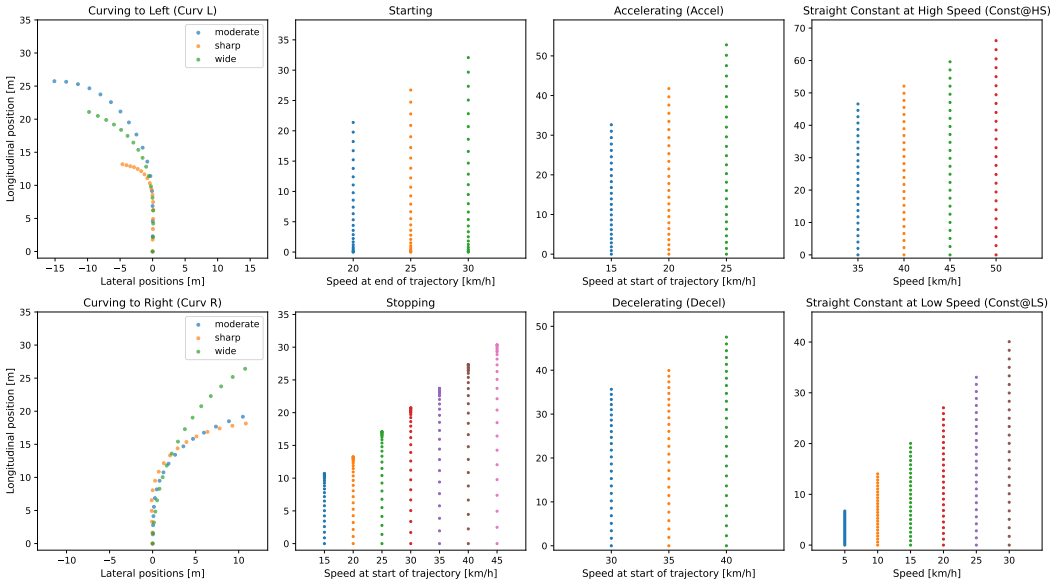

Figure 8: **Template Instruction Trajectories** used in the proposed ACT-BENCH. The trajectories represent eight categories with 32 variations in total, showcasing diverse movement patterns. Each trajectory is manually selected and associated with a corresponding scene in ACT-BENCH to ensure alignment with its intended instruction. These trajectories are carefully curated from CoVLA dataset through a manual selection process to capture representative and meaningful motion behaviors.

C TEMPLATE INSTRUCTION TRAJECTORY

Template instruction trajectories are essential for defining the ground-truth vehicle movements in ACT-BENCH and evaluating the fidelity of generated driving scenes. These trajectories act as ground-truth references, paired with context videos extracted from nuScenes dataset. Figure 8 illustrates all the instruction trajectories used in ACT-BENCH. To ensure consistency, the initial speed of each trajectory is matched with the starting speed of the vehicle in the context video.

To achieve comprehensive evaluation, we defined eight categories of instruction trajectories, such as Curving to Left (Curv L), Curving to Right (Curv R), Starting, Stopping, Accelerating (Accel), Decelerating (Decel), Straight Constant at High Speed, and Straight Constant at Low Speed. Each category includes multiple variations based on curvature, speed, or displacement, resulting in 32 distinct trajectories. This diversity ensures that ACT-BENCH effectively assesses world models' ability to generate realistic and instruction-adherent driving scenarios.

D TERRA WORLD MODEL DESIGN

As illustrated in Figure 3, TERRA is an autoregressive Transformer-based World Model that takes a sequence of discretized image tokens and a vector sequence of trajectory instructions as input, predicting the sequence of image tokens at future time steps. An Image Tokenizer is employed to convert a sequence of image frames into a sequence of discrete tokens, while a frame-wise Decoder is used to transform the sequence of discrete tokens back into image frames. These components correspond to Encoder and Decoder of an Autoencoder, respectively. The vector sequence of trajectory instructions is processed through Action Embedder for input representation. Additionally, TERRA incorporates a post-hoc Video Refiner to enhance temporal consistency and resolution of the videos predicted by the frame-wise Decoder. The following section, D.1, provides a detailed description of the core components of the world model, while Section D.2 focuses specifically on the design and functionality of the Video Refiner.

D.1 WORLD MODELING THROUGH TOKEN PREDICTION

Image Tokenization. Given that our world model is constructed using an autoregressive Transformer, which works well with discrete token representations, we opt to represent the latent codes as sequences of discrete tokens. Formally, we employ CNN based encoder network E_θ to tokenize a sequence of frames $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T) \in \mathbb{R}^{T \times H \times W \times 3}$ into discrete latent codes $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_T) \in \{1, 2, \ldots, K\}^{T \times H' \times W'}$. Here, H denotes the height of the image, W the width of the image, and K the codebook (vocabulary) size. Let H' and W' represent the downscaled dimensions, defined as $H' = H/D$ and $W' = W/D$, where D is the downscaling factor. Each $\mathbf{c}_t$ is subsequently flattened into a one-dimensional sequence of discrete tokens in raster-scan order before being input into the autoregressive Transformer.

Action-conditioning with Sequence of Trajectories. Since TERRA aims to be utilized as a simulator for autonomous driving, it is designed to accept future vehicle trajectories as input, a format commonly adopted as the output by many autonomous driving planning algorithms (Hu et al., 2023c; Jiang et al., 2023; Chen et al., 2024b; Weng et al., 2024). A trajectory is provided at each time step t, represented as a sequence $\mathbf{a}_t = (\mathbf{a}_1^t, \ldots, \mathbf{a}_L^t)$, where each point $\mathbf{a}_l^t = (x_l, y_l, t_l)$ indicates the vehicle's position in a vehicle-centered coordinate system t_l seconds into the future, with the vehicle's position at time step t as the origin.

Interleaved Inputs. During training phase, it is assumed that trajectories $\mathbf{a}_1, \ldots, \mathbf{a}_T$ are available at each time step corresponding to the discrete codes $\mathbf{c}_1, \ldots, \mathbf{c}_T$ of the T frames. In cases where corresponding trajectory data is not available, a special trajectory $\mathbf{a}^* = (\mathbf{a}_1^*, \ldots, \mathbf{a}_L^*)$ representing an empty trajectory is used for all time steps t. Latent codes and trajectories are then arranged in an interleaved format as $(\mathbf{c}_1, \mathbf{a}_1, \mathbf{c}_2, \mathbf{a}_2, \ldots, \mathbf{c}_T, \mathbf{a}_T)$. On the other hand, in the inference phase, we autoregressively predict the discrete codes $\hat{\mathbf{c}}_{T'+1}, \ldots, \hat{\mathbf{c}}_T$ of the subsequent frames using the discrete codes $\mathbf{c}_1, \ldots, \mathbf{c}_{T'}$ of the frame sequence provided as context, along with the trajectories $\mathbf{a}_1, \ldots, \mathbf{a}_T$ for $T(> T')$ time steps. Initially, we provide $(\mathbf{c}_1, \mathbf{a}_1, \ldots, \mathbf{c}_{T'}, \mathbf{a}_{T'})$ and predict the discrete code sequence $\hat{\mathbf{c}}_{T'+1}$, generating one token at a time. After predicting $N = H' \times W'$ tokens, we insert $\mathbf{a}_{T'+1}$ afterward, reformulating the sequence as $(\mathbf{c}_1, \mathbf{a}_1, \ldots, \mathbf{c}_{T'}, \mathbf{a}_{T'}, \hat{\mathbf{c}}_{T'+1}, \mathbf{a}_{T'+1})$, thereby enabling the prediction of discrete tokens for the next frame.

Before being fed into the autoregressive Transformer, the data are first transformed into embeddings of d dimensions. The token sequences representing image frames, given their discrete nature, are embedded through a learnable lookup table. In contrast, each trajectory $\mathbf{a}_t$ consists of L three dimensional vectors representing future positions and timestamps, and is therefore converted into embeddings via a multi-layer perceptron.

Learnable Positional Embedding. We apply learnable positional encodings decomposed into temporal and spatial components. The temporal positional encoding provides d-dimensional embeddings that assign unique values at each time step t for the image frames. In contrast, the spatial positional encoding assigns unique values to each of the $N + L$ tokens within the same time steps.

Training Objective. The autoregressive Transformer is trained on next token prediction task. In this process, the loss is computed only for the token sequences representing image frames, while tokens representing trajectories are excluded from loss calculation. The loss function is formalized as follows.

$$\mathcal{L}_{\text{world model}} = -\sum_{t=1}^{T} \sum_{n=1}^{N} \log p(c_{t,n} | \mathbf{c}_{<t}, c_{t,m<n}, \mathbf{a}_{<t}) \tag{5}$$

D.2 VIDEO REFINER

When employing the world model as a simulator for camera-based autonomous driving systems, it becomes necessary to decode predicted future states, represented as discrete token sequences, back into video sequences. A straightforward approach to achieve this is to utilize the decoder from the image tokenizer, which is typically adopt an Autoencoder architecture, to decode each frame individually. However, with this approach, the resulting video may exhibit low temporal consistency, even

if the quality of individual frames is high. Furthermore, when downscaling is applied to the images to reduce the sequence length input to the autoregressive model, the decoded images are also downscaled, which is suboptimal for use as a neural simulator. To address these issues and improve both image resolution and temporal consistency, we employ a latent diffusion model (LDM) (Podell et al., 2023; Rombach et al., 2022) based Video Refiner. Specifically, the Video Refiner is constructed by fine-tuning the pre-trained model of Stable Video Diffusion (SVD) Blattmann et al. (2023), an image-to-video model. In SVD, the conditioning image is first transformed into a latent representation $\mathbf{z} \in \mathbb{R}^{C_r \times H_r \times W_r}$, which is then concatenated along the channel axis with each frame of the noise $\mathbf{n} \in \mathbb{R}^{T_r \times C_r \times H_r \times W_r}$, resulting in a combined latent representation $\mathbf{n}' \in \mathbb{R}^{T_r \times 2C_r \times H_r \times W_r}$. By iteratively denoising $\mathbf{n}'$ using the U-net model D_θ, a video is generated with reference to conditioning image. On the other hand, we first decode images using the Autoencoder's decoder, then upscale it to the desired resolution, and use this sequence of frames as conditioning. Conditioning images are transformed into latent representations $\mathbf{z}' \in \mathbb{R}^{T_r \times C_r \times H_r \times W_r}$ by the VAE encoder of SVD, which are then concatenated to the noise $\mathbf{n}$. The training and inference process follows the same flow as SVD.

E Implementation Details of Terra

E.1 Hyper-parameter Settings

We set the size of the input images before passing them into the Image Tokenizer to $H = 288$ and $W = 512$. During training, we process 25 frames at a time ($T = 25$). Since we handle videos at a frame rate of 10 Hz, this corresponds to 2.5 seconds of video. For algorithms that convert image(s) into sequence(s) of discrete tokens, VQ-VAE (Van Den Oord et al., 2017) is widely known; however, we employ a more expressive approach using Lookup-Free Quantization (Yu et al., 2024). Specifically, we utilize the pre-trained weights[1] of Open-MAGVIT2 (Luo et al., 2024) as our tokenizer. The Image Tokenizer we utilize is configured with a codebook size of $K = 262,144$ and the downscaling parameter of $D = 16$. As a result, the number of discrete tokens used to represent a single image is $N = 288/16 \times 512/16 = 576$. The length of the vector sequence representing actions is $L = 6$, resulting in a sequence length during training of $(N + L) \times T = (576 + 6) \times 25 = 14,550$. The dimensionality of the embedding input to the Transformer is set to $d = 2048$. As a special trajectory $\mathbf{a}^*$ used for padding, we employ a matrix where all elements are set to -1.0:

$$\begin{bmatrix} -1. & -1. & -1. \\ -1. & -1. & -1. \\ -1. & -1. & -1. \\ -1. & -1. & -1. \\ -1. & -1. & -1. \\ -1. & -1. & -1. \end{bmatrix}$$

The values of t_l vary depending on the training dataset, as shown below:

$$t_l = \begin{cases} 0.45 + 0.5 \times (l-1), & l = 1, 2, \ldots, 6. \quad \text{(CoVLA)} \\ 0.5 \times l, & l = 1, 2, \ldots, 6. \quad \text{(nuScenes)} \end{cases}$$

In the Video Refiner, the images are first upscaled from 288×512 to 384×640. Subsequently, the latent variables compressed to $H_r = 48$ and $W_r = 80$ using the pre-trained Autoencoder from SVD are utilized. The settings for T_r and C_r are kept consistent with those of SVD, using $T_r = 25$ and $C_r = 4$.

E.2 Training Procedure

We conduct the training of the world model and the Video Refiner separately. As a preparation step for training both models, videos from OpenDV-YouTube, nuScenes, and CoVLA dataset are converted into sequences of image frames at 10 Hz. Each image frame is subsequently transformed

[1] https://huggingface.co/TencentARC/Open-MAGVIT2/blob/2f7982b9d1d4c540645a5fb2c39e5892ebea15b7/imagenet_256_B.ckpt

into a sequence of tokens using the pre-trained Image Tokenizer. For CoVLA dataset, since trajectory data in the vehicle-centric coordinate systems is available for each time step up to 2.95 seconds ahead, trajectory instruction data is created by sampling six (x, y) coordinates. Similarly, for nuScenes dataset, trajectory data in the vehicle-centric coordinate system is generated based on the `ego_pose` up to 3 seconds ahead, from which six (x, y) coordinates is sampled to create the trajectory instruction data. The data is segmented into non-overlapping chunks of 25 frames each and stored. Ultimately, OpenDV-YouTube dataset is divided into 1.67 million chunks, nuScenes dataset into 25,000 chunks, and CoVLA dataset into 0.23 million chunks.

We employ a Transformer based on Llama (Touvron et al., 2023) architecture as the world model, which is trained from a randomly initialized state. The training is conducted over 40k steps using 56 H100 80GB GPUs, with a per-GPU batch size of 1. Gradient accumulation steps is set to 4. The world model is optimized using AdamW (Loshchilov, 2017) optimizer in combination with a Cosine Decay learning rate schedule. The detailed parameter settings for the world model training are provided in the Table 7.

The training of the Video Refiner is based on the first-stage training setup of Vista (Gao et al., 2024a). In Vista, a dynamic prior is provided for the first three frames, and the initial frame is used as a conditioning frame by concatenating it with a noise tensor $\mathbf{n}$ along the channel axis. However, in our case, the goal is to refine the coarse predictions mode by the frame-wise Decoder. Therefore, we do not include a dynamic prior. Instead, we concatenate the latent variables of the coarse predictions for each frame, as predicted by the frame-wise Decoder, with the noise tensor $\mathbf{n}$ along the channel axis. The training is conducted over 800k steps on 8 H100 80GB GPUs with a per-GPU batch size of 1.

F Video Generation Settings

For video generation with Vista, we refer to the `sample.py`[2] script in Vista and use the parameter settings listed in the Table 8. However, in the case of multi-round generation with Vista, the same instructions are repeatedly used for each generation round. In our dataset, corrected target trajectories are provided for each future frame, representing the position and orientation at each timestep if the vehicle were to move faithfully along the target trajectory. Therefore, we modify the process of multi-round generation to use the trajectory assigned to the frame at the start of each round as illustrated in Figure 6.

In video generation with Terra, trajectory instructions are incorporated by appending the trajectory instruction $\mathbf{a}_t$ corresponding to each frame to the sequence of image tokens $\hat{\mathbf{c}}_t$ generated for that timestep. This approach is repeated for every frame during the generation process. To accelerate inference, video generation is performed using vLLM (Kwon et al., 2023). We conduct generation with the generation parameter settings $\texttt{temperature} = 0.9$, $\texttt{top\_p} = 1.0$ and $\texttt{top\_k} = -1$.

G Visualization

Figure 9 visualizes videos generated by Terra. While the movements do not exactly follow the instructed trajectory, they demonstrate a reasonable level of adherence to the given instructions.

In Figure 10, the first row illustrates a case where the preceding and oncoming vehicles begin moving unnaturally as the ego vehicle approaches. In contrast, the example in the second row depicts a scenario where a parallel vehicle accelerates unnaturally. In Vista, instructions are inserted at the transitions between rounds, making these transitions particularly prone to noticeable irregularities.

Figure 11 demonstrates an example where the oncoming vehicle gradually decelerates and comes to a stop in response to the ego vehicle's deceleration in the first-row example. In the second-row example, the parallel vehicle, initially moving faster than the ego vehicle, similarly decelerates and eventually stops as the ego vehicle reduces its speed.

[2] https://github.com/OpenDriveLab/Vista/blob/main/sample.py

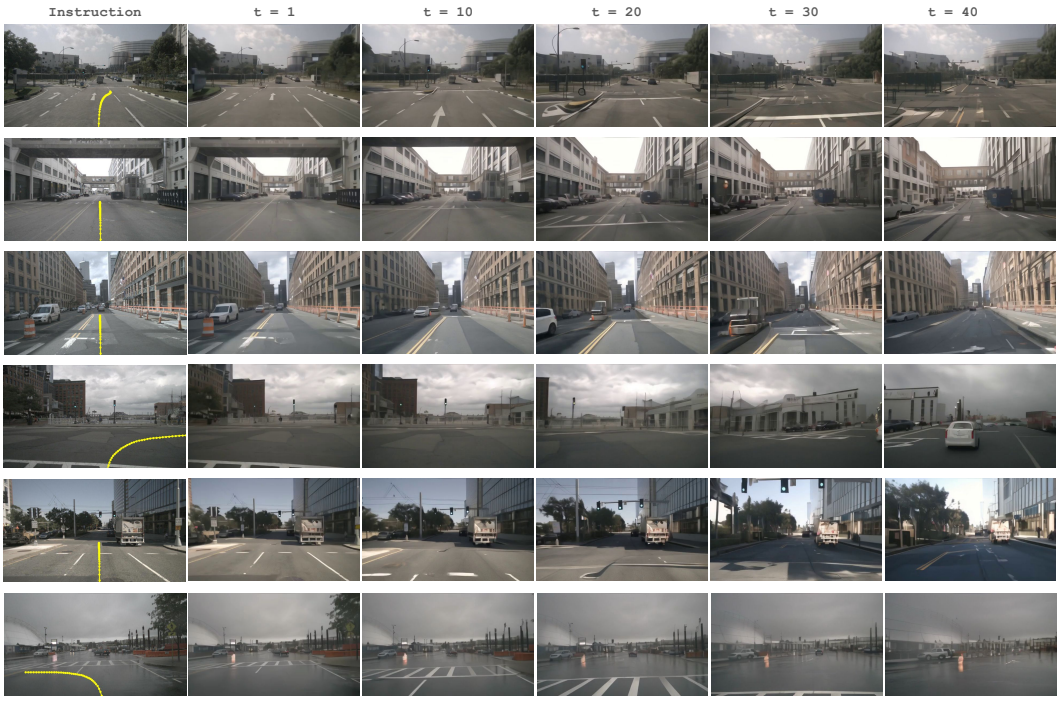

Figure 9: **Generation Capability of TERRA.** Examples of video generation results by TERRA, showing its ability to generate realistic driving scenes that adhere to specific instructions. The leftmost column visualizes the provided instruction trajectory, and the subsequent columns depict generated frames corresponding to the instruction at various time steps.

Table 7: Hyper-parameter settings for the world model training

Model Parameters	**Value**
vocab_size	262145
hidden_size	2048
intermediate_size	5632
num_hidden_layers	22
num_attention_heads	32
num_key_value_heads	4
max_position_embeddings	14550
activation_function	"relu"
attention_dropout	0.0
attn_implementation	"flash_attention_2"
pad_token_id	262144
bos_token_id	262144
eos_token_id	262144
Optimizer Parameters	**Value**
type	AdamW
learning_rate	1.0e-4
betas	(0.9, 0.999)
weight_decay	0.0
eps	1e-8
Learning Rate Scheduler Parameters	**Value**
type	cosine
num_warmup_steps	0
num_training_steps	172440

Table 8: Hyper-parameter Settings for Video Generation with Vista

Parameter	Value
action	"traj"
n_rounds	2
n_frames	25
n_conds	1
seed	23
height	576
width	1024
cfg_scale	2.5
cond_aug	0.0
n_steps	50

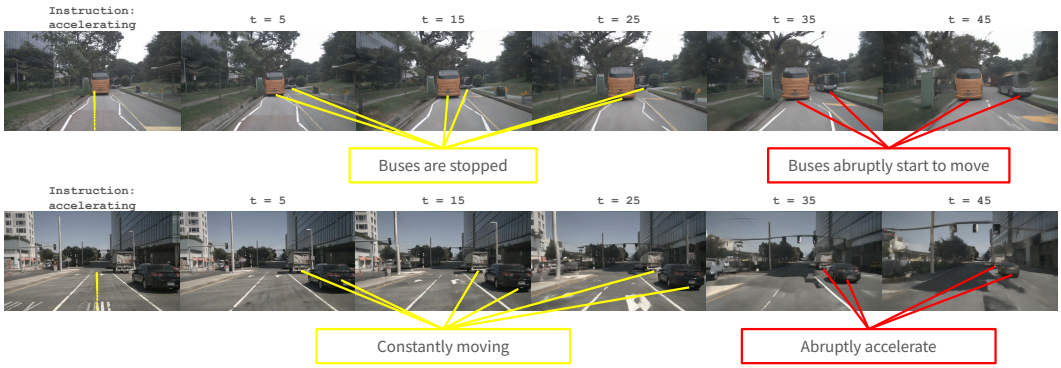

Figure 10: ***Causal Misalignment* examples in generated videos of Vista.** Example in the first row illustrates a case where the preceding and oncoming vehicle starts moving unnaturally as the ego vehicle approaches. In contrast, example in the second row depicts a scenario where a parallel vehicle accelerates unnaturally. In Vista, instructions are inserted at the transitions between rounds, making these transitions particularly prone to noticeable changes.

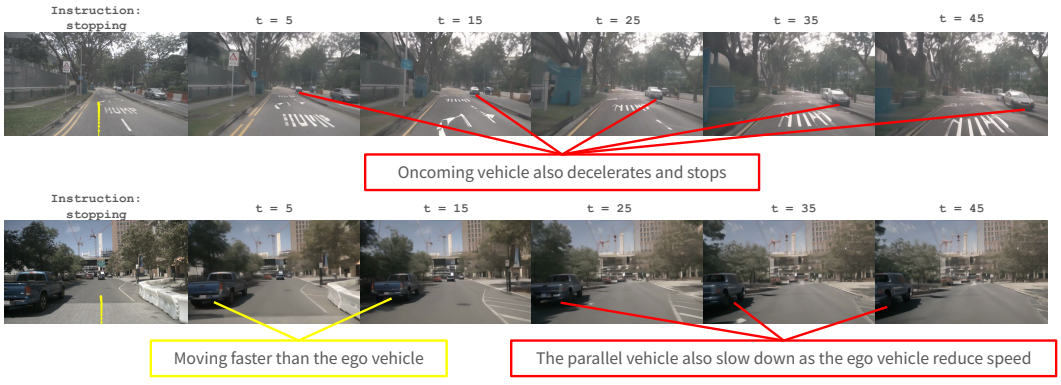

Figure 11: ***Causal Misalignment* examples in generated videos of TERRA.** The example in the first row demonstrates a scenario where the oncoming vehicle gradually decelerates and comes to a stop in response to the deceleration of the ego vehicle. In the second-row example, the parallel vehicle, initially moving faster than the ego vehicle, similarly decelerates and eventually stops as the ego vehicle reduces its speed.