Generalized Predictive Model for Autonomous Driving
Supplementary Material

Jiazhi Yang[1*] Shenyuan Gao[2,1*] Yihang Qiu[1*] Li Chen[3,1†] Tianyu Li[1] Bo Dai[1]
Kashyap Chitta[4,5] Penghao Wu[1] Jia Zeng[1] Ping Luo[3] Jun Zhang[2‡]
Andreas Geiger[4,5‡] Yu Qiao[1‡] Hongyang Li[1†]

[1] OpenDriveLab and Shanghai AI Lab [2] Hong Kong University of Science and Technology
[3] University of Hong Kong [4] University of Tübingen [5] Tübingen AI Center

[*]Equal contribution [‡]Equal co-advising [†]Project lead

Appendix

A. Discussions

To assist a better understanding of our work, we supplement discussions on intuitive questions that one may raise.

Q1. *Why do you propose the training target of the predictive model as videos?*

Video is a particularly universal and scalable target given a wealth of uncalibrated driving videos. Different from BEV representations [25, 37] that require camera extrinsic parameters and point clouds [43, 95] that are restricted by different LiDAR configurations, video prediction can be performed in a pose-agnostic manner. This characteristic offers significant advantages in scalability to more diverse data sources, which are key for the generalization ability of the learned model.

Q2. *Why do you predict multiple frames simultaneously with historical frames as input? How about alternatively using an auto-regressive design, i.e., predicting future frames one by one?*

Indeed, auto-regressive prediction can further stabilize the prediction process by leveraging conditional dependencies on previously generated frames, thereby enhancing consistency. Nevertheless, we still choose to employ a joint denoising procedure for two primary reasons. To start, diffusion models are typically computationally expensive, and our model is no exception. For videos comprised of multiple frames, predicting them auto-regressively would multiply their computational intensity, making it inefficient for implementation and deployment.

Moreover, conducting auto-regressive predictions makes it challenging to effectively apply conditions that require significant changes. Consider the scenario of a driver making a turn, which typically takes several seconds and involves a long sequence of frames. If the prediction duration is too short, the model may struggle to follow the given instructions, as it is impossible to achieve substantial changes within a single frame. Instead, it might simply continue the tendency of previously determined frames and completely disregard the provided instructions. Therefore, joint prediction also allows us to effectively apply complex controls and facilitate more coherent action generation.

Q3. *What is the criterion to prove good generalization ability of your model? How much data do we need to guarantee generalization?*

Currently, it is hard to define a specific criterion to assess the generalization ability of our predictive models for the reason that the quality judgment is subjective [58] and it is impossible to find an aligned method that is available to compare. However, through our exhaustive exploitation of public data, we have discovered that increasing the scale of the data is advantageous for zero-shot generation on existing datasets. It is also important to note that our method is easily scalable, offering opportunities to continuously enhance its generalization ability by leveraging vast amounts of unlabeled data.

Q4. *Why not evaluate models using typical video prediction metrics? What are the appropriate metrics to evaluate the performance of the driving video prediction model with multiple conditions?*

Common practices in the task of video prediction use Structural Similarity Index Measure (SSIM) [94] and a perceptual metric LPIPS [110] for quantitative evaluation. These two metrics calculate frame-wise similarities between predicted frames and corresponding ground truth frames. They are designed to assess the model's ability to *exactly* follow the recorded events. Consequently, models optimized for these metrics tend to copy certain patterns and could overfit the small datasets adopted, thereby restricting their potential for diverse future generations. This limitation is particularly problematic for predictive models in driving scenarios, where multiple futures may occur and proactive preparation is essential for each of these cases.

We sought to use the distribution-based metrics, including FVD [88] and CLIPSIM which are widely adopted by diffusion-based generation approaches [5, 92, 107]. However, for the image-to-video generation models [12, 111] in our comparison, they do not directly compose the input image as any specific video frames they generate, mainly preserving semantics and contents from input images. Thus, it becomes challenging to align the comparison settings with ours for metrics like FVD, which measures the distribution distance of consecutive frames, or CLIPSIM, which can be used to evaluate the semantic similarity between the conditional frame and generated frames. Moreover, these metrics are not perfect. For instance, FVD could be blind to unrealistic repetition and prefer small-scale motion, as discussed in [5, 7].

In short, from the existing metrics, it is hard to quantitatively evaluate the prediction abilities of a generalized model for real-world driving, which encompasses multi-modal conditions and requires temporal consistency. There still needs to be effort made to design an appropriate metric that can effectively evaluate such models.

Q5. Broader impact. *What are potential applications and future directions with the provided large-scale OpenDV-2K data and the GenAD model, for both academia and industry?*

To the best of our knowledge, OpenDV-2K is the largest available data corpus that we can collect from public sources. It significantly enhances the quantity and diversity of driving video footage in multiple dimensions, providing the research community with a massive high-quality resource for exploring open avenues in autonomous driving. In addition to video prediction, we hope our dataset can also benefit the community to enable broader applications [19, 71, 80, 90, 102].

In this work, we have demonstrated that the strong representation of GenAD can be beneficial for planning. Similarly, it is also promising to adapt it to a broader range of downstream tasks such as perception [113]. To improve the flexibility and efficiency for deployment, transferring the knowledge of generative models via distillation [51] is also worth investigating. Except for its powerful representations, the prediction futures conditioned on actions also open the opportunities for model-predictive control [30, 49] and inverse dynamics model [1, 3, 19] to enable trajectory planning, which are beyond the scope of this paper. Note that our model will be made publicly available to benefit the community and it is flexible to further fine-tune it on in-house data for the industry.

Q6. Limitations. *What are the issues with current designs, and corresponding preliminary solutions?*

It is known that the captions used for training have a great impact on generation quality [4, 13]. Currently, the context description of OpenDV-2K is automatically annotated by BLIP-2 [53]. However, we empirically find that the generated captions have two main limitations. First, the BLIP-2 captions tend to be short and plain, lacking enough details about the complicated driving scenes and becoming indistinguishable from one another. Second, the alignment between the image and caption still needs to be improved. The BLIP-2 captions are mostly centric on a single object and thus fail to include the majority of important content in the scene. In addition, affected by its fine-tuning samples [56], BLIP-2 is unaware of the state of the image observer itself. Hence, it fails to infer ego intentions, which may lead to conflicts with high-level commands. To overcome these limitations, it is promising to utilize more advanced vision-language models that have a more comprehensive understanding and text-rich description of the whole scene [23, 57, 66], and have temporal awareness [50].

We opt for SDXL [68] as our starting point to inherit its merits in high quality of visual details, large capacity of model size, and better rendering abilities of text encoders. On the other hand, we have noticed that SDXL is slow to sample and computationally expensive. Our model does suffer from that as well. However, as a pioneering work exploring how to build a generalized predictive model on internet-scale driving data, the main focus of this work is the generalization ability to diverse unseen driving scenarios instead of computation overhead. Future works may include trying faster sampling methods [62, 77, 112] and transferring our general recipe to more efficient diffusion models [75].

While there is not a silver bullet yet we hope that future work takes a deep and grounded look at these discussions, identifying what more downstream applications could be applied - and more importantly, why they work or fail. Our hope is that GenAD serves as a starting point, as the main paper argues, a generalized video pre-trained paradigm that is built on top of the largest available driving videos and excels at a wide spectrum of autonomous driving tasks.

B. Related Work

Related work is introduced below due to the limited space in the main paper.

B.1. Driving Scene Generation

Over the past few years, scene generation has gained increasing popularity due to its importance for safety-critical domains like autonomous driving. One family of works [16, 99, 103, 104] perform 3D-aware rendering for sensor simulation. In particular, GeoSim [16] augments the existing images by borrowing objects from other scenes and rendering them at novel poses. UniSim [103] creates digital twins of driving logs with manipulable foreground objects to enable close-loop simulation. However, these methods can only manipulate objects from the collected assets, and novel objects cannot be created unless further collected. Recent advancements [14, 24, 46, 55, 86, 91, 101] use diffusion models to synthesize scenes with novel content beyond the collected data. As a dual task of perception, several works simulate realistic sensor data controlled by input layouts such as 2D bounding boxes [14] and bird's-eye view (BEV) segmentation maps [46, 86, 101]. More recent works [24, 55, 91] choose 3D bounding boxes for better geometry control. These methods also have potential to serve as data engine [14, 24, 55, 86, 101, 103], *i.e.*, the simulated data can be further adopted as augmented samples to boost the performance of existing perception models. However, their control abilities are acquired from manually annotated datasets, preventing them from scaling to more unlabelled data and increasing both diversity and generalization.

Besides layout-controlled sensor simulation, another thread of progresses [25, 37, 38, 45, 91] focuses on simulating the temporal dynamics of the driving scenarios. Specifically, MILE [37] firstly introduces a model of the world incorporating the BEV representation. By imagining the world within the designed space, the world dynamics can be implicitly encoded and

the behaviors of vehicles can be interpretably decoded. This opens the opportunity for executing planning policies without having access to real observations. Differently, inspired by the advances in video generation, DriveDreamer [91] and GAIA-1 [38] propose to build a realistic world model in the form of video frames. Particularly, GAIA-1 is scaled up to about 10B model parameters on 4700 hours of in-house videos, showing highly appealing results. However, the diversity of their generation is still limited by the datasets they adopt. To be specific, the nuScenes [9] used by DriveDreamer is collected in Singapore and Boston, while GAIA-1's driving logs are recorded within London. Both of them use fixed or similar camera settings. The distribution of their data sources limits their generalization abilities to unseen scenarios, different camera poses, and other settings. Moreover, how to utilize the learned knowledge for downstream applications, *e.g*., planning, is still rarely mentioned and explored.

B.2. Video Generation and Prediction

Video generation and prediction are effective ways to model the real world. Several practices [30, 40, 105] have been made to synthesize future driving videos. With the renaissance of diffusion models [36, 83], recent progresses [65, 73, 75, 76] have demonstrated that diffusion models show a great advantage over other generative methods [27, 47, 74] in both fidelity and diversity. These advantages have also been extended to the temporal domain by numerous works in video generation [5, 29, 33, 54, 92]. Among them, many works [5, 31, 60, 89] include public driving datasets [9, 18, 26] as touchstones for their evaluation. However, none of these methods have proposed effective designs that are specialized for driving scenarios, which are known to be more complex and challenging [30] as we discussed in the main paper. In addition, due to their exclusive training strategy, the model capability is greatly limited by each small and simple dataset [9, 18, 26], hindering the generalization ability to diverse driving scenes in the real world. In contrast, we explore the first practice of building a generalized prediction model via training on large-scale driving videos in a joint manner.

B.3. Learning from Web Driving Videos

Learning the general capabilities from large-scale data has been well studied in the field of both vision and language [8, 70, 79]. It is also promising to exploit the internet-scale videos for autonomous driving. However, due to the unlabeled nature of the web data, there exist great challenges and there are only a few methods that leverage this idea to driving tasks for different purposes. SelfD [108] learns driving policies via semi-supervised learning on YouTube videos. The policy network is pre-trained with pseudo trajectories and then transferred to the target datasets via fine-tuning. Instead of directly pre-training the policy, ACO [109] introduces an action contrastive learning method to obtain action-related representations for downstream tasks. However, both SelfD and ACO rely on pseudo-labeling of trajectories or actions on vast amounts of driving videos. This could be highly sensitive to domain changes, thus compromising their reliability. More recently, PPGeo [96] proposes a fully self-supervised learning pipeline to learn a motion-aware encoder through geometric reconstruction. The encoder can be further fine-tuned to benefit downstream tasks. However, their pipeline requires separating each component into different training stages. Instead, our method directly conducts self-supervised learning via future prediction, which is more intuitive and flexible. This allows us to easily apply it to such massive and diverse uncalibrated driving videos for the first time. In addition, our predictions generate interpretable visual outputs that implicitly perform the planning process and seamlessly serve as a real-world driving simulator.

B.4. Video Datasets from the Internet

Large-scale datasets have been proven to be a core component for generalizable foundation models [70, 79]. For video tasks, collecting data in laboratories or through crowd-sourcing is a common strategy for specific tasks, such as robotics [6] and ego-centric perception [28]. However, the collection and annotation process is costly and hard to scale. Therefore, researchers have sought YouTube or similar websites as video sources as they cover diverse topics and environments, and support academic usage licenses. For example, some pioneering works manually annotate YouTube videos for action classifications [42, 69, 84], action descriptions or captions [20, 115], and hand-object intersections [22, 81]. Recently, researchers have begun to leverage alt-text [2], automatic speech recognition [63, 100, 106], original image captions [64], or paired subtitles [63, 78] to enlarge the annotation scale for video captions. With the development of foundation models, Wang *et al*. [93] employ image captioning models and language models to generate video captions. These video-text pairs have demonstrated great help for general-domain video-language pre-training. ACO [109] and SelfD [108] are the only two that collect 120 and 100 hours of driving videos from YouTube, respectively, to pre-train an encoder for policy learning (Details in Appendix B.3). In contrast, we exhaustively mine driving videos from YouTube and construct the largest driving video datasets publicly available, accumulating over 1700 hours. Besides, our videos are paired with descriptions and command labels which can be used for broader applications such as language-guided autonomous driving [15, 52, 82, 116].

C. OpenDV-2K Dataset

Our data suite, OpenDV-2K, the *largest* public driving dataset to date, contains 2059 hours of driving video along with diverse text conditions, including *contexts* and *commands*. In this section, we detail the YouTube video collection process (Appendix C.1.1), language annotation method (Appendix C.1.2 for OpenDV-YouTube and Appendix C.2 for other public datasets), more examples and analysis to illustrate the diversity of OpenDV-2K (Appendix C.1.3 and Appendix C.1.4).

C.1. OpenDV-YouTube

C.1.1 Data Collection

Data Acquisition. We first search for videos of driving tours on YouTube and select 43 video uploaders worldwide, *i.e.*, YouTubers, who continuously post high-quality driving videos. We further check the quality of videos from these YouTubers in terms of resolution, frame rate, scene transition frequency, *etc.*, resulting in 2139 high-quality front-view driving videos. We take all videos from 3 selected YouTubers as the validation set, including `Pete Drives USA`, `KenoVelicanstveni`, and `Driving Experience`, while the other videos are used for training. We illustrate the diversity of the OpenDV-YouTube in Fig. 1.

Format Conversion. To simplify the data usage for training both image and video models, We pre-process all videos into sets of consecutive frames in image format using `decord` and `opencv` packages. We sample videos with resolutions no less than 720p (*e.g.*, 1280×720 for $16:9$ videos) at 10Hz.

Data Cleaning. To ensure the quality of our dataset, we exclude non-driving frames which are commonly shown in each video and introduce unwanted noise. Specifically, we discard the first 90 seconds and the last 30 seconds for most videos to remove the channel introduction at the beginning and the subscription reminder at the end. For YouTubers with longer video introductions, we discard the first 180 or 300 seconds from their videos. We further detect and remove black frames and transition frames with the help of vision-language models. As depicted in Fig. 2, we first search for frames with phrases like `words`, `watermark`, `dark night`, `dark street`, and `blur` in their BLIP-2 [53] -generated contexts, followed by the manual quality check to determine their removal. For details on BLIP-2 descriptions, please refer to Appendix C.1.2.

C.1.2 Language Annotation

Our OpenDV-YouTube possesses two types of annotations, frame descriptions (contexts) and ego-driver commands. The context aims to benefit text-to-image learning, helping the model understand the concepts of open-world objects and scenarios, whereas the command is designed to correlate the future predictions with ego actions and further enables the language as control signals. We show some examples in Fig. 3 and introduce the annotation method below.

Frame Descriptions (Contexts). We leverage the established BLIP-2 [53] to describe the main objects or scenarios in each frame with the following prompt. The language annotations are also used in data cleaning, as mentioned in Appendix C.1.1.

```
Prompt = "Question: Describe the image of a driving scenario concisely. Answer: "
```

Table 1. **BLIP-2 Prompt for generating context of each frame**.

Driver Commands. Similar to the conventional behavior planning approach [72], we classify the commands for ego vehicle into 13 categories, *i.e.*, {`forward`, `intersection passing`, `left turn`, `right turn`, `left lane change`, `right lane change`, `left lane branch`, `right lane branch`, `crosswalk passing`, `rail passing`, `merge`, `U-Turn`, `stop/decelerate`, `deviate`}. We train an action model based on optical flow to annotate the command for the unlabeled YouTube dataset. Specifically, we leverage the pre-trained GMFlow [97, 98] to extract optical flow between adjacent frames of a driving video sequence. Taking as input both the optical flow and its distance map [114], we train a ResNet-18 [32] to classify the action of each 4s video clip. The training is conducted on the merged dataset of Honda-HDD-Action and Honda-HDD-Cause [72], which provides specified action annotations. For each type of action, we match it with multiple expressions to enrich language understanding. During training, we randomly select one text from the matched caption set for each action. The dictionary for paraphrasing is shown in Tab. 3 below.

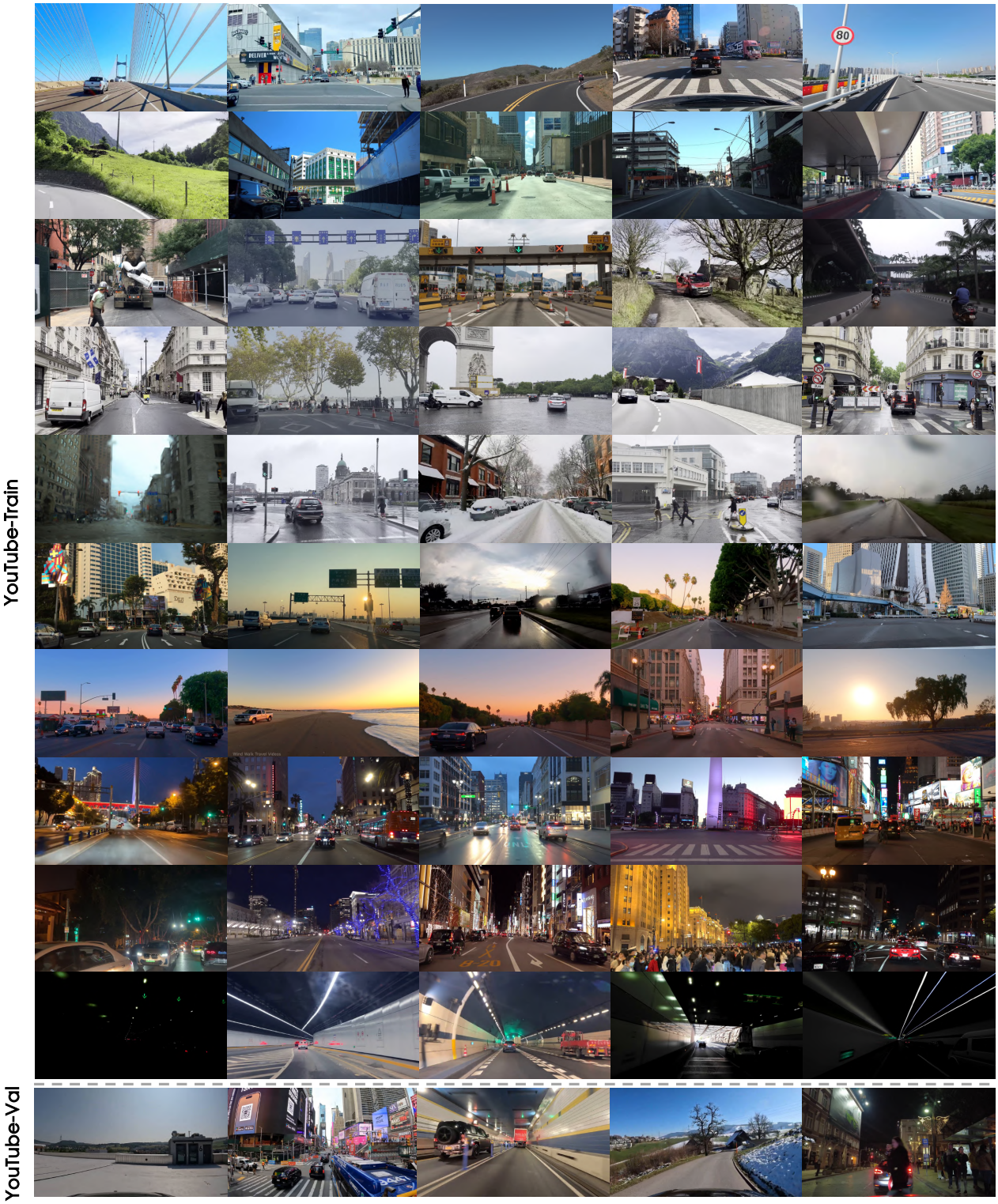

Figure 1. **Diverse video samples in OpenDV-YouTube**. We only showcase certain frames from videos due to space limits. OpenDV-YouTube covers a wide spectrum of diversity in multiple axes, including geographic locations, traffic scenarios, time periods, weather conditions, *etc*. We strictly construct the Train/Val split from different YouTubers for zero-shot evaluation.

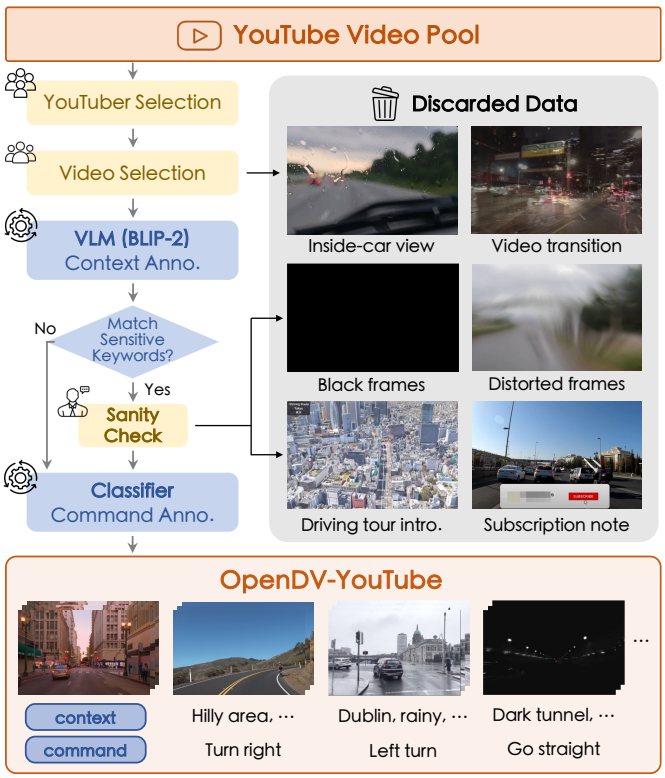

Figure 2. **Dataset construction of OpenDV-YouTube with quality check in the loop.** We collect videos from YouTubers with qualified driving videos, and dispose of those with inappropriate viewpoints or involving scene transitions. Then each frame is described with language contexts using VLM followed by keyword checks on texts, such as "words", "watermark", "dark", "blurry", *etc*. Through this process, distorted or entirely black images are wiped out. A classifier tags videos with high-level intentions as commands, incubating the final data corpus of high-quality video-text pairs being 1747 hours long.

```
command_caption_dict = {

    0:  [   "Move forward.",                    "Move steady.",                 "Go forward.",
            "Go straight.",                     "Proceed.",                     "Drive forward.",
            "Drive straight.",                  "Drive steady.",                "Keep the direction.",
            "Maintain the direction."                                                                   ],

    1:  [   "Pass the intersection.",           "Cross the intersection.",      "Traverse the intersection.",
            "Drive through the intersection.",  "Move past the intersection.",  "Pass the junction.",
            "Cross the junction.",              "Traverse the junction.",       "Drive through the junction.",
            "Move past the junction.",          "Pass the crossroad.",          "Cross the crossroad.",
            "Traverse the crossroad.",          "Drive through the crossroad.", "Move past the crossroad."  ],

    2:  [   "Turn left.",                       "Turn to the left.",            "Make a left turn.",
            "Take a left turn.",                "Turn to the left.",            "Left turn.",
            "Steer left.",                      "Steer to the left."                                    ],

    3:  [   "Turn right.",                      "Turn to the right.",           "Make a right turn.",
            "Take a right turn.",               "Turn to the right.",           "Right turn.",
            "Steer right.",                     "Steer to the right."                                   ],

    4:  [   "Make a left lane change.",         "Change to the left lane.",     "Switch to the left lane.",
            "Shift to the left lane.",          "Move to the left lane."                                ],
```

```
    5:  [   "Make a right lane change.",     "Change to the right lane.",     "Switch to the right lane.",
            "Shift to the right lane.",      "Move to the right lane."                                    ],

    6:  [   "Go to the left lane branch.",   "Take the left lane branch.",    "Move into the left lane branch.",
            "Follow the left lane branch.",  "Follow the left side road."                                 ],

    7:  [   "Go to the right lane branch.",  Take the right lane branch.",    "Move into the right lane branch.",
            "Follow the right lane branch.", "Follow the right side road."                                ],

    8:  [   "Pass the crosswalk.",           "Cross the crosswalk.",          "Traverse the crosswalk.",
            "Drive through the crosswalk.",  "Move past the crosswalk.",      "Pass the crossing area.",
            "Cross the crossing area.",      "Traverse the crossing area.",   "Drive through the crossing area.",
            "Move past the crossing area."                                                                ],

    9:  [   "Pass the railroad.",            "Cross the railroad.",           "Traverse the railroad.",
            "Drive through the railroad.",   "Move past the railroad.",       "Pass the railway.",
            "Cross the railway.",            "Traverse the railway.",         "Drive through the railway.",
            "Move past the railway."                                                                      ],

    10: [   "Merge.",                        "Merge traffic.",                "Merge into traffic.",
            "Merge into the traffic.",       "Join the traffic.",             "Merge into the traffic flow.",
            "Join the traffic flow.",        "Merge into the traffic stream.", "Join the traffic stream.",
            "Merge into the lane.",                                                                       ],

    11: [   "Make a U-turn.",                "Make a 180-degree turn.",       "Turn 180 degree.",
            "Turn around.",                  "Drive in a U-turn."                                         ],

    12: [   "Stop.",                         "Halt.",                         "Decelerate.",
            "Slow down.",                    "Brake."                                                     ],

    13: [   "Deviate.",                      "Deviate from the path.",        "Deviate from the lane.",
            "Change the direction.",         "Shift the direction."                                       ]

}
```

Table 3. **Paraphrasing dictionary for command generation**. Each index corresponds to one of the 13 actions inferred by the classifier.

C.1.3 Analyses Methods

In this section, we elaborate on the means of data analysis for OpenDV-YouTube. The analysis results are reported in the main paper and Appendix C.1.4.

Geographic Diversity Analysis. We take GPT-3.5-turbo [67] to infer the geographic information of each video from its title. We also apply handmade rules to post-process the results from GPT-3.5-turbo to deal with multiple aliases of one city or one country. The prompts are shown in Tab. 4 where `title` denotes the video title to be inferred. For simplicity, we assume that all clips of a video are taken in the same place. For videos with multiple inferred locations, we assume that all clips included in that video are uniformly distributed in these locations. For a video composed of M clips with N inferred locations, we assume there are $\frac{M}{N}$ clips taken in each site.

```
Messages = [

{ "role": "system", "content": f""" You are a helpful assistant, who is a geography expert and is also good at recog-
nizing different languages. """ },

{ "role": "user", "content": f""" Try to infer in which city or state a video is taken from its title. Please answer the
city name, the state name, and the country name in English respectively and briefly, in the following form: \

“Country: {{the name of the country}}
```

State: {{the name of the state or the province}}
City: {{the name of the city}}". \

If something cannot be inferred, fill the corresponding blank with "N/A". If there is more than one city in the video, first check if all the answers are valid, i.e. the name of cities, instead of the names of districts or towns. If there are multiple cities after checking the validity, use "," to separate different cities. \

You should also try to infer the state or province where the cities belong and fill the answer into the blank of "State". Note that you must infer the country where the video might be taken. Moreover, please discard meaningless words like "city", "country", "province" or "state" when filling in the blanks. \

The title of the video is as follows: {`title`}"""}]

Table 4. **Prompt for geographic inference of videos**.

Scenario Diversity Analysis. For scene analysis, we visualize the frequency of different scenes in frame descriptions generated in Appendix C.1.2. For analyses on weather and time period, we observe that some language hints such as "foggy" and "night" are often present in videos' titles, thus we prompt GPT-3.5-turbo [67] to infer the weather and photographed period of the video from its title. The prompt is shown in Tab. 5 where `title` denotes the video title to be inferred.

`Messages` = [

{ `"role": "system", "content": f"` You are a helpful assistant, who has a good command of multiple languages. `"},`

{ `"role": "user", "content": f"""` Try to infer in which weather and period a video is taken from its title. Please answer the weather and period in English respectively and briefly, in the following form: \

"Weather: {{the weather}}
Period: {{the period}}". \

If something cannot be inferred, fill the corresponding blank with "N/A". The weather must be one of the following: "sunny", "rainy", "foggy", "snowy", "cloudy", "storm". The period must be one of the following: "daytime", "dusk", "dawn", "nighttime". \

The title of the video is as follows: {`title`}"""}]

Table 5. **Prompt for weather and time period inference of videos**.

C.1.4 Diversity Highlights

Geographic Distribution. As indicated by the human-refined GPT inference results, YouTube videos are taken from over 244 cities in more than 40 countries, covering considerably more areas than any existing public driving datasets, as shown in Tab. 1 and Fig. 2 in the main paper. Note that the result is still *underestimated* since the geographic information may not be included in the title for some videos and cannot be inferred. Taking the two most popular areas as an example, OpenDV-YouTube contains 36.4M clips in the US, covering 40 out of 50 states, and 12.9M clips in China, covering 26 out of 34 provinces. Moreover, to test the zero-shot performance of a model in its unseen locations, our YouTube-Val subset contains videos from 3 countries that are not included in YouTube-Train, *i.e.*, `Bosna i Hercegovina`, `Denmark`, and `Hungary`. There are also videos from 1 state of the US unseen in YouTube-Train, *i.e.*, `Maine`.

Camera Settings. Considering that the online videos are sourced from different YouTubers around the globe, our dataset enjoys high diversity in photography equipment, leading to plentiful color settings, camera intrinsic parameters, and camera poses. For instance, a front-view video on a double-deck bus (see the second left picture in the last row of Fig. 1) is provided in our YouTube-Val subset while no similar cases are included in the YouTube-Train subset.

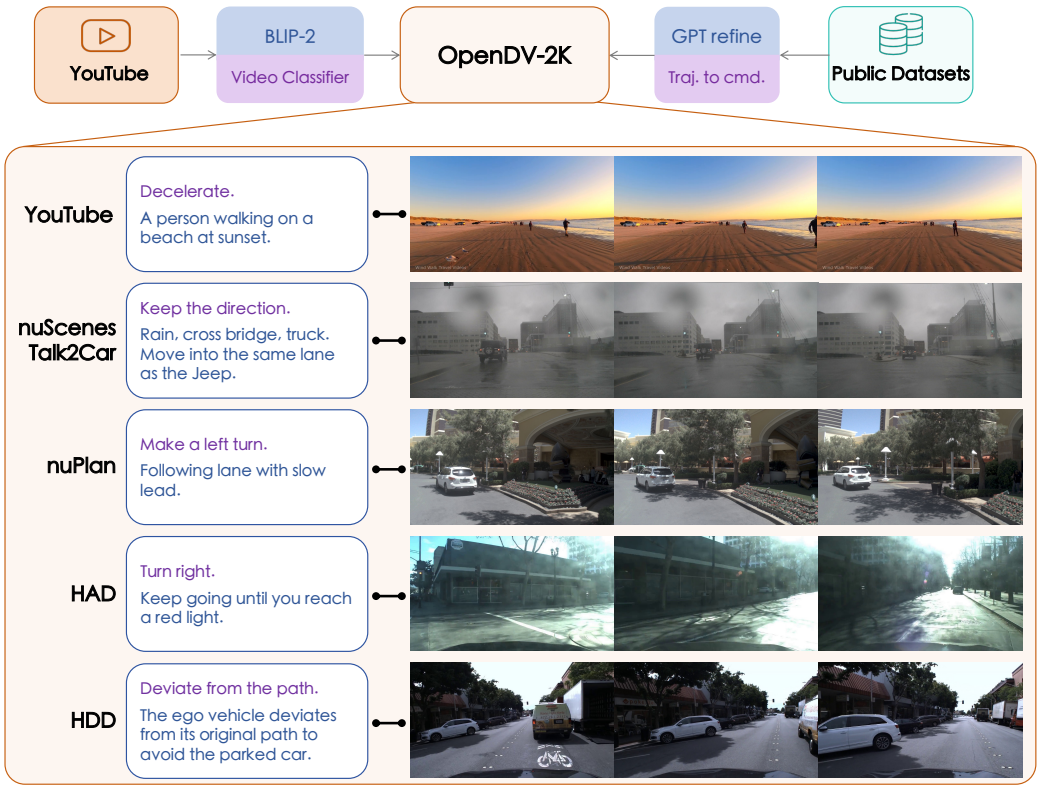

Figure 3. **Examples of language annotations for different data sources in OpenDV-2K**. We unify the paired text as command and context for all data sources after careful pre-processing. The command represents the action of the ego vehicle, whereas the context covers various aspects of information in the driving scenario. For details on how to merge public driving datasets, please refer to Appendix C.2.

Scenarios. We claim that there is sufficient data of diverse driver actions, weather conditions, photographed periods, and scenes in our OpenDV-YouTube. Results are shown in Tab. 10, Tab. 12, Tab. 11, and Fig. 4, respectively. Note that according to the analysis process in Appendix C.1.3, the diversity of scenarios in our dataset is *estimated* values since not all videos provide weather and filming periods in their titles.

Corner Cases. YouTube videos also contain corner cases and safety-critical cases. Several special cases from OpenDV-YouTube are given in Fig. 1, *e.g.*, dark tunnels with limited lighting (the leftmost and the rightmost in the 2nd row from bottom), intersections crowded with numerous pedestrians during nighttime (the 2nd right in the 3rd row from bottom), beaches at sunset (the 2nd left in the 5th row from bottom), rooftop (the leftmost in the last row), and videos captured with raindrops on the camera lens (the rightmost in the 5th row from top).

C.2. Merged Public Datasets

Though the annotations in the OpenDV-YouTube are on a large scale, annotations are subject to limited patterns. *Contexts* from BLIP-2 follow certain syntax while *commands* are generated by the paraphrase dictionary. To provide more diverse expressions of contexts and commands, we merge annotations and sensor data from existing public datasets after converting their labels into complete sentences with correct grammar and format.

C.2.1 Contexts Generation

nuScenes & nuPlan. Contexts are directly inherited from the scenario description of its belonging scenario in nuScenes [9] or nuPlan [10].

ONCE. In the metadata of ONCE [61], weather condition and filming time period are provided. These annotations are directly inherited in OpenDV-ONCE as contexts.

Honda-HAD. Diverse contexts are generated by refining and paraphrasing driving events provided by Honda-HAD [44]. The prompt for refinement and paraphrasing is as follows.

```
Messages = [

{ "role": "system", "content": f""" You are a helpful assistant. """ },

{ "role": "user", "content": f"""Generate {NUM_GEN} descriptions with exactly the same meanings as the following reference sentence, REF: {current_caption}. \

Please write these sentences concisely in diverse ways, and try to use common and simple words if possible. \

Each generated sentence denotes a short description of noteworthy elements (e.g. pedestrians, traffic lights, cars) in this driving scenario. \

There might be some typos, grammar errors, or unnatural expressions in the REF sentence, and you might need to correct these issues in the generated sentences. Each generated sentence should be correct in grammar and spelling, easy to understand, in natural and smooth expression. All sentences have the same meaning with the reference sentence REF, and the only difference is the wording. \

Your complete response is only a python list including {NUM_GEN} strings (No other text needed), each one is an example sentence with an identifier '\n' in the end. """}]
```

Table 6. **Prompt for paraphrasing contexts in Honda-HAD dataset**.

C.2.2 Commands Annotation

nuScenes & nuPlan. Since vehicle trajectory is given in nuScenes [9] and nuPlan [10], ego-vehicle commands can be easily calculated from trajectories by mathematical methods. After commands are generated, we can refer to Tab. 3 to provide diverse expressions of driver commands.

Talk2Car. Talk2Car [21] provides texts of possible human intentions for each scene in nuScenes. These annotations are inherited after they are refined by GPT-3.5-turbo to be grammatically correct and in appropriate formats. The prompt used for refinement is as follows.

```
Messages = [

{ "role": "system", "content": f""" You are a helpful assistant. """ },

{ "role": "user", "content": f""" Please correct the capitalization and punctuation issues in this sentence: "{current_caption}". The original characters and words should be exactly the same without any changes. Do not add quotation marks. """}]
```

Table 7. **Prompt for refining texts in Talk2Car dataset**.

ONCE. Behaviours of the ego-vehicle can be obtained from the change in camera pose provided in ONCE [61]. They are further converted to natural language using Tab. 3.

Honda-HAD. Since driver behaviours are not directly provided in Honda-HAD [44], we implement the video classifier trained in Appendix C.1.2 and refer to Tab. 3 to generate ego-vehicle behaviours. Moreover, Honda-HAD does provide sufficient driving advice for each scene. We use GPT-3.5-turbo to refine and paraphrase these annotations so that diverse expressions are contained in our OpenDV-2K. Prompts for GPT-3.5-turbo are as follows.

```
Messages = [

{ "role": "system", "content": f""" You are a helpful assistant. """ },
```

{ "role": "user", "content": f"""Generate {NUM_GEN} driving commands with exactly the same meanings as the following sentence: {current_caption}. \

Please write these sentences concisely in diverse ways, and try to use common and simple words if possible. Remember all sentences have the same meaning, which is an instruction or intention for the planning of the ego vehicle. \

Your complete response is only a python list including {NUM_GEN} strings (No other text needed), each one is an example sentence with an identifier '\n' in the end. """}]

Table 8. **Prompt for paraphrasing command annotations from driving advice in Honda-HAD dataset**.

Honda-HDD-Action. Honda-HDD-Action [72] contains 104 hours of videos with corresponding labels of driving commands. Since some clips begin with transitions from a completely green frame, we remove the first 30 frames from all clips. Moreover, driving events in videos with a duration too long might be inconsistent with human-annotated behaviors. Therefore, we have to discard all video clips longer than 20 seconds. Meanwhile, since in the training stage, our model takes videos no shorter than 4 seconds as input, we also remove all videos shorter than 4 seconds. Only 32 hours of videos are left after this cleaning process. For the remaining clips, we directly use the labels as driver commands and use Tab. 3 to generate command texts.

Honda-HDD-Cause. There are 12 hours of videos in Honda-HDD-Cause [72], as well as corresponding human-annotated driving behaviors and human explanations. Similar to Honda-HDD-Action, we apply the same cleaning process on Honda-HDD-Cause, with about 1 hour of cleaned driving videos preserved. To align with OpenDV-YouTube, we convert these videos into frame sets by sampling the sensor videos at 10Hz. For command annotations, causal explanations in the form of phrases in the original dataset are inherited after refining and paraphrasing by GPT-3.5-turbo. The prompts used are as follows.

elements= "sign, congestion, traffic light, pedestrian, parked car"

Messages = [

{ "role": "system", "content": f""" You are a helpful AI driving assistant, who gives commands to the ego vehicle in natural language for safe driving. \

You are provided with one of the following elements of the driving scenario, namely, {elements}. Based on the given element, produce a driving command indicating either 'stop' or 'deviate' to the ego vehicle. Specifically, sign, congestion, traffic light, crossing vehicle, and pedestrian simulates a 'stop' command, and only the parked car leads to a 'deviate' command. \

You should write {NUM_GEN} fluent, concise, and diverse sentences for each command, using common and simple words. Half of these sentences are descriptions of the action of the ego vehicle (or driver), and the other half should be imperative sentences. All sentences should have the same meaning, and the only difference is the wording. """ },

{ "role": "user", "content": current_caption}]

Table 9. **Prompt for refining driving commands in Honda-HDD-Cause dataset**.

D. Implementation Details of GenAD

D.1. Model Design

D.1.1 GenAD

GenAD is built upon 2.7B SDXL [68], which is a large-scale text-to-image generation model. We first fine-tune it in the first stage to transfer its domain knowledge to driving view synthesis. After that, we freeze the original blocks in the denoising UNet, and interleave them with our proposed temporal reasoning blocks, in total 2.5B, to allow for modeling on video sequences in video prediction pre-training. Following the original SDXL, the language conditions are encoded by two frozen

Figure 4. **Word cloud of frame descriptions for OpenDV-2K**. Only the top 500 most frequently mentioned objects, agents, or scenarios are included in the word cloud.

Driver Action	Forward	Stop	Left Turn	Right Turn	U-Turn	Lane Change	Intersection Passing
Estimated Proportion	81.39%	8.85%	1.89%	1.81%	0.27%	0.30%	5.49%

Table 10. **Driver action distribution of OpenDV-YouTube**.

Period	Daytime	Dawn	Dusk	Nighttime
Estimated Frame Count	54M	425K	2M	4M

Table 11. **Time period distribution of OpenDV-YouTube**.

Weather	Normal	Rainy	Cloudy	Foggy	Snowy	Storm
Estimated Frame Count	58M	690K	503K	284K	503K	117K

Table 12. **Weather distribution of OpenDV-YouTube**.

CLIP variants with 817M parameters, namely, CLIP ViT-L [70] and OpenCLIP ViT-bigG [17], and the projection between the pixel space and latent space is performed by a pre-trained autoencoder with 83.7M parameters. As a result, GenAD has 5.9B parameters in total. The computational complexity is 5.27 TFLOPs.

D.1.2 Extension on Action-condition Prediction

Besides the text conditions and past-frame conditions, we introduce the future trajectory of the ego vehicle as an additional condition signal to guide the denoising and therefore control the future imagination. We implement it by transforming the low-dimensional future waypoints into high-dimensional continuous embeddings [87], then projecting it with a zero-initialized linear layer into the same dimension with the text conditions $\mathbf{c}$. With zero initialization, the knowledge of future trajectory could be gradually injected into the model through the conditional cross-attention layer alongside $\mathbf{c}$, avoiding disturbing the learned prior on other conditions in the first place. It further controls the future simulation to be consistent with the ego intentions. Here the conditional future trajectory includes 6 waypoints at 2Hz.

D.1.3 Extension on Planning

Since GenAD is capable of predicting reasonable futures given past observations, it encodes past frames in a meaningful way to guide the denoising of future frames. Therefore, we take the pre-trained GenAD as a strong feature extractor to obtain the spatiotemporal representations from past frames for downstream policy learning. We only utilize the encoder part of GenAD's denoising UNet to extract intermediate semantic features rather than acquiring the noise from the decoder part. Specifically, given the past two frames and a high-level command generated in the same way as in [39, 41], the frozen GenAD encoder extracts the spatiotemporal features, which are passed to a randomly initialized multi-layer perceptron (MLP) to project them into the future trajectory of ego vehicle. The MLP is composed of 6 linear layers and 5 ReLU activations, containing only 0.8M parameters in total. The first two linear layers downsample the features channel-wise, then the features of two frames are concatenated in channel dimension and further downsampled by the third linear layer. After that, the features are average-pooled in spatial dimensions, and the resulting vector is projected to the future trajectory, which is composed of 6 waypoints at 2Hz (3s), by the last two linear layers of the MLP.

D.2. Training Details

GenAD is trained in two phases, *i.e.*, image domain transfer and video prediction pre-training. In the first stage, we fine-tune the pre-trained SDXL on per-image denoising with 2.7B trainable parameters of its denoising UNet. It is trained on 65.1M image-text pairs of OpenDV-2K. Each text condition is unified as "command, context". For some commands and contexts that are originally labeled for video sequences rather than static images, we simply associate them with all image frames included in that video sequence. We train the model for 300K iterations on 32 GPUs with a total batch size of 256 with AdamW [59]. We linearly warm up the learning rate for 10^4 steps in the beginning then keep it constant at 1.25×10^{-6}. The default GPUs in most of our experiments are NVIDIA Tesla A100 devices unless otherwise specified.

In the second stage, we train the model on video-level denoising using video-text pairs lifting it to predict the future iteratively during inference. For compute efficiency, we freeze all blocks of the fine-tuned image model and only optimize our introduced temporal reasoning blocks, resulting in 2.5B trainable parameters in this stage. To maximize the data efficiency for constructing video clips, we take each frame of a 10Hz YouTube video as a starting frame to form a 4s training sequence at 2Hz, resulting in 65M video sequences for training. For each sequence with 8 frames at 2Hz, we randomly take the leading $m \in \{1, 2\}$ frames as conditional frames and the remaining $n \in \{7, 6\}$ frames to be corrupted for video denoising, with probabilities $p \in \{0.1, 0.9\}$, respectively. We do not add noise on conditional frames since there is no need to generate *past observations*. The text condition is structured in the same way as the first stage, and we acquire the context from the middle frame of the sequence. GenAD is trained on 64 GPUs for 112.5K iterations with a total batch size of 64. The learning rate is set as 1.25×10^{-5} after 10^4 warm-up steps.

In both stages, the input frames are resized to 256×448, and the text condition $\mathbf{c}$ is dropped at a probability of $p = 0.1$ to enable classifier-free guidance [35] in sampling. Both CLIP text encoders and the autoencoder are kept frozen throughout our experiments.

For extensions on action-conditioned prediction, we fine-tune the pre-trained GenAD as well as the linear projection layer for trajectory conditions on nuScenes. We conduct training on 16 GPUs for 100K steps with a total batch size of 16. Other training protocols such as the learning rate are the same with video prediction pre-training. For extensions on planning, we adapt a lightweight MLP to project the spatiotemporal features from frozen GenAD to future trajectory. We only optimize the MLP with 0.8M trainable parameters to adapt to planning. The MLP is trained for 12 epochs with a batch size of 16 and a learning rate of 5×10^{-4}, taking only 10 minutes to converge on a single NVIDIA Tesla V100 device.

D.3. Sampling Details

Given two types of conditions including the past two frames and text, GenAD simulates 6 future frames accordingly via iteratively denoising its input latent, which starts from random Gaussian noises. The image resolution is 256×448 and the video sequence is at 2Hz. The sampling process is performed by Denoising Diffusion Implicit Models (DDIM) [83]. We use 100 sampling steps and set the scale of classifier-free guidance to 7.5. The sampling speed is 539.41 ms/step.

E. Experimental Setup

E.1. Data Preparation

We conduct extensive experiments on multiple datasets to evaluate the performance of our method. Specifically, the experiments of zero-shot transfer (Appendix F.2) are conducted on OpenDV-YouTube, Waymo [85], KITTI [26] and Cityscapes [18]. Experiments of action-condition prediction (Appendix F.3) and motion planning (Main Sec. 4.3) are established on nuScenes [9]. The results of text-to-image generation (Appendix F.1) are shown in OpenDV-YouTube. As for failure case studies (Appendix F.4, Fig. 9), there are three cases in OpenDV-YouTube (a, b, d) and one case in Waymo (c). All results are reported in the validation set, which is completely unseen in the training of GenAD. All images and video frames are resized to 256×448 before being fed into GenAD. For tasks based on video prediction, we construct 2 frames in 1s at 2Hz as conditional frames. Each video sequence is paired with text conditions composed of command and context. For zero-shot datasets, the command and context are generated by the BLIP-2 model and video classifier respectively, following the preparation of training data. For nuScenes, we generate the command from logged trajectory following [39, 41] and map them to language using dictionary in Tab. 3, and we take the scenario descriptions as the context, which are officially provided in the dataset.

E.2. Metrics

We use various metrics in multiple aspects for quantitative evaluation. These metrics include Fréchet Inception Distance (FID) [34] , Fréchet Video Distance (FVD) [88], CLIP-Similarity (CLIPSIM), Action Prediction Error, Average Displacement Error (ADE) and Final Displacement Error (FDE). For video prediction tasks, all predicted future frames are at 2Hz. We refer readers for discussions on metrics in Appendix A (Q4).

FID: It evaluates the generation quality of images, which are video frames in our experiments, by measuring the distribution distance of features between the predictions and original frames in the dataset. The features are extracted by a pre-trained Inception model. For quantitative comparison on nuScenes, FID is evaluated on 6019 generated frames and ground-truth frames. For experiments on YouTube, FID is calculated on 18000 frames from both generation and the dataset.

FVD: It measures the semantic similarity between real and synthesized videos with a pre-trained I3D action classification model [11] as the feature extractor. We evaluate 4369 video clips for the nuScenes comparison experiment, and 3000 video

clips for YouTube.

CLIPSIM: We use the CLIP ViT-L/14 [70] to evaluate the consistency and coherence of the predicted video by computing the average similarity score of CLIP features between 6 generated frames and the first conditional frame. We take 3000 video sequences for evaluation.

Action Prediction Error: For experiments of action-condition prediction on nuScenes, it measures the consistency between the input trajectory $\mathbf{w}$ and predicted future frames of GenAD. We transform the future frames into trajectory $\hat{\mathbf{w}}$ using an inverse dynamics model (IDM), which is trained on nuScenes to project a video sequence into a trajectory following the design in [48]. This metric is then calculated as the mean L2 distance between all corresponding waypoints of $\mathbf{w}$ and $\hat{\mathbf{w}}$. Here both $\mathbf{w}$ and $\hat{\mathbf{w}}$ include 6 waypoints in 2 Hz, and $\mathbf{w}$ is generated from the logged trajectory in ego coordinate.

ADE/FDE: To evaluate the performance of planning on nuScenes, we calculate the ADE and FDE between the predicted trajectory and ground-truth trajectory in an open-loop setting. Here, ADE is the mean L2 distance between all waypoints of these two trajectories, and FDE is the L2 distance between the final waypoints of them.

F. More Visualizations

F.1. Image Generation in Driving Domain

After image domain transferring, the fine-tuned image model now focuses on synthesizing images in realistic driving views. Given text prompts in Tab. 13, the corresponding generated images are shown in Fig. 5 where the generated samples greatly reflect the abundant visual details in complex and driving scenes. The ability of high-quality driving-view generation laid the foundation for simulating a realistic futuristic driving world, which is learned through video prediction pre-training.

```
1.  Take a left turn.  A city at night with a lot of lights.
2.  Move steady.  A car driving down a highway with a view of the sky.
3.  Move steady.  A car driving through a tunnel.
4.  Drive steady.  A city street at night with cars and taxis.
5.  Keep the direction.  A city street with a crosswalk and tall buildings.
6.  Go straight.  A car driving down a mountain road.
7.  Maintain the direction.  A city street with parked cars.
8.  Turn to the left.  A car driving down a city street.
9.  Steer right.  A car driving on a mountain road.
10.  Make a right turn.  A car driving down a mountain road.
11.  Drive steady.  A car driving down a city street.
12.  Move steady.  A car driving down a road in a small village.
13.  Proceed.  A car driving on a highway with a sun in the sky.
14.  Drive steady.  A car driving down a snowy road.
15.  Take a left turn.  A car driving down a hill with houses on the side.
16.  Drive through the junction.  A red car is driving down a street in Boston.
17.  Brake.  A city street with cars and tall buildings.
18.  Proceed.  A car is driving down a hill with parked cars on the side.
19.  Decelerate.  A car is driving on a highway with cars behind it.
20.  Drive straight.
21.  Move forward.
22.  Move forward.  A car driving on a mountain road.
23.  Keep the direction.  A red double decker bus driving down a city street.
24.  Stop.  A car driving on a busy street.
25.  Drive straight.  A tram on a street at night.
26.  Drive forward.  A city street with a crosswalk.
27.  Proceed.  A green light on a street with cars and pedestrians.
28.  Move steady.  A view of a highway with a city in the background.
29.  Maintain the direction.  A view of a city street with buildings and mountains in the background.
30.  Drive straight.  A city street with a lot of cars and buildings.
31.  Steer right.  A car driving down a cobblestone street in a city.
32.  Drive forward.  A car driving on a dark road at night.
33.  Move forward.  A car driving down a busy street at night.
34.  Proceed.  A car driving through a tunnel.
35.  Drive straight.  A white van driving down a city street.
36.  Maintain the direction.  A city street at night with a ferris wheel.
```

Table 13. **Prompts for image generation in Fig. 5**, in the sequential order (from left to right and top to bottom).

F.2. Zero-shot Transfer

With a strong capability on video prediction, the pre-trained GenAD can generalize to multiple unseen datasets in a zero-shot manner. In Fig. 6, we showcase multiple zero-shot video prediction results on OpenDV-YouTube. In Fig. 7, we illustrate

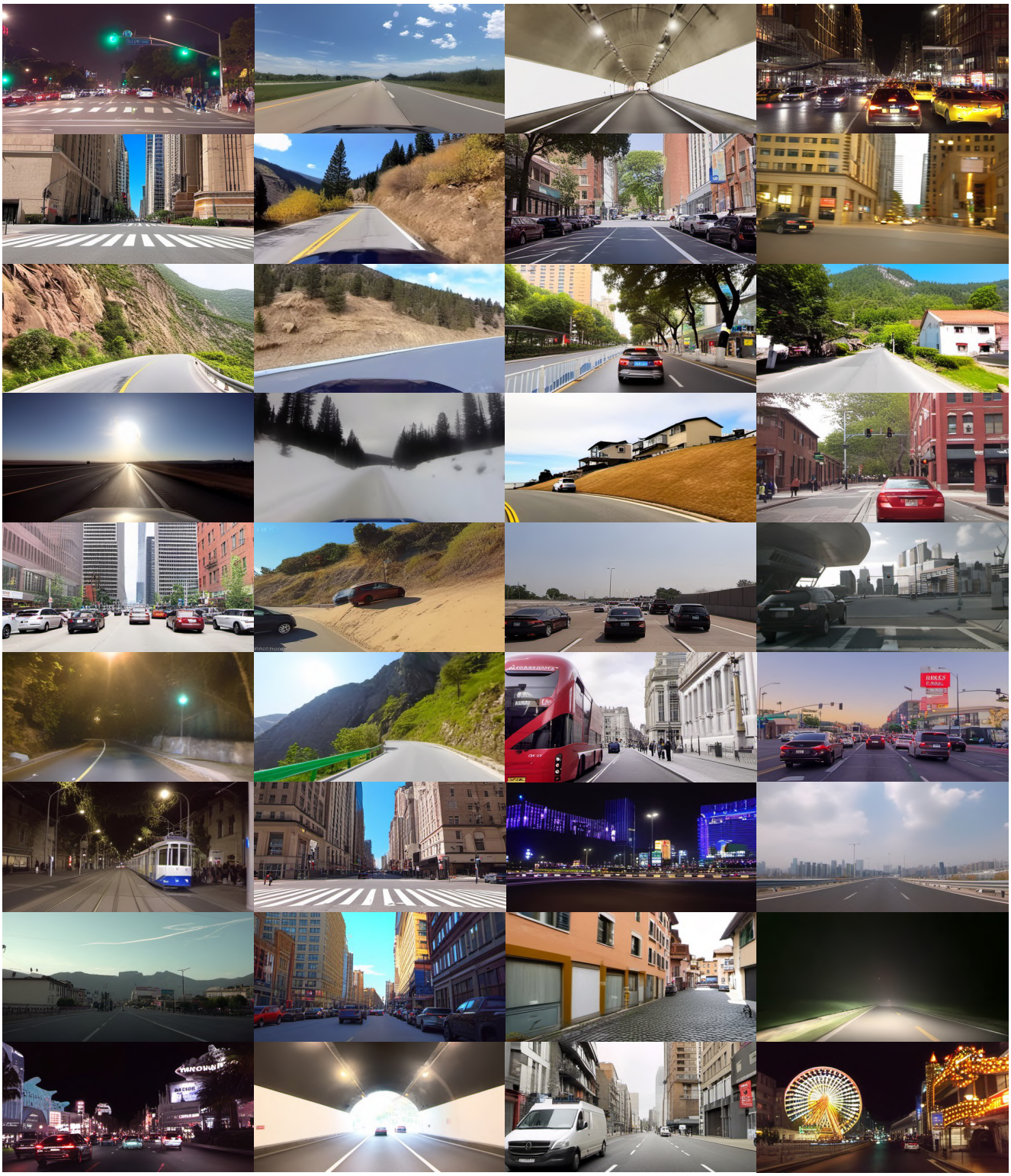

Figure 5. **Generated images by the fine-tuned image model.** Corresponding text prompts are listed in Tab. 13.

the superiority of our method by comparing it to the previous state-of-the-arts on 4 datasets, including OpenDV-YouTube, Waymo [85], Cityscapes [18] and KITTI [26].

F.3. Action-conditioned Prediction

By introducing an additional trajectory condition, the fine-tuned GenAD-act can be controlled to simulate different futures according to the input trajectory. We show four groups of action-conditioned prediction in Fig. 8. Both the input trajectory conditions (shown in the left bird's-eye view map) and imagined future frames are in 3s at 2Hz.

F.4. Failure Cases

We showcase four failure cases generated by our model in Fig. 9. The model is sometimes disturbed by misleading contexts and is not strong enough to produce high-quality human details, as discussed in the Appendix A Q6. In some cases, the motion is not smooth enough. Meanwhile, the model fails to keep up with out-of-distribution camera height for 3s, even though succeeds in the first 2 seconds. These cases are worth future explorations.

Figure 6. **Zero-shot video prediction on OpenDV-YouTube** (the YouTube-Val subset from different YouTubers with strict geofence). The corresponding text conditions from top to bottom are as follows. 1. “Move steady. A car driving down a highway with cars behind it.”, 2. “Turn to the left. A car is driving on a roof.”, 3. “Maintain the direction. A taxi driving on a city street at night.”, 4. “Drive forward. A car driving down a city street.”, 5. “Proceed. A car driving on a bridge at night.”, 6. “Steer left. A car driving on a road with trees and a blue sky.”, 7. “Slow down. A street in a city with buildings and cars.”, 8. “Go straight. A blue car driving down a city street.”, 9. “Decelerate. A car driving on a city street.”, 10. “Keep the direction. A view of a city street from the driver’s seat.”, 11. “Brake. A car driving down a street with trees and buses.”, 12. “Proceed. A car driving on a city street at night.”, 13. “Move forward. A car driving down a road near a river.”, 14. “Drive straight. A van is driving down a highway with tall buildings in the background.”.

Figure 7. **Zero-shot video prediction on public datasets compared with state-of-the-art video generation/prediction models**. Videos generated by I2VGen-XL are inconsistent with the condition frame. VideoCrafter1 appears to generate static scenarios. DMVFN suffers from huge image distortions. Meanwhile, all the other 3 models fail to generate videos when the ego vehicle should turn to the left and follow the lane (see the rightmost case in the last row). Our model manages to succeed in predictive video generation with great consistency with the conditional frames. We only show the first, third, and fifth frames from 6 predicted frames of our model due to space limits.

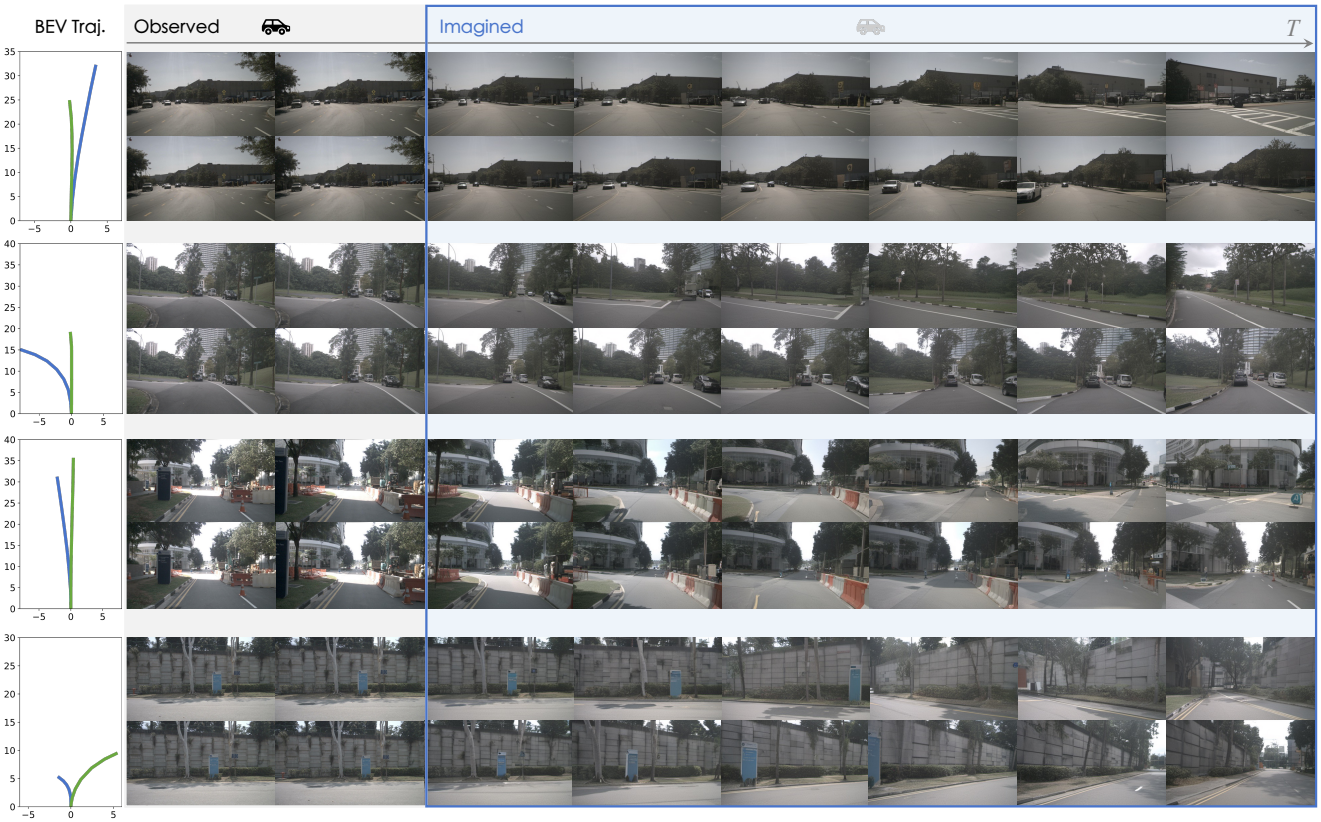

Figure 8. **Action-conditioned prediction on nuScenes**. We show four groups of video predictions for comparison, where each group is conditioned on the same two starting frames and different trajectories. In each group, the results in the first and second row are conditioned on the blue and green trajectories shown in the leftmost bird's-eye view, respectively.

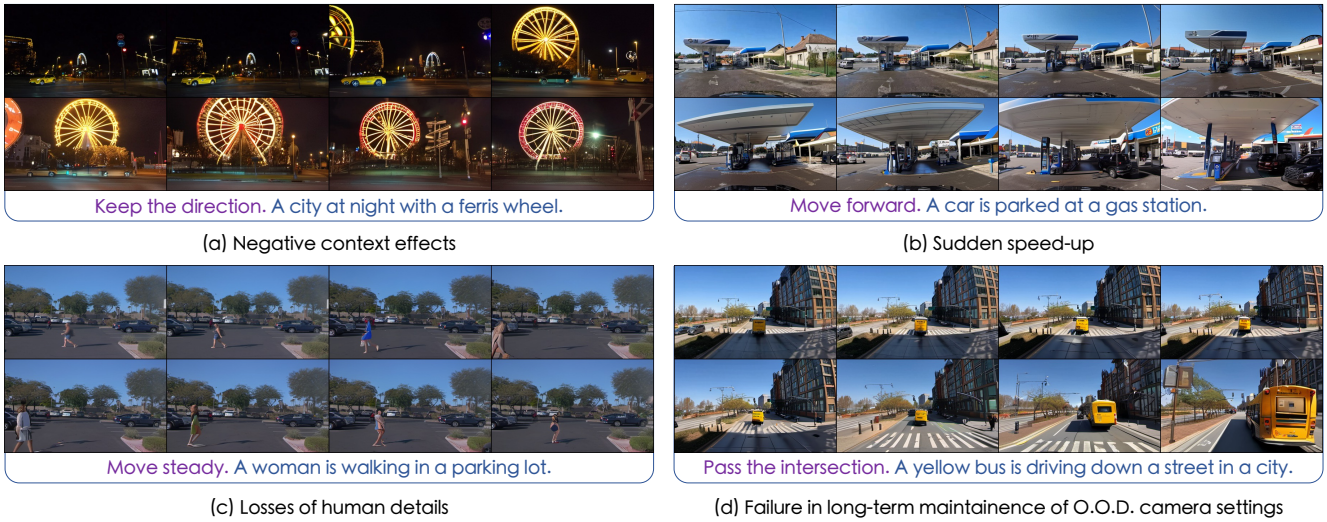

Figure 9. **Examples of failure cases**. Examples (a, b, d) are from OpenDV-YouTube, and example (c) is from Waymo. We notice that sometimes contexts exert negative impacts on generated videos since the model tends to sacrifice temporal consistency to explicitly generate the object in the context under some circumstances (see example (a)). In examples (b) and (c), the model faces challenges in generating smooth motion and human details, respectively. In example (d), the model succeeds in holding on to the out-of-distribution camera setting, *i.e.*, on a double-deck bus, for the first 4 frames. But the camera height gradually falls down as normal in the last 2 frames.

References

[1] Anurag Ajay, Seungwook Han, Yilun Du, Shaung Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. In *NeurIPS*, 2023. 3

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 4

[3] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (VPT): Learning to act by watching unlabeled online videos. In *NeurIPS*, 2022. 3

[4] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions, 2023. 3

[5] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 2, 4

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 4

[7] Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei Efros, and Tero Karras. Generating long videos of dynamic scenes. In *NeurIPS*, 2022. 2

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 4

[9] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yuxin Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 4, 10, 11, 14

[10] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric M. Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuPlan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *CVPR Workshops*, 2021. 10, 11

[11] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 14

[12] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. VideoCrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 2

[13] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 3

[14] Kai Chen, Enze Xie, Zhe Chen, Yibo Wang, Lanqing Hong, Zhenguo Li, and Dit-Yan Yeung. GeoDiffusion: Text-prompted geometric control for object detection data generation. *arXiv preprint arXiv:2306.04607*, 2023. 3

[15] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *arXiv preprint arXiv:2306.16927*, 2023. 4

[16] Yun Chen, Frieda Rong, Shivam Duggal, Shenlong Wang, Xinchen Yan, Sivabalan Manivasagam, Shangjie Xue, Ersin Yumer, and Raquel Urtasun. GeoSim: Realistic video simulation via geometry-aware composition for self-driving. In *CVPR*, 2021. 3

[17] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 13

[18] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 4, 14, 17

[19] Yilun Dai, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *NeurIPS*, 2023. 3

[20] Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*, 2013. 4

[21] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. Talk2Car: Taking control of your self-driving car. In *EMNLP*, 2019. 11

[22] David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *CVPR*, 2018. 4

[23] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. LLaMA-Adapter V2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 3

[24] Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. MagicDrive: Street view generation with diverse 3d geometry control. *arXiv preprint arXiv:2310.02601*, 2023. 3

[25] Zeyu Gao, Yao Mu, Ruoyan Shen, Chen Chen, Yangang Ren, Jianyu Chen, Shengbo Eben Li, Ping Luo, and Yanfeng Lu. Enhance sample efficiency and robustness of end-to-end urban autonomous driving via semantic masked world model. *arXiv preprint arXiv:2210.04017*, 2022. 2, 3

[26] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *IJRR*, 2013. 4, 14, 17

[27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 4

[28] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 4

[29] Jiaxi Gu, Shicong Wang, Haoyu Zhao, Tianyi Lu, Xing Zhang, Zuxuan Wu, Songcen Xu, Wei Zhang, Yu-Gang Jiang, and Hang Xu. Reuse and diffuse: Iterative denoising for text-to-video generation. *arXiv preprint arXiv:2309.03549*, 2023. 4

[30] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. MaskViT: Masked visual pre-training for video prediction. In *ICLR*, 2023. 3, 4

[31] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. In *NeurIPS*, 2022. 4

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[33] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022. 4

[34] 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. In *NeurIPS*, 2017. 14

[35] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshops*, 2021. 14

[36] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 4

[37] 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. In *NeurIPS*, 2022. 2, 3

[38] 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*, 2023. 3, 4

[39] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022. 13, 14

[40] Xiaotao Hu, Zhewei Huang, Ailin Huang, Jun Xu, and Shuchang Zhou. A dynamic multi-scale voxel flow network for video prediction. In *CVPR*, 2023. 4

[41] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *CVPR*, 2023. 13, 14

[42] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

[43] Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. Point cloud forecasting as a proxy for 4d occupancy forecasting. In *CVPR*, 2023. 2

[44] Jinkyu Kim, Teruhisa Misu, Yi-Ting Chen, Ashish Tawari, and John Canny. Grounding human-to-vehicle advice for self-driving vehicles. In *CVPR*, 2019. 11

[45] Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. DriveGAN: Towards a controllable high-quality neural simulation. In *CVPR*, 2021. 3

[46] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. NeuralField-LDM: Scene generation with hierarchical latent diffusion models. In *CVPR*, 2023. 3

[47] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[48] Lei Lai, Zhongkai Shangguan, Jimuyang Zhang, and Eshed Ohn-Bar. XVO: Generalized visual odometry via cross-modal self-training. In *ICCV*, 2023. 15

[49] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62, 2022. 3

[50] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 3

[51] Daiqing Li, Huan Ling, Amlan Kar, David Acuna, Seung Wook Kim, Karsten Kreis, Antonio Torralba, and Sanja Fidler. DreamTeacher: Pretraining image backbones with deep generative models. In *ICCV*, 2023. 3

[52] Hongyang Li, Yang Li, Huijie Wang, Jia Zeng, Pinlong Cai, Dahua Lin, Junchi Yan, Feng Xu, Lu Xiong, Jingdong Wang, Futang Zhu, Kai Yan, Chunjing Xu, Tiancai Wang, Beipeng Mu, Shaoqing Ren, Zhihui Peng, and Yu Qiao. Open-sourced data ecosystem in autonomous driving: the present and future. 2023. 4

[53] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 3, 5

[54] Xin Li, Wenqing Chu, Ye Wu, Weihang Yuan, Fanglong Liu, Qi Zhang, Fu Li, Haocheng Feng, Errui Ding, and Jingdong Wang. VideoGen: A reference-guided latent diffusion approach for high definition text-to-video generation. *arXiv preprint arXiv:2309.00398*, 2023. 4

[55] Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. DrivingDiffusion: Layout-guided multi-view driving scene video generation with latent diffusion model. *arXiv preprint arXiv:2310.07771*, 2023. 3

[56] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 3

[57] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 3

[58] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. *arXiv preprint arXiv:2310.11440*, 2023. 2

[59] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 13

[60] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. VDT: An empirical study on video diffusion with transformers. *arXiv preprint arXiv:2305.13311*, 2023. 4

[61] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Xiaodan Liang, Yamin Li, Chao Ye, Wei Zhang, Zhenguo Li, Jie Yu, Hang Xu, and Chunjing Xu. One million scenes for autonomous driving: ONCE dataset. In *NeurIPS Datasets and Benchmarks*, 2021. 10, 11

[62] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *CVPR*, 2023. 3

[63] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 4

[64] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions. In *ECCV*, 2022. 4

[65] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 4

[66] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3

[67] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. 8, 9

[68] 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. 3, 12

[69] Shengyi Qian, Linyi Jin, Chris Rockwell, Siyi Chen, and David F Fouhey. Understanding 3d object articulation in internet videos. In *CVPR*, 2022. 4

[70] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4, 13, 15

[71] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *CoRL*, 2023. 3

[72] Vasili Ramanishka, Yi-Ting Chen, Teruhisa Misu, and Kate Saenko. Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *CVPR*, 2018. 5, 12

[73] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 4

[74] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015. 4

[75] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3, 4

[76] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 4

[77] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 3

[78] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: a large-scale dataset for multimodal language understanding. *arXiv preprint arXiv:1811.00347*, 2018. 4

[79] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks*, 2022. 4

[80] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. ViNT: A foundation model for visual navigation. In *CoRL*, 2023. 3

[81] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 4

[82] 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. 4

[83] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 4, 14

[84] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 4

[85] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott M. Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 14, 17

[86] Alexander Swerdlow, Runsheng Xu, and Bolei Zhou. Street-view image generation from a bird's-eye view layout. *arXiv preprint arXiv:2301.04634*, 2023. 3

[87] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, 2020. 13

[88] 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. 2, 14

[89] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. MCVD-masked conditional video diffusion for prediction, generation, and interpolation. In *NeurIPS*, 2022. 4

[90] Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Kevin Qinghong Lin, Satoshi Tsutsui, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. In *CVPR*, 2023. 3

[91] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, and Jiwen Lu. DriveDreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023. 3, 4

[92] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, Yuwei Guo, Tianxing Wu, Chenyang Si, Yuming Jiang, Cunjian Chen, Chen Change Loy, Bo Dai, Dahua Lin, Yu Qiao, and Ziwei Liu. LAVIE: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023. 2, 4

[93] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. InternVid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023. 4

[94] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *ACSSC*, 2003. 2

[95] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nicholas Rhinehart. Inverting the pose forecasting pipeline with SPF2: Sequential pointcloud forecasting for sequential pose forecasting. In *CoRL*, 2021. 2

[96] Penghao Wu, Li Chen, Hongyang Li, Xiaosong Jia, Junchi Yan, and Yu Qiao. Policy pre-training for autonomous driving via self-supervised geometric modeling. In *ICLR*, 2023. 4

[97] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: Learning optical flow via global matching. In *CVPR*, 2022. 5

[98] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE TPAMI*, 2023. 5

[99] Yinghao Xu, Menglei Chai, Zifan Shi, Sida Peng, Ivan Skorokhodov, Aliaksandr Siarohin, Ceyuan Yang, Yujun Shen, Hsin-Ying Lee, Bolei Zhou, and Sergey Tulyakov. DisCoScene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis. In *CVPR*, 2023. 3

[100] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *CVPR*, 2022. 4

[101] Kairui Yang, Enhui Ma, Jibin Peng, Qing Guo, Di Lin, and Kaicheng Yu. BEVControl: Accurately controlling street-view elements with multi-perspective consistency via bev sketch layout. *arXiv preprint arXiv:2308.01661*, 2023. 3

[102] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023. 3

[103] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. UniSim: A neural closed-loop sensor simulator. In *CVPR*, 2023. 3

[104] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *CVPR*, 2024. 3

[105] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G. Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, and Lu Jiang. MAGVIT: Masked generative video transformer. In *CVPR*, 2023. 4

[106] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. MERLOT: Multimodal neural script knowledge models. In *NeurIPS*, 2021. 4

[107] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arXiv:2309.15818*, 2023. 2

[108] Jimuyang Zhang, Ruizhao Zhu, and Eshed Ohn-Bar. SelfD: self-learning large-scale driving policies from the web. In *CVPR*, 2022. 4

[109] Qihang Zhang, Zhenghao Peng, and Bolei Zhou. Learning to drive by watching youtube videos: Action-conditioned contrastive policy pretraining. In *ECCV*, 2022. 4

[110] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 2

[111] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2VGen-XL: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023. 2

[112] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In *NeurIPS*, 2023. 3

[113] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *ICCV*, 2023. 3

[114] Jilai Zheng, Chao Ma, Houwen Peng, and Xiaokang Yang. Learning to track objects from unlabeled videos. In *ICCV*, 2021. 5

[115] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 4

[116] Yunsong Zhou, Linyan Huang, Qingwen Bu, Jia Zeng, Tianyu Li, Hang Qiu, Hongzi Zhu, Minyi Guo, Yu Qiao, and Hongyang Li. Embodied understanding of driving scenarios. *arXiv preprint arXiv:2403.04593*, 2024. 4