OpenViCA: Video Continuation for Automotive Driving Scenes by Streamlining and Fine-Tuning Open Source Models with Public Data

Björn Möller[1], Zhengyang Li[1], Malte Stelzer[1], Thomas Graave[1],
Fabian Bettels[1], Muaaz Ataya[1] and Tim Fingscheidt[1]

***Abstract*—Recent successful video generation systems that predict and create realistic automotive driving scenes from short video inputs assign tokenization, future state prediction (world model), and video decoding to dedicated models. These approaches often utilize large models that require significant training resources, offer limited insight into design choices, and lack publicly available code and datasets. In this work, we address these deficiencies and present OpenViCA, an open video continuation system for automotive driving scenes. Our contributions are: Unlike several earlier works for video generation, such as GAIA-1, we provide a deep analysis of the three components of our system by separate quantitative and qualitative evaluation: Image tokenizer, world model, video decoder. Second, we purely build upon powerful pre-trained open-source models from various domains, which we fine-tune by public automotive data (BDD100K) on GPU hardware at academic scale. Third, we build a coherent video continuation system by streamlining interfaces of our components. Fourth, due to public availability of the underlying models and data, we allow full reproducibility. Finally, we publish our code and models at https://github.com/ifnspaml/OpenViCA. For an image size of 256x256 at 4 fps, we predict realistic driving scene videos frame-by-frame from any two prompted frames.**

***Index Terms*—video generation, automotive, open source.**

I. Introduction

Video generation models [1] have improved significantly due to recent advances in large language models (LLMs) [2], [3] and generative AI [4], enabling the creation of realistic visual content from limited input data. Some successful world model-based systems [5], [6], which learn structured representations of dynamic environments, enable the generation of long, temporally consistent video sequences through autoregressive next-token prediction. Such systems typically operate in three stages: A tokenizer (TOK) encodes the input into discrete representations, a world model (WM) predicts a future token sequence, and finally a video decoder (VDEC) reconstructs the predicted token into video frames [5].

In autonomous driving, video generation offers a scalable approach to improve safety [7], [8] by simulating diverse driving scenarios, including rare and harmful events such as evasive maneuvers, addressing the scarcity of critical training data [9]. Despite the demonstrated potential of video generation models such as GAIA-1 [5] and GAIA-2 [10], the lack of open-source implementations or training code, sparse description of architectures, and non-public datasets limit reproducibility. In contrast, *we propose OpenViCA, an open video continuation system for automotive driving scenes, see Fig.1, based purely on open-source models and public data, allowing for full reproducibility.*

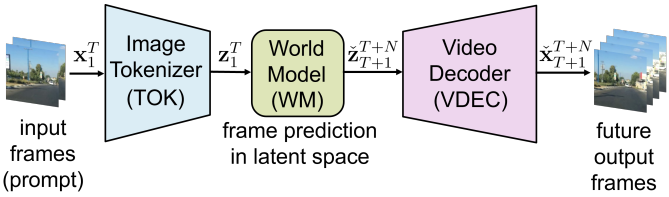

Fig. 1: **Proposed video generation system**: It consists of an image tokenizer, encoding T input frames $\mathbf{x}_1^T$ into a latent representation of discrete tokens $\mathbf{z}_1^T$, a world model then predicting N subsequent frames as latent image tokens $\check{\mathbf{z}}_{T+1}^{T+N}$, and a video decoder, generating N output frames $\check{\mathbf{x}}_{T+1}^{T+N}$. Models are pre-trained, open-source, and then fine-tuned on public automotive data.

Open-source foundation LLMs [2], [3], having learned temporal prediction from vast corpora of text data, provide a powerful basis for WMs in video generation. Extending them to multi-modal token prediction using image, video, and text data, enables them to model spatio-temporal structures. However, training such a WM is demanding with respect to training time, dataset size, and computational resources. Therefore, we build upon the pre-trained 7B parameter `LWM` [11] model and employ low-rank adaptation (LoRA) [12] to efficiently fine-tune it. For our image tokenizer and decoder, we also leverage a pre-trained `VQGAN` [13] model. However, both these open-source models were originally trained on general-domain data causing deficiencies for automotive video generation. Accordingly, we transfer them to the automotive domain by fine-tuning on public automotive data.

We utilize the large public driving dataset BDD100K [14] to fine-tune our TOK on its images, and our VDEC and WM on its videos. We also evaluate on the official image data splits. To process the dataset's video content with limited GPU memory, we have to streamline all components towards a coherent system in terms of predicted token sequence length (WM), frame rate (WM), token per image (TOK), and input frame resolution (all).

In this paper, we first build a video continuation system for automotive driving scenes (OpenViCA) and provide insights

[1]Institute for Communications Technology, Technische Universität Braunschweig, Schleinitzstr. 22, 38106 Braunschweig, Germany {bjoern.moeller, zhengyang.li, malte.stelzer, thomas.graave, f.bettels, m.ataya, t.fingscheidt}@tu-bs.de

into its components by qualitative and quantitative evaluation. We investigate loss architectures for image tokenizer and image decoder fine-tuning, evaluate the entire system, and investigate the world model's top-k selection hyperparameter, controlling its creativity. Second, we build upon general-purpose open-source models and fine-tune them on public automotive driving data, using limited GPU resources. Third, we describe streamlining of individual open-source system components. For full reproducibility, we publish our training and inference code.

II. Related Work

a) Image Autoencoder Models: Autoencoder (AE) models learn an efficient representation of data by encoding the input into a lower-dimensional representation and decoding the original data as accurately as possible. Incorporating convolutional layers [15], regularization [16], or probabilistic inference [17], image AEs are widely used to compress images into a compact latent representation for efficient processing in subsequent models. Introducing quantization into the latent space [18], [19] results in discrete representations that enhance image generation and interpretability. Following leading video generation approaches [5], [11], we fine-tune a `VQGAN` [19] to encode patches of input frames into discrete tokens, which are then processed by our world model.

b) World Models: World models (WMs) estimate missing information and predict future states of dynamic environments [20], supporting applications as reinforcement learning [21], long-sequence prediction [11], and controllable video generation [6]. In automotive video generation, early work explored an LSTM-based WM [22]. More recent approaches employ an autoregressive transformer as WM to perform next-token prediction over discrete visual tokens [5], [23], [24]. In contrast, diffusion-based WMs predict videos by iteratively denoising continuous latent representations conditioned on frames, actions, maps or bounding boxes [10], [25]–[27], with some works integrating LLMs for trajectory planning [9] or action priors [27]. While diffusion approaches provide high temporal consistency [28], [29], recent discrete next-token prediction strategies have shown strong performance for long-horizon temporal modeling [24], [30], which is particularly relevant for driving scenarios. Nevertheless, their application to this domain remains limited, hindered by the lack of publicly available code [5] and datasets [5], [24], limited analysis of architectural design choices [5], [23], [24] and substantial computing demand [5], [23], [24]. *To address these shortcomings, we choose a discrete next-token prediction strategy for our WM and employ a pre-trained open-source autoregressive transformer [11], adapt it to the automotive domain using public data, and, in contrast to prior works, publish our inference and training code alongside model weights to ensure reproducibility.*

c) Video Generation Models: Video generation models synthesize temporally consistent video sequences from input prompts. While GAN-based [31] and autoregressive models [32] show good results, recent video diffusion models (VDMs) [1] further enhance temporal consistency and quality in generated videos. However, training such VDMs requires extensive hardware resources. *Accordingly, we decide for a lower complexity attractive frame-by-frame video continuation approach by extending our fine-tuned `VQGAN` image decoder with temporal input context and fine-tune it on video data.*

III. OpenViCA

Following Fig. 1, our proposed video generation system predicts N future frames $\check{\mathbf{x}}_{T+1}^{T+N} = (\check{\mathbf{x}}_t)$, with $\check{\mathbf{x}}_t \in \mathbb{I}^{H\times W\times 3}$, of automotive driving scenes based on an input sequence $\mathbf{x}_1^T = (\mathbf{x}_t)$ of T initial images, where $\mathbf{x}_t \in \mathbb{I}^{H\times W\times 3}$, with height H, width W, 3 color channels, and gray values normalized to the range $\mathbb{I} = [-1, 1]$. Also, a fixed text prompt is utilized for this video continuation task, see Fig. 2. In the following, we describe the system components, the open-source models used, and our approach of streamlining and fine-tuning those models to automotive data.

A. System Components

a) Image Tokenizer: The image tokenizer (TOK) in Fig. 2 converts a preprocessed image $\mathbf{x}_t$ to a latent representation $\mathbf{z}_t = (\mathbf{z}_{t,\nu}) \in \mathbb{R}^{n\times d}$ of $n = 16 \times 16 = 256$ discrete tokens $\mathbf{z}_{t,\nu}$. It consists of an encoder (ENC), which encodes an image into n latent space representations of an image patch, and a vector quantizer (VQ), which outputs token vectors $\mathbf{z}_{t,\nu} \in \mathbb{R}^d$ from a learnable codebook $\mathbf{CB} \in \mathbb{R}^{K\times d}$ with K being the codebook size and codevector dimension d. We deploy an ENC and VQ from a `VQGAN` [19] as image tokenizer (TOK).

b) World Model: Based on prompted image tokens $\mathbf{z}_1^T = (\mathbf{z}_t)$ of T initial frames and a text prompt $c_1^M = (c_m)$ consisting of M text tokens, the world model (WM) predicts the discrete tokens of N future frames, $\check{\mathbf{z}}_{T+1}^{T+N} = (\check{\mathbf{z}}_t)$. Since the prompted image tokens $\mathbf{z}_t = (\mathbf{z}_{t,\nu})$ are discrete, a vector-index mapping converts $\mathbf{z}_1^T$ to a codebook index sequence $k_1^{T\cdot n'} = (k_\nu)$, with indices $k_\nu \in \mathcal{K} = \{0, ..., K\}$, and patch position $\nu \in \{1, ..., T \cdot n'\}$ of these codebook vectors. Also, end-of-image tokens ($k_{\tau\cdot n'} = 0, \quad \tau \in \{1, 2, ..., T\}$), with $n' = n{+}1 = 257$, are inserted as structural information after each block of n codebook indices that represent a prompted image. Focusing on video continuation, we fix the text prompt to "`You are a helpful assistant. USER: Generate a video of driving vehicles. ASSISTANT: <VISION>`", which is byte-pair encoded to M text tokens c_1^M. Here, `<VISION>` marks the beginning of the codebook index sequence $k_1^{T\cdot n'} = (k_\nu)$ of T tokenized initial ground-truth frames. From its concatenated inputs $(c_1^M,\ k_1^{T\cdot n'})$, the WM predicts N future frames in an autoregressive loop, selecting the next image token in each of the $N \cdot n'$ total iterations. The next image token is selected by first predicting a probability distribution and then sampling from the top-k most probable entries.

c) Video Decoder: The video decoder (VDEC) generates N frames $\check{\mathbf{x}}_{T+1}^{T+N} = (\check{\mathbf{x}}_t)$ from the world model's temporal predictions $\check{\mathbf{z}}_{T+1}^{T+N} = (\check{\mathbf{z}}_t)$ one frame at a time, using a bidirectional input context window over three frames. Consequently, our VDEC has only one-frame algorithmic latency (lookahead) during decoding. We deploy a 3D CNN as VDEC, derived from

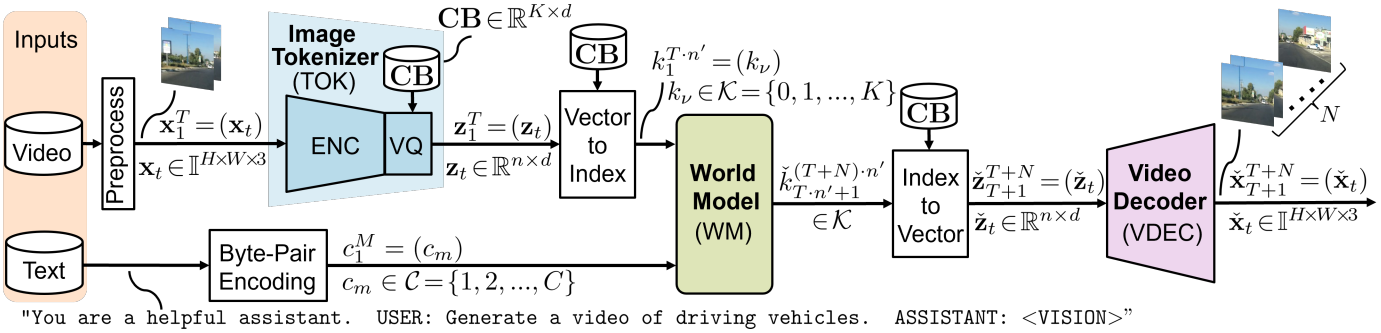

Fig. 2: **Inference** of the proposed **video generation system**: A sequence of N future frames $\check{\mathbf{x}}_{T+1}^{T+N}$ is predicted based on T initial input images $\mathbf{x}_1^T$ and a fixed text prompt c_1^M. The tokenizer encodes the prompted image sequence $\mathbf{x}_1^T$ into a discrete token sequence $\mathbf{z}_1^T$. The world model predicts future tokens $\check{\mathbf{z}}_{T+1}^{T+N}$ for the video decoder to generate the corresponding image sequence $\check{\mathbf{x}}_{T+1}^{T+N}$. Details of the image tokenizer (TOK, i.e., ENC and VQ) are shown in Fig. 3.

our fine-tuned 2D VQGAN image decoder (DEC) by 3D central inflation [33].

B. Open Source Models

a) VQGAN: The vector-quantized generative adversarial network (VQGAN) [19] integrates vector quantization [18], autoencoders [17], and GANs [34] to generate high-quality images with compact latent representations. Our approach builds on an open-source general-purpose VQGAN model [13] originally trained for 2.5M steps on diverse real-world images at 256×256 resolution. The TOK module (ENC + VQ) has 59.2M parameters, about 20% of GAIA-1's tokenizer, and the DEC block contains 87M. *However, deploying it for front-facing camera videos of automotive driving scenes, the model exhibits limited reconstruction quality.*

b) LWM: The large world model (LWM) [11] is a 7B-parameter multimodal autoregressive decoder-only transformer based on LLaMA-2 [3] capable of processing sequences up to 1M tokens of both video and text modalities with 32 decoder blocks. As our WM, we adopt the open-source LWM-Chat-1M model, originally trained on text and four vision-language tasks with VQGAN-encoded inputs [13] and 495B tokens from diverse web datasets [35], [36]. *However, no dedicated driving video dataset containing recordings from forward-facing cameras was utilized. Also, the task of predicting future latent frames based on initial latent frames was not addressed in LWM training. Moreover, LWM was trained with 256 tokens per image defining the image tokenizer output size and thus limits the video frame resolution, which we have to streamline to fit to the BDD100K automotive dataset.*

C. Our Streamlining and Fine-Tuning

a) Streamlining: Before separately fine-tuning the pre-trained models, we streamline the components within the system for processing automotive data with limited hardware resources. While the LWM is capable of processing long token sequences yet with highly parallelized hardware, in practice, sequence length is limited by available GPU memory. The open-source LWM-Chat-1M was originally trained on video frames quantized to 256 tokens each and a frame rate of 4 fps, both of which we retain for our fine-tuning. On a single Nvidia A100 GPU with 80GB of VRAM, we were able to predict image tokens 4 seconds into the future during fine-tuning, yielding a total predicted sequence length of $N \cdot n' = 4{,}112$ tokens, which is the product of $N = 4$ fps$\cdot$ 4 $s = 16$ frames, and $n' = n+1 = 257$ image tokens including a manually inserted end-of-image token. Given that automotive datasets typically provide frames at a resolution of 1.280×720 pixels, the 256-token per-frame constraint imposed by the world model presents a substantial challenge for tokenizing spatial features across the entire scene. To address this, we first select an image tokenizer (as part of VQGAN) that allows for a relatively high input image size of $H \times W \times 3 = 256 \times 256 \times 3$ producing $\frac{H=256}{16} \times \frac{W=256}{16} = 256$ image patch tokens. To grasp a meaningful section of the automotive scene, we downscale the original BDD100K image $\mathring{\mathbf{x}}_t$ by a factor of 0.5 in height and width using bilinear interpolation [37]. This consequently increases the density of spatial detail, which adds difficulty in learning precise reconstructions during VQGAN fine-tuning. The streamlined system takes a normalized image crop $\mathbf{x}_t$ of size $H \times W \times 3 = 256 \times 256 \times 3$ from the image center as input, modeling the relevant portion of the driving scene.

Also, for later VQGAN fine-tuning (shown in Fig. 3), a teacher network is used to distill relevant information into the discrete tokens $\mathbf{z}_t$ by self-supervised learning (SSL). To match the dimensions for the pre-trained VQGAN's quantized representations $\mathbf{z}_t$ and the teacher's encoded latent representation $\tilde{\mathbf{z}}_t^{\mathrm{SSL}}$, a fully connected adapter layer $\mathrm{FC}(384)$ is inserted.

b) Fine-Tuning the Tokenizer and Image Decoder: We fine-tune the image tokenizer (TOK) jointly with the image decoder (DEC) as parts of a VQGAN model utilizing a combination of various loss functions, see Fig. 3. As the TOK compresses each image independently, we fine-tune on an *image* training dataset $\mathcal{D}_{\mathrm{BDDimg/5s}}^{\mathrm{train}}$ (see Table I) and present the process for a single raw image $\mathring{\mathbf{x}}_t$, which is first downscaled, cropped, and normalized, yielding our ground-truth training image $\mathbf{x}_t$.

The TOK and DEC optimization can be expressed as total

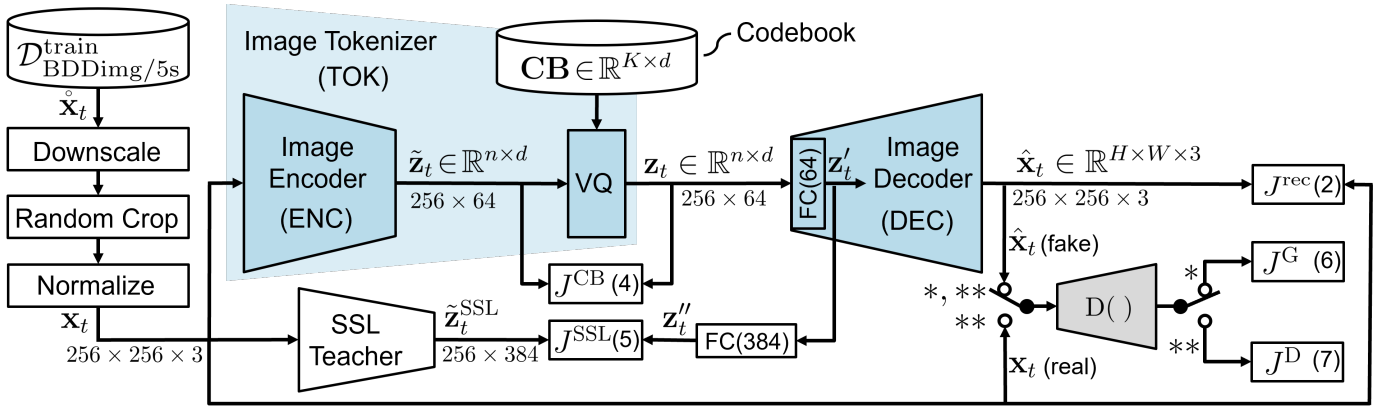

Fig. 3: **Fine-tuning** of the VQGAN **image tokenizer** and **image decoder** for a single training image $\mathbf{x}_t$ (selectors in positions $*$, as drawn). The image tokenizer consists of an encoder (ENC) and a vector quantizer (VQ) and is jointly trained with the image decoder (DEC), as part of a VQGAN model. ENC and DEC are optimized by a combination of reconstruction loss J^{rec}, GAN loss (generator component J^{G}) and self-supervised learning loss J^{SSL}. The VQ's codebook $\mathbf{CB}$ and ENC are additionally optimized with a codebook loss J^{CB}. The **discriminator** $\text{D}(\,)$ is alternatingly optimized on the same batch (selector positions $**$), using discriminator loss J^{D}.

loss with hyperparameters λ:

$$J^{\text{total}} = J^{\text{rec}} + \lambda^{\text{CB}} J^{\text{CB}} + \lambda^{\text{SSL}} J^{\text{SSL}} + \lambda^{\text{G}} J^{\text{G}}. \tag{1}$$

Our reconstruction loss combines the pixel-wise L_1, L_2, and the perceptual loss $J'(\,)$ [38] in a weighted sum

$$J^{\text{rec}}(\hat{\mathbf{x}}_t, \mathbf{x}_t) = \lambda_1 ||\hat{\mathbf{x}}_t - \mathbf{x}_t||_1 + \lambda_2 ||\hat{\mathbf{x}}_t - \mathbf{x}_t||_2^2 + \lambda' J'(\hat{\mathbf{x}}_t, \mathbf{x}_t). \tag{2}$$

The perceptual loss encourages reconstructions $\hat{\mathbf{x}}_t$ that are perceptually similar to the reference image $\mathbf{x}_t$ by similar feature activation $\boldsymbol{\phi}_\ell(\hat{\mathbf{x}}_t)$ and $\boldsymbol{\phi}_\ell(\mathbf{x}_t)$ for an ImageNet-pre-trained loss network $\boldsymbol{\phi}$ with

$$J'(\hat{\mathbf{x}}_t, \mathbf{x}_t) = \sum_\ell \|\boldsymbol{\phi}_\ell(\hat{\mathbf{x}}_t) - \boldsymbol{\phi}_\ell(\mathbf{x}_t)\|_2^2, \tag{3}$$

where ℓ refers to specific model layers [38].

The codebook CB is optimized using an embedding loss that utilizes the L_2 error to align the codebook vectors and the encoder output $\tilde{\mathbf{z}}$, while the commitment loss [18] forces the encoder to stick to a particular embedding, jointly forming the codebook loss

$$J^{\text{CB}}(\mathbf{z}_t, \tilde{\mathbf{z}}_t) = \underbrace{\|\text{sg}(\tilde{\mathbf{z}}_t) - \mathbf{z}_t\|_2^2}_{\text{embedding loss}} + \beta \cdot \underbrace{\|\tilde{\mathbf{z}}_t - \text{sg}(\mathbf{z}_t)\|_2^2}_{\text{commitment loss}}, \tag{4}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator and $\beta = 0.25$ is a commitment weight. As the quantization function (VQ) is non-differentiable, gradients of J^{rec}, J^{SSL}, and J^{G} are backwards copied from $\mathbf{z}_t$ to $\tilde{\mathbf{z}}_t$, influencing the codebook only indirectly [19].

Also, a teacher model is leveraged to distill semantic meaning into the image patch tokens $\mathbf{z}_t$ by enforcing cosine similarity between a teacher's encoding $\tilde{\mathbf{z}}_t^{\text{SSL}}$ and the quantized and transformed representation $\mathbf{z}'_t$, serving as a self-supervised learning loss

$$J^{\text{SSL}}(\mathbf{z}'_t, \mathbf{z}_t^{\text{SSL}}) = 1 - \frac{(\tilde{\mathbf{z}}_t^{\text{SSL}})^{\text{T}} \cdot \mathbf{z}''_t}{\max(\|\tilde{\mathbf{z}}_t^{\text{SSL}}\|_2 \cdot \|\mathbf{z}''_t\|_2, \epsilon)}, \tag{5}$$

where $\epsilon = 10^{-8}$ and $\mathbf{z}''_t \in \mathbb{R}^{384}$ is the output of a learnable fully connected layer with input $\mathbf{z}'_t \in \mathbb{R}^{64}$. We adopt ViT-s DINOv2 [39] as the teacher, extending GAIA-1's use of DINOv1 [40], with experimental validation.

To generate realistic-looking images, a GAN loss approach is adopted, consisting of two loss components with complementary objectives for discriminator $\text{D}(\,)$ and generator (TOK+DEC). Included in (1) is the generator loss

$$J^{\text{G}}(\hat{\mathbf{x}}_t) = -\text{D}(\hat{\mathbf{x}}_t), \tag{6}$$

depicted with selector positions $*$ in Fig. 3, which optimizes the generator (TOK+DEC) to produce fake images, misleading the discriminator $\text{D}(\,)$ into classifying them as real.

Complementing the total loss J^{total} (1), a patch-based discriminator $\text{D}(\,)$ is trained from scratch to distinguish real and generated images using a separate discriminator loss

$$J^{\text{D}}(\hat{\mathbf{x}}_t, \mathbf{x}_t) = 0.5(\text{ReLU}(1-\text{D}(\mathbf{x}_t)) + \text{ReLU}(1+\text{D}(\hat{\mathbf{x}}_t))), \tag{7}$$

encouraging $\text{D}(\mathbf{x}_t) \approx 1$ for real samples and $\text{D}(\hat{\mathbf{x}}_t) \approx -1$ for fake (i.e., auto-encoded) samples, see Fig. 3 (selector positions $**$). We adapt the patch-based discriminator's architecture from Esser et al. [19], reducing the number of patches to 64, validated empirically ("our $\text{D}(\,)$").

c) Fine-Tuning the World Model: We fine-tune the world model (WM) on a 4 fps *video* training dataset $\mathcal{D}^{\text{train-70k}}_{\text{BDDvid-4fps}}$ (cf. Table I), with each sample consisting of text token index sequence $c_1^M = (c_m)$ and an image patch token index sequence $k_1^{(T+N)\cdot n'} = (k_\nu)$, where each of the $T = 2$ initial and $N = 14$ future video frames is represented by $n' = 257$ indices. The input sequence $\mathbf{y}_1^L = (\mathbf{y}_\ell)$, formed by embedding and concatenating c_1^M and $k_1^{(T+N)\cdot n'-1} = (k_\nu)$, is processed by 32 decoder blocks under causal attention to predict the next image token index. Fine-tuning uses teacher forcing and cross-entropy loss over the predicted probability distributions $\mathbf{P}_{M+1}^{L+1} = (\mathbf{P}_\ell)$

TABLE I: **Amounts of videos and images** used in this work.

Notation	Data format	Fps	#train	#val	#test
$\mathcal{D}_{\text{BDDvid}}$	video	30	70k	10k	20k
$\mathcal{D}^{\text{train}-70\text{k}}_{\text{BDDvid}-4\text{fps}}$	video	4	70k	-	-
$\mathcal{D}^{\text{val}-500}_{\text{BDDvid}-4\text{fps}}$	video	4	-	500	-
$\mathcal{D}_{\text{BDDimg}}$	images	-	70k	10k	20k
$\mathcal{D}_{\text{BDDimg/5s}}$	images	-	$\sim$ 538k	-	-

for all target image patch tokens k_ν. Due to the size of the WM, full fine-tuning is impractical, so we use the parameter-efficient LoRA [12] method, fine-tuning only a small set of additional adapter weights for all linear and embedding layers, with full fine-tuning limited to normalization layer weights. This results in 6.97B total model parameters, with only 2.39% of these being fine-tuneable. To further reduce memory usage, frozen parameters are stored in the lower-precision format `bfloat16`, while only fine-tunable parameters remain in `FP32`.

d) Fine-Tuning the Video Decoder: We fine-tune our video decoder (VDEC) on the 4 fps *video* training dataset $\mathcal{D}^{\text{train}-70\text{k}}_{\text{BDDvid}-4\text{fps}}$ (cf. Table I), using samples of 3 consecutive frames $\mathbf{x}^{t+1}_{t-1}$. The frames are preprocessed and then tokenized separately using the fine-tuned and then frozen TOK, and jointly decoded to $\hat{\mathbf{x}}^{t+1}_{t-1}$. For optimization, we employ the image reconstruction loss J^{rec} (2) and the generator component $J^{\text{G}-3\text{D}}$ of a sequence-based GAN loss in analogy to (6), as video decoder loss

$$J^{\text{VDEC}}(\hat{\mathbf{x}}^{t+1}_{t-1}, \mathbf{x}^{t+1}_{t-1}) = \sum_{\tau=t-1}^{t+1} J^{\text{rec}}(\hat{\mathbf{x}}_\tau, \mathbf{x}_\tau) + J^{\text{G}-3\text{D}}(\hat{\mathbf{x}}^{t+1}_{t-1}). \tag{8}$$

Consequently, we use the discriminator loss component $J^{\text{D}-3\text{D}}(\hat{\mathbf{x}}^{t+1}_{t-1}, \mathbf{x}^{t+1}_{t-1})$ after (7) to train a 3D patch-based discriminator $\text{D}^{\text{VDEC}}(\ )$ from scratch, which jointly classifies sequences of three reconstructed frames $\hat{\mathbf{x}}^{t+1}_{t-1}$. We adopt our $\text{D}(\ )$ architecture from TOK+DEC training and inflate it to 3D, inspired by Yu et al. [33].

IV. Experiments and Discussion

A. Experimental Setup

Dataset: The Berkeley Deep Drive Dataset (BDD100K) [14] is the largest open source driving video dataset containing 100,000 videos, most of them with a duration of 40 seconds. These have been recorded in various conditions, such as weather, lighting, and driving routes, captured at 30 fps with image size $\mathring{H} \times \mathring{W} = 720 \times 1,280$. As detailed in Table I, the BDD100K video dataset $\mathcal{D}_{\text{BDDvid}}$ comprises a training, validation, and test split, with 70k, 10k, and 20k videos, respectively. For WM and VDEC fine-tuning and validation, we create a subset of 70k training videos $\mathcal{D}^{\text{train}-70\text{k}}_{\text{BDDvid}-4\text{fps}}$ and a subset of 500 validation videos, $\mathcal{D}^{\text{val}-500}_{\text{BDDvid}-4\text{fps}}$, both sub-sampled to 4 fps. BDD100K images $\mathcal{D}_{\text{BDDimg}}$, a subset of $\mathcal{D}_{\text{BDDvid}}$, contains one frame per video extracted just after 10 seconds. For fine-tuning the image TOK (and image DEC), we created a larger custom image subset $\mathcal{D}^{\text{train}}_{\text{BDDimg/5s}}$ with about 538k training images, sampled at 0.2 fps from $\mathcal{D}^{\text{train}}_{\text{BDDvid}}$, while we still validate on the official validation split $\mathcal{D}^{\text{val}}_{\text{BDDimg}}$. The BDD100K video training set $\mathcal{D}^{\text{train}}_{\text{BDDvid}}$ provides 778 hours of training data *with approximately 84M unique images, which is only about 15% of the 5,100 hours of GAIA-1's unpublished dataset with about 420M unique images* [5].

Metrics: To assess image reconstruction quality, we use PSNR [41], SSIM [42], MS-SSIM [43], and LPIPS [44], comparing decoded images to ground-truth references. For *generated* image/video frames, direct ground-truth comparison is not reasonable, accordingly, we report the Fréchet inception distance (FID) [45], the CLIP maximum mean discrepancy (CMMD) [46], and for video data also the Fréchet video distance (FVD) [47]. The number of evaluated video frames is noted in the subscript (e.g., FVD_{14}). These perceptual metrics enable quality assessment of the WM and image/video decoder by evaluating predicted video frames while allowing them to diverge from corresponding ground-truth frames. Thereby, we provide a *quantitative* analysis of image/video quality, while GAIA-1 [5] authors and GAIA-2 [10] authors only report qualitative results.

Training Details: All our models are fine-tuned using the AdamW optimizer, distributing the batch across four `Nvidia H100` 94GB GPUs. Our TOK+DEC (`VQGAN`) is fine-tuned for 200k steps with a batch size of 80, comprising 2k steps of linear warm-up to an initial learning rate of 5×10^{-5} and 150k steps of cosine decay to a final learning rate of 5×10^{-7}, *resulting in only 24 hours of fine-tuning*. The discriminator loss J^D (7) and generator loss J^{G} (6) are applied after 20k steps. Our WM is fine-tuned for 28.3k steps with a batch size of 24, comprising 250 steps of linear warm-up to an initial learning rate of 6×10^{-4} and 15k steps of cosine decay to a final learning rate of 6×10^{-5}, *resulting in 65 hours of fine-tuning*. Other configurations follow LLaMA-2 [2]. Our VDEC is fine-tuned for 100k steps with a batch size of 48, comprising 100 steps of linear warm-up to an initial learning rate of 5×10^{-5} followed by a cosine decay schedule to 5×10^{-7}, *resulting in 65 hours of fine-tuning.* The discriminator loss J^{D-3D} and generator loss $J^{\text{G}-3\text{D}}$ are applied after 2k steps. All code is based on the `PyTorch` [48] and `jax` [49] frameworks.

B. Results and Discussion

Image Tokenizer: Table II shows the results of fine-tuning the pre-trained `VQGAN` (TOK+DEC) using total loss J^{total} (1) and discriminator loss J^{D} (7), and explores the effects of individual loss components by omitting them ($-J$) during fine-tuning. Although we employ a smaller tokenizer and partly different loss definitions, we start our investigations with the exact GAIA-1 loss weights [5]: $\lambda^{\text{L1}}=0.2$, $\lambda^{\text{L2}}=2.0$, $\lambda' = 0.1$, $\lambda^{\text{G}} = 1.0$, $\lambda^{\text{CB}} = 1.0$, $\lambda^{\text{SSL}} = 0.1$. Omitting J^{SSL} improves four metrics, but on the important perceptual FID and particularly CMMD it falls behind J^{total}. Removing the perceptual contribution J' of J^{total} expectedly leads to catastrophic perceptual metrics, which are also not improved

TABLE II: **Tokenizer** and **image decoder loss function** ablations: Quantitative results for *omitting* (-J) loss components during fine-tuning with the total loss J^{total} (1), employing J^{D}(7) for discriminator training, with D( ) from [19] and J^{SSL} (5) using `DINOv1` [40], evaluated on $\mathcal{D}^{\text{val}}_{\text{BDDimg}}$. `VQGAN` models (TOK + DEC) were fine-tuned on $\mathcal{D}^{\text{train}}_{\text{BDDimg/5s}}$ for 200k iterations. Weights of omitted losses are set to zero. Best results are in bold font, second best underlined. PSNR in (dB).

Loss functions	PSNR ↑	SSIM ↑	MS-SSIM ↑	LPIPS ↓	FID ↓	CMMD ↓
J^{total} (1),J^{D} (7)	25.75	0.7630	0.9022	<u>0.1170</u>	**5.48**	**0.074**
- J^{SSL} (1), (5)	<u>26.31</u>	<u>0.7794</u>	<u>0.9140</u>	**0.1096**	<u>5.79</u>	0.122
- J' of J^{rec} (1), (2), (3)	25.75	0.7487	0.8943	0.1756	18.64	0.594
- J^{L2} of J^{rec} (1), (2)	23.87	0.7304	0.8713	0.1307	6.10	<u>0.107</u>
- J^{G} (1), (6)	**27.11**	**0.8041**	**0.9178**	0.1759	17.97	0.393
No fine-tuning	25.08	0.7690	0.9018	0.1207	5.82	0.385

TABLE III: **Tokenizer** and **image decoder discriminator and loss weight** ablations: Quantitative results for using our D( ) with varying weights λ', λ^{G} for perceptual loss and generator loss during fine-tuning with the total loss J^{total} (1), employing J^{D}(7) for discriminator training, with J^{SSL} (5) using `DINOv1` [40], evaluated on $\mathcal{D}^{\text{val}}_{\text{BDDimg}}$. `VQGAN` models (TOK + DEC) were fine-tuned on $\mathcal{D}^{\text{train}}_{\text{BDDimg/5s}}$ for 200k iterations. Best results are in bold font, second best underlined. PSNR in (dB).

Fine-tuning with D( )	λ'	λ^{G}	$\frac{\lambda^{\text{G}}}{\lambda'}$	PSNR ↑	SSIM ↑	MS-SSIM ↑	LPIPS ↓	FID ↓	CMMD ↓
D( ) from [19]	0.1	1.0	10	25.75	0.7630	0.9022	0.1170	5.48	0.074
Our D( ) (Sec. III-C)	0.1	1.0	10	<u>25.88</u>	0.7685	0.9039	0.1148	5.14	0.065
.	0.1	1.5	15	**25.89**	0.7694	0.9044	0.1146	5.32	0.064
.	0.3	1.0	3.3	25.68	<u>0.7704</u>	<u>0.9049</u>	0.1050	4.43	0.053
.	0.3	1.2	4	25.68	**0.7712**	**0.9050**	0.1053	4.45	0.059
(**proposed**)	1.0	1.0	1	25.14	0.7649	0.8991	**0.1035**	3.97	**0.048**
	2.0	1.0	0.5	24.74	0.7616	0.8941	0.1042	**3.55**	<u>0.050</u>
	2.0	2.0	1	24.77	0.7623	0.8954	<u>0.1040</u>	<u>3.93</u>	0.051

a) Orig. image $\mathbf{x}_t$ b) Orig. RoI c) No fine-tuning d) **proposed**

Fig. 4: **Example** of a region of interest (RoI) for a transcoded (ENC+VQ+DEC = TOK+DEC) image from $\mathcal{D}^{\text{val}}_{\text{BDDimg}}$. (a) The RoI is marked in the original image $\mathbf{x}_t$, and (b) magnified for comparison. (c) Without domain-specific fine-tuning, the pre-trained `VQGAN` produces poor reconstructions of automotive objects (here: parked cars). (d) Proposed fine-tuning significantly improves image quality. Best viewed on screen.

vs. J^{total} by omitting J^{L2}. Due to the strong PSNR, SSIM, MS-SSIM, one might be tempted to omit J^{G}, but the price is again significantly worse perceptual metrics. For all subsequent investigations, we adopt the total loss J^{total} (1), as it overall yields best perceptual performance (LPIPS, FID, CMMD).

In Table III, we analyze and optimize the loss weights λ' and λ^{G} of the important J^{total} contributions J' and J^{G}, respectively. We particularly show fine-tuning results using the discriminator from [19], and in more detail *our* discriminator D( ) (cf. Sec. III-C) modified to classify fewer patches, each with an increased receptive field. We finally propose and further use the TOK+DEC fine-tuning configuration of J^{total} (1), our D( ), and $\lambda' = 1$, $\lambda^{\text{G}} = 1$, as it is overall strongest in the perceptual metrics and therefore shows highest potential for application in a generative time-predictive framework as ours.

Fig. 4 shows successful TOK+DEC fine-tuning, presenting an original image of an automotive scene (Fig. 4a), a zoomed-in region of interest highlighting parked cars (Fig. 4b), and corresponding transcoded model outputs (Fig. 4c,d). The non-fine-tuned `VQGAN` (TOK+DEC) exhibits poor reconstructions of car parts (Fig. 4c). Fine-tuning with our proposed configuration from Table III restores structure and details, thereby improving perceptual quality.

World Model: Table IV presents the system performance with WM, but still using the image decoder DEC. Different teacher models are used for self-supervised learning loss J^{SSL} (5) during TOK + DEC fine-tuning. Evaluation is performed on video data $\mathcal{D}^{\text{val}-500}_{\text{BDDvid}}$ for $N = 14$ predicted frames based on $T = 2$ conditioning frames. While GAIA-1 uses a probably larger `DINO` (Version 1) for J^{SSL}, our best FVD score is achieved with $J^{\text{SSL}-\text{DINOv2}}$. This demonstrates the effectiveness of `DINOv2`'s representations over `DINO` or no J^{SSL} guidance, motivating its use as the teacher for our finally proposed TOK+DEC fine-tuning, therefore used in all consecutive experiments.

In Table V, we investigate the world model's performance depending on the top-k hyperparameter, which controls the breadth of token selection during inference. We show

TABLE IV: **Distillation teacher model** ablations: Quantitative results for different teacher models used with J^{SSL} during fine-tuning of the tokenizer (TOK) and image decoder (DEC). Results were calculated for $N = 14$ on the $\mathcal{D}^{\text{val}-500}_{\text{BDDvid}}$ subset. Best results are in bold font. Top-k value provided.

TOK/DEC fine-tuned with	k	Validation: TOK/WM/DEC FID$_{14}$ ↓	CMMD$_{14}$ ↓	FVD$_{14}$ ↓
No J^{SSL}	50	14.65	0.105	188.59
$J^{\text{SSL}-\text{DINO}}$	50	**14.25**	0.099	181.55
$J^{\text{SSL}-\text{DINOv2}}$	50	14.72	**0.083**	**178.97**

TABLE V: **World model top-k** ablations: Quantitative results for various configurations of top-k sampling on the $\mathcal{D}^{\text{val}-500}_{\text{BDDvid}}$ subset with $N = 14$, $J^{\text{SSL}-\text{DINOv2}}$ used for DEC optimization; VDEC builds upon that. Best results are in bold font. Top-k value provided.

System	k	FID$_{14}$ ↓	CMMD$_{14}$ ↓	FVD$_{14}$ ↓
TOK+DEC	-	7.19	0.048	104.35
TOK+WM+DEC	1	28.04	0.255	644.71
	5	20.26	0.114	296.95
	10	18.23	0.092	243.97
	50	14.72	0.083	178.97
	200	12.22	**0.081**	**160.48**
	1000	**11.29**	0.090	163.80
TOK+VDEC	-	8.85	0.155	74.29
OpenViCA (TOK+WM+VDEC)	1	28.61	0.411	646.03
	5	22.14	0.244	276.93
	10	19.94	**0.215**	210.71
	50	16.67	0.229	153.19
	200	14.22	0.241	136.41
	1000	**13.29**	0.248	**132.16**

TOK+DEC and TOK+VDEC as reference, which perform no prediction (i.e., no WM), instead, here, we only transcode the video frames. VDEC is based on DEC trained with $J^{\text{SSL}-\text{DINOv2}}$ and our $\text{D}(\ )$. We observe that temporal prediction by the WM of course harms the perceptual metrics, but values in the range $k = 10...1000$ improve the metrics. For our finally proposed OpenViCA system, we recommend $k = 1000$ as this choice provides the best perceptive video quality (FVD$_{14}$). We further find that higher top-k values lead to increasingly diverse and creative videos generated by our OpenViCA system.

Image/Video Decoder: Table VI presents our final quantitative comparison of the image decoder (DEC) and the video decoder (VDEC) for temporal predictive video generation using the WM. We compare results of both systems using their strongest top-k results in terms of FVD$_{14}$. We observe that using our VDEC yields superior FVD$_{14}$ performance compared to our DEC (132.16 vs. 160.48), justifying its adoption in our finally proposed OpenViCA system.

TABLE VI: **Image/video decoder** and **final results with strongest top-k**: Quantitative results for deploying the *image* decoder (DEC) or *video* decoder (VDEC) for generating video frames from WM output. Results were calculated for $N = 14$ on the $\mathcal{D}^{\text{val}-500}_{\text{BDDvid}}$ subset. VDEC is based on DEC trained with $J^{\text{SSL}-\text{DINOv2}}$. OpenViCA employs TOK/WM/VDEC. Top-k value provided. Best results in bold.

System	k	FID$_{14}$ ↓	CMMD$_{14}$ ↓	FVD$_{14}$ ↓
TOK+WM+DEC	200	**12.22**	**0.081**	160.48
OpenViCA	1000	13.29	0.248	**132.16**

V. Limitations

Our 3D CNN video decoder, trained with GAN and reconstruction losses, likely produces lower-quality frames than modern diffusion-based methods. However, it offers a simple-to-train solution that effectively handles quantized latent image patch tokens and seamlessly integrates into our system. While also being effective in its frame-by-frame processing, extending its temporal context might allow to enhance visual coherence beyond our current achievements.

VI. Conclusions

We introduced OpenViCA, an open video continuation system for automotive driving scenes. Our method is build entirely on open-source pre-trained general-purpose models, which we fine-tune to the automotive domain using the public driving videos dataset BDD100K. OpenViCA employs dedicated models for image tokenization, world modeling, and video decoding allowing for coherent video continuation with 256x256-sized videos being generated frame-by-frame at 4 fps. We motivate design choices by quantitative and qualitative analysis, and publish our training and inference code, thereby allowing for full reproducibility and deployment at academic scale.

Acknowledgment: Computational resources were provided by the German AI Service Center WestAI.

References

[1] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video Diffusion Models," in Proc. of NeurIPS, New Orleans, LA, USA, Nov. 2022, pp. 8633–8646.

[2] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and Efficient Foundation Language Models," Feb. 2023, arXiv:2302.13971 [cs].

[3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models," Jul. 2023, arXiv:2307.09288 [cs].

[4] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in Proc. of NeurIPS, vol. 33, virtual, Dec. 2020, pp. 6840–6851.

[5] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado, "GAIA-1: A Generative World Model for Autonomous Driving," Sep. 2023, arXiv:2309.17080 [cs].

[6] J. Wu, S. Yin, N. Feng, X. He, D. Li, J. Hao, and M. Long, "iVideoGPT: Interactive VideoGPTs are Scalable World Models," Oct. 2024, arXiv:2405.15223 [cs].

[7] T. Fingscheidt, H. Gottschalk, and S. Houben, Eds., Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety. Cham: Springer Nature, 2022.

[8] S. Houben, S. Abrecht, M. Akila, A. Bär, F. Brockherde, P. Feifel et al., "Inspect, Understand, Overcome: A Survey of Practical Methods for AI Safety," in Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety. Cham: Springer Nature, 2022, pp. 3–78.

[9] G. Zhao, X. Wang, Z. Zhu, X. Chen, G. Huang, X. Bao, and X. Wang, "DriveDreamer-2: LLM-Enhanced World Models for Diverse Driving Video Generation," Apr. 2024, arXiv:2403.06845 [cs].

[10] L. Russell, A. Hu, L. Bertoni, G. Fedoseev, J. Shotton, E. Arani, and G. Corrado, "GAIA-2: A Controllable Multi-View Generative World Model for Autonomous Driving," Mar. 2025, arXiv:2503.20523 [cs].

[11] H. Liu, W. Yan, M. Zaharia, and P. Abbeel, "World Model on Million-Length Video And Language With Blockwise RingAttention," Jul. 2024, arXiv:2402.08268 [cs].

[12] E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in Proc. of ICLR, virtual, Apr. 2022, pp. 1–26.

[13] S. Patil, W. Berman, R. Rombach, and P. von Platen, "aMUSEd: An Open MUSE Reproduction," Jan. 2024, arXiv:2401.01808 [cs].

[14] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," in Proc. of CVPR, virtual, Jun. 2020, pp. 2636–2645.

[15] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction," in Proc. of Artificial Neural Networks and Machine Learning, Espoo, Finnland, Jun. 2011, pp. 52–59.

[16] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow, "Adversarial Autoencoders," in Proc. of ICLR Workshops, San Juan, Puerto Rico, May 2016, pp. 1–10.

[17] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in Proc. of ICLR, Banff, AB, Canada, Apr. 2014, pp. 1–9.

[18] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural Discrete Representation Learning," in Proc. of NIPS, vol. 30, Long Beach, CA, USA, Dec. 2017, pp. 6309 – 6318.

[19] P. Esser, R. Rombach, and B. Ommer, "Taming Transformers for High-Resolution Image Synthesis," in Proc. of CVPR, Nashville, TN, USA, Jun. 2021, pp. 12 868–12 878.

[20] Y. Guan, H. Liao, Z. Li, J. Hu, R. Yuan, Y. Li, G. Zhang, and C. Xu, "World Models for Autonomous Driving: An Initial Survey," IEEE Transactions on Intelligent Vehicles, pp. 1–17, May 2024.

[21] W. Zhang, G. Wang, J. Sun, Y. Yuan, and G. Huang, "STORM: Efficient Stochastic Transformer based World Models for Reinforcement Learning," in Proc. of NeurIPS, New Orleans, LA, USA, Dec. 2023, pp. 27 147–27 166.

[22] S. W. Kim, J. Philion, A. Torralba, and S. Fidler, "DriveGAN: Towards a Controllable High-Quality Neural Simulation," in Proc. of CVPR, Nashville, TN, USA, Jun. 2021, pp. 5816–5825.

[23] F. Bartoccioni, E. Ramzi, V. Besnier, S. Venkataramanan, T.-H. Vu, Y. Xu, L. Chambon, S. Gidaris, S. Odabas, D. Hurych, R. Marlet, A. Boulch, M. Chen, Zablocki, A. Bursuc, E. Valle, and M. Cord, "VaViM and VaVAM: Autonomous Driving through Video Generative Modeling," Feb. 2025, arXiv:2502.15672 [cs].

[24] X. Hu, W. Yin, M. Jia, J. Deng, X. Guo, Q. Zhang, X. Long, and P. Tan, "DrivingWorld: Constructing World Model for Autonomous Driving via Video GPT," Dec. 2024, arXiv:2412.19505 [cs].

[25] Y. Wang, J. He, L. Fan, H. Li, Y. Chen, and Z. Zhang, "Driving into the Future: Multiview Visual Forecasting and Planning with World Model for Autonomous Driving," in Proc. of CVPR, Seattle, WA, USA, Jun. 2024, pp. 14 749–14 759.

[26] X. Wang, Z. Zhu, G. Huang, X. Chen, J. Zhu, and J. Lu, "DriveDreamer: Towards Real-World-Driven World Models for Autonomous Driving," in Proc. of ECCV, Milan, Italy, Sep. 2024, pp. 55–72.

[27] F. Jia, W. Mao, Y. Liu, Y. Zhao, Y. Wen, C. Zhang, X. Zhang, and T. Wang, "ADriver-I: A General World Model for Autonomous Driving," Nov. 2023.

[28] S. Gao, J. Yang, L. Chen, K. Chitta, Y. Qiu, A. Geiger et al., "Vista: A Generalizable Driving World Model with High Fidelity and Versatile Controllability," Oct. 2024, arXiv:2405.17398 [cs].

[29] M. Hassan, S. Stapf, A. Rahimi, P. M. B. Rezende, Y. Haghighi, D. Brüggemann et al., "GEM: A Generalizable Ego-Vision Multimodal World Model for Fine-Grained Ego-Motion, Object Dynamics, and Scene Composition Control," Dec. 2024, arXiv:2412.11198 [cs].

[30] Y. Wang, T. Xiong, D. Zhou, Z. Lin, Y. Zhao, B. Kang, J. Feng, and X. Liu, "Loong: Generating Minute-level Long Videos with Autoregressive Language Models," Apr. 2025, arXiv:2410.02757 [cs].

[31] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating Videos with Scene Dynamics," in Proc. of NIPS, vol. 29, Barcelona, Spain, Dec. 2016, pp. 1–9.

[32] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, "VideoGPT: Video Generation using VQ-VAE and Transformers," Sep. 2021, arXiv:2104.10157 [cs].

[33] L. Yu, Y. Cheng, K. Sohn, J. Lezama, H. Zhang, H. Chang, A. G. Hauptmann, M.-H. Yang, Y. Hao, I. Essa, and L. Jiang, "MAGVIT: Masked Generative Video Transformer," in Proc. of CVPR, Vancouver, BC, Canada, Jun. 2023, pp. 10 459–10 469.

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in Proc. of NIPS, vol. 27, Montreal, QC, Canada, Dec. 2014, pp. 1–9.

[35] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models," in Proc. of NeurIPS, New Orleans, LA, USA, Dec. 2022, pp. 25 278–25 294.

[36] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval," in Proc. of ICCV, Montreal, QC, Canada, Oct. 2021, pp. 1708–1718.

[37] R. C. Gonzales and R. E. Woods, Digital Image Processing. Prentice Hall, 2008.

[38] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," in Proc. of ECCV, The Netherlands, Amsterdam, Oct. 2016, pp. 694–711.

[39] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning Robust Visual Features without Supervision," Transactions on Machine Learning Research, pp. 1–32, Jan. 2024.

[40] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging Properties in Self-Supervised Vision Transformers," in Proc. of ICCV, virtual, Oct. 2021, pp. 9650–9660.

[41] D. Salomon, Data Compression: The Complete Reference. Springer Science & Business Media, 2004.

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: from Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[43] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale Structural Similarity for Image Quality Assessment," in Proc. of ACSSC, Pacific Grove, CA, USA, Nov. 2003, pp. 1398–1402.

[44] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in Proc. of CVPR, Salt Lake City, UT, Jun. 2018, pp. 586–595.

[45] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," in Proc. of NIPS, vol. 30, Long Beach, CA, USA, Dec. 2017, pp. 6626–6637.

[46] S. Jayasumana, S. Ramalingam, A. Veit, D. Glasner, A. Chakrabarti, and S. Kumar, "Rethinking FID: Towards a Better Evaluation Metric for Image Generation," in Proc. of CVPR, Seattle, WA, USA, Jun. 2024, pp. 9307–9315.

[47] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "FVD: A New Metric for Video Generation," in Proc. of ICLR Workshops, New Orleans, LA, USA, May 2019, pp. 1–9.

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc., 2019.

[49] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: Composable Transformations of Python+NumPy Programs," 2018.