The Matrix: Infinite-Horizon World Generation with Real-Time Moving Control

Ruili Feng*‡, Han Zhang*, Zhantao Yang*, Jie Xiao*, Zhilei Shu*, Zhiheng Liu, Andy Zheng, Yukun Huang, Yu Liu†, Hongyang Zhang‡

Alibaba Group, University of Hong Kong, University of Waterloo, Vector Insititute

*Equal Contribution, †Engineer Advisor, ‡Project Leader

Correspondence to: thematrix1999.ai@gmail.com

https://thematrix1999.github.io/

Abstract

*We present The Matrix, the first foundational **realistic world simulator** capable of generating **infinitely long** 720p **high-fidelity real-scene** video streams with **real-time**, responsive control in both first- and third-person perspectives, enabling immersive exploration of richly dynamic environments. Trained on limited supervised data from AAA games like Forza Horizon 5 and Cyberpunk 2077, complemented by large-scale unsupervised footage from real-world settings like Tokyo streets, The Matrix allows users to traverse diverse terrains—deserts, grasslands, water bodies, and urban landscapes—in continuous, uncut hour-long sequences. With speeds of up to 16 FPS, the system supports real-time interactivity and demonstrates **zero-shot generalization**, translating virtual game environments to real-world contexts where collecting continuous movement data is often infeasible. For example, The Matrix can simulate a BMW X3 driving through an office setting—an environment present in neither gaming data nor real-world sources. This approach showcases the potential of AAA game data to advance robust world models, bridging the gap between simulations and real-world applications in scenarios with limited data. All the codes, data, and model checkpoints in this paper will be open sourced.*

> "This is the world that you know; the world as it was at the end of the 20th century. It exists now only as part of a neural-interactive simulation that we call the Matrix."
>
> Morpheus to Neo

1. Introduction

Neural-interactive simulation, a concept popularized by *The Matrix* (1999), envisions a world fully constructed by AI to replicate 20th-century human society. This paper takes an initial step toward realizing this vision by developing a world model that enables neural networks to 'dream' visually authentic environments. The result is an infinite-horizon, high-resolution (720p) simulation that supports real-time (8 - 16 FPS) interactive exploration across diverse landscapes, including deserts, grasslands, water terrains, and urban settings. Responding to real-time control signals, the world model predicts future frames in these environments in a streaming and auto-regressive fashion.

Figure 1. *The Matrix* is a foundational **realistic world simulator** capable of generating **infinitely long** 720p **high-fidelity real-scene** video streams with **real-time**, precise moving control. **Click to play with Adobe Acrobat Reader!** The upper 1-minute demo may need flushing time.

World models offer a promising solution to the overwhelming costs of AAA game development, which can easily run into tens or even hundreds of millions of dollars. Traditional game creation depends on engines such as Unity 3D, Unreal Engine, and Blender, each requiring substantial expertise, intensive asset preparation, and meticulous hyperparameter tuning. Furthermore, games built with these engines are often limited in reusability, as each new title demands a comprehensive redesign. In contrast, data-driven world models tackle these issues by minimizing the need for manual configuration, simplifying development work-

flows, and boosting scalability across projects.

Despite extensive research in world models [30], key challenges remain. First, prior studies have predominantly focused on non-AAA video games, such as Atari [1, 10, 31], Mario [25], Minecraft [5, 11], Counter-Strike: Global Offensive (CS:GO) [1], and DOOM [36], which fall short in replicating real-world fidelity. Second, current video generation techniques, like Sora [23], are constrained to short sequences of about 1 minute, forcing existing world models to assemble independently generated clips with noticeable transitions. Finally, achieving real-time generation remains a major hurdle. For example, state-of-the-art 2D platformer game generator Genie [2] runs as slow as 1 FPS. This paper addresses these limitations by introducing the first scalable, high-fidelity (1280×720 pixels) world model in real time that enhances simulation realism and bridges the gap between virtual environments and reality. Notably, our world model is the first with strong domain generalization and real-time control. For example, our foundation model allows us to control BMW X3 driving through an indoor setting or in the sea—an environment present in neither gaming data nor real-world sources.

1.1. Our Contributions

Our contributions are as follows:

- We introduce *The Matrix*, the first foundational simulator for realistic worlds, capable of generating infinitely long, high-fidelity 720p real-scene video streams with real-time, interactive controls and strong domain generalization. The model is light and consists of 2.7B parameters.
- At the core of *The Matrix* is a novel diffusion technique, the Shift-Window Denoising Process Model (Swin-DPM), enabling pre-trained DiT models [24] to extrapolate seamlessly for smooth, continuous, and infinitely extendable video creation. This technique holds potential for broader applications in long-form video generation.
- Additionally, we introduce *GameData*, a platform that autonomously captures paired in-game states—extracted from CPU memory—alongside corresponding video frames, significantly reducing labeling costs and complexity. This platform produces *Source*, a new training dataset for world models with action-frame paired data.

1.2. Technical Advantages of *The Matrix*

Tab. 1 highlights a comparison between *The Matrix* and other game generation models across six key features. Our work advances the state-of-the-art of world models in the following aspects:

- **Infinite Video Generation:** *The Matrix* generates consistent, infinitely long video sequences using a streaming, auto-regressive approach.
- **High-Quality Rendering:** *The Matrix* delivers AAA-level, realistic rendering at a resolution of 1280×720.
- **Real-Time, Frame-Level Control:** *The Matrix* operates with speeds of 8 - 16 FPS, providing real-time, frame-level control for interactive applications.
- **Domain Generalization:** Trained with small amounts of supervised AAA game data and large amounts of unsupervised internet videos, *The Matrix* achieves strong domain generalization to real-world settings.

2. Related Work

World Model for Agent Learning. Developing world models for training agents has been a long-standing research focus, aimed at enhancing policy learning within simulated environments rather than solely achieving high-fidelity reconstructions of observations. This research involves two primary stages: 1) modeling the training environment by reconstructing observations, rewards, and continuation signals, often through a recurrent state-space model; and 2) utilizing this model to predict future states, enabling reinforcement learning to optimize robust policy functions. Studies indicate that this method provides sample efficiency gain of over 1000% compared to directly learning policies from real environments, shows resilience across diverse domains, and can outperform fine-tuned expert agents on a range of benchmarks and data budgets [11]. Key contributions in this area include Recurrent World Models [8], Dreamer (v1 [9], v2 [10], and v3 [11]), TD-MPC (v1 [12] and v2 [13]), DayDreamer [37], Safe-Dreamer [18], and MuDreamer [3]. Notably, MuZero [31] runs the self-play of Monte Carlo tree search to build world models for Atari, Go, chess and shogi, without external data.

World Simulation. Distinct from world models designed for agent learning, another research direction emphasizes world simulation, focusing on human interaction with neural networks through high-quality rendering, robust control, and strong domain generalization to real-world scenarios. This research explores two types of control: video-level and frame-level. In video-level control, a control signal is given at the start, and the model generates a responsive video sequence; notable examples include UniSim [39], Pandora [38], GameGen-X [4], MicroVGG [25], and GAIA-1 [16]. To approximate continuous control, this approach often stitches together independently generated clips, which may result in visible transitions. In contrast, frame-level control provides fine-grained adjustments every few frames, enabling more precise, responsive interactions similar to gameplay, as seen in examples like Genie [2], DIAMOND [1], GameNGen [36], and Oasis [5]. Prior work in world simulation has typically focused on one of three aspects—video length, high resolution, or domain generaliza-

Table 1. Comparison of recent generative models for game simulation. *The Matrix* distinguishes itself as a foundation model capable of generating infinitely long videos with AAA game quality, high resolution, frame-level real-time control, and robust domain generalization. Here, * indicates concurrent work with *The Matrix*, and supervised/unsupervised refers to the video data with/without true control signal.

Feature	Genie	DIAMOND	MarioVGG*	GameNGen*	Oasis*	GameGen-X*	The Matrix
Video Length	2s	Infinite	6 Frames	Infinite	Infinite	4s–16s	Infinite
Training Corpus	2D Games (unsupervised)	Atari CS:GO	Mario	DOOM	Minecraft	AAA Games	AAA Games (supervised, small) Internet Videos (unsupervised, large)
Resolution	360p	280 × 150	64 × 48	240p	720p	720p	720p
Control	Frame-Level	Frame-Level	Video-Level	Frame-Level	Frame-Level	Video-Level	Frame-Level
Real-Time	No	Yes	No	Yes	Yes	No	Yes
Control Generalization	Yes	No	No	No	No	No	Yes

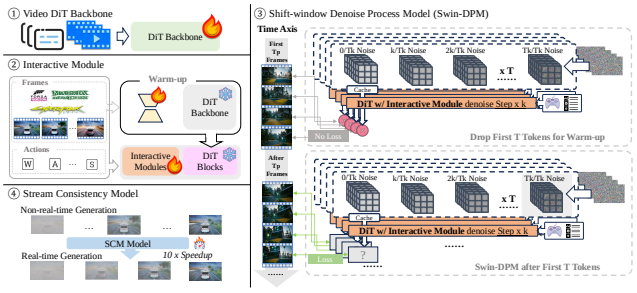


Figure 2. The training process of *The Matrix* begins with a pre-trained video DiT backbone. First, the Interactive Module is warmed up using Synthesized Observations of Unreal Rendered Contextual Environments data with unsupervised LoRA to make subsequent training focus on movement, not visuals. Then, we train the Interactive Module for precise frame-level control. Swin-DPM enables infinite-length generation, and Stream Consistency Model is introduced to accelerate sampling to real-time speeds.

tion—without addressing all three simultaneously. Table 1 presents a comparison between *The Matrix* and prior works. *The Matrix* uniquely stands out as a foundation model capable of generating infinitely long, AAA-quality videos with high resolution, frame-level real-time control, and strong generalization to real-world contexts.

3. Methods

Achieving granular control is notoriously challenging, as labeling actions at the frame level is typically cost-prohibitive. To address this, we develop the *GameData* platform, which autonomously captures paired data of in-game states (extracted directly from CPU memory) alongside corresponding video frames, significantly reducing labeling costs and complexity. Additionally, *The Matrix* incorporates an advanced Interactive Module that learns and generalizes game movement interactions from a limited amount of labeled data combined with extensive unlabeled data from both games and real-world environments. This enables *The Matrix* to deliver exceptional accuracy across diverse scenarios, while maintaining robust performance in the gaming domain.

Generating high-quality, real-time, and generalizable video simulations for infinite sequences presents additional technical challenges, often forcing previous simulators to compromise on one or more essential aspects. *The Matrix* overcomes these limitations by adapting the world model from a pre-trained video Diffusion Transformer (DiT) model [24], leveraging its extensive pre-existing knowledge and generation quality. To enable infinite-length generation, *The Matrix* introduces a novel diffusion approach, the Shift-Window Denoising Process Model (Swin-DPM), which allows the DiT model to extrapolate for smooth, continuous, and indefinitely long video creation. Finally, to achieve real-time efficiency, we fine-tune a Stream Consistency Model (SCM), accelerating inference to real-time.

Video DiT Backbone. As a preliminary, we introduce the video DiT backbone, adapted from the publicly available DiT models [41]. It employs a 3D Variational Auto-Encoder (VAE) to encode $T \times p$ video frame into T video tokens. The backbone consists of 32 attention blocks, followed by a linear output head with LayerNorm. Each attention block includes a self-attention layer operating on network features, a cross-attention layer linking conditions with self-attention outputs, and an FFN layer composed of two linear layers with a GELU activation [14] in between. See *Appendix Section A.1* for further details.

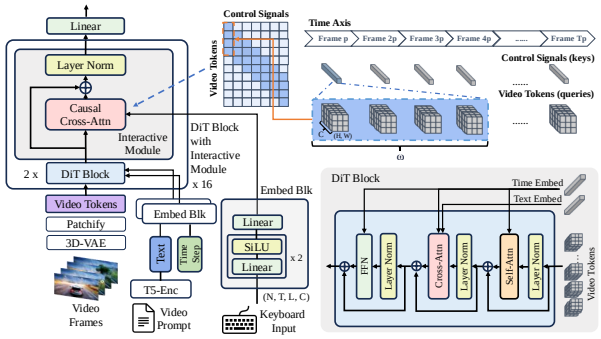
3.1. Model Components

The Matrix comprises three main components: a) an **Interactive Module** that interprets user intentions (e.g., keyboard inputs) and integrates them into video token generation; b) a **Shift-Window Denoising Process Model** (Swin-DPM) that enables infinite-length video generation; and c) a **Stream Consistency Model** (SCM) that accelerates sampling to achieve real-time performance. As shown in Fig. 2, the model is fine-tuned from a pre-trained video DiT model through a three-stage process: first, we fix the DiT model parameters and train the Interactive Module; next, we train the Interactive Module and the DiT together following the Swin-DPM; finally, we optimize an SCM to accelerate inference to real-time speeds. The first two stages leverage both labeled gaming and unlabeled internet video data to enhance generalization, while the final SCM training focuses on labeled gaming data to reduce optimization complexity.

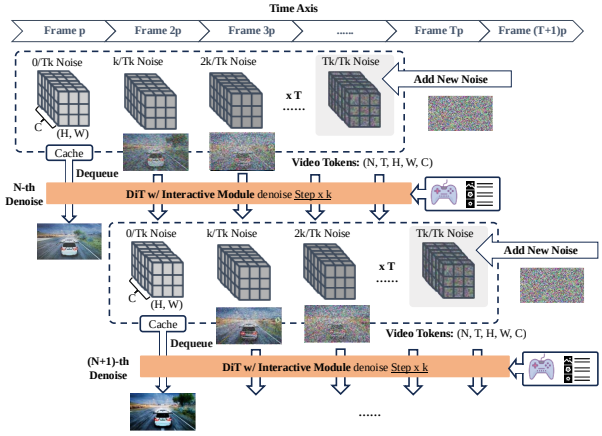
Interactive Module. The Interactive Module consists of an Embedding block (see Fig. 3a) and a cross-attention layer. Its primary function is to translate keyboard inputs into natural language that guides video generation. For example,

pressing 'W' is interpreted as "The car is driving forward" in the *Forza Horizon 5* scenario, or as "The man is moving forward and looking up" when combined with an upward mouse movement in *Cyberpunk 2077*. For unlabeled real or game data, we apply a default description: "The camera is moving in an unknown way." To enhance robustness, we randomly replace labeled keyboard inputs with this default sentence during training with probability $q = 0.1$.

To prepare for training, we first warmup the base DiT model for a few epochs using collected game and real-world data, fine-tuning a LoRA weight [17]. This process ensures that the Interactive Module focuses on learning interactions and movement patterns rather than simply fitting the video.



(a) **The Interactive Module:** After every two DiT blocks, the module merges the keyboard inputs into the video token feature through a **Causal Cross-Attention Layer**, where each keyboard input is limited to influence only the current and subsequent ω tokens. Here, every p frames are condensed into a single token.



(b) **Shift-Window Denoising Process Model:** The Swin-DPM transforms the traditional diffusion process into a streaming one, where T video tokens with different noise levels are denoised simultaneously. After each token is fully denoised and dequeued for decoding, a new token of pure noise is added to the queue. The dequeued token is then copied to the cache, allowing it to continue participating in attention computations until the next token is dequeued.

Figure 3. Main components of *The Matrix*.

Once translated, these natural language descriptions are processed by a T5 encoder [27] and transformed into a vector embedding through two linear layers and a SiLU layer [6] between them. This vector embedding is then concatenated with its corresponding video token and the next ω video tokens, where ω is a pre-defined causal relation range, typically set to $\omega = 4$, as is shown in Fig. 3a.

We perform this cross-attention operation each time the DiT model completes an odd-numbered self-attention step, enabling effective information exchange across frames and achieving precise, frame-level control for video generation.

Shift-Window Denoising Process Model. Typical DiT models are limited to generating only a few seconds of video, even when substantial spatial and temporal compression is applied via VAEs. This limitation is largely due to the high computational cost and memory demands of attention mechanisms over extended time durations. To address this, it becomes crucial to assume that temporal dependencies are confined within a limited time window, beyond which attention computations are unnecessary. Building on this idea, we propose the Shift-Window Denoising Process Model (Swin-DPM), which leverages a sliding temporal window to manage dependencies effectively and enables the generation of long or even infinite videos by producing tokens with a stride of $s = 1$. As is shown in Fig. 3b, within each window, a queue of video tokens undergoes denoising at various noise levels. After k denoising steps (where $k \times T$ is the number of diffusion solver steps), the leftmost, lowest-noisy token is dequeued into a cache. To maintain the queue length, a new token with Gaussian noise will be then added to the rightmost position. Each cached token is re-appended to the window's token queue at noise level 0 until the next token is cached, allowing it to continue participating in denoising and ensuring continuity between different windows. The network of Swin-DPM is fine-tuned from a pre-trained DiT model. During training, we sample $2w$ video tokens, where w is the window size. We usually set $w = T$. The first w tokens are used solely for warming up Swin-DPM and do not participate in backpropagation; loss is computed only on the last w tokens. At inference time, we follow the same setup: the first w tokens are for warmup and are discarded, with the generated video starting from the $(w + 1)$-th token.

Stream Consistency Model. After extending the DiT model to Swin-DPM, we further address the need for achieving real-time rendering of the simulated world. A promising approach is to combine Swin-DPM with Consistency Models [32, 33], a leading method for accelerating diffusion. We use the Stream Consistency Model (SCM) [22], which distills the original diffusion process and its class-free guidance into a four-step consistency model while incorporating the denoising window design from Swin-DPM. The training procedure is illustrated in Fig. 2. This integration results in a 10 - 20× acceleration in inference speed, reaching a rendering rate of 8 - 16 FPS.

Table 2. Ablation study on the components of *The Matrix*. Note that there is a trade-off between inference speed, control precision, and rendering quality. **Move-LPIPS** and **Move-PSNR** are computed between the generated videos and test videos with ground truth movements.

Component	Scene	#Params	Inference Speed	FVD↓	FID↓	CLIP↑	Move-LPIPS↓	Move-PSNR↑
DiT Backbone	-	2.3B	48 frames / 34 Seconds	1016.30	318.10	0.30	-	-
+ Warmup	*Cyberpunk 2077*	2.3B	64 frames / 34 Seconds	1429.45	183.24	0.28	0.125	27.80
	DROID	2.3B	48 frames / 34 Seconds	1133.16	224.98	0.29	0.191	27.72
	Forza Horizon 5	2.3B	48 frames / 34 Seconds	1891.67	141.11	0.31	0.128	26.89
+ Interactive Module	*Cyberpunk 2077*	2.7B	48 frames / 55 Seconds	1112.49	173.31	0.28	0.129	28.24
	DROID	2.7B	48 frames / 55 Seconds	1200.82	237.66	0.30	0.180	27.90
	Forza Horizon 5	2.7B	48 frames / 55 Seconds	1211.30	119.20	0.27	0.125	28.98
+ Swin-DPM	*Forza Horizon 5*	2.7B	0.8 FPS	1651.50	163.27	0.24	0.113	29.90
+ SCM	*Forza Horizon 5*	2.7B	8 - 16 FPS	1936.79	153.80	0.23	0.109	29.73

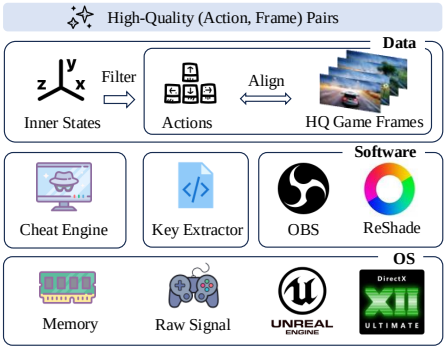


Figure 4. **The *GameData* Platform** that creates the *Source* dataset. It uses *CheatEngine* to capture in-game status from CPU memory and filter out unsatisfactory frames, such as those with stuck characters or irregular movements. *Reshade* removes game UIs and HUDs to ensure a more consistent data distribution. A Key Extractor then captures keyboard inputs and aligns them with frames recorded by *OBS*.

3.2. Construction of the *Source* Dataset

To train *The Matrix* model, we construct the Synthesized Observations of Unreal Rendered Contextual Environments (*Source*) dataset, which consists of two components: synthetic game data from Unreal Engine and real-world, unlabeled footage. The synthetic game data, collected using the *GameData* Platform, serves as supervised training data for precise motion control, while the real-world footage improves the model's visual quality and generalization to real-world scenarios.

After collection, the data is segmented into 6-second clips of continuous scenes and captioned using GPT-4o [19], resulting in a dataset of 750k labeled samples and 1.2 million unlabeled samples, all with 60 FPS. The labeled game data is further refined to ensure a balanced distribution of all possible game states. For more details on the dataset, see *Appendix Section B.2*.

The *GameData* Platform. As shown in Fig. 4, the *GameData* Platform is built on open-source tools: *Cheat Engine* software [7], the *Reshade* plugin [29] for DirectX, and *OBS Recording* software [28]. *Cheat Engine* is used to capture in-game world status data, such as character (x, y, z) positions and camera movements. This status data is aligned with recorded video frames to create per-frame action-video pairs and is also used to check if the character or camera is stuck and requires a reboot. We employ the *Reshade* plugin to remove all game UIs and HUDs and to standardize shading styles, providing a more consistent, low-complexity data source. Data for *Forza Horizon 5* is collected using autonomous scripts with random walking algorithms, while *Cyberpunk 2077* data is gathered manually with human operators running the *GameData* Platform. See *Appendix Section B.1* for more details on the *GameData* Platform.

4. Experiments

Training Details. We train *The Matrix* on the *Source* dataset, using a pre-trained 2.3B parameter DiT model as the backbone, which generates 4 video tokens per second, each decoded into 4 frames by the VAE decoder [21]. To match this generation rate, we downsample the videos and keyboard inputs in the *Source* dataset accordingly. For all training cases, we first warm up the base DiT model on unlabeled *Source* data for 20,000 steps with a batch size of 32. Following this, we train the Interactive Module on labeled *Source* data for an additional 20,000 steps with the same batch size, introducing another 0.4B parameter. Next, we fine-tune *The Matrix* model using Swin-DPM over 60,000 steps, also with a batch size of 32. For the final Consistency Model distillation, we use the Swin-DPM checkpoints as a teacher model and train the student network for 10,000 steps with a batch size of 32. More details can be found in *Appendix Section A.2*.

Metrics. We evaluate performance using metrics for both general visual quality and movement control precision. For general visual quality, we use Fréchet Inception Distance (FID) [15], Fréchet Video Distance (FVD) [35], and CLIP Score [26] to assess text alignment. All metrics are evaluated on 2,048 seconds of randomly generated videos. To evaluate movement control precision, we generate 2,048 seconds of video based on keyboard inputs and text prompts from a fixed test set, then measure the Peak Signal-to-Noise Ratio (Move-PSNR) [34] and Learned Perceptual Image

Figure 5. The results demonstrate frame-level precise control achieved by the Interactive Module across diverse scenes, weather conditions, and movement modes. **Click the first frame to play with Adobe Acrobat Reader!**

Patch Similarity (Move-LPIPS) [40] between the generated videos and real videos with ground truth movements.

4.1. Precise Frame-Level Interactions

In this section, we evaluate the effectiveness of the Interactive Module by testing its performance in three distinct scenarios: the *Forza Horizon 5* car driving scenario, the *Cyberpunk 2077* city walking scenario, and a robotic arm task from the *DROID* dataset [20]. We select 50,000 6-second clips from the *DROID* dataset, along with per-frame action labels of joint angles for seven joints, to form the training dataset. More details can be found in *Appendix Section B.3*. The third scenario is specifically designed to assess the effectiveness of *The Matrix* in embodied AI tasks. For all scenarios, we follow the same training strategy: starting with a pre-trained DiT model, we first perform a warm-up using unlabeled data, followed by fine-tuning the Interactive Module with labeled data.

Qualitative Results. Fig. 5 illustrates examples of *The Matrix*'s generated outputs across all scenarios. *The Matrix* demonstrates the ability to create vivid and dynamic worlds, accurately reflecting user interactions and intentions. It also models the physical behaviors within these environments, such as dust being kicked up when a car drives through a dry desert, or water splashing when it travels through a river. Additional examples of *The Matrix*'s generation capabilities are provided in *Appendix Section C.1*.

Quantitative Results. The last two columns of Tab. 2 present the quantitative evaluation of interaction precision, using LPIPS and PSNR metrics. The results demonstrate that Interactive Module significantly improves control precision, and this enhancement is maintained throughout the subsequent Swin-DPM and SCM processes.

4.2. Infinete-Horizon World Generation

Traditional world simulators focused on precise control often rely on small, auto-regressive generators trained from scratch to minimize the significant memory and time costs associated with pre-trained DiT models. However, this approach compromises visual quality and limits the full potential of world simulators. In this work, we introduce the first world simulator leveraging pre-trained video diffusion models, enabling infinite-length world generation with real-time rendering capabilities. In this section, we present our evaluation of these advancements.

(a) Long 1-minute video generated by *The Matrix*.

(b) A continuous 2.5-minute video generated by *The Matrix*, spanning multiple diverse scenes controlled through DiT text prompts.

Figure 6. Long worlds generation results by *The Matrix*. More examples are included in *Supplementary Videos*.

Generating Infinitely Long Videos. Fig. 6 showcases examples of generating 1-minute long worlds across diverse scenarios, including desert, river, grassland, snow, and day-to-night transitions. During generation, we switch the DiT prompt to adapt the environment, as shown in Fig. 6b. *The Matrix*'s capability extends beyond this; it can generate truly **infinite-length videos**, with additional **half-hour** examples available in *Supplementary Videos*. Tab. 2 reports the video quality and control precision of *The Matrix* after training with Swin-DPM. While some visual quality is sacrificed, control precision remains strong, and the visual quality still surpasses previous world simulators, achieving a realistic AAA-level standard.

Real-Time Rendering. We further investigate integrating SCM with *The Matrix*. As reported in Tab. 2, this integration highlights *The Matrix*'s real-time rendering capability,

with a slight trade-off in visual quality and minimal loss in control precision, while significantly improving rendering speed from 0.8 FPS to 8 - 16 FPS.

4.3. Generalization to Out-of-Distribution Worlds

In addition to superior visual quality, a key advantage of using pre-trained video DiTs is their inherent ability to generalize across diverse scenes. We observe impressive generalization in *The Matrix*, showcasing the potential of future research into building world simulators with pre-trained DiTs.

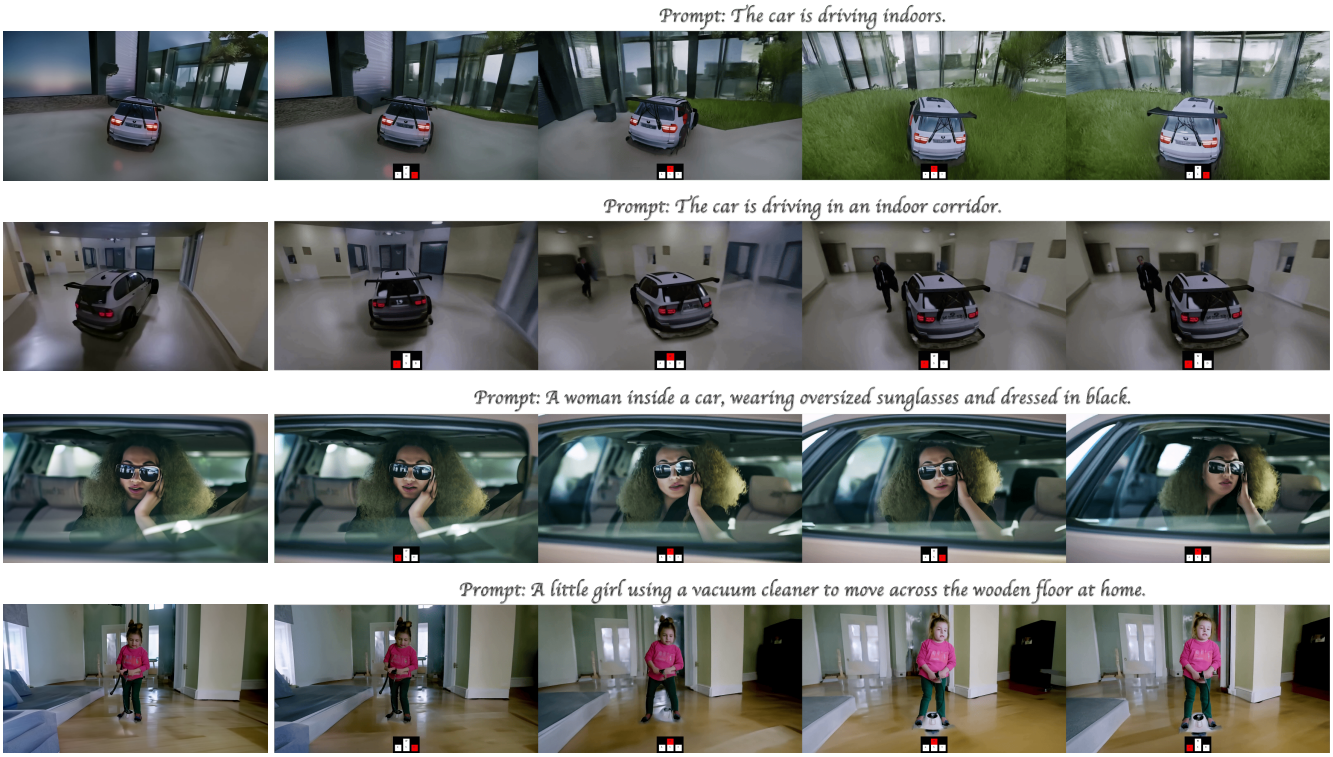
Generating Unseen Scenes. With *The Matrix*, we can control a car in previously unseen scenes by describing the scenario in the prompt. The first two rows of Fig. 7a demonstrate this capability, where the car is driven through indoor environments, which were not part of the *Source* dataset.

Interacting with Unseen Objects. A more remarkable feature is *The Matrix*'s ability to generalize interaction with real-world objects. As shown in the last two rows of Fig. 7a, by specifying a human as the center object in the DiT prompt, we can make the person move in response to keyboard inputs.

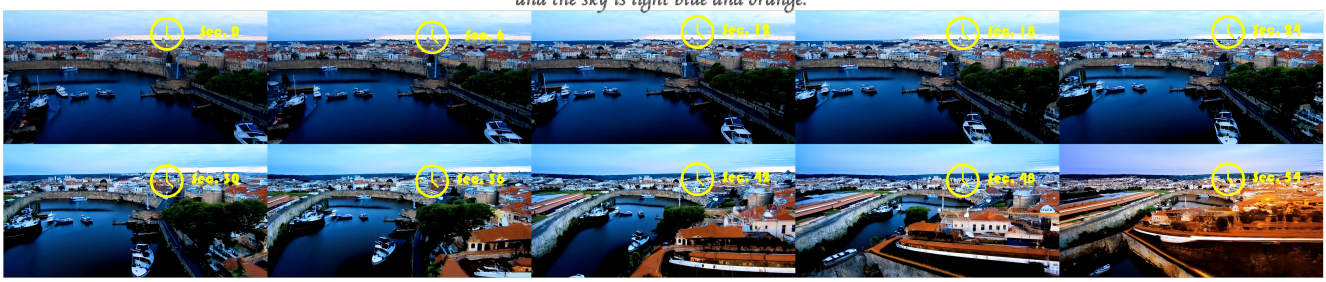
Generating Long Videos without Moving Control. Though *The Matrix* is trained on the *Source* dataset, it can also function as a general long video generator. By disabling the Interactive Module and using only the DiT backbone trained after Swin-DPM, *The Matrix* can generate long videos corresponding to ordinary prompts. Fig. 7b shows such an example, further proving *The Matrix*'s strength as a realistic world simulator.

5. Conclusion

We introduce *The Matrix*, a real-world simulator capable of generating infinitely long, high-fidelity video streams with precise real-time control. Trained on a blend of AAA game data and real-world footage, *The Matrix* supports immersive exploration of dynamic environments, with zero-shot generalization to unseen scenarios. Operating at 8 - 16 FPS, it enables continuous, interactive simulations across diverse terrains, bridging the gap between virtual and real-world applications. This work highlights the potential of using game data to build robust world models with minimal supervision, and showcases the power of pre-trained video DiTs in enabling realistic, large-scale simulations.



(a) *The Matrix* can generalize its precise movement control to unlabeled scenes and objects, such as driving indoors or making people move as instructed. **Click the first frame to play with Adobe Acrobat Reader!**



(b) *The Matrix* can also generate long, general videos by disabling the Interactive Module, acting as a powerful video generator.

Figure 7. Generalization ability of *The Matrix* on unseen scenes and objects.

References

[1] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. *arXiv preprint arXiv:2405.12399*, 2024. 2

[2] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning*, 2024. 2

[3] Maxime Burchi and Radu Timofte. MuDreamer: Learning predictive world models without reconstruction. *arXiv preprint arXiv:2405.15083*, 2024. 2

[4] Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. GameGen-X: Interactive open-world game video generation. *arXiv preprint arXiv:2411.00769*, 2024. 2

[5] Decart. Oasis: A universe in a transformer. *Preprint*, 2024. 2

[6] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018. 4

[7] Cheat Engine. https://www.cheatengine.org/, 2024. Software. 5

[8] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018. 2

[9] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019. 2

[10] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020. 2

[11] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. 2

[12] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022. 2

[13] Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023. 2

[14] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[16] 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. 2

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 4

[18] Weidong Huang, Jiaming Ji, Borong Zhang, Chunhe Xia, and Yaodong Yang. SafeDreamer: Safe reinforcement learning with world models. *arXiv preprint arXiv:2307.07176*, 2023. 2

[19] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 5

[20] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 6

[21] Diederik P Kingma. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013. 5

[22] Akio Kodaira, Chenfeng Xu, Toshiki Hazama, Takanori Yoshimoto, Kohei Ohno, Shogo Mitsuhori, Soichi Sugano, Hanying Cho, Zhijian Liu, and Kurt Keutzer. Streamdiffusion: A pipeline-level solution for real-time interactive generation. *arXiv preprint arXiv:2312.12491*, 2023. 4

[23] OpenAI. Sora: Creating video from text. *Preprint*, 2024. 2

[24] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 3

[25] Virtuals Protocol. Video game generation: A practical study using Mario. *Preprint*, 2024. 2

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5

[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 4

[28] OBS Recording. https://obsproject.com/, 2024. Software. 5

[29] Reshade. https://reshade.me/, 2024. Software. 5

[30] Jürgen Schmidhuber. On learning to think: Algorithmic information theory for novel combinations of reinforcement learning controllers and recurrent neural world models. *arXiv preprint arXiv:1511.09249*, 2015. 2

[31] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. 2

[32] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, 2024. 4

[33] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 4

[34] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis and machine vision*. Springer, 2013. 5

[35] 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. 5

[36] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024. 2

[37] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. DayDreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023. 2

[38] Jiannan Xiang, Guangyi Liu, Yi Gu, Qiyue Gao, Yuting Ning, Yuheng Zha, Zeyu Feng, Tianhua Tao, Shibo Hao, Yemin Shi, et al. Pandora: Towards general world model with natural language actions and video states. *arXiv preprint arXiv:2406.09455*, 2024. 2

[39] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023. 2

[40] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[41] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-Sora: Democratizing efficient video production for all, 2024. 3