SUPPLEMENTARY MATERIAL FOR SLOWFAST-VGEN: SLOW-FAST LEARNING FOR ACTION-DRIVEN LONG VIDEO GENERATION

Yining Hong[1], **Beide Liu**[1]†, **Maxine Wu**[1]†, **Yuanhao Zhai**[3], **Kai-Wei Chang**[1], **Linjie Li**[2], **Kevin Lin**[2], **Chung-Ching Lin**[2], **Jianfeng Wang**[2], **Zhengyuan Yang**[2,††], **Yingnian Wu**[1,††], **Lijuan Wang**[2,††]
[1] UCLA, [2] Microsoft Research, [3] State University of New York at Buffalo
† Co-second Contribution, †† Equal Advising

CONTENTS

A CONTRIBUTION STATEMENT

Yining Hong was responsible for all of the code development, paper writing, and experiments. She also collected the data for Minecraft.

Beide Liu contributed to most of the data collection with regard to Unreal data. He was in charge of setting up the Unreal Engine, purchasing assets online, writing the Python scripts for automate agent control, and recording first-person and third-person videos of Unreal data.

Maxine Wu collected the data of Google 3D Tiles. She was also responsible for the task setup of RLBench and the data collection of RLBench. She also curated part of the driving data.

Yuanhao Zhai wrote the codes for AnimateDiff, which was one of the baseline models.

The other people took on the advising roles, contributing extensively to the project by offering innovative ideas, providing detailed technical recommendations, assisting with troubleshooting code issues, and conducting multiple rounds of thorough paper reviews. They provided valuable expertise on video diffusion models. **Zhengyuan Yang, Yingnian Wu and Lijuan Wang** were involved in brainstorming and critical review throughout the project. Specifically, **Zhengyuan Yang** provided much technical support. **Yingnian Wu** came up with the idea of TEMP-LORA for modelling episodic memory as well as the masked video diffusion model. **Lijuan Wang** provided valuable insights throughout the project.

B MORE DETAILS ABOUT THE METHOD

B.1 PRELIMINARIES ON LATENT DIFFUSION MODELS

Stable Diffusion (Rombach et al., 2022), operates in the compressed latent space of an autoencoder obtained by a pre-trained VAE . Given an input x_0, the process begins by encoding it into a latent representation: $z_0 = E(x_0)$ where E is the VAE encoder function. Noise is then progressively added to the latent codes through a Gaussian diffusion process:

$$q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{1-\beta_t}z_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

for $t = 1, \ldots, T$, where T is the total number of diffusion steps and β_t are noise schedule parameters. This iterative process can be expressed in a simpler form:

$$z_t = \sqrt{\bar{\alpha}_t}z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{2}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, and $\alpha_i = 1 - \beta_i$. Stable Diffusion employs an ϵ-prediction approach, training a neural network ϵ_θ to predict the noise added to the latent representation. The loss function is defined as:

$$L = \mathbb{E}_{t, z_0 \sim p_{\text{data}}, \epsilon \sim \mathcal{N}(0,1), c}\left[||\epsilon - \epsilon_\theta(z_t, t, c)||_2^2\right] \tag{3}$$

Here, c represents the conditioning (e.g., text), and θ denotes the neural network parameters, typically implemented as a U-Net (Ronneberger et al., 2015).

During inference, the model iteratively denoises random Gaussian noise, guided by the learned ϵ_θ, to generate latent representations. These are then decoded to produce high-quality images consistent with the given textual descriptions.

Video diffusion models (Ho et al., 2022) typically build upon LDMs by utilizing a 3D U-Net architecture, which enhances the standard 2D structure by adding temporal convolutions after each spatial convolution and temporal attention blocks following spatial attention blocks.

B.2 PRELIMINARIES ON LOW-RANK ADAPTATION (LORA)

LoRA Hu et al. (2021) transforms the fine-tuning process for large-scale models by avoiding the need to adjust all parameters. Instead, it utilizes compact, low-rank matrices to modify only a subset of the model's weights. This approach keeps the original model parameters fixed, addressing the problem of catastrophic forgetting, where new learning can overwrite existing knowledge. LoRA

utilizes compact, low-rank matrices to modify only a subset of the model's weights, therefore avoiding the need to adjust all parameters. In LoRA, the weight matrix $W \in \mathbb{R}^{m \times n}$ is updated by adding a learnable residual. The modified weight matrix W' is:

$$W' = W + \Delta W = W + AB^T$$

where $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{n \times r}$ are low-rank matrices, and r is the rank parameter that determines their size. In this paper, we denote the LoRA finetuning as the fast learning process and the pre-training as slow learning process. The equation then becomes:

$$W' = W + \Delta W = W_{\text{slow}} + W_{\text{fast}} = \Phi + \Theta \tag{4}$$

where Φ corresponds to the pre-trained slow-learning weights, and Θ corresponds to the LoRA paraemters in the fast learning phase.

B.3 MODELSCOPET2V DETAILS

We base our slow learning model on ModelscopeT2V (Wang et al., 2023). Here, we introduce the details of this model.

Given a text prompt p, the model generates a video v_{pr} through a latent video diffusion model that aligns with the semantic meaning of the prompt. The architecture is composed of a visual space where the training video v_{gt} and generated video v_{pr} reside, while the diffusion process and denoising UNet ϵ_θ operate in a latent space. Utilizing VQGAN, which facilitates data conversion between visual and latent spaces, the model encodes a training video $v_{gt} = [f_1, \ldots, f_F]$ into its latent representation $Z_{gt} = [E(f_1), \ldots, E(f_F)]$. During the training phase, the diffusion process introduces Gaussian noise to the latent variable, ultimately allowing the model to predict and denoise these latent representations during inference.

To ensure that ModelScopeT2V generates videos that adhere to given text prompts, it incorporates a text conditioning mechanism that effectively injects textual information into the generative process. Inspired by Stable Diffusion, the model augments the UNet structure with a cross-attention mechanism that allows for conditioning of visual content based on textual input. The text embedding c derived from the prompt p is utilized as the key and value in the multi-head attention layer, enabling the intermediate UNet features to integrate text features. The text encoder from the pre-trained CLIP ViT-H/14 converts the prompt into a text embedding, ensuring a strong alignment between language and vision embeddings.

The core of the latent video diffusion model lies in the denoising UNet, which encompasses various blocks, including the initial block, downsampling block, spatio-temporal block, and upsampling block. Most of the model's parameters are concentrated in the denoising UNet ϵ_θ, which is tasked with the diffusion process in the latent space. The model aims to minimize the discrepancy between the predicted noise and the ground-truth noise, thereby achieving effective video synthesis through denoising. ModelScopeT2V's architecture also includes a spatio-temporal block, which captures complex spatial and temporal dependencies to enhance video synthesis quality. The spatio-temporal block is comprised of spatial convolutions, temporal convolutions, and attention mechanisms. By effectively synthesizing videos through this structure, ModelScopeT2V learns comprehensive spatio-temporal representations, allowing it to generate high-quality videos. The model implements a combination of self-attention and cross-attention mechanisms, facilitating both cross-modal interactions and spatial modeling to capture correlations across frames effectively.

C DATASET STATISTICS

We provide the dataset statistics in Figure 1.

D MORE EXPERIMENTAL DETAILS

D.1 EXPERIMENTAL SETUP AND IMPLEMENTATION DETAILS

We utilize approximately 64 V100 GPUs for the pre-training of SLOWFAST-VGEN, with a batch size of 128. The slow learning rate is set to 5e-6, while the fast learning rate is 1e-4. Training

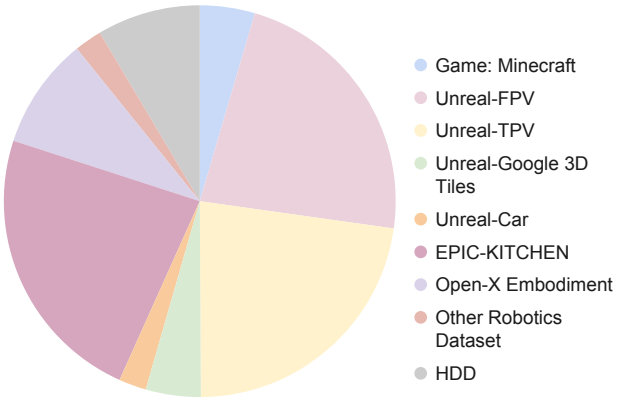

Figure 1: **Statistics of our Training Dataset.**

videos of mixed lengths are used, all within the context window of 32 frames. During training, we freeze the VAE and CLIP Encoder, allowing only the UNet to be trained. For inference and fast learning, we employ a single V100 GPU. For TEMP-LORA, a LoRA rank of 32 is used, and the Adam optimizer is employed in both learning phases.

D.2 COMPUTATION COSTS

In Table 1, we show the computation costs with and without TEMP-LORA. While the inclusion of TEMP-LORA does introduce some additional computation during the inference process, the difference is relatively minor and remains within acceptable limit.

	Ours wo TEMP-LORA	Ours w TEMP-LORA
Average Inference Time per Sample (seconds)	12.9305	13.8066
Inference Memory Usage (MB)	9579	9931

Table 1: **Comparison of Computation Costs with and without TEMP-LORA**

D.3 HUMAN EVALUATION DETAILS

In our human evaluation session for action-conditioned long video generation, 30 participants assessed the generated video samples (50 videos per person) based on three criteria:

- Video Quality (0 to 1): Participants evaluated the overall visual quality, considering aspects such as resolution, clarity, and aesthetic appeal.
- Coherence (0 to 1): They examined the logical flow of actions and whether the events progressed seamlessly throughout the video, ensuring there were no abrupt changes or disconnections.
- Adherence to Actions (0 to 1): Participants judged how accurately the generated videos reflected the specified action prompts, assessing whether the actions were effectively depicted.

Each video was rated by at least three different individuals to ensure reliability. The collected ratings were then compiled for analysis, with average scores calculated to assess performance across the different criteria.

E EXPERIMENTS ON ABLATIONS AND VARIATIONS OF SLOWFAST-VGEN

We introduce several variations of SLOWFAST-VGEN, including:

	SCuts ↓	SRC ↑
Our (w original TEMP-LORA)	0.55	92.24
Ours (wo Local Learning Rule)	**0.36**	90.27
Ours (wo Chunk Input)	1.24	90.01
Ours (wo/ Temp-LoRA)	1.88	89.04
Ours SLOWFAST-VGEN	0.37	**93.71**

Table 2: **Scene Cuts and SRC Scores**. Comparison of scene cuts and SRC scores for our method with and without Temp-LoRA.

- **Ours wo Chunk Input** that only conditions on single-frame images instead of previous chunk
- **Ours wo Local Learning Rule** that samples over the whole generated sequence for training TEMP-LORA, instead of using local inputs and outputs to train.
- **Ours w original TEMP-LORA** that uses the original TEMP-LORA structure that were designed for long text generation.

We show the results below. From the table, we can see that SLOWFAST-VGEN trained over sampled full sequence also shows good performances. However, our observation indicates that this method tends to over-smooth the generated sequences, leading to blurry videos for later frames.

F MORE QUALITATIVE EXAMPLES

F.1 MORE QUALITATIVE EXAMPLES OF SLOW LEARNING

In Figure 2 and Figure 3, we include more qualitative examples with regard to slow learning.

F.2 MORE QUALITATIVE EXAMPLES OF FAST LEARNING

In Figure 4 and Figure 5, we include more qualitative examples with regard to fast learning.

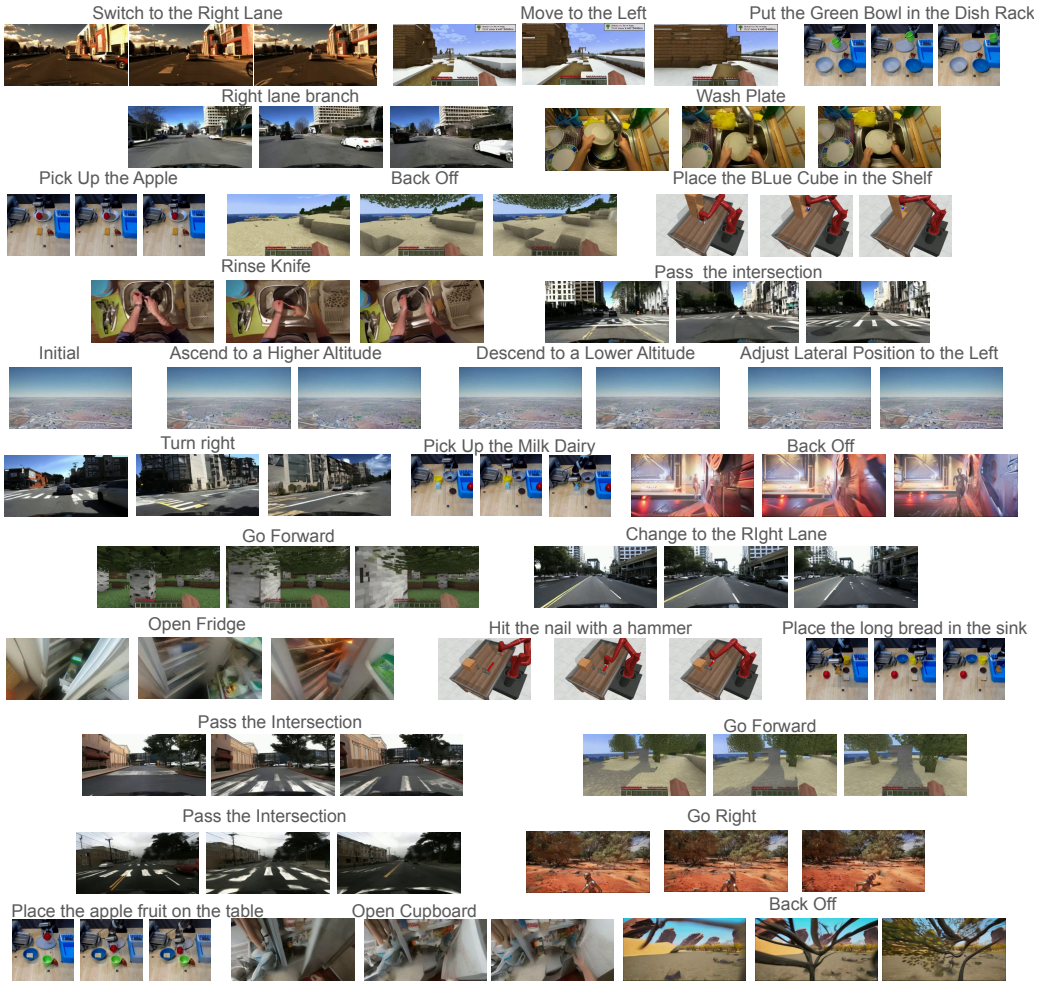

Figure 2: Qualitative Examples on Slow Learning. Part 1.

Figure 3: Qualitative Examples on Slow Learning. Part 2.

Figure 4: Qualitative Examples on Fast Learning. Part 1. We mark consistent objects / frames in green bounding boxes and inconsistent ones in red.

Figure 5: Qualitative Examples on Fast Learning. Part 2. We mark consistent objects / frames in green bounding boxes and inconsistent ones in red.

REFERENCES

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. URL https://arxiv.org/abs/2204.03458.

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, 2021. URL https://arxiv.org/abs/2106.09685.

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

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report, 2023. URL https://arxiv.org/abs/2308.06571.