Real-Time Atari Game Simulation With Diffusion Models

組員： 李逵安　　組別： B-3　　指導教授： 朱威達

Abstract

We implement **GameNGen**[1], a neural game engine based on diffusion models, for real-time simulation in resource-limited settings. Unlike the original DOOM-on-TPU setup, we target simpler Atari games like **Pong** and **Breakout**. Using **RL-collected**[2] (frame, action) data, we train a modified **Stable Diffusion**[2] model to predict future frames. Techniques like noise-augmented conditioning and truncated context enable stable long-horizon generation. Our results show that diffusion-based game simulation is feasible even on limited hardware. We release our source code at https://github.com/jerrykal/GameNGen-Atari.

Method

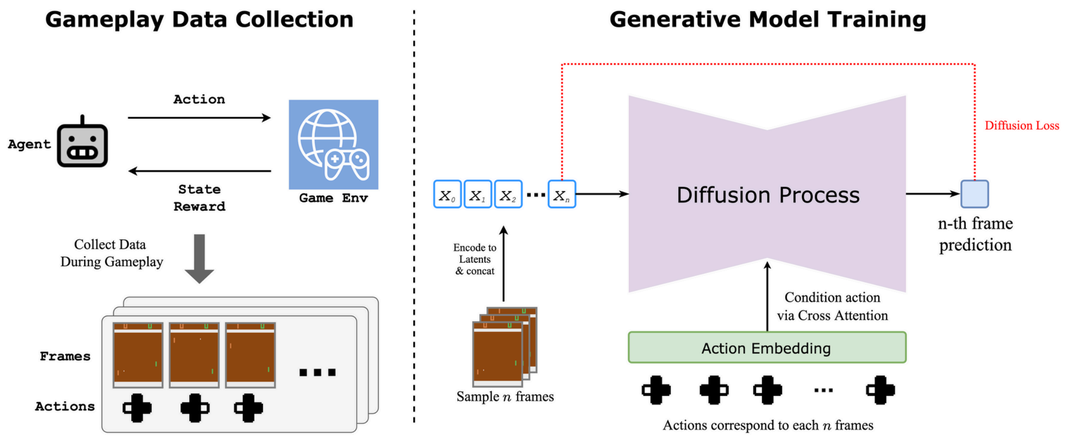

Figure 1: Training overview

Our training pipeline consists of two stages: **data collection** and **generative model training** (see **Figure 1**).

1. Data Collection

We train an RL agent to interact with Atari games (e.g., Pong), recording sequences of frames and actions as training data.

2. Generative Model Training

A modified **Stable Diffusion** model is trained to predict the next frame given past frames and actions.

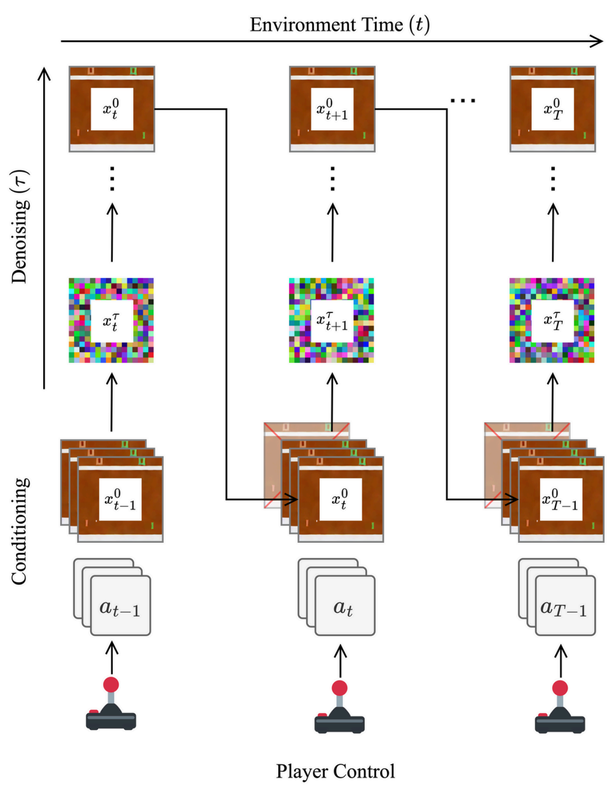

Figure 2: Inference overview

- **Frames** are encoded into latents and concatenated; **actions** are embedded and conditioned via cross-attention.
- **Noise augmentation** is applied to context frames to improve autoregressive stability during training.

During **inference** (see **Figure 2**), the model recursively generates future frames by denoising latent predictions, conditioned on its own previous outputs and player actions.

Results

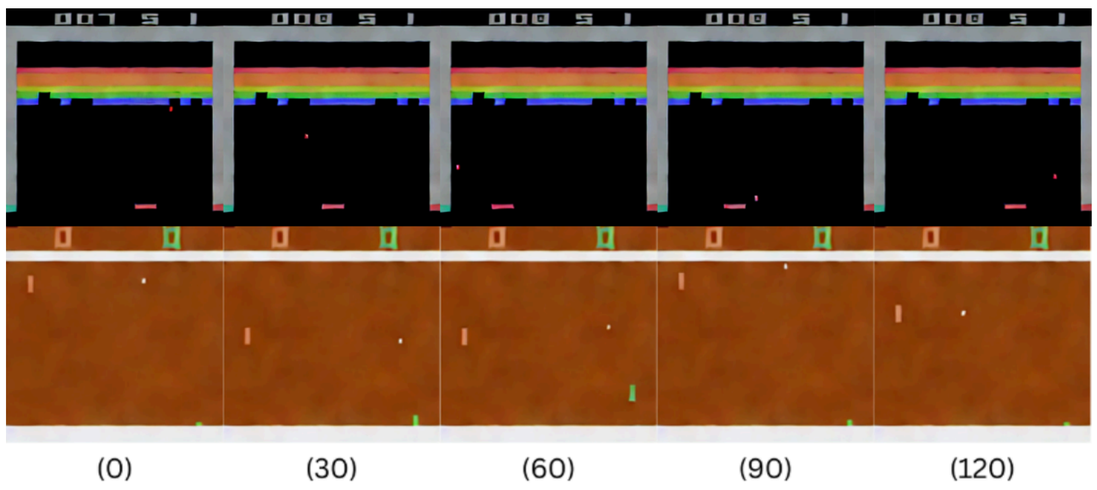

Figure 3: A trajectory of simulated Pong and Breakout gameplay conditioned on groudtruth data, with every 30th frame displayed across a total of 120 frames.

	Ours(4 steps)	Ours(8 steps)	JPEG Comp.
Breakout			
PSNR ↑	34.7± 0.571	34.67± 0.525	**35.66± 0.462**
LPIPS ↓	0.083± 0.012	**0.05± 0.011**	0.116± 0.011
Pong			
PSNR ↑	33.59 ± 0.115	34.65 ± 0.133	**35.63 ± 0.074**
LPIPS ↓	0.128 ± 0.006	**0.047 ± 0.005**	0.07 ± 0.003

Table 1: PSNR and LPIPS scores of our model under different denoising steps compared with lossy JPEG compression (quality ≈ 20).

We evaluated our diffusion-based simulator on Pong and Breakout. The model was trained for 70,000 steps with a batch size of 16 on a single NVIDIA RTX 4090 GPU machine. Visual fidelity was assessed on 2,048 held-out sequences using PSNR and LPIPS.

As shown in **Figure 2** and **Table 1**, increasing the number of denoising steps improves perceptual quality. With just 4 to 8 steps, our model produces predictions comparable to JPEG compression (quality level 20), achieving high PSNR and low LPIPS across both games.

Conclusion

We show that diffusion models can simulate game environments in real time with high visual quality. Our results on Atari games match JPEG-level fidelity with just a few denoising steps.

Due to hardware limits, our model uses only **4 context frames**, leading to occasional inconsistent **physics** and missed **internal states** like round timers. Extending context remains a key challenge.

References:

[1] D. Valevski, Y. Leviathan, M. Arar, and S. Fruchter, "Diffusion Models Are Real-Time Game Engines," Aug. 27, 2024, arXiv: arXiv:2408.14837. doi: 10.48550/arXiv.2408.14837.

[2] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," Journal of Machine Learning Research, vol. 22, no. 268, pp. 1–8, 2021.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA: IEEE, Jun. 2022, pp. 10674–10685. doi: 10.1109/CVPR52688.2022.01042.