Exploration-Driven Generative Interactive Environments

Nedko Savov[1,†] Naser Kazemi[1] Mohammad Mahdi[1] Danda Pani Paudel[1]
Xi Wang[1,2,3] Luc Van Gool[1]
[1] INSAIT, Sofia University "St. Kliment Ohridski" [2] ETH Zurich [3] TU Munich

Abstract

Modern world models require costly and time-consuming collection of large video datasets with action demonstrations by people or by environment-specific agents. To simplify training, we focus on using many virtual environments for inexpensive, automatically collected interaction data. Genie [5], a recent multi-environment world model, demonstrates simulation abilities of many environments with shared behavior. Unfortunately, training their model requires expensive demonstrations. Therefore, we propose a training framework merely using a random agent in virtual environments. While the model trained in this manner exhibits good controls, it is limited by the random exploration possibilities. To address this limitation, we propose AutoExplore Agent - an exploration agent that entirely relies on the uncertainty of the world model, delivering diverse data from which it can learn the best. Our agent is fully independent of environment-specific rewards and thus adapts easily to new environments. With this approach, the pretrained multi-environment model can quickly adapt to new environments achieving video fidelity and controllability improvement.

In order to obtain automatically large-scale interaction datasets for pretraining, we group environments with similar behavior and controls. To this end, we annotate the behavior and controls of 974 virtual environments - a dataset that we name `RetroAct`*. For building our model, we first create an open implementation of Genie - GenieRedux and apply enhancements and adaptations in our version GenieRedux-G. Our code and data are available at https://github.com/insait-institute/GenieRedux.*

1. Introduction

Learning from interactive environments allows us to understand and represent the rules, the possible actions, and the consequences that govern them. As an alternative to laboriously hand-coded synthetic simulators, world models have emerged as deep learning tools for realistic environment modeling entirely from observations, commonly images of the observed environment [1, 5, 37, 65].

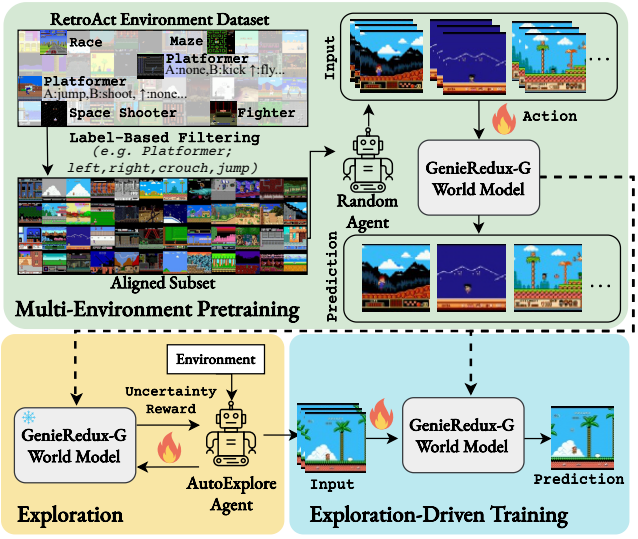

Figure 1. **Our proposed world model training framework.** It consists of a pretrained multi-environment world model on random agent data, and a new AutoExplore Agent that explores an environment and delivers diverse data for fine-tuning.

Previous work such as [19, 23, 33] uses light world models to support goal-driven agents with goal-specific state representations. The focus is on coarse future predictions, not on their high visual quality. In contrast, the objective of recent world models is to achieve high-quality future predictions given past observations and actions. Such recent models are able to offer realistic action execution and even real-time interaction with people [1, 58]. This has become possible with the rise of diffusion, transformers [14, 59], and state space models [17], and by borrowing architectural choices from video generation pipelines [51, 60]. Typically, these generative models are designed to closely match a single selected environment. One of the state-of-the-art models, Genie, distinguishes itself by being trained on many visually diverse environments with similar dynamics, thus

†Corresponding author. *firstname.secondname*@insait.ai

demonstrating generalization across new visuals.

Building these high-quality statistical simulators requires diverse observations of the environment as well as of the actions to simulate. Some obtain this data by costly video dataset collection and curation with human demonstrations of the actions [1, 5, 63]. If actions are unavailable, an extra component is designed to predict them, which can introduce uncertainty compared to ground truth labels [5, 37]. Extension to new environments with new types of actions in this setting is difficult as it requires again an expensive data collection process. Others, such as [58] have explored retrieving data with an environment-specific agent, in their case - the game Doom.

In this work, we propose a framework for accessible and effort-free training of world models in multiple environments. To this end, we first build `RetroAct`- an annotated and curated large dataset of retro game environments (based on the environments of Stable Retro [48]). We group them based on behavior labels and control descriptions. This grouping allows us to generate large-scale interaction datasets across environments with similar behaviors. Next, we pretrain a multi-environment world model GenieRedux - our open implementation of Genie [5], using a random agent. Unlike [66], which reports the agent's improved behavior from pretraining, we aim to improve the world model. For this, we adapt GenieRedux to virtual environments and implement architectural and training procedure enhancements, resulting in the GenieRedux-G model. We observe that just by training GenieRedux-G on random interactions from subsets of 200 environments and 50 environments with mapped controls, automatically collected from `RetroAct`, we are able to obtain control behavior (0.450 ΔPSNR in 50 environments) and reasonable visual fidelity (26.36 PSNR in 50 environments).

As random actions are limited in their ability to explore the environment, we develop a method to obtain more diverse interaction data to improve the control behavior and visual fidelity of our model. To this end, inspired by [53], we develop our own environment-independent reward function, allowing an agent to explore different environments, entirely without relying on predefined environment rewards. While they aim at a high-performing goal-driven agent, we base our design on improving the underlying large world model for the simulation of environments in terms of higher visual fidelity and improved controllability. For graphical illustrations, see Fig. 1. The objective of our exploration-driven agent is to maximize the world model's uncertainty, estimated by the classification entropy available in the observation prediction stage of GenieRedux-G. Once the diverse data is obtained, we fine-tune GenieRedux-G. We show that this method leads to significant visual (up to 7.4 PSNR) and control (up to 1.4 ΔPSNR) improvements, compared to random agent pretraining.

Our contributions are as follows:

- A framework for training world models with cheap data collection - by training an exploration agent based on our world's model uncertainty.
- The implementation and release of GenieRedux and GenieRedux-G - open Pytorch models based on [5].
- Architectural and loss changes to the model leading to fidelity improvements, based on our tokenizer representation study.
- Preparing a large scale environment dataset for multi-environment world model training.

2. Related Work

World models. Initially built as rough imagination models assisting reinforcement learning (RL) agents [10, 19, 21, 24, 53], world models have evolved into independent realistic video generation models conditioned on actions [9, 39, 50, 64]. They facilitate task-specific agent training by providing predictive representations of the environment. Inspired by [20], Ha and Schmidhuber [18] use a VAE to encode visual observations into latent states, with an MDN-RNN predicting future states based on prior states, actions, and VAE outputs to facilitate policy learning. DreamerV2 [22] introduce an RL agent, achieving human-level performance in Atari. It encodes images with a CNN and computes posterior and prior stochastic states using recurrent states. Unlike our work, though, it does not assess the agent's impact on world model improvement nor generalize task rewards across different environments.

World models also aim to generate realistic video conditioned on actions [27, 37, 65]. Genie [5] trains a video tokenizer and a Latent Action Model (LAM) for dynamic next-frame generation. GAIA-1 [27] tackle autonomous driving in unstructured settings by encoding multi-modal inputs into a unified representation and predicting image tokens based on prior inputs, using an autoregressive transformer. Menapace et al. [37] employ an encoder-decoder architecture in which the predicted action labels act as a bottleneck, allowing a user to control the generated video by a discrete action. The key gap in these works is automatic data collection, which is addressed in our approach.

Efficient exploration. The importance of efficient exploration in RL is highlighted by [28]. Early methods enhanced exploration by adding noise [16, 34] or using entropy regularization [40], but they have action space limitations and often fail with complex dynamics, where varied actions do not always drive meaningful exploration. A more direct approach uses heterogeneous actors [26, 29, 52] with diverse exploration strategies to enhance environment exploration. Bayesian methods [54, 57] have also been introduced to create acquisition functions for uncertainty-driven exploration [2, 38, 41–44], but often struggle to generalize to high-dimensional inputs like images.

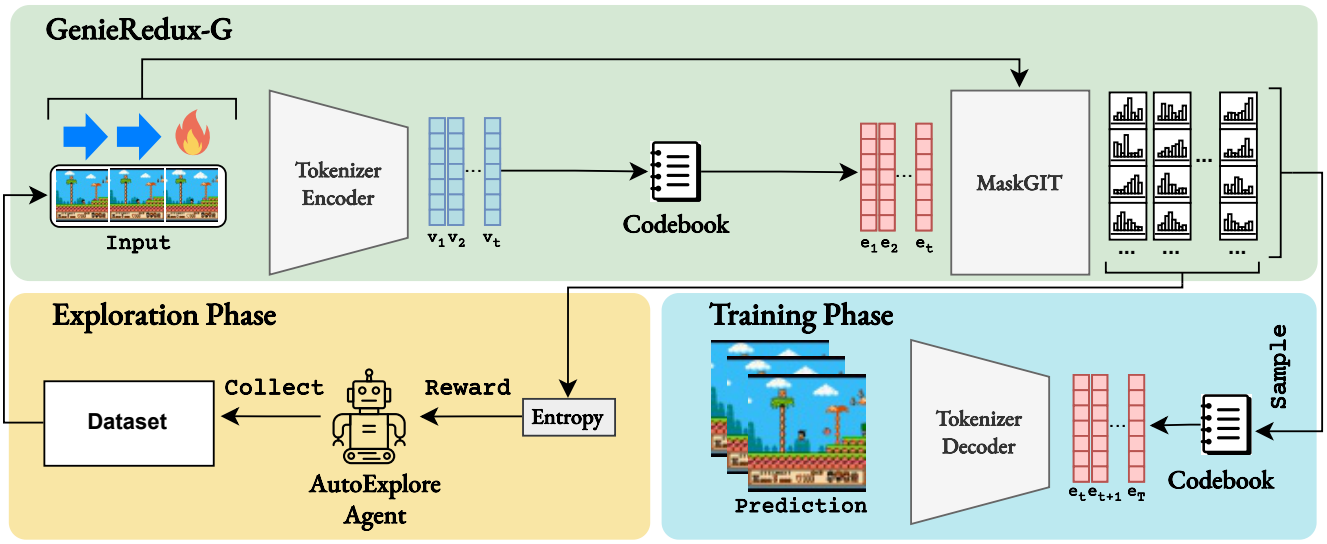

Figure 2. **Method Overview.** We propose an alternative to costly human interaction data collection - by exploring environments with an agent. The reward is solely based on the classification uncertainty of our model.

Recent exploration methods emphasize state novelty [3, 7, 11, 35, 36, 45, 55, 67], focusing on encouraging agents to assess novelty only after visiting states. In contrast, our approach, inspired by [7, 47, 53], uses model disagreement to proactively guide agents to states with the highest potential without environment target-driven reward. [6, 46] propose exploration agents driven by uncertainty in state transition and simple feature extractors. Instead, we propose an exploration agent designed to improve a world model that does not model states. Plan2Explore [53] enables agents to seek novel states using a reward that maximizes the state entropy of an RSSM model. While Plan2Explore improves goal-driven agents with their framework, we improve a modern transformer world model with a novel exploration-based reward using token uncertainty.

Rather than relying on world models, EX2 [15] learns a classifier to distinguish visited states, providing intrinsic rewards for states that are difficult for the classifier to differentiate. KL-divergence-based approaches [30–32], guide exploration by comparing distributions. For example, SMM [32] computes the KL divergence between the policy-induced state distribution and a uniform target. Tao et al. [56] propose an intrinsic reward based on the distance between a state and its nearest neighbors in a low-dimensional feature space. However, low-dimensionality leads to information loss, restricting full state space exploration — an issue we address by the use of a world model.

3. RetroAct Dataset

We first tackle the problem of accessible training of multi-environment world models by building a framework for cheaply acquiring multi-environment interaction data. In particular, we aim to collect interactions of similar actions

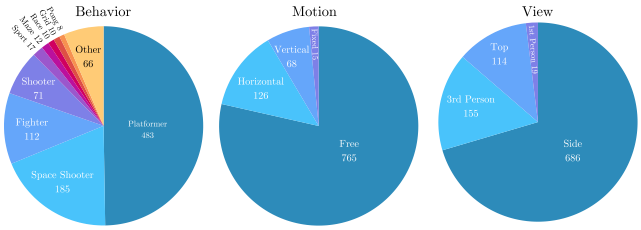

Figure 3. **`RetroAct` Annotation.** Description of environments in `RetroAct` by annotated attribute. Better viewed zoomed.

in many environments. Instead of relying on expensive human interaction, we obtain and curate a collection of virtual environments. As a source, we use the Stable Retro framework by [48], which is a collection of retro games across multiple platforms, with an accompanying starting state. We make no use of the defined rewards. We obtain almost all the supported games (974).

This raw set contains an environment mix of very different visuals and behaviors. However, in our setting of learning from similar dynamics, it is required to establish correspondence between the environments' behaviors. We perform annotation where for each environment three aspects are classified. The motion style classifies the general style of what and how is moved by the controls, closely relating to game genre; the camera viewpoint; the control axis describing in which direction the player can be moved. The label distributions are shown in Fig. 3. In Tab. 1 we compare our `RetroAct` with other related datasets. `RetroAct` distinguishes itself by providing behavior and control annotations, while maintaining a high number of environments.

It is discovered that the most prevalent type of environment in our set is the platformer - 483 titles. As the largest subset, we filter only these games for further use, as it is required to have many environments exhibiting similar con-

Table 1. **Comparison of `RetroAct` dataset to others.**

Dataset	Type	#Environments	Diverse Behaviors	Open	Behavior Annotation	Control Annotation
`Coinrun` [13]	Environments	1	✗	✓	✗	✗
`ALE` [4]	Environments	57	✓	✓	✗	✗
`Stable Retro` [48]	Environments	1003	✓	✓	✗	✗
`Platformers` [5]	Videos	Unknown	✓	✗	✗	✗
`RetroAct`(Ours)	Environments	974	✓	✓	✓	✓

trols. Five motion actions are defined for our model - moving `left`, `right`, `up`, `down` and `jump`. Each game has its own mapping of buttons to actions. Therefore, we generate a short clip of each of the 5 selected actions for each of the 483 titles and build an annotation tool to observe and annotate the executed action. Eventually, we annotated 2,925 behavior and 2,898 control labels.

After experimenting, we observed that models require more training with a higher number of environments, so we defined two subsets of our large set to handle computational cost: a subset consisting of the first 200 games of 483 behavior-filtered games for pretraining, and another subset of 50 randomly selected action-consistent games using RetroAct's action labels for fine-tuning.

We collect a large scale dataset by launching a random agent in all of the environments, collecting actions and observations. From the 200-game set we build `Platformers-200` - a dataset with 10,000 episodes (50 episodes per game) with 500 frames each at most, resulting in 4.6mln images. From the 50-game set we obtain `Platformers-50` - 5000 episodes (100 per game) of length at most 1000, resulting in 4.8mln images. In our protocol, we take 1% of the sessions of each environment as a validation set. We show that using a random agent is already sufficient to learn a level of controllability and later build on top with an exploration agent of our design.

To validate our GenieRedux implementation, we implement the CoinRun case study in [5]. Using the protocol from above, we obtain a dataset of 10k episodes with a maximum length of 500, resulting in 4mln images.

4. Multi-Environment World Model

Given virtual environments, our first goal is to automatically obtain a dataset of image sequences $I_1, ..., I_N$ and corresponding actions $a_1, ..., a_{N-1}$. Given a sequence $I_1, ..., I_N$ and past and future actions $a_1, ..., a_{N+T-1}$, our world model aims to predict the future T frames $I_{N+1}, ..., I_T$, corresponding to the actions executed.

GenieRedux. As Genie [5] is not made available by the authors, we create an open implementation and call it GenieRedux. We validate our implementation quantitatively and qualitatively in Sec. 5 and Sup.Mat F. It consists of three components. A video **Tokenizer** encodes input frame sequences into spatio-temporal tokens: $e_1, ..., e_N = T_{enc}(I_1, ..., I_N)$, and decodes back to images: $I_1, ..., I_N = T_{dec}(e_1, ..., e_N)$. A **Latent Action Model** encodes input frame sequences into spatio-temporal tokens: $a_1, ..., a_{N-1} = LAM_{enc}(I_1, ..., I_{N-1})$, and decodes them to reconstruct future prediction $I_2, ..., I_N = LAM_{dec}(a_1, ..., a_{N-1})$. A **Dynamics module** predicts the next frames based on partially masked frame tokens and actions: $I_2, ..., I_{N+T-1} = D(e_1, ..., e_N, ..., e_{N+T-1}; a_1, ..., a_{N+T-1})$, where in inference $e_N, .., e_{N+T-1}$ are masked. We adhere closely to Genie's specifications for implementing these components.

All components use the causal Spatial Temporal Transformer (STTN) [62]. We use Position Encoding Generator (PEG) [12] for spatial and temporal attention, and Attention with Linear Biases (ALiBi) [49] for temporal attention.

We train our models with a sequence size of 16 frames and resolution of 64x64 to address computational limitations. We train a U-Net-based superresolution network on 50K data samples to upscale the output to 256x256. (Sup.Mat. B)

GenieRedux-G. Building upon the base model, we offer a variant - GenieRedux-G, which is adapted to virtual environments and contains architectural and training improvements. While GenieRedux uses an indispensable LAM model to obtain the actions, we discard it, as ground truth actions are available from our agent. Instead, the one-hot actions are concatenated to each layer of the Dynamics module for conditioning. In this way, we avoid the uncertainty of a prediction.

The Dynamics module consists of an ST-ViViT encoder, followed by a MaskGIT architecture [8], which predicts indices from the tokenizer's codebook for randomly masked input tokens during training, according to a schedule. As standard cross-entropy is used, token classification has the drawback to penalize equally any prediction different from the ground truth. However, close tokens in the codebook result in significantly fewer changes than far tokens, as also shown in Sec. 5. To enable this concept of a distance between tokens in the classification of N_E tokens, we design a Token Distance Cross-Entropy (TDCE) Loss:

$$TDCE(x, y) = (y^T K) \cdot softmax(x) + CE(x, y) \quad (1)$$

Here $x \in \mathcal{R}^{N_E}$ is the prediction logits, $y \in \mathcal{R}^{N_E}$ is the ground truth one-hot class. $K \in \mathcal{R}^{N_E \times N_E}$ is a precomputed table at the start of training of the cosine distances between all tokens ; $CE(.)$ denotes standard Cross-Entropy Loss. When an incorrect token class is given probability, it is penalized based on its distance to the ground truth class.

MaskGIT's design is to take as input learnable embeddings, indexed by the tokens predicted by the Tokenizer. They are randomly initialized, and therefore contain none of the content of the tokens. Given that that the encoding itself and the distance between tokens can contribute to Dynamic module's performance, we add a skip connection by adding the embedding to the token itself, which improves visual fidelity and controllability of the model.

AutoExplore Agent We extend our framework with an exploration agent that obtains data by going deeper into the environments. We name it AutoExplore Agent. The reward of the agent is entirely based on the world model performance and operates without any environment rewards. Therefore, it can be trained in various environments without tuning to their specifics or relying on a reward definition.

The design of our reward is based on the fact that GenieRedux-G employs classification for token prediction. Each token is predicted by sampling from a categorical distribution over the codebook. We first obtain all N_T token prediction distributions by running GenieRedux-G-50 5 steps back from the current observation I_c for which we want to estimate the reward. We provide 2 images I_{c-4}, I_{c-3}, predict 3 images - $I_{c-2}, ..., I_c$, and take the distributions of the predicted tokens of I_c to obtain $x = [x_1, ..., x_t, ..., x_{N_T}]$. We evaluate the uncertainty per predicted token u_t by calculating the entropy over the categorical distribution and normalize it in the range $[0, 2]$:

$$u_t = \frac{2 \cdot \sum_i^{N_T} x_i \cdot log(x_i)}{N_e} \tag{2}$$

Studying the properties of the Tokenizer representation, we find that a prevalent token is learned representing static parts of the environment. Only the changing parts generate high uncertainty and, therefore, we take the subset S_{top} of 25% highest uncertainties of the entire set of uncertainties $S = \{u_t\}$. The reward, shown in Eq. 3, establishes the agent's goal to collect data that maximizes uncertainty of the world model.

$$S_{25\%} = \{u \in S \mid u \geq Q_{75}(S)\} \tag{3}$$

$$R(I_c) = \frac{1}{|S_{25\%}|} \sum_{u \in S_{25\%}} u \tag{4}$$

Our agent is an actor-critic, trained with the Policy Gradient method. For the agent architecture, we follow [39]. It consists of a CNN encoder followed by an LSTM. As standard in RL, 4 frames are stacked, max-pooled, and the result is the input to the agent for a single time step.

Exploration-driven World Model Training. We initially pretrain GenieRedux-G on `Platformers-200` and fine-tune on `Platformers-50` to obtain the model GenieRedux-G-50. Then, we train AutoExplore Agent by using GenieRedux-G-50, using it as a source of reward. The details of training the agent are presented in Sup.Mat A.3.

Running the trained exploratory agent on a selected environment, we obtain a new diverse dataset with action demonstrations under unseen scenes. We first fine-tune the decoder of the Tokenizer for 1,000 iterations to adapt to the new unseen scenes. The Dynamic module of GenieRedux-G is then fine-tuned on the new data to achieve greater visual fidelity and controllability under new conditions. In order to build test sets to evaluate our approach, we train an Agent-57 model for each of the environments we explored, using the available environment rewards. More details on the test setup are provided in Sup.Mat A.2.

For visual fidelity evaluation, we use FID (Fréchet inception distance) Heusel et al. [25], PSNR (signal-to-noise ratio) and SSIM (structural similarity index measure) Wang et al. [61]. To evaluate controllability, we use the recently proposed Δ_tPSNR metric [5], which compares the visual effect of the ground truth action ($\hat{x}_t$) versus a random action ($\hat{x}'_t$): $\Delta_t\text{PSNR} = \text{PSNR}(x_t, \hat{x}_t) - \text{PSNR}(x_t, \hat{x}'_t)$,, where x_t is the ground truth frame at time t. A higher Δ_tPSNR indicates a higher level of controllability. As in Bruce et al. [5], for all experiments we report Δ_tPSNR with $t = 4$.

5. Experiments

Comparing GenieRedux and GenieRedux-G. We implement the original CoinRun case study with a random agent, as advised by [5], in order to validate and compare GenieRedux with LAM, and GenieRedux-G which uses agent-provided actions instead. In this study, the presence of LAM is the only difference between the models. We first train on a dataset, collected by a random agent. Visual fidelity results are in Tab. 2. Our GenieRedux implementation exhibits high visual quality and matches all seven CoinRun environment actions, as well as progressing environment motions (demonstrated in Sup.Mat. F). However, as demonstrated by the metrics, GenieRedux-G shows superior visual fidelity and controllability (more in Sup.Mat. F), as it avoids the uncertainty of LAM prediction. This study demonstrates that even using a random agent can result in action performance abilities in the world model.

Next, we train an actor-critic agent with PPO on the environment reward, following [13] to collect data and train GenieRedux-TA and GenieRedux-G-TA. Tab. 3 shows evaluation on a test set collected by a trained agent.

Table 2. **Comparison of GenieRedux and GenieRedux-G on Basic Test Set.** Peformed on a test set, collected from the Coinrun environment with randomly sampled actions.

Model	Basic Test Set		
	FID↓	PSNR↑	SSIM↑
Tokenizer	18.14	38.25	0.96
LAM	37.01	33.97	0.92
GenieRedux	21.88	25.51	0.77
GenieRedux-G	**18.88**	**33.41**	**0.92**

Table 3. **Comparison of GenieRedux and GenieRedux-G on Diverse Test Set.** The models are trained with data collected by random agent and trained agent (-TA), and tested on data collected by a trained agent from the Coinrun environment.

Model	Diverse Test Set		
	FID↓	PSNR↑	SSIM↑
Tokenizer	19.13	35.85	0.94
Tokenizer-TA	**11.63**	**40.62**	**0.97**
GenieRedux	23.97	23.82	0.73
GenieRedux-G	19.51	31.66	0.90
GenieRedux-TA	12.57	31.97	0.90
GenieRedux-G-TA	**12.40**	**34.44**	**0.92**

GenieRedux-G outperforms GenieRedux on all settings. Furthermore, models trained on diverse agent-collected data are visually superior to those trained on random agents. The higher ΔPSNR of 1.89 for GenieRedux-G-TA compared to 0.70 for GenieRedux-G shows the superiority of diverse data training in controllability. (more in Sup.Mat. F)

Multi-Environment Models. Here, we evaluate the models we initially train on many environments from `RetroAct`. GenieRedux-G-200 is pretrained on the `Platformers-200` dataset for 180k iterations. On the validation set, we obtain 23.32 PSNR and 17.12 FID. Using this model as a base, GenieRedux-G-50 is trained on `Platformers-50`. Its quantitative evaluation on a test set of 10k sessions separately generated from the selected 50 environments is at the start of Tab. 4. As the 50 environments are selected with corresponding action controls between each other, we see a boost in the quality of prediction. Fig. 4 demonstrates that the instructed action is executed successfully by GenieRedux-G. As the up action is rarely used, it serves more as a no-operation action. (more in Sup.Mat C.1)

Ablation Study. In this experiment we evaluate the additive gain of each proposed improvement in GenieRedux-G - the additive token input and training with the Token Distance Cross-Entropy Loss. The ablation is performed on a generated test set of 10k sessions, each 500 frames long.

Table 4. **Ablation study on improvements in GenieRedux-G.**

Model	FID↓	PSNR↑	SSIM↑
GenieRedux-G-200	22.31	25.11	0.80
GenieRedux-G-50	23.80	26.36	0.84
+ Token Input	22.96	26.65	0.84
+ TDCE Loss	22.95	27.06	0.85
Autoregressive	**22.11**	**28.07**	**0.88**

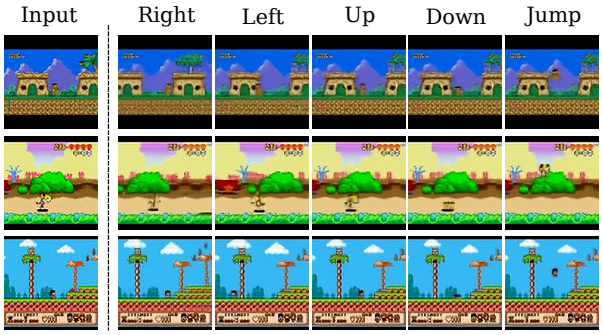

Figure 4. **Control Of GenieRedux-G-50.** Demonstrating all controls of our multi-environment model on multiple games.

The data is collected using a random action policy from the environments in `Platformers-50`. Visual fidelity evaluation is provided in Tab. 4. It can be seen that each component gives our model a benefit in terms of visual fidelity. Finally, we perform an autoregressive evaluation of the best model to achieve our highest score.

Tokenizer Representation Study. This experiment provides insights into the inner workings of GenieRedux-G to motivate our proposed changes. As the Dynamics module operates entirely on the token representation, we examine it closely. Fig. 5 shows the reconstructions of an input sequence (first row) and the visualized token representation (last row), where each predicted token index is assigned a different color. The visual features of the first frame are captured by various tokens. Starting with the second frame, the representation drastically changes - a token is specialized in representing the static frame regions compared to the past, while all motion regions are updated with new content. Observing that visually similar patches predict identical or similar tokens, we replace each predicted token with its closest in the codebook. We only keep the special background token unchanged. In the second row of Fig. 5 we show the resulting reconstruction - while some blurriness appears, the image remains largely the same. Conversely, replacing each token with its furthest in the codebook (third row) results in a significantly different image. This property - closer tokens having more similar appearance - motivates our Token Distance Cross-Entropy Loss, which penalizes predicting tokens further away from the ground truth.

Fig. 6 visualizes the uncertainty of GenieRedux-G-50 for each predicted token of its Dynamics module given a

Table 5. **Quantitative Results on 3 environments.** We evaluate the benefit of the data from the propose AutoExplore Agent to our models. GenieRedux-G-50 is our pretrained world model on 50 environments. GenieRedux-G-50-ft are fine-tuned models using data from a random agent or AutoExplore (Exploration). GenieRedux-G denotes a non-fine-tuned model, trained with the exploration data.

Environment	Strategy	Model	FID↓	PSNR↑	SSIM↑	ΔPSNR↑
Adventure Island II	Random	GenieRedux-G-50	41.99	26.32	0.81	0.83
		GenieRedux-G-50-ft	42.34	27.04	0.81	1.19
	Exploration	Tokenizer-ft	11.01	38.95	0.98	-
		GenieRedux-G	11.94	28.33	0.88	0.37
		GenieRedux-G-50-ft	**12.77**	**30.60**	**0.90**	**1.47**
	Random Autoregressive	GenieRedux-G-50-ft	41.55	27.82	0.83	1.24
	Exploration Autoregressive	GenieRedux-G-50-ft	**11.33**	**33.61**	**0.94**	**2.09**
Super Mario Bros	Random	GenieRedux-G-50	29.83	34.24	0.94	**0.56**
		GenieRedux-G-50-ft	30.13	34.54	0.94	**0.54**
	Exploration	Tokenizer	8.09	42.00	0.99	-
		GenieRedux-G	**9.56**	34.00	0.95	0.09
		GenieRedux-G-50-ft	**9.55**	**36.13**	**0.97**	**0.57**
	Random Autoregressive	GenieRedux-G-50-ft	30.84	34.85	0.95	0.57
	Exploration Autoregressive	GenieRedux-G-50-ft	**9.33**	**37.77**	**0.97**	**0.76**
Smurfs	Random	GenieRedux-G-50	79.51	21.47	0.69	0.47
		GenieRedux-G-50-ft	80.61	21.83	0.70	0.65
	Exploration	Tokenizer	17.86	35.61	0.98	-
		GenieRedux-G	20.43	35.42	0.80	0.85
		GenieRedux-G-50-ft	**20.01**	**27.45**	**0.85**	**1.55**
	Random Autoregressive	GenieRedux-G-50-ft	80.16	22.16	0.71	0.69
	Exploration Autoregressive	GenieRedux-G-50-ft	**18.97**	**29.53**	**0.90**	**2.06**

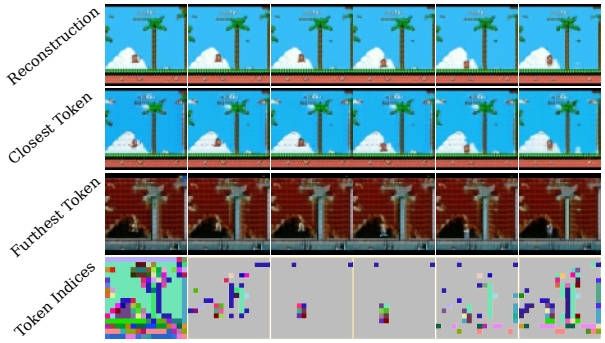

Figure 5. **Tokenizer Representation.** Reconstruction images from the tokenizer, and the effect of replacing each token with its closest and furthest in the codebook. Lastly, we visualize the indices of the predicted tokens.

sequence. The uncertainty metric is the entropy of the classification over 1024 codebook tokens. Tokens corresponding to motion have the highest uncertainty; other regions are mostly classified as the "static" token. Thus, minimal character movement yields low uncertainty, while forward motion increases it. This motivates us to build AutoExplore Agent's reward based on this uncertainty.

Exploration-based training. We demonstrate our exploration-based training of GenieRedux-G. We perform the procedure on 3 environments - `AdventureIslandII`, which provides an easy setting for the agent to learn (single platform with no enemies at the start), `SuperMarioBros` provides an enemy and obstacles soon after the start and `Smurfs` provides a more complex background imagery and different action dynamics. For each of the environments, we train an AutoExplorer Agent. We observe that the agent learns to move forward and navigate obstacles to maximize reward. (more in Sup.Mat. D)

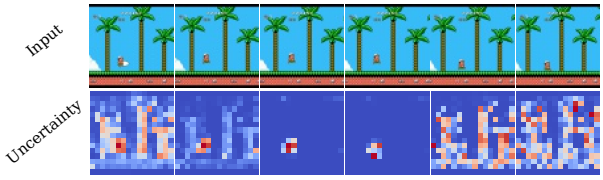

Figure 6. **Dynamics Uncertainty.** Shown is the uncertainty per token predicted for each image of an example sequence. Uncertainty is generated in the regions of motion.

We use our pretrained GenieRedux-G-50 model as a baseline and fine-tune it for each environment in two settings - a dataset collected on the selected environment by a random agent and by our AutoExplorer Agent. Each dataset consists of 10k sessions, each 700 frames long. We fine-tune (GenieRedux-G-50-ft) for 10k iterations and pick the best performing model. In our comparison, we also include a GenieRedux-G model trained from scratch on the diverse exploration datasets for 15k iterations to show the effect of pretraining. We perform single-pass generation for all mod-

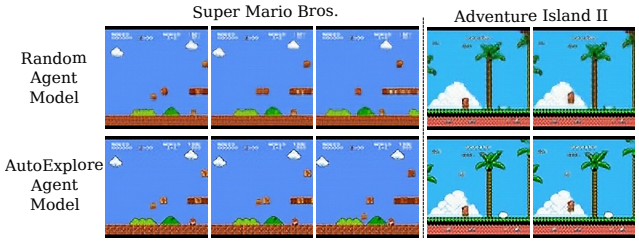

Figure 7. **AutoExplore Agent vs Random Agent Qualitative Comparison.** We show that AutoExplore exhibits better visual quality and avoids losing track of the agent.

Table 6. **Comparison of AutoExplore Agent with others.**

Agent	SuperMarioBros			AdventureIslandII		
	PSNR↓	**SSIM↓**	**ΔPSNR↓**	**PSNR↓**	**SSIM↓**	**ΔPSNR↓**
RF	28.58	0.94	0.181	24.82	0.78	0.44
VAE	24.40	0.86	0.087	16.57	0.5	**0.072**
Ours	**23.81**	**0.85**	**0.065**	**15.20**	**0.41**	**0.070**

els and the more computationally heavy autoregressive evaluation for the fine-tuned models on data from random and AutoExplore Agent's datasets. Tab. 5 shows visual fidelity and controllability metrics for each environment, confirming the effectiveness of our exploration method. The model fine-tuned on AutoExplore Agent's data consistently outperforms the models trained on random actions in terms of visual fidelity. Exploration-based fine-tuning also improves controllability. Environments with small characters and uniform backgrounds can be more challenging for all models to learn. However, the gain in controllability in this case remains noticeable during autoregressive evaluation. Fig. 7 demonstrates the superior quality of our method. In addition, we observe that the multienvironment pretraining leads to significant gains in both studied aspects compared to the nonpretrained model. (more in Sup.Mat. C)

AutoExplore Agent Evaluation. We compare AutoExplore Agent with exploration-based methods in [6]. We train agents based on SSE of RF and VAE features on top of GenieRedux and compare with ours on Tab. 6. AutoExplore Agent's reward results in maximum world model visual and controllability errors (on 1k episodes of agent actions), fulfilling its intended role in our framework.

User Studies. To validate the quality of our final results, we perform a user study in which we ask people to rate from 1 to 5 the quality of samples produced respectively by GenieRedux-G trained on random agent's data and on AutoExplore Agent's data. Each sample in our study consists of two 16-frame clips playing in a synchronized manner - the ground truth clip and our GenieRedux-G-50-ft reconstruction, given two initial frames and generating the rest autoregressively. We provide a total of 120 samples to the users - 40 samples per model and 40 samples of two ground-truth samples, to establish scale. We give 20 samples from each of the two selected games - SuperMarioBros and AdventureIslandII. We get reviews from 19 participants. The results are shown in Fig. 8. The model, trained on data from AutoExplore Agent is clearly rated closer to the ground truth, establishing the quality of our method.

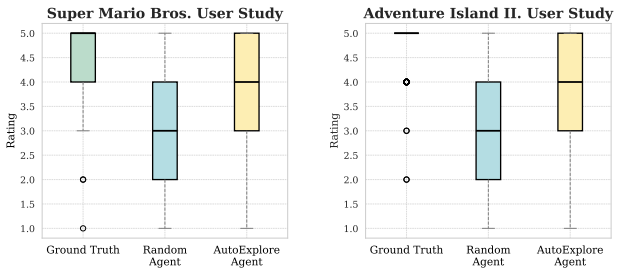

Figure 8. **User study results.** Our user study on two games shows that our model trained with AutoExplore Agent's data is consistently rated higher.

With a second user study, we evaluate the action accuracy of the generated frames. We use ambiguous single input cases (character starting mid-air) and generate 60 clips with 3 actions on AdventureIslandII. Users prefer our exploration-trained model, rating it **0.75 ± 0.019** on a scale from 0 (random preferred) to 1 (exploration preferred). (more in Sup.Mat. E.2)

6. Conclusion

As world models have developed into large models with impressive simulation properties, they require large interaction datasets, complete with diverse observations and actions. Genie [5] demonstrates impressive abilities by training on multiple environments, however, requiring the collection of a large video dataset and a model to infer actions.

In this work, we address the heavy burden of data collection and curation by building a new framework for training large world models by collecting interaction data from a large number of virtual environments. We first build an open implementation of Genie - GenieRedux and enhance it into its version GenieRedux-G. We obtain models exhibiting control by pretraining on a large set of virtual environments. We address the overfitting limitations of random data collection policy by proposing AutoExplore Agent, an agent entirely independent of the environment reward, maximizing the uncertainty of GenieRedux-G. After fine-tuning on the explored environment, our model is able to improve its visual fidelity and controllability much better than training solely on random agent's data. Demonstrating this on multiple environments, we show the potential of our framework to make training of next-generation world models more accessible, cost-effective, and effort-free.

7. Acknowledgments

INSAIT, Sofia University "St. Kliment Ohridski". Partially funded by the Ministry of Education and Science of Bulgaria's support for INSAIT as part of the Bulgarian National Roadmap for Research Infrastructure. This project was supported with computational resources provided by Google Cloud Platform (GCP).

References

[1] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37: 58757–58791, 2024.

[2] Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE, 2018.

[3] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[4] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[5] 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 *Forty-first International Conference on Machine Learning*, 2024.

[6] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2019.

[7] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

[8] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[9] Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

[10] Silvia Chiappa, Sébastien Racaniere, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. In *International Conference on Learning Representations*, 2017.

[11] Leshem Choshen, Lior Fox, and Yonatan Loewenstein. Dora the explorer: Directed outreaching reinforcement action-selection. *arXiv preprint arXiv:1804.04012*, 2018.

[12] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[13] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International conference on machine learning*, pages 1282–1289. PMLR, 2019.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[15] Justin Fu, John Co-Reyes, and Sergey Levine. Ex2: Exploration with exemplar models for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[16] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[17] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

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

[19] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[20] David Ha, Jonas Jongejan, and Ian Johnson. Draw together with a neural network. *Retrieved Oct*, 5:2021, 2017.

[21] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[22] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

[23] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

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

[25] 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.

[26] Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. Distributed prioritized experience replay. In *International Conference on Learning Representations*, 2018.

[27] 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.

[28] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.

[29] Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *International conference on learning representations*, 2018.

[30] Youngjin Kim, Wontae Nam, Hyunwoo Kim, Ji-Hoon Kim, and Gunhee Kim. Curiosity-bottleneck: Exploration by distilling task-specific novelty. In *International conference on machine learning*, pages 3379–3388. PMLR, 2019.

[31] Martin Klissarov, Riashat Islam, Khimya Khetarpal, and Doina Precup. Variational state encoding as intrinsic motivation in reinforcement learning. In *Task-Agnostic Reinforcement Learning Workshop at Proceedings of the International Conference on Learning Representations*, pages 16–32, 2019.

[32] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

[33] Ian Lenz, Ross A Knepper, and Ashutosh Saxena. Deepmpc: Learning deep latent features for model predictive control. In *Robotics: Science and Systems*, page 25. Rome, Italy, 2015.

[34] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[35] Marlos C Machado, Marc G Bellemare, and Michael Bowling. Count-based exploration with the successor representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5125–5133, 2020.

[36] Jarryd Martin, Suraj Narayanan Sasikumar, Tom Everitt, and Marcus Hutter. Count-based exploration in feature space for reinforcement learning. *arXiv preprint arXiv:1706.08090*, 2017.

[37] Willi Menapace, Stephane Lathuiliere, Sergey Tulyakov, Aliaksandr Siarohin, and Elisa Ricci. Playable video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10061–10070, 2021.

[38] Alberto Maria Metelli, Amarildo Likmeta, and Marcello Restelli. Propagating uncertainty in reinforcement learning via wasserstein barycenters. *Advances in Neural Information Processing Systems*, 32, 2019.

[39] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023.

[40] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PmLR, 2016.

[41] Ian Osband and Benjamin Van Roy. Bootstrapped thompson sampling and deep exploration. *arXiv preprint arXiv:1507.00300*, 2015.

[42] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

[43] Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, pages 2377–2386. PMLR, 2016.

[44] Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[45] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR, 2017.

[46] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[47] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pages 5062–5071. PMLR, 2019.

[48] Mathieu Poliquin. Stable retro, a maintained fork of openai's gym-retro. https://github.com/Farama-Foundation/stable-retro, 2024.

[49] Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.

[50] Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. In *The Eleventh International Conference on Learning Representations*, 2023.

[51] 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*, pages 10684–10695, 2022.

[52] Tom Schaul. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[53] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International conference on machine learning*, pages 8583–8592. PMLR, 2020.

[54] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 1015–1022, 2010.

[55] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[56] Ruo Yu Tao, Vincent François-Lavet, and Joelle Pineau. Novelty search in representational space for sample efficient

exploration. *Advances in Neural Information Processing Systems*, 33:8114–8126, 2020.

[57] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

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

[59] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[60] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2022.

[61] Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.

[62] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

[63] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2023.

[64] Sherry Yang, Jacob Walker, Jack Parker-Holder, Yilun Du, Jake Bruce, Andre Barreto, Pieter Abbeel, and Dale Schuurmans. Video as the new language for real-world decision making. *arXiv preprint arXiv:2402.17139*, 2024.

[65] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023.

[66] Lixuan Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Prelar: World model pre-training with learnable action representation. In *European Conference on Computer Vision*, pages 185–201, 2024.

[67] Tianjun Zhang, Paria Rashidinejad, Jiantao Jiao, Yuandong Tian, Joseph E Gonzalez, and Stuart Russell. Made: Exploration via maximizing deviation from explored regions. *Advances in Neural Information Processing Systems*, 34:9663–9680, 2021.