Learning Generative Interactive Environments By Trained Agent Exploration

Naser Kazemi[1][*]   Nedko Savov[1][*][†]   Danda Pani Paudel[1]   Luc Van Gool[1]

[1] INSAIT, Sofia University "St. Kliment Ohridski"

{firstname.lastname}@insait.ai

Abstract

World models are increasing in importance for interpreting and simulating the rules and actions of complex environments. Genie, a recent model, excels at learning from visually diverse environments but relies on costly human-collected data. We observe that their alternative method of using random agents is too limited to explore the environment. We propose to improve the model by employing reinforcement learning based agents for data generation. This approach produces diverse datasets that enhance the model's ability to adapt and perform well across various scenarios and realistic actions within the environment. In this paper, we first build, evaluate and release the model GenieRedux - a complete reproduction of Genie. Additionally, we introduce GenieRedux-G, a variant that uses the agent's readily available actions to factor out action prediction uncertainty during validation. Our evaluation, including a replication of the Coinrun case study, shows that GenieRedux-G achieves superior visual fidelity and controllability using the trained agent exploration. The proposed approach is reproducable, scalable and adaptable to new types of environments. Our codebase is available at https://github.com/insait-institute/GenieRedux.

1 Introduction

Recently, world models have emerged as tools for understanding rules, meaning and consequences of actions in increasingly complex environments. World models have developed from rough imagination models assisting reinforcement learning agents Chiappa et al. (2017), Ha and Schmidhuber (2018), Hafner et al. (2019), Hafner et al. (2023), Sekar et al. (2020) to independent realistic video generation models conditioned on actions Micheli et al. (2022), Chen et al. (2022), Yang et al. (2024), Robine et al. (2023). For example, works like Menapace et al. (2021), Yang et al. (2023), Bruce et al. (2024), Hu et al. (2023), simulate real-world environments.

Notably, Bruce et al. (2024) propose Genie - a model capable of learning from many visually different environments with the same behavior - particularly platformer games. This allows the model to apply the learned per-frame motion controls to new unseen images. Moreover, Genie incorporates a Latent Action Model predicting actions and enabling the model to be trained on action-free data. We recognize that using multiple environments is an important step towards generalizable world models.

However, Genie's approach is to use human demonstrations of exploring environments - they obtain a large scale dataset by collecting and cleaning online playthrough videos of platformer games. Such datasets are difficult to build and switching to a different kind of environment requires another costly human action data collection or recording. As an alternative to human demonstrations, the authors only provide a small-scale case study where a random agent is used to obtain data from a virtual

---

[*]Indicates equal contribution.

[†]Corresponding author.

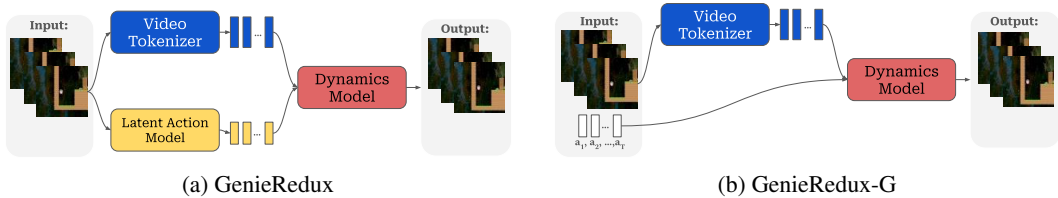|	
(a) GenieRedux	(b) GenieRedux-G

Figure 1: **Architecture of our models.** GenieRedux shares the architecture of Genie; GenieRedux-G takes agent actions as input instead of predicting them.

environment. However, a random agent cannot progress and explore far in the environment. This causes the model to overfit on the seen start scenes of the environment. Instead of a random agent, we propose to use an RL-based trained agent on the environment to produce more diverse data. Training on this diverse data overcomes the aforementioned overfitting problem. Note that collecting data using a trained agent is significantly cheaper than through human demonstrations.

In this work, we first reproduce the Genie model Bruce et al. (2024), as Genie's official codebase is not available. The resulting model we release under the name GenieRedux. As the trained agent gives us agent actions, we use a guided variant of the model named GenieRedux-G where the next frame prediction is conditioned on agent actions rather than on predictions from the Latent Action Model. This allows us to evaluate our proposed environment exploration while we factor out any action prediction noise. Architectures are shown on Fig. 1. We show that our model performs well both on visual fidelity and controllability. We implement the Coinrun Cobbe et al. (2019) case study, proposed by Bruce et al. (2024), with both a random agent and a trained agent and show that the latter produces a model able to perform better in diverse situations in the environment. Our setup is easily reproducable and scales up when extending to different types of environments for training.

Our contributions are as follows:

- The implementation and release of GenieRedux and GenieRedux-G - Pytorch open source models based on Bruce et al. (2024).
- Generating diverse data through trained agent exploration and using it to train world models enhancing visual fidelity and controllability. Conditioning the world model on this data and its available agent actions (GenieRedux-G), instead of in-model predictions, leading to improved performance.
- Performing video fidelity and controllability studies on all relevant components.

2 Methodology

GenieRedux consists of three components, as shown in Fig. 1. A **video tokenizer** encodes input frame sequences into spatio-temporal tokens. A **Latent Action Model** encodes input frame sequences into spatio-temporal tokens. A **dynamics model** predicts the next frame based on frame tokens and actions. We adhere closely to Genie's specifications for implementing these components.

ST-ViViT. All components use the Spatiotemporal Transformer (STTN) architecture Xu et al. (2020), with ST-Blocks that capture spatial and temporal patterns using separate attention layers for efficiency. Causal temporal attention allows for multiple future predictions at once. ST-ViViT is an encoder-decoder model with a VQ-VAE objective Van Den Oord et al. (2017) for generating discrete tokens, inspired by C-ViViT Villegas et al. (2022) but with more efficient ST-Blocks. The encoder alternates spatial and temporal attention, mirrored by the decoder. Position Encoding Generator (PEG) Chu et al. (2021) is used for spatial and temporal attention, while Attention with Linear Biases (ALiBi) Press et al. (2021) is used for temporal attention.

GenieRedux. The **video tokenizer** is an ST-ViViT autoencoder, while the **Latent Action Model** (LAM) is an ST-ViViT encoder-decoder predicting the next frame by generating a token for the action between the last two frames (with a linear layer at the encoder). We offer two **dynamics model** variants: GenieRedux, which follows Genie by summing LAM encoded actions with tokenized frames, and GenieRedux-G, which uses the concatenation of frame tokens with one-hot agent actions, which are readily available and eliminate LAM prediction uncertainty evaluations of the trained agent exploration evaluation. The architectures are shown on Fig. 1.

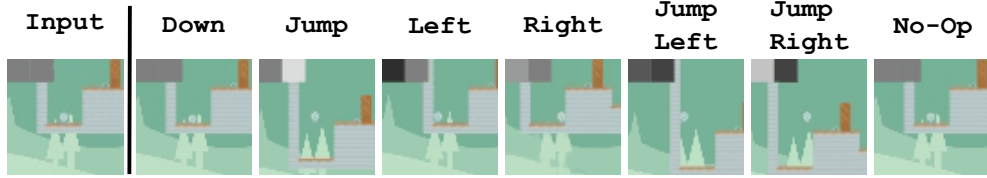| Input	Down	Jump	Left	Right	Jump Left	Jump Right	No-Op

Figure 2: **GenieRedux-G-TA Control Demonstration.** GenieRedux-G-TA is able to consistently perform all environment actions. Here we demonstrate all of them as generated by the model.

The dynamics model consists of an ST-ViViT encoder, followed by a MaskGIT architecture Chang et al. (2022), which predicts indices from the tokenizer's codebook for randomly masked input tokens during training, according to the schedule described for Genie.

Experimental Setup. We use Genie's case study setup with random exploration in the Coinrun environment Cobbe et al. (2019) with 7 actions. We obtain a dataset with 88k episodes on random hard levels (10% validation) with up to 500 frames each and a separate test set with 1000 episodes that we call Basic Test Set. The random agent shows limited progression beyond the start of levels. In addition, we train a CNN agent with Proximal Policy Optimization according to Cobbe et al. (2019) on the easy Coinrun levels. With the trained agent, we collect 10k episodes (10% validation) and a separate 1000-episode test set named Diverse Test Set. These episodes are much more content-wise diverse than those from random exploration.

Training. All our models are trained on 64x64 resolution with sequence size of 16, with a patch size 4. For evaluation, we use a sequence size of 10. We first train the tokenizer. We then train the LAM and dynamics together, using frame tokens and predicted actions for GenieRedux or ground truth agent actions (no LAM) for GenieRedux-G. The random exploration dataset is used to obtain the **GenieRedux-Base** and **GenieRedux-G-Base** baseline models. We then fine-tune the tokenizer and LAM on the trained agent dataset, and fine-tune the dynamics to create the **GenieRedux-TA** and **GenieRedux-G-TA** models. Further details are in App. A.

3 Experiments

Baseline Evaluation. In this experiment we repeat the original case study with a random agent, as advised by Bruce et al. (2024) and evaluate our implementation of the GenieRedux-Base and GenieRedux-G-Base models and their components on the Basic Test Set. We show visual fidelity results on Tab. 1. We note that in the original case study of Genie scores are not reported. However, we compare our tokenizer's 38.25 PSNR with the reported tokenizer's 35.7 PSNR in their Appendix C.2. Our LAM is able to learn environment actions, leading to the visual fidelity of GenieRedux-Base, validating the correctness of our implementation. However, GenieRedux-G-Base demonstrates superior visual fidelity, controllability and ability to progress motions over time (demonstrated in App. B), as it avoids the uncertainty of LAM. Note that the evaluation of dynamics consists of predicting 10 images in the future, given a single image and the actions to perform. The prediction on a single step is with 25 MaskGIT iterations.

Trained Agent Exploration Models Evaluation. In this experiment, we evaluate our models trained with the trained agent exploration, rather than the random agent - GenieRedux-TA and GenieRedux-G-TA. The evaluation set is the Basic Test Set to match the classic case study. Visual fidelity results are shown in Tab. 2. Tokenizer-TA shows significantly improved visual fidelity compared

Table 1: **Visual Fidelity** of baseline models.

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

Table 2: **Visual Fidelity** of TA models.

Model	Basic Test Set		
	FID↓	PSNR↑	SSIM↑
Tokenizer-TA	12.10	39.53	0.97
LAM-TA	47.73	28.24	0.85
GenieRedux-TA	13.26	25.47	0.82
GenieRedux-G-TA	**13.01**	**32.09**	**0.94**

Table 3: **Visual Fidelity Evaluation** of GenieRedux, GenieRedux-G and their tokenizer, trained with random agent exploration (-Base), compared to training with trained agent exploration (-TA). Evaluation is done on Diverse Test Set.

Figure 3: **GenieRedux-G-TA Controllability Across Horizons.**

Model	Diverse Test Set			
	FID↓	PSNR↑	SSIM↑	Δ_tPSNR↑
Tokenizer-Base	19.13	35.85	0.94	-
Tokenizer-TA	**11.63**	**40.62**	**0.97**	-
GenieRedux-Base	23.97	23.82	0.73	-
GenieRedux-G-Base	19.51	31.66	0.90	0.70
GenieRedux-TA	12.57	31.97	0.90	-
GenieRedux-G-TA	**12.40**	**34.44**	**0.92**	**1.89**

to the Base model. LAM-TA shows reduced visual fidelity which does not affect GenieRedux-TA, as performance is on-par with Base - a sign for a good predicted action quality. (see App. C). Meanwhile, GenieRedux-G-TA, unaffected by LAM's uncertainty, shows significantly better visual quality and is consistently able to enact all environment actions and progress motions, as seen on Fig. 4 (more in App. E). All actions are demonstrated on Fig. 2.

Comparison between Trained and Random Exploration. Here we compare all our models on the various scenarios in the Diverse Test Set. Tab. 3 shows that both trained agent exploration models outperform the random exploration models in terms of visual fidelity. Moreover, trained agent exploration offers a significant gain in controllability, represented by the Δ_tPSNR metric, defined in Bruce et al. (2024). This is also demonstrated with our best model GenieRedux-G-TA on Fig. 2.

Comparison with Jafar. We compare with Jafar Willi et al. (2024) - a concurrent with ours implementation of Genie (in JAX). We obtain and train their model as instructed. We train GenieRedux-Base with Jafar's model parameters and like them separate LAM from Dynamics in training. The latter significantly worsened GenieRedux-Base's action representation. Despite that, GenieRedux-Base shows significantly better visual fidelity metrics, achieving 17.91 PSNR (46.12 FID), compared to Jafar's 12.66 PSNR (154.12 FID) . GenieRedux-Base does not exhibit Jafar's artifacts or the reported problematic "hole digging" behavior (more in App. D). Moreover, we observe that Jafar lacks causality which we find problematic.

Prediction Horizon Evaluations. We evaluate our best model's controllability (at 50k iterations) over varying prediction horizons on Fig. 3. As expected, predictions become more challenging further into the future. The first prediction is also difficult due to insufficient motion information - we obtain 0.4 Δ_tPSNR for $t = 1$. To address this issue, we provide the model with 4 frames and actions (predicting 10), and observe an improvement of our best model (GenieRedux-G-TA) from 34.79 PSNR (12.75 FID) on Tab. 3 to **38.31** PSNR (12.29 FID) on Diverse Test Set.

4 Conclusion

In this work, we revisited Bruce et al. (2024)'s Genie - while achieving strong results, we note it relies on costly human data and limited random agent exploration. We address these limitations by demonstrating that RL-based exploration provides a scalable, effective alternative, enhancing the generalizability and efficiency of world models in complex environments.

Figure 4: **GenieRedux-G-TA Qualitative Result.** We give a single frame and actions from the test set and we generate 10 frames. In this example our model first successfully progresses the motion of falling. Then, it performs a jump. Ground truth frames are at the top; generated - at the bottom.
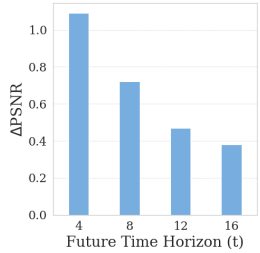
5   Acknowledgements

References

Bruce, J., Dennis, M. D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al. (2024). Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*.

Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. (2022). Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325.

Chen, C., Wu, Y.-F., Yoon, J., and Ahn, S. (2022). Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*.

Chiappa, S., Racaniere, S., Wierstra, D., and Mohamed, S. (2017). Recurrent environment simulators. *arXiv preprint arXiv:1704.02254*.

Chu, X., Tian, Z., Zhang, B., Wang, X., and Shen, C. (2021). Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*.

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, J. (2019). Quantifying generalization in reinforcement learning. In *International conference on machine learning*, pages 1282–1289. PMLR.

Ha, D. and Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. (2019). Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. (2023). Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*.

Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., and Corrado, G. (2023). Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*.

Menapace, W., Lathuiliere, S., Tulyakov, S., Siarohin, A., and Ricci, E. (2021). Playable video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10061–10070.

Micheli, V., Alonso, E., and Fleuret, F. (2022). Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*.

Press, O., Smith, N. A., and Lewis, M. (2021). Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Robine, J., Höftmann, M., Uelwer, T., and Harmeling, S. (2023). Transformer-based world models are happy with 100k interactions. *arXiv preprint arXiv:2303.07109*.

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. (2020). Planning to explore via self-supervised world models. In *International conference on machine learning*, pages 8583–8592. PMLR.

Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Villegas, R., Babaeizadeh, M., Kindermans, P.-J., Moraldo, H., Zhang, H., Saffar, M. T., Castro, S., Kunze, J., and Erhan, D. (2022). Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*.

Willi, T., Jackson, M. T., and Foerster, J. N. (2024). Jafar: An open-source genie reimplemention in jax. In *First Workshop on Controllable Video Generation @ ICML 2024*.

Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., and Xiong, H. (2020). Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*.

Yang, S., Walker, J., Parker-Holder, J., Du, Y., Bruce, J., Barreto, A., Abbeel, P., and Schuurmans, D. (2024). Video as the new language for real-world decision making. *arXiv preprint arXiv:2402.17139*.

Yang, Z., Chen, Y., Wang, J., Manivasagam, S., Ma, W.-C., Yang, A. J., and Urtasun, R. (2023). Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399.

A Appendix: Training Setup

The architecture and training parameters of the tokenizer, LAM and dynamics model are shown respectively on Tab. 4, Tab. 5, Tab. 6.

We train the tokenizer on 6 A100 GPUs for 100k iterations - 4 days. We finetune it on the trained exploration data for 150k iterations - 2 days. We train GenieRedux and GenieRedux-G models on 8 A100 GPUs for 150k iterations - 4 days.

For training the agent for exploration, we enable velocity maps on Coinrun. These maps need to also be enabled for the agent during data collection. When evaluating models trained on different datasets, to be fair, we exclude the velocity map regions by setting their pixels to black.

Throughout the training, we use a batch size of 84 and a patch size of 4 for all components. We use the Adam Optimizer with a linear warm-up and cosine annealing strategy.

Component	Parameter	Value
Encoder	num_layers	8
	d_model	512
	num_heads	8
Decoder	num_layers	8
	d_model	512
	num_heads	8
Codebook	num_codes	1024
	latent_dim	32

Table 4: Tokenizer hyperparameters

Component	Parameter	Value
Encoder	num_layers	8
	d_model	512
	num_heads	8
Decoder	num_layers	8
	d_model	512
	num_heads	8
Codebook	num_codes	7
	latent_dim	32

Table 5: LAM hyperparameters

Component	Parameter	Value
Architecture	num_layers	12
	d_model	512
	num_heads	8
Sampling	temperature	1.0
	maskgit_steps	25

Table 6: Dynamics hyperparameters

Parameter	Value
max_lr	1×10^{-4}
min_lr	5×10^{-5}
β_1	0.9
β_2	0.99
weight_decay	1×10^{-4}
linear_warmup_start_factor	0.5
warmup_steps	5000

Table 7: Optimizer Hyperparameters

B   Appendix: GenieRedux-G-Base Qualitative Evaluation

On Fig. 5 we show quantitative results demonstrating that GenieRedux-G-Base can perform motion progression and action execution.
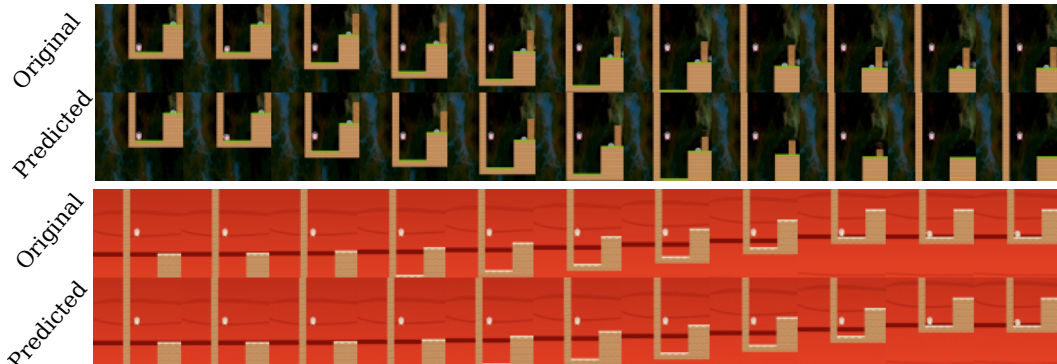
Figure 5: **GenieRedux-Base Quantitative Evaluation.** We present a few sequences from the test set with predictions from GenieRedux-Base. On the example at the top we show a successful jump action. On the example at the bottom we show a successful motion progression.

C   Appendix: GenieRedux-TA Qualitative Evaluation

On Fig. 6 we demonstrate that GenieRedux-TA is able to execute actions and complete motion. On Fig. 7 we show that the model is capable of executing all actions of the environment.
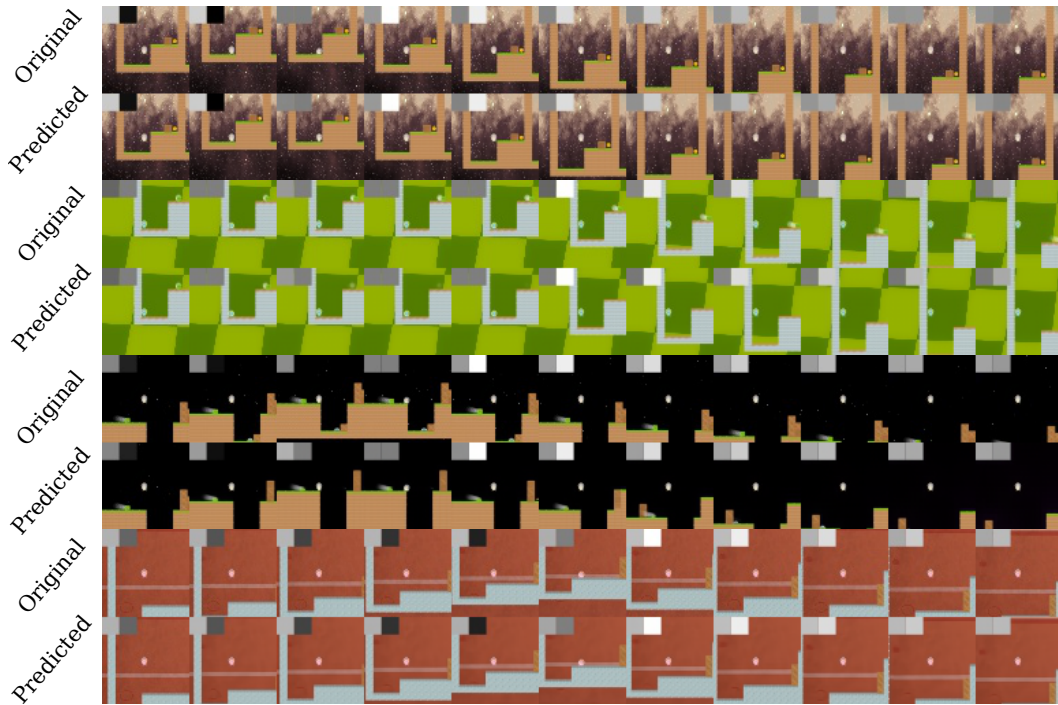
Figure 6: **GenieRedux-TA Qualitative Compatison.** We present a few samples from the test set with various actions. We demonstrate that GenieRedux-G-TA performs the actions correctly.
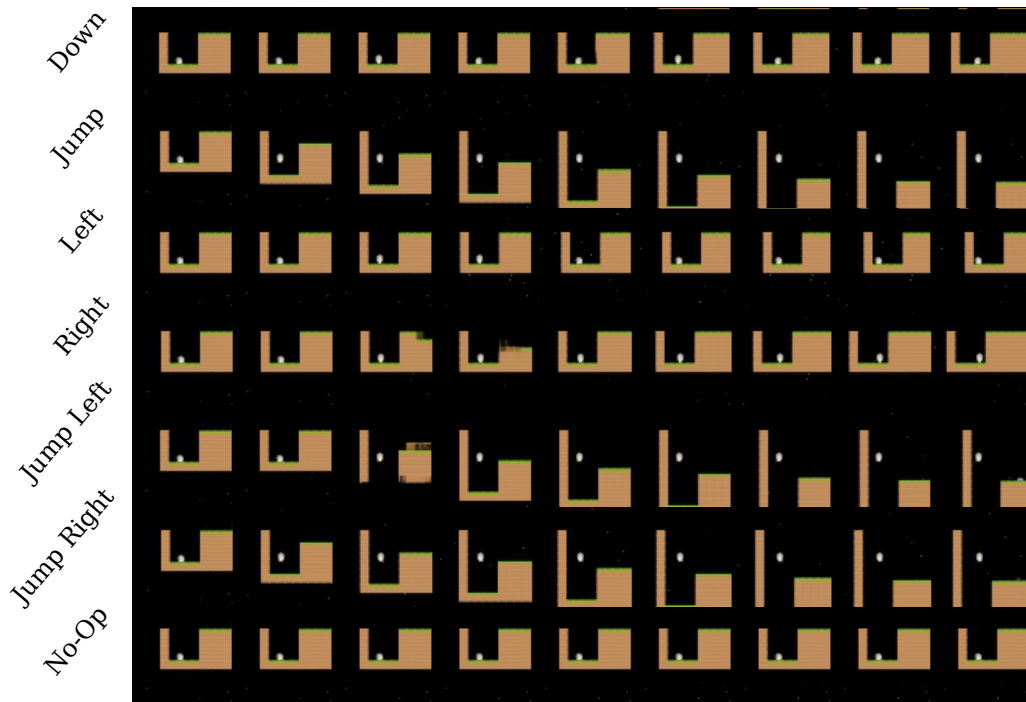

Figure 7: **GenieRedux-TA Controllability.** We show predictions for all environment actions of GenieRedux-TA.

D   Appendix: Jafar Qualitative Comparison

On Fig. 8 we show Jafar's reconstruction of 10 frames into the future, given the first frame and a sequence of actions. The results are on the validation set after training. We observe an abundance of artifacts. We note that if we provide the images instead of providing the first frame we get much less artifacts. This seems to hint that Jafar relies on future images to make predictions for the current frame, which might be an inherent problem of the model not being causal.

We additionally report to the numbers reported in the main text, test set results for Jafar - 0.48 SSIM and for GenieRedux(with Jafar parameters) - 0.62 SSIM.
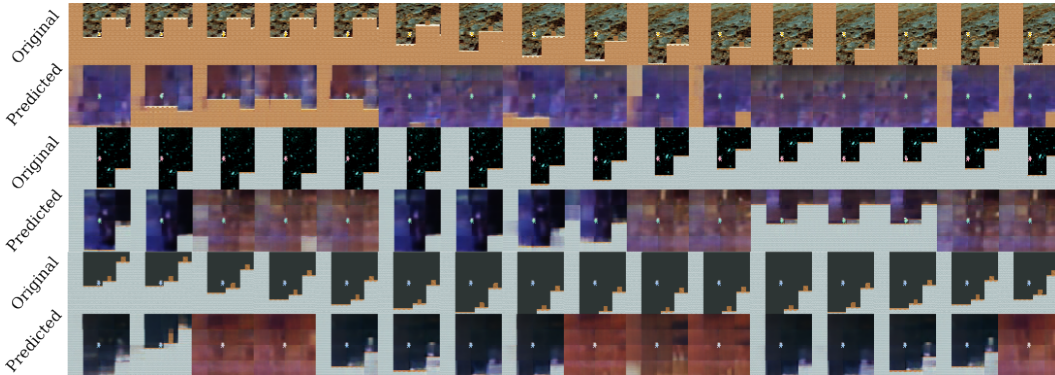

Figure 8: **Jafar Qualitative Results.** The results are on the validation set. We give only a single image and actions and predict 15 frames in the future.

In addition we show the version of GenieRedux that we trained to match Jafar on Fig. 9. While it can be noticed that the model prefers inaction when encountering actions, it successfully progresses motion - e.g. moving a character through the air. We also notice fairly good visual quality.
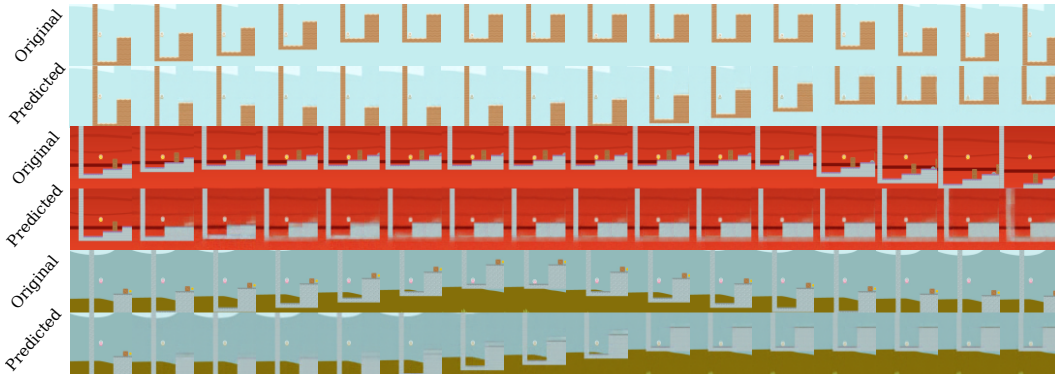

Figure 9: **GenieRedux with Jafar's Parameters Qualitative Results.** We show 15 frames into the future given actions and an initial frame of our model.

E   Appendix: Additional GenieRedux-G-TA Qualitative Results

We provide additional visuals of our best performing GenieRedux-G-TA on Fig. 10. We see that our model performs well under different actions and scenarios.

Next, we discuss the limitations of GenieRedux-G-TA and we visualize the known cases on Fig. 11. One possible failure case occurs whenever the environment state or the actions suggest a major exploration of the environment will unfold - for example, when falling down from mid-jump. As the agent is only given a single frame and cannot possibly know the layout of the level, it attempts to reconstruct something that is not guaranteed to be the actual level. Often, the agent exhibits uncertainty in these cases, as shown in the results.
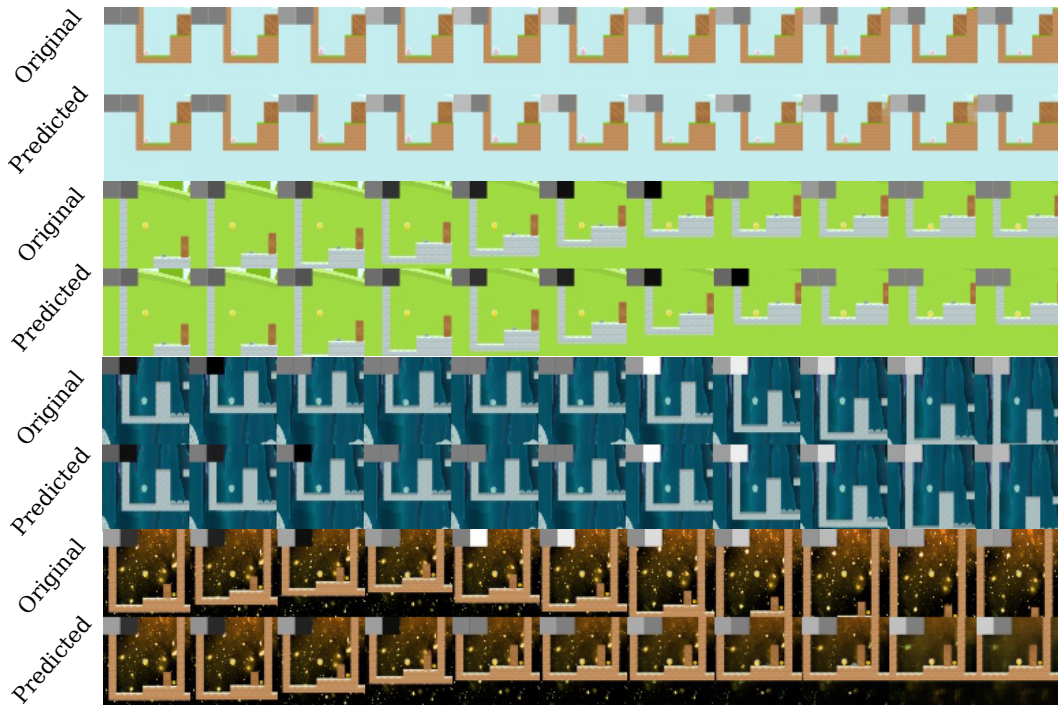
Figure 10: **GenieRedux-G-TA Extra Qualitative Results.** More sampled sequences from the test set, showing good match with the ground truth when enacting actions.

Another possible weakness occurs whenever on the first frame a motion is already in progress - for example, in progress of jumping. In that case the model observes a single frame with the agent in the air and has no information about which direction the agent is heading - going up or going down. In that case the model could exhibit uncertainty in the form of artifacts suggesting that the agent is both landing and jumping up, or alternatively not perform an action at all. This is a state that the agent often recovers from in a few steps. Still, we find that it can be avoided by providing more input frames to the model that can give motion information.
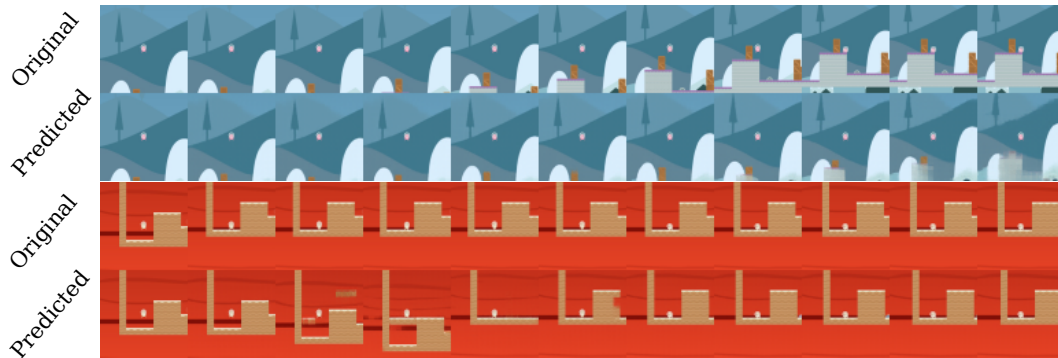

Figure 11: **GenieRedux-G-TA Limitations.** Two failure cases of GenieRedux-G-TA - whenever a sizeable new unknown part of the environment is revealed; whenever an in-progress motion is ambiguous.