A Atari 100k curves

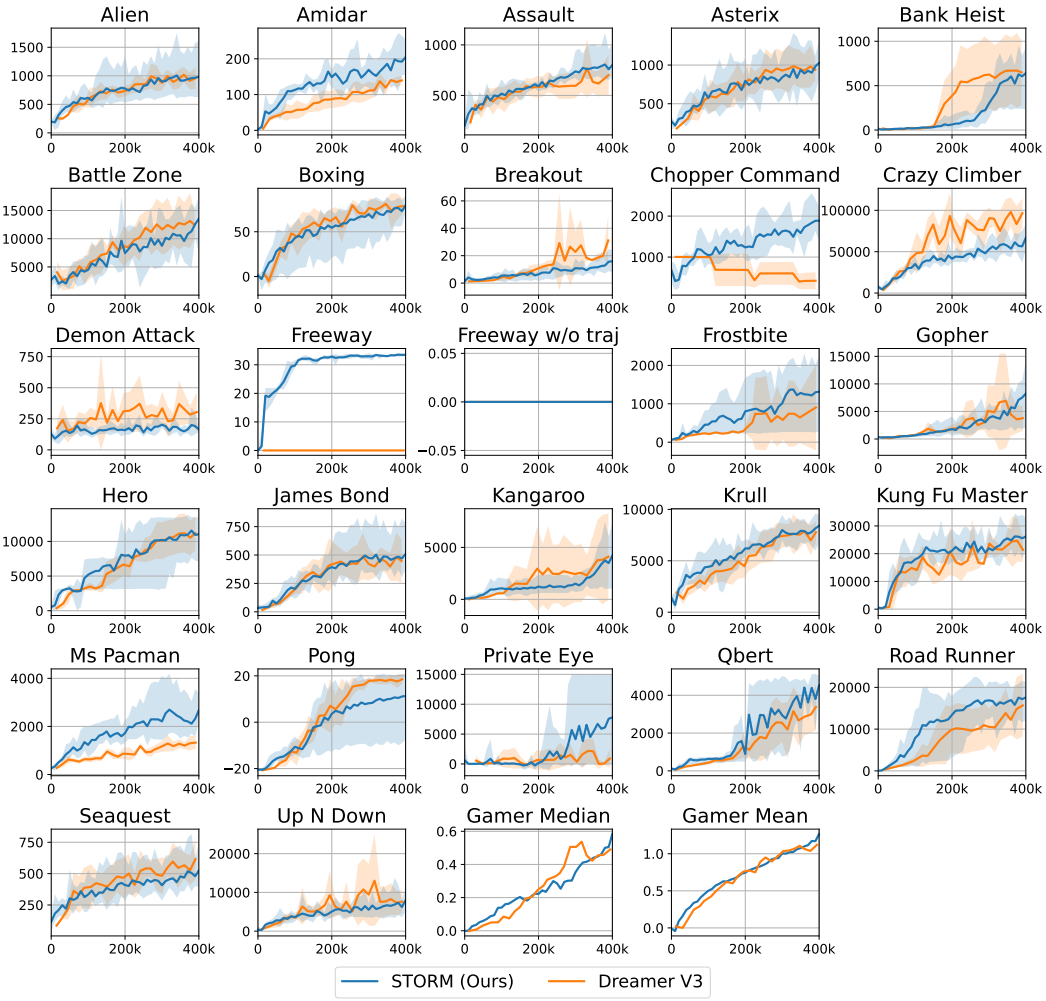

Figure 6: Performance comparison on the Atari 100k benchmark. Our method is represented in blue, while DreamerV3 [8] is in orange. The solid line represents the average result over 5 seeds, and the filled area indicates the range between the maximum and minimum results across these 5 seeds.

B Details of model structure

Table 2: Structure of the image encoder. The size of the submodules is omitted and can be derived from the shape of the tensors. ReLU refers to the rectified linear units used for activation, while Linear represents a fully-connected layer. Flatten and Reshape operations are employed to alter the indexing method of the tensor while preserving the data and their original order. Conv denotes a CNN layer [28], characterized by kernel $= 4$, stride $= 2$, and padding $= 1$. BN denotes the batch normalization layer[35].

Submodule	Output tensor shape
Input image (o_t)	$3 \times 64 \times 64$
Conv1 + BN1 + ReLU	$32 \times 32 \times 32$
Conv2 + BN2 + ReLU	$64 \times 16 \times 16$
Conv3 + BN3 + ReLU	$128 \times 8 \times 8$
Conv4 + BN4 + ReLU	$256 \times 4 \times 4$
Flatten	4096
Linear	1024
Reshape (produce $\mathcal{Z}_t$)	32×32

Table 3: Structure of the image decoder. DeConv denotes a transpose CNN layer [36], characterized by kernel $= 4$, stride $= 2$, and padding $= 1$.

Submodule	Output tensor shape
Random sample (z_t)	32×32
Flatten	1024
Linear + BN0 + ReLU	4096
Reshape	$256 \times 4 \times 4$
DeConv1 + BN1 + ReLU	$128 \times 8 \times 8$
DeConv2 + BN2 + ReLU	$64 \times 16 \times 16$
DeConv3 + BN3 + ReLU	$32 \times 32 \times 32$
DeConv4 (produce $\hat{o}_t$)	$3 \times 64 \times 64$

Table 4: Action mixer $e_t = m_\phi(z_t, a_t)$. Concatenate denotes combining the last dimension of two tensors and merging them into one new tensor. The variable A represents the action dimension, which ranges from 3 to 18 across different games. D denotes the feature dimension of the Transformer. LN is an abbreviation for layer normalization [37].

Submodule	Output tensor shape
Random sample (z_t), Action (a_t)	32×32, A
Reshape and concatenate	$1024 + A$
Linear1 + LN1 + ReLU	D
Linear2 + LN2 (output e_t)	D

Table 5: Positional encoding module. $w_{1:T}$ is a learnable parameter matrix with shape $T \times D$, and T refers to the sequence length.

Submodule	Output tensor shape
Input ($e_{1:T}$) Add ($e_{1:T} + w_{1:T}$) LN	$T \times D$

Table 6: Transformer block. Dropout mechanism [38] can prevent overfitting.

Submodule	Module alias	Output tensor shape
Input features (label as x_1)		$T \times D$
Multi-head self attention Linear1 + Dropout(p) Residual (add x_1) LN1 (label as x_2)	MHSA	$T \times D$
Linear2 + ReLU Linear3 + Dropout(p) Residual (add x_2) LN2	FFN	$T \times 2D$ $T \times D$ $T \times D$ $T \times D$

Table 7: Transformer based sequence model $h_{1:T} = f_\phi(e_{1:T})$. Positional encoding is explained in Table 5 and Transformer block is explained in Table 6.

Submodule	Output tensor shape
Input ($e_{1:T}$) Positional encoding Transformer blocks $\times K$ Output ($h_{1:T}$)	$T \times D$

Table 8: Pure MLP structures. A 1-layer MLP corresponds to a fully-connected layer. 255 is the size of the bucket of symlog two-hot loss [8].

Module name	Symbol	MLP layers	Input/ MLP hidden/ Output dimension
Dynamics head	g_ϕ^D	1	D/ -/ 1024
Reward predictor	g_ϕ^R	3	D/ D/ 255
Continuation predictor	g_ϕ^C	3	D/ D/ 1
Policy network	$\pi_\theta(a_t \mid s_t)$	3	D/ D/ A
Critic network	$V_\psi(s_t)$	3	D/ D/ 255

C Hyperparameters

Table 9: Hyerparameters. Note that the environment will provide a 'done' signal when losing a life, but will continue running until the actual reset occurs. This configuration aligns with the setup used in IRIS [12].

Hyperparameter	Symbol	Value
Transformer layers	K	2
Transformer feature dimension	D	512
Transformer heads	-	8
Dropout probability	p	0.1
World model training batch size	B	16
World model training batch length	T	64
Imagination batch size	B	1024
Imagination context length	C	8
Imagination horizon	L	16
Update world model every env step	-	1
Update agent every env step	-	1
Environment context length	-	16
Gamma	γ	0.985
Lambda	λ	0.95
Entropy coefficiency	η	3×10^{-4}
Critic EMA decay	σ	0.98
Optimizer	-	Adam [39]
World model learning rate	-	1.0×10^{-4}
World model gradient clipping	-	1000
Actor-critic learning rate	-	3.0×10^{-5}
Actor-critic gradient clipping	-	100
Gray scale input	-	False
Frame stacking	-	False
Frame skipping	-	4 (max over last 2 frames)
Use of life information	-	True

D Demonstration trajectory information

Table 10: To account for frame skipping, the frame count is multiplied by 4. These trajectories were gathered using pre-trained DQN agents [40].

Game	Episode return	Frames
Ms Pacman	5860	1612×4
Pong	18	2079×4
Freeway	27	2048×4

E Computational cost details and comparison

Table 11: Computational comparison. In the V100 column, an item marked with a star indicates extrapolation based on other graphics cards, while items without a star are tested using actual devices. The extrapolation method employed aligns with the setup used in DreamerV3 [8], where it assumes the P100 is twice as slow and the A100 is twice as fast.

Method	Original computing resource	V100 hours
SimPLe [10]	NVIDIA P100, 20 days	240^*
TWM [11]	NVIDIA A100, 10 hours NVIDIA GeForce RTX 3090, 12.5 hours	20^*
IRIS [12]	NVIDIA A100, 7 days for two runs	168^*
DreamerV3 [8]	NVIDIA V100, 12 hours	12
STORM	NVIDIA GeForce RTX 3090, 4.3 hours	9.3

F Atari video predictions

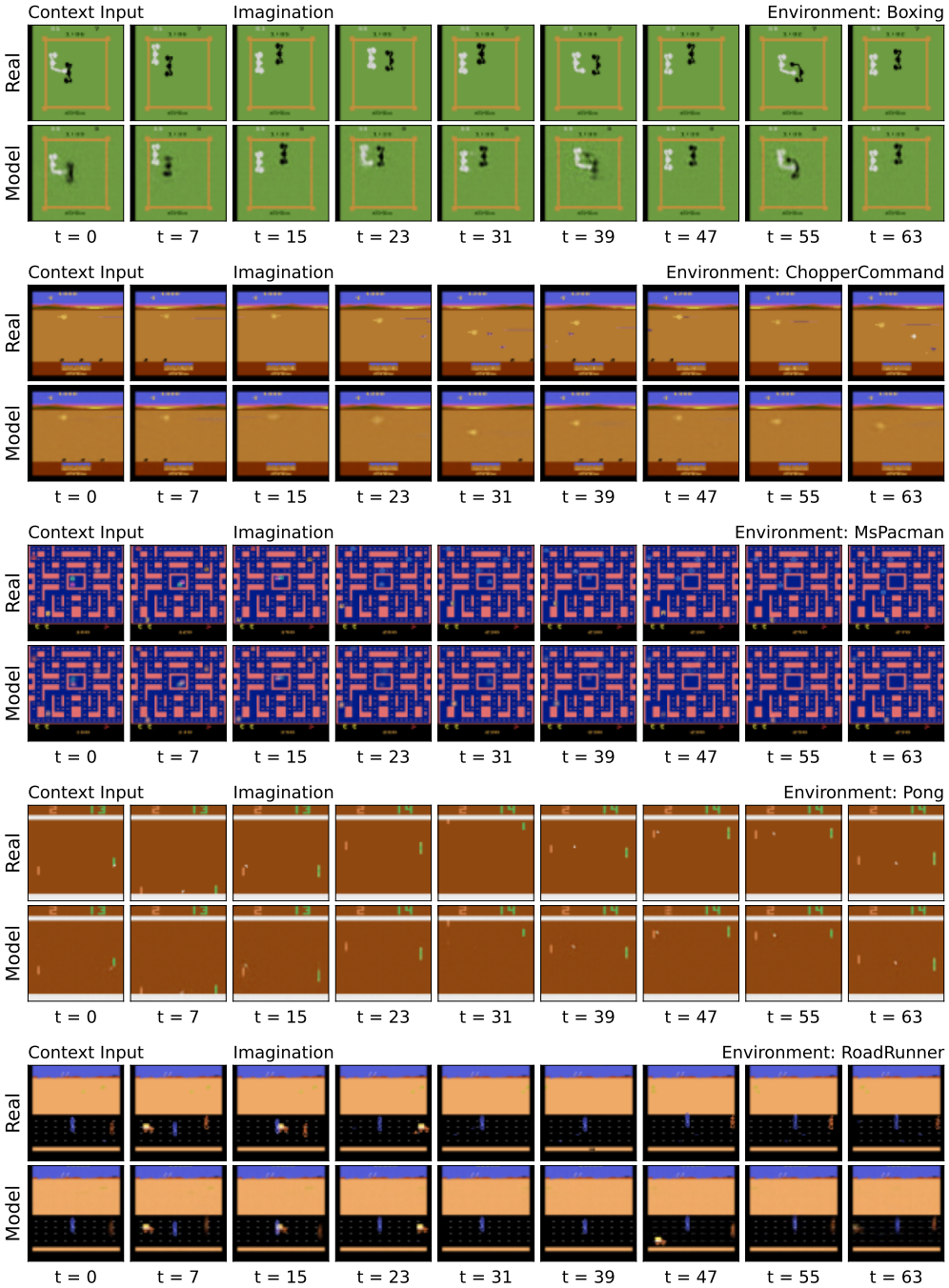

Figure 7: Multi-step predictions on several environments in Atari games. The world model utilizes 8 observations and actions as contextual input, enabling the imagination of future events spanning 56 frames in an auto-regressive manner.