Facing Off World Model Backbones: RNNs, Transformers, and S4

May 7, 2025

Reference

Fei Deng, Junyeong Park, and Sungjin Ahn. Facing off world model backbones: Rnns, transformers, and s4, 2023. URL https://arxiv.org/abs/2307.02064.

Linear State Space Models (SSMs)

- **Definition:** Maps 1-D input signal $u(t)$ to 1-D output signal $y(t)$
- **Formulation:**
  - **Continuous-time:**

$$\mathbf{s}'(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{B}u(t)$$
$$y(t) = \mathbf{C}\mathbf{s}(t) + \mathbf{D}u(t)$$

  - **Discrete-time:**

$$\mathbf{s}_k = \overline{\mathbf{A}}\mathbf{s}_{k-1} + \overline{\mathbf{B}}u_k$$
$$y_k = \overline{\mathbf{C}}\mathbf{s}_k + \overline{\mathbf{D}}u_k$$

- **Parallelizable SSMs (PSSMs):**
  - Combines benefits of RNNs and Transformers
  - Efficient autoregressive generation like RNNs
  - Parallelizable computation like Transformers
  - Computed via discrete convolution or parallel associative scan
- **PSSM Interface:**
  - **Single step:** $\mathbf{y}_k, \mathbf{s}_k = \text{PSSM}(\mathbf{u}_k, \mathbf{s}_{k-1})$
  - **Parallel:** $\mathbf{y}_{1:T}, \mathbf{s}_T = \text{PSSM}(\mathbf{u}_{1:T}, \mathbf{s}_0)$

The S4 Model

- **Core Idea:** SSMs with learnable parameters for sequence modeling
  - Matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ and step size Δ learned via gradient descent
  - Challenge: Expensive computation of $\overline{\mathbf{A}}$ powers
- **Key Innovation:** Diagonal Plus Low-Rank (DPLR) parameterization
  - $\mathbf{A} = \mathbf{\Lambda} - \mathbf{P}\mathbf{P}^*$ with diagonal $\mathbf{\Lambda}$
  - HiPPO-LegS: $\mathbf{\Lambda}_{ii} = -(2i+1)$, $P_i = \sqrt{2i+1}$
  - DPLR enables $O(N)$ computation of powers and inverse via Woodbury identity
  - Negative diagonal ensures stability and uniform recall across timescales
- **Implementation:** HiPPO initialization, multiple SSMs for vector sequences
- **Extensions:** S4D (diagonal $\mathbf{A}$), S5 (multi-input/output SSM)

S4WM [Deng et al., 2023]: World Modeling with Structured State Space Models

- **Key Innovation:** First world-model framework compatible with any PSSM
  - Works with S4, S5, and their variants
  - Operates in compact latent space rather than raw pixels
  - Efficiently learns long-range dynamics in image sequences
- **Architecture Components:**
  - **Encoder:** Compresses raw observations into latent representations
  - **PSSM Backbone:** Models temporal dynamics in latent space
  - **Decoder:** Reconstructs observations from predicted latent states
- **Advantages:**
  - **Efficiency:** Lower computational cost than pixel-space modeling
  - **Long-range dependencies:** Captures complex temporal patterns
  - **Flexibility:** Compatible with various PSSM architectures
  - **Scalability:** Handles longer sequences than traditional approaches

S4WM Architecture

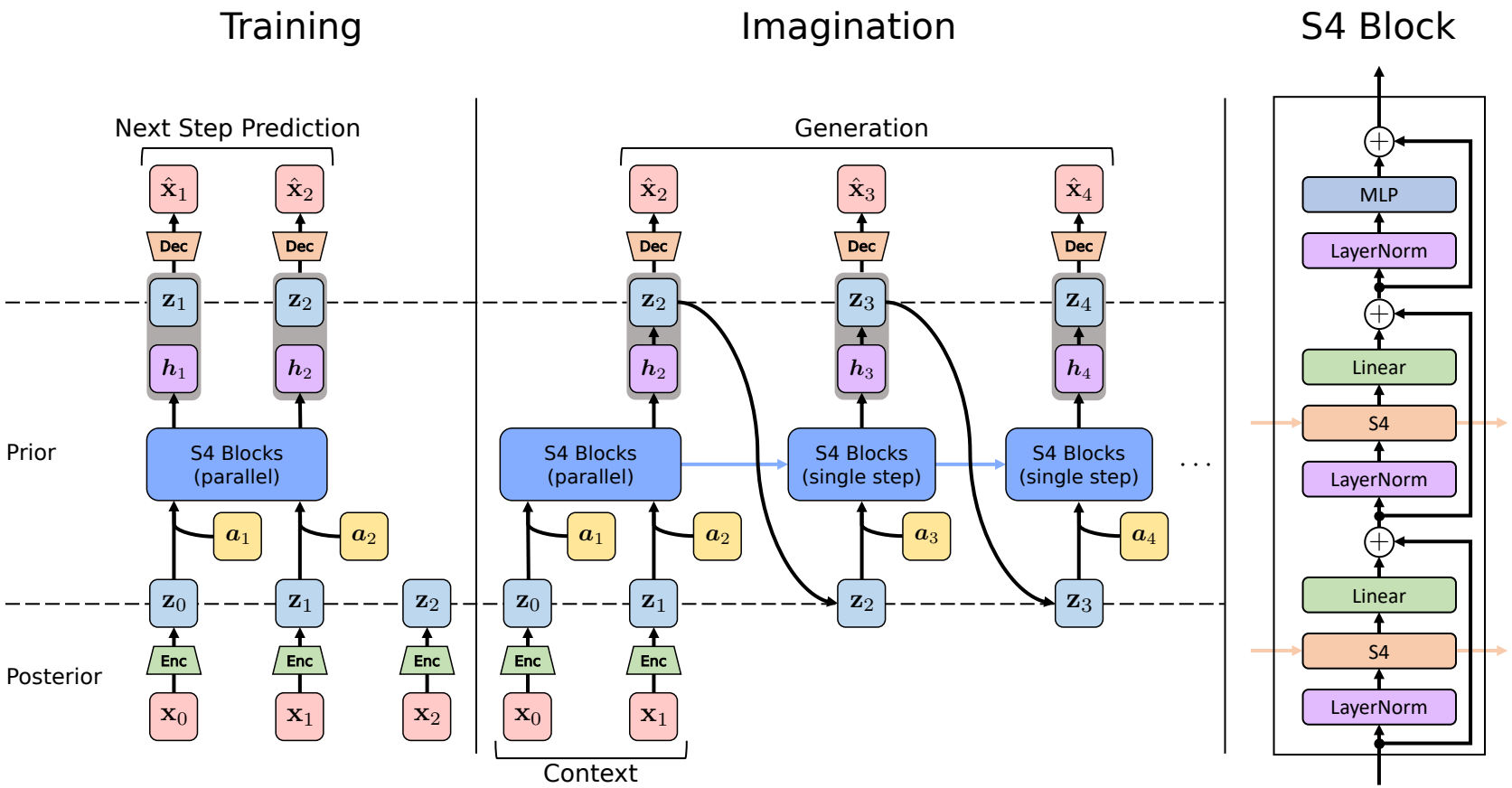

Figure: S4WM, the first S4-based world model for improving long-term memory. S4WM efficiently models the long-range dependencies of environment dynamics in a compact latent space, using a stack of S4 blocks. This crucially allows fully parallelized training and fast recurrent latent imagination. S4WM is a general framework that is compatible with any parallelizable SSM including S5 and other S4 variants.

Latent-Space Generative Process in S4WM

- **Stochastic Latent Variables:** Model future observations through latent space
  - Introduces stochastic latents $\mathbf{z}_{0:T}$ to model observations $\mathbf{x}_{1:T}$
  - Conditioned on initial observation $\mathbf{x}_0$ and actions $\mathbf{a}_{1:T}$
- **Generative Process:**

$$p(\mathbf{x}_{1:T} \mid \mathbf{x}_0, \mathbf{a}_{1:T}) = \int p(\mathbf{z}_0 \mid \mathbf{x}_0) \prod_{t=1}^{T} \underbrace{p(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{a}_{\leq t})}_{\text{prior}} \underbrace{p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{a}_{\leq t})}_{\text{likelihood}} \, d\mathbf{z}_{0:T}.$$

- **Key Advantage:** Dimensionality Reduction
  - Factorization through latent variables avoids direct modeling of high-dimensional $\mathbf{x}_t$
  - Enables efficient learning and inference in complex environments
  - PSSM backbone effectively captures temporal dependencies in compact latent space

History Encoding via PSSM Blocks

- To capture dependencies in $\{\mathbf{z}_{<t}, \mathbf{a}_{\leq t}\}$, S4WM feeds

$$\mathbf{g}_t = \mathrm{MLP}\big([\mathbf{z}_{t-1};\ \mathbf{a}_t]\big)$$

  into a **stack** of PSSM blocks (e.g. S4 layers).
- These blocks compute in parallel during training and sequentially during rollout:

$$\textbf{(single step)} \qquad \mathbf{h}_t,\ \mathbf{s}_t = \mathrm{PSSM\_Blocks}(\mathbf{g}_t,\ \mathbf{s}_{t-1})$$

$$\textbf{(parallel)} \qquad \mathbf{h}_{1:T},\ \mathbf{s}_T = \mathrm{PSSM\_Blocks}(\mathbf{g}_{1:T},\ \mathbf{s}_0)$$

- Here $\mathbf{h}_t$ is a fixed-size embedding summarizing all past latents and actions

Probabilistic Components

- **Prior**

$$p(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{a}_{\leq t}) = \mathrm{MLP}(\mathbf{h}_t),$$

  which outputs e.g. mean and variance for a Gaussian latent prior.

- **Likelihood**

$$p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{a}_{\leq t}) = \mathcal{N}\big(\hat{\mathbf{x}}_t, \mathbf{I}\big), \quad \hat{\mathbf{x}}_t = \mathrm{Decoder}\big([\mathbf{h}_t; \mathbf{z}_t]\big).$$

  This lets the model reconstruct pixels from the latent history

Variational Training

- We introduce an CNN **encoder** that maps each $\mathbf{x}_t$ to a posterior over $\mathbf{z}_t$:

$$q(\mathbf{z}_{0:T} \mid \mathbf{x}_{0:T}, \mathbf{a}_{1:T}) = \prod_{t=0}^{T} q(\mathbf{z}_t \mid \mathbf{x}_t), \quad q(\mathbf{z}_0 \mid \mathbf{x}_0) = p(\mathbf{z}_0 \mid \mathbf{x}_0).$$

- The **ELBO** objective is

$$\log p(\mathbf{x}_{1:T} \mid \mathbf{x}_0, \mathbf{a}_{1:T}) \geq$$
$$\mathbb{E}_q \left[\sum_{t=1}^{T} \log p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{a}_{\leq t}) \; - \; \mathrm{KL}\big(q(\mathbf{z}_t \mid \mathbf{x}_t) \,\|\, p(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{a}_{\leq t})\big) \right].$$

- All $\mathbf{z}_t$ samples and PSSM computations run in parallel during training

Training Algorithm

Algorithm 1: S4WM Training

Input: $(\mathbf{x}_0, \mathbf{a}_1, \mathbf{x}_1, \ldots, \mathbf{a}_T, \mathbf{x}_T)$

▷ Obtain posterior latents

for *time step* $t = 0, \ldots, T$ ***parallel*** **do**

  $\mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{x}_t) = \text{Encoder}(\mathbf{x}_t)$

▷ Prepare inputs to S4 blocks

for *time step* $t = 1, \ldots, T$ ***parallel*** **do**

  $\mathbf{g}_t = \text{MLP}([\mathbf{z}_{t-1}; \mathbf{a}_t])$

▷ Encode history by S4 blocks

$\mathbf{h}_{1:T}, \mathbf{s}_T = \text{S4Blocks}(\mathbf{g}_{1:T}, \mathbf{s}_0)$

▷ Compute prior and decode posterior latents

for *time step* $t = 1, \ldots, T$ ***parallel*** **do**

  $p(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{a}_{\leq t}) = \text{MLP}(\mathbf{h}_t)$

  $\hat{\mathbf{x}}_t = \text{Decoder}([\mathbf{h}_t; \mathbf{z}_t])$

Compute objective by ELBO Equation

Imagination Algorithm

Algorithm 2: S4WM Imagination

Input: context $(\mathbf{x}_{0:C}, \mathbf{a}_{1:C})$, query $\mathbf{a}_{C+1:T}$

▷ Encode context

for *context step* $t = 0, \ldots, C$ ***parallel*** **do**

$\mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{x}_t) = \text{Encoder}(\mathbf{x}_t)$

$\mathbf{g}_{t+1} = \text{MLP}([\mathbf{z}_t; \mathbf{a}_{t+1}])$

$\mathbf{h}_{1:C+1}, \mathbf{s}_{C+1} = \text{S4Blocks}(\mathbf{g}_{1:C+1}, \mathbf{s}_0)$

▷ Imagine in latent space

$\mathbf{z}_{C+1} \sim p(\mathbf{z}_{C+1}|\mathbf{z}_{0:C}, \mathbf{a}_{1:C+1}) = \text{MLP}(\mathbf{h}_{C+1})$

for *query step* $t = C + 2, \ldots, T$ **do**

$\mathbf{g}_t = \text{MLP}([\mathbf{z}_{t-1}; \mathbf{a}_t])$

$\mathbf{h}_t, \mathbf{s}_t = \text{S4Blocks}(\mathbf{g}_t, \mathbf{s}_{t-1})$

$\mathbf{z}_t \sim p(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{a}_{\leq t}) = \text{MLP}(\mathbf{h}_t)$

▷ Decode imagined latents

for *query step* $t = C + 1, \ldots, T$ ***parallel*** **do**

$\hat{\mathbf{x}}_t = \text{Decoder}([\mathbf{h}_t; \mathbf{z}_t])$

Environments for Evaluating World Model Memory Capabilities

- **Evaluation Focus:** Memory capabilities in world models
  - Long-term imagination
  - Context-dependent recall
  - Reward prediction
  - Memory-based reasoning
- **Environment Design:**
  - 3D Memory Maze and 2D MiniGrid environments with partial observability
  - Each environment targets specific memory capability
  - Deterministic environments with moderate visual complexity
- **Evaluation Protocol:**
  - World model observes context phase
  - Evaluated on imagination given query phase actions
  - Mean squared error (MSE) as primary metric
  - Evaluation on unseen episodes from same policy distribution
- **Advantages over Prior Work:**
  - Provides deeper insights than final agent performance
  - Isolates world model capabilities from policy learning
  - Enables focused assessment of specific memory aspects
  - Paves way for improved model-based agents with better memory

World Model Baselines and Performance Comparison

- **Baseline Models:**
  - **RSSM-TBTT:** RNN-based world model with truncated backpropagation
  - **TSSM-XL:** Transformer-based model using Transformer-XL for longer sequences
  - **S4WM:** Structured State Space Model-based world model
- **Computational Efficiency:**
  - **Training Speed:** S4WM and TSSM-XL train faster due to parallel computation
  - **Memory Usage:** RSSM-TBTT is most memory-efficient during training
  - **Imagination Throughput:** RSSM-TBTT achieves $\sim 10\times$ faster inference

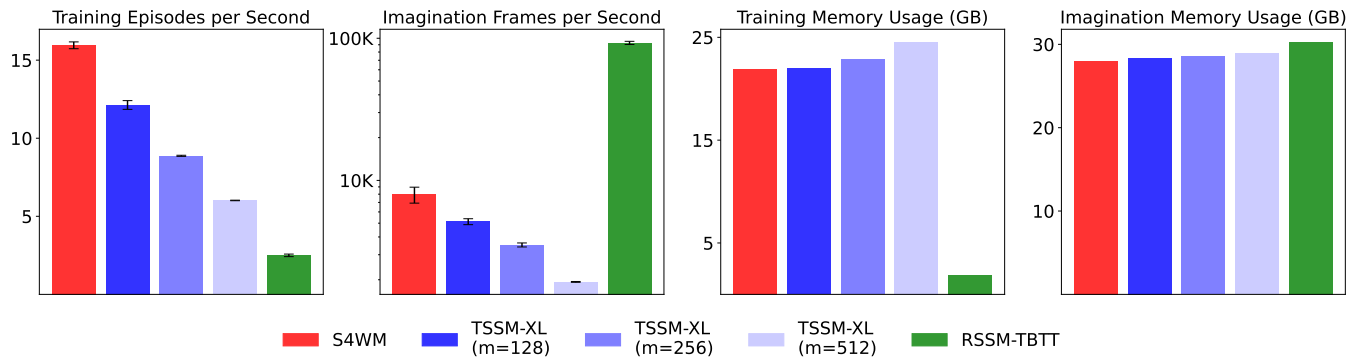

Figure: Comparison of speed and memory usage during training and imagination. S4WM is the fastest to train, while RSSM-TBTT is the most memory-efficient during training and has the highest throughput during imagination.

Long-Term Imagination

	Two Rooms (301 | 200)		Four Rooms (501 | 500)		Ten Rooms (1101 | 900)	
	Recon. MSE (↓)	Gen. MSE (↓)	Recon. MSE (↓)	Gen. MSE (↓)	Recon. MSE (↓)	Gen. MSE (↓)
RSSM-TBTT	**1.7**	62.2	**1.5**	219.4	**1.5**	323.1
TSSM-XL	2.5	62.9	2.4	224.4	2.6	360.4
S4WM	1.8	**27.3**	1.7	**44.0**	1.8	**224.4**

Table: Long-term imagination evaluation. Format: (context | query steps). All models show good reconstruction, but S4WM excels at generation up to 500 steps. Ten Rooms remains challenging for all models.

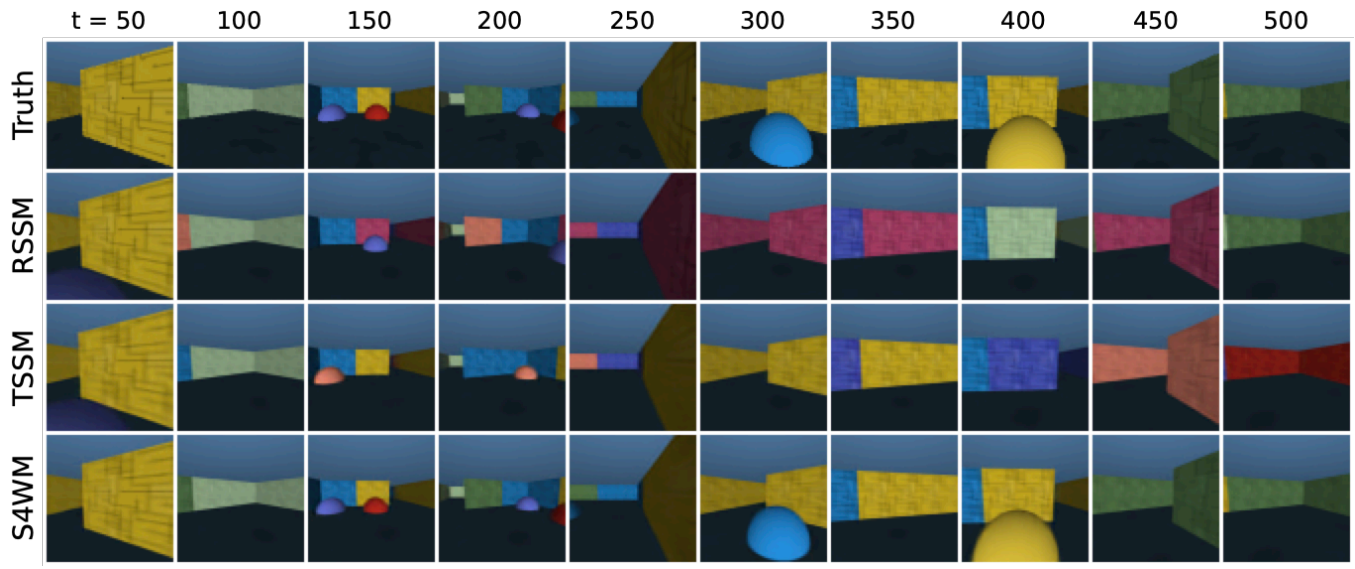

Figure: Long-term imagination in Four Rooms. RSSM-TBTT and TSSM-XL show significant errors in walls, objects, and positions, while S4WM generates more accurately with only minor position errors.

Context-Dependent Recall

- **Teleport Experiment:** Agent observes initial context, then is teleported to random point in history and must recall subsequent events using the same action sequence
- **Evaluates:** Models' ability to access specific memories from arbitrary points in history

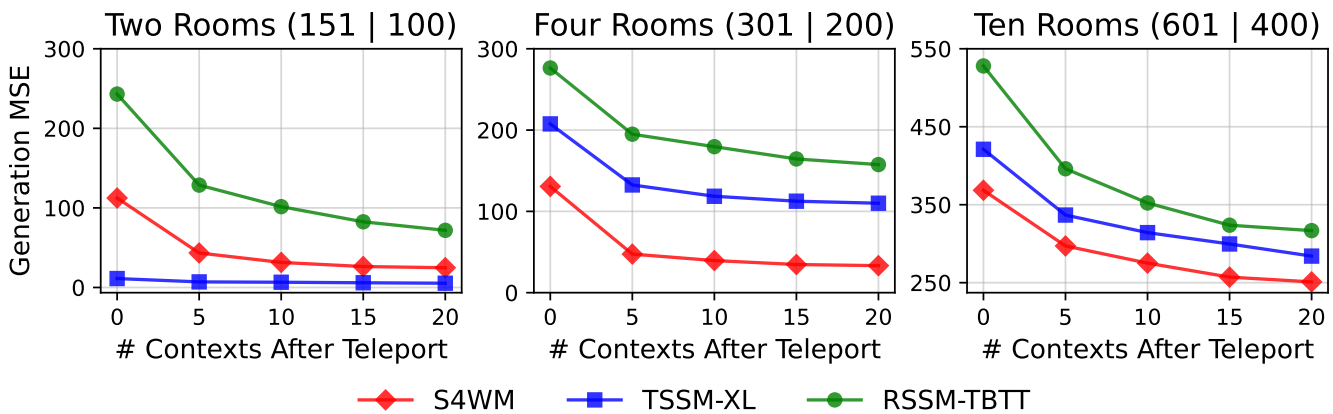

Figure: Context-dependent recall in teleport environments (format: context | query steps). TSSM-XL excels with short contexts, requiring no additional observations, while S4WM performs best with longer context phases.

Reward Prediction

- To facilitate policy learning within imagination, world models need to accurately predict the rewards
- We evaluate reward prediction accuracy over long time horizons using the visually simpler 2D MiniGrid environment
- **Distracting Memory environment:**
  - More challenging than original MiniGird Memory environment
  - Distractors of random colors placed in hallway
  - Reward of 1 given if square reached matches color of square in left room
  - Model must ignore distractors while tracking agent's position
- **Results:**
  - Only S4WM accurately predicts rewards within imagination
  - TSSM-XL succeeds with full sequence but fails in imagination
  - RSSM-TBTT completely fails (close to random guessing)
  - Failures mainly due to inability to track agent's position

Reward Prediction (cont.)

Table: Reward prediction accuracy in the Distracting Memory environments. Each environment is labeled with (context steps | query steps). S4WM succeeds in all environments. TSSM-XL has limited success when observing the full sequence, but fails to predict rewards within imagination. RSSM-TBTT completely fails.

	Width = 100 (199 | 51)		Width = 200 (399 | 101)		Width = 400 (799 | 201)	
	Inference Accuracy (↑)	Imagination Accuracy (↑)	Inference Accuracy (↑)	Imagination Accuracy (↑)	Inference Accuracy (↑)	Imagination Accuracy (↑)
RSSM-TBTT	47.9%	49.6%	48.7%	48.4%	50.4%	52.2%
TSSM-XL	**100.0**%	51.2%	99.9%	51.3%	50.4%	51.0%
S4WM	**100.0**%	**100.0**%	**100.0**%	**100.0**%	**100.0**%	**100.0**%

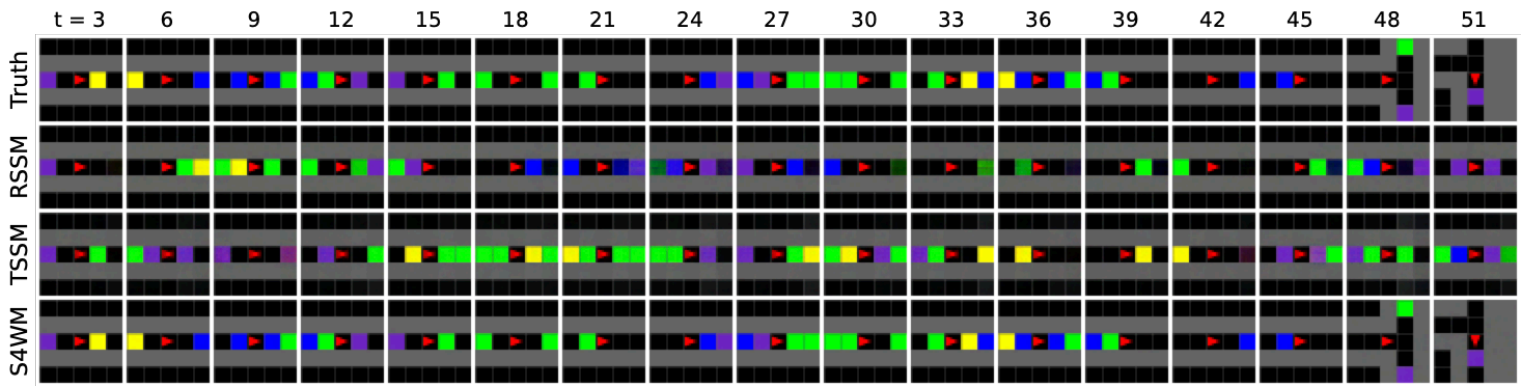

Figure: Imagination in the Distracting Memory environment of width 100. RSSM-TBTT and TSSM-XL fail to keep track of the agent's position, leading to inaccurate reward prediction within imagination.