NEURAL INFORMATION PROCESSING SYSTEMS

Parallelizing Model-based Reinforcement Learning Over the Sequence Length

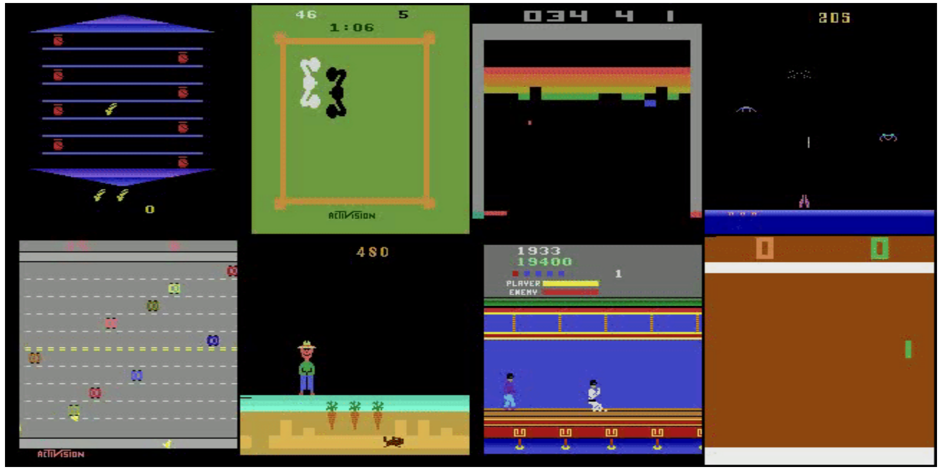
Zirui Wang, Yue Deng, Junfeng Long, Yin Zhang

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

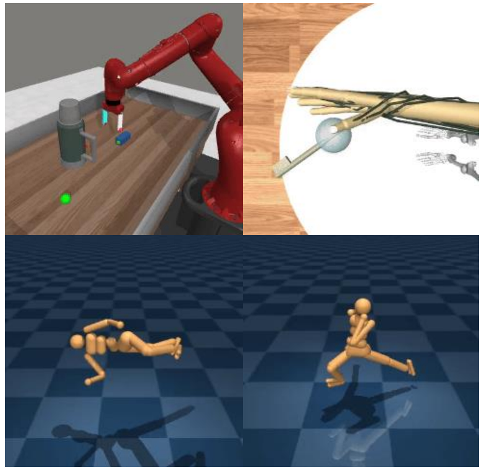
Motivations

MBRL has demonstrated stunning **Sample Efficiency** in:

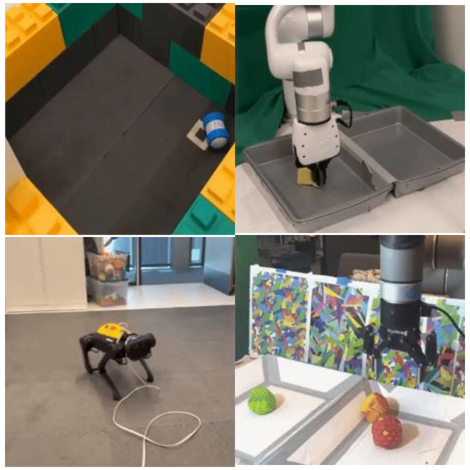
- **Continuous** and **Discrete** Action Space
- **Proprioception** and **Pixel** Observation
- **Simulator**, **Game**, and **Robots in Real-World**



(Micheli et al, 2023)



(Hansen et al, 2023)



(Wu et al, 2022)

Motivations

MBRL works so well, but what must we give it return?

Extra **Computations**, **Memory**, and **Training Time**

Training overhead in Atari100K benchmark (roughly 2 hours of gameplay)

MBRL Method	Wall-clock Time	Hardware Requirement
SimPLE [KBM+'2019]	5 days	1 x P100
EfficientZero [YLK+'2021]	7 hours	4 x RTX3090
IRIS [MAF+'2023]	7 days	1 x A100
TWM [RHU+'2023]	10 hours	1 x A100
DreamerV3 [HPB+'2023]	12 hours	1 x V100
STORM [ZWS+'2023]	9.3 hours	1 x RTX3090

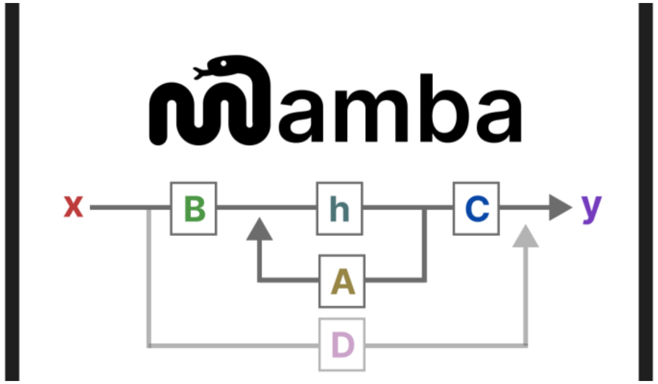
Motivations

How can we overcome these challenges in MBRL?

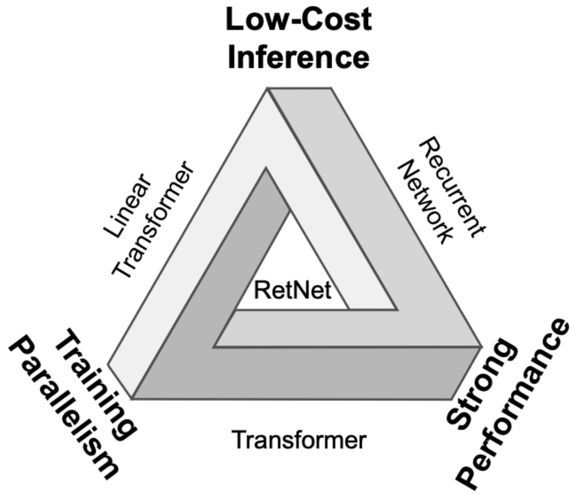
Let’ s take a glimpse at the challengers of Transformers in LLM domain

Key idea: **Parallel Training** and **Recurrent Inference**

Key technology: **Parallel Scan**



(Gu et al, 2023)



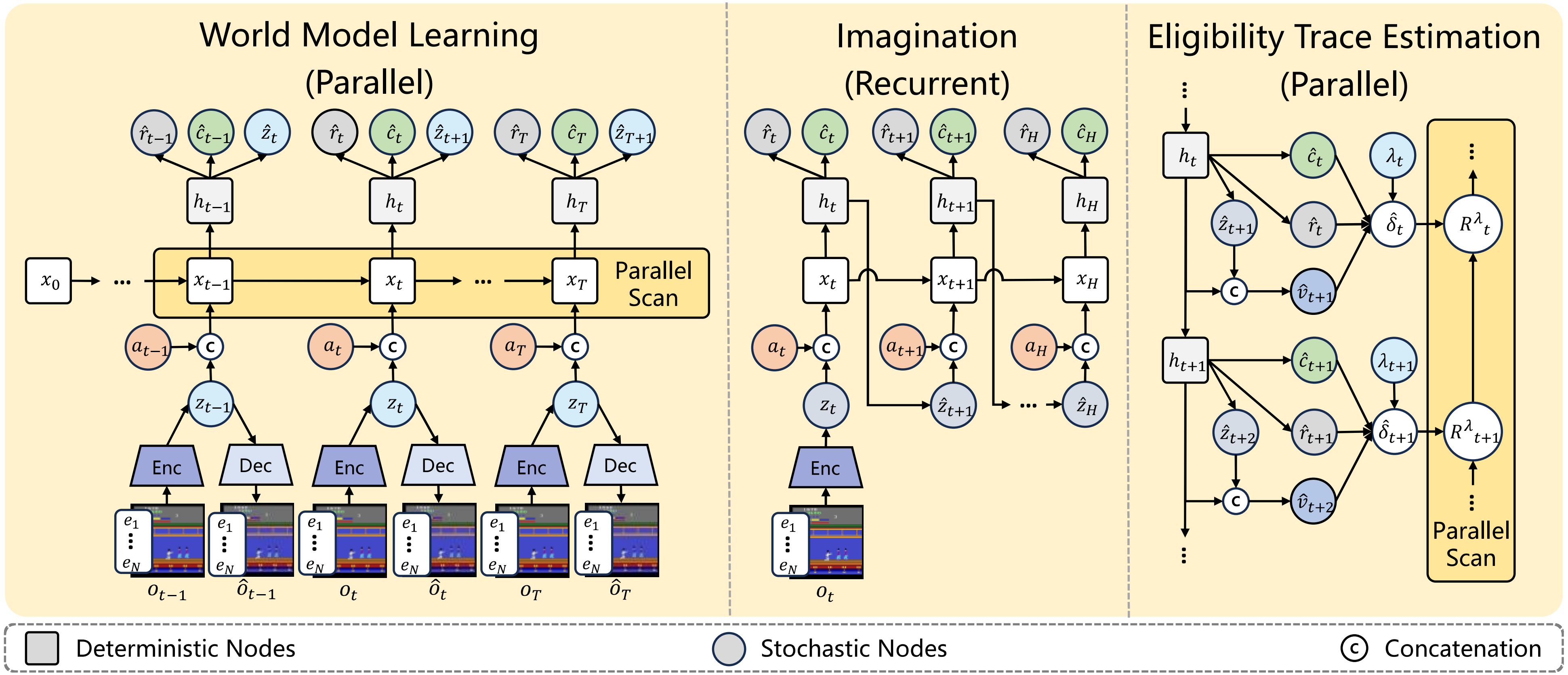
(Sun et al, 2023)

(Peng et al, 2023)

PaMoRL

We propose **Pa**rallelized **Mo**del-based **RL** framework

Apply parallel scan for both **World Model Learning & Policy Learning**



Parallel World Model

浙江大学 ZHEJIANG UNIVERSITY

We employ the **modified Linear Attention** in our world model

Kernel function in Linear Attention

Token mixing module

Data-dependent decay rate

(i.e. forget gate)

Post-Norm

$$
\begin{aligned}
q_t, k_t &= \boxed{1 + \mathrm{ELU}(u_t W_q), 1 + \mathrm{ELU}(u_t W_k)}, \\
v_t &= \mathrm{Sigmoid}(u_t W_r) \odot u_t W_v, \\
g_t &= \mathrm{Sigmoid}(\boxed{\mu \odot u_t + (1 - \mu) \odot u_{t-1}} W_g), \\
x_t &= \boxed{g_t \odot x_{t-1}} + k_t^\top v_t, \\
y_t &= \boxed{\mathrm{RMSNorm}(q_t x_t)} W_h + u_t, \\
h_t &= \mathrm{SiLU}(y_t W_g) \odot y_t W_y + y_t.
\end{aligned}
$$

Parallel World Model

	Minimum complexity			More expressive		
Architecture	Training	Inference step	Imagination step	Parallel	Resettable	Selective
Atten	$\mathcal{O}(L^2)$	$\mathcal{O}(L^2)$	$\mathcal{O}((L+H)^2)$	✓	✓	✓
RNN	$\mathcal{O}(L)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	✗	✓	✓
SSM (FFT)	$\mathcal{O}(L \log L)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	✓	✗	✗
SSM (Scan)	$\mathcal{O}(L)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	✓	✓	✗
Lin-Atten (Scan)	$\mathcal{O}(L)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	✓	✓	✓

Parallelized Eligibility Trace Estimation

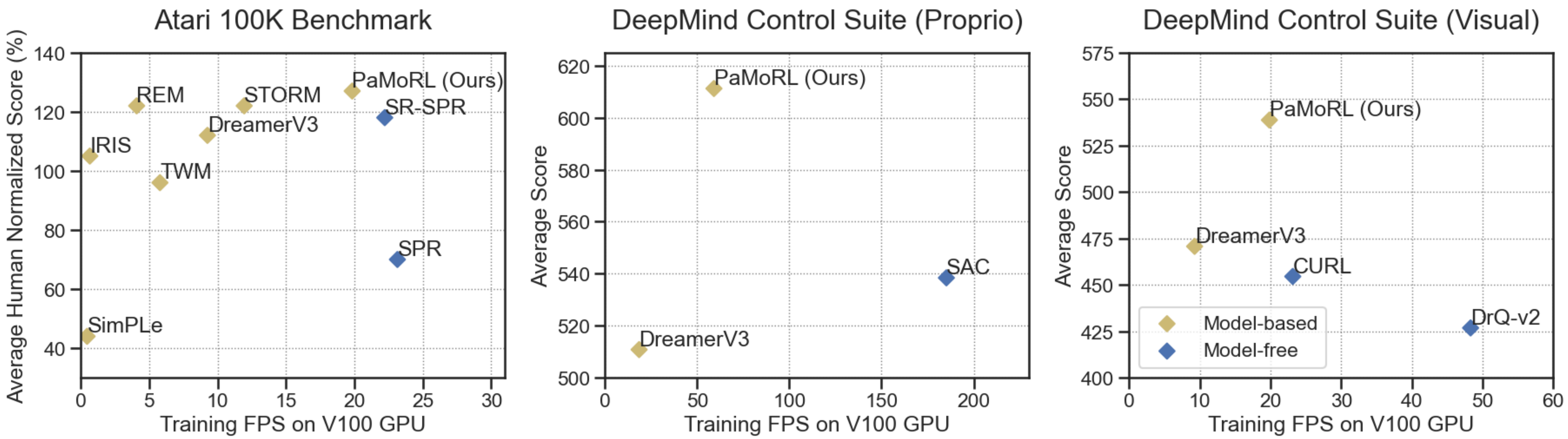
浙江大学 ZHEJIANG UNIVERSITY

TD-λ: $$R_t^\lambda = \hat{r}_t + (\gamma \hat{c}_t)\left[(1-\lambda)v_\phi(s_{t+1}) + \lambda R_{t+1}^\lambda\right]$$
$$= (\lambda \gamma \hat{c}_t) R_{t+1}^\lambda + [\hat{r}_t + (1-\lambda)(\gamma \hat{c}_t) v_\phi(s_{t+1})]$$

Eligibility Trace: $$R_t = v_\phi(s_t) + E_t,$$
$$E_t = (\gamma \hat{c}_t \lambda_t) E_{t+1} + \rho_t \left[\hat{r}_t + (\gamma \hat{c}_t) v_\phi(s_{t+1}) - v_\phi(s_t)\right]$$

We notice that the computation of **Eligibility Trace Estimations** can also be speeded up by using parallel scan 🥳

Results

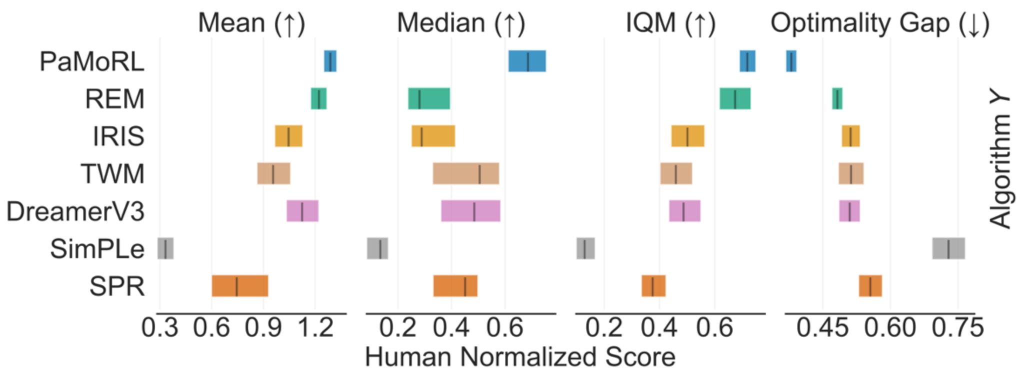


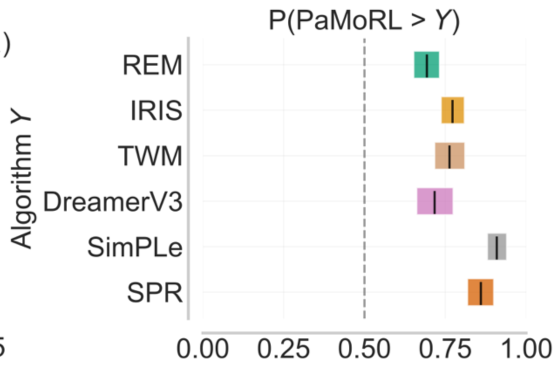
Task	Atari 100K	Proprio (easy)	Proprio (hard)	Vision (easy)	Vision (hard)
Runtime	3.5 hours	0.94 hours	1.88 hours	2.74 hours	7.1 hours

PaMoRL achieves **MBRL-level sample efficiency**,

and **MFRL-level computational efficiency**

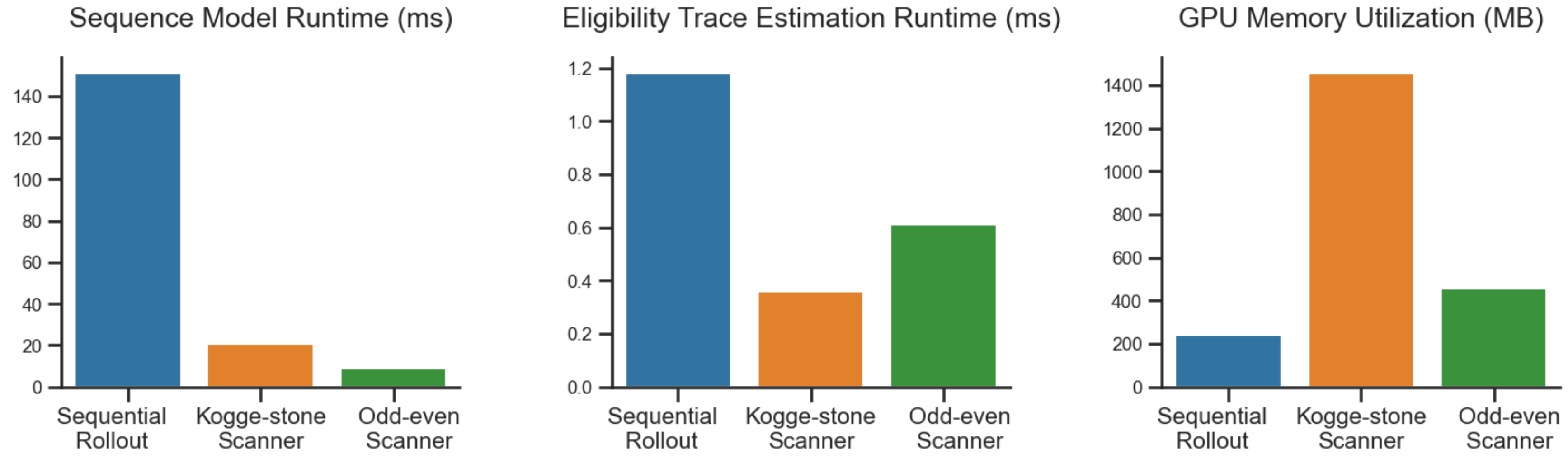
Results

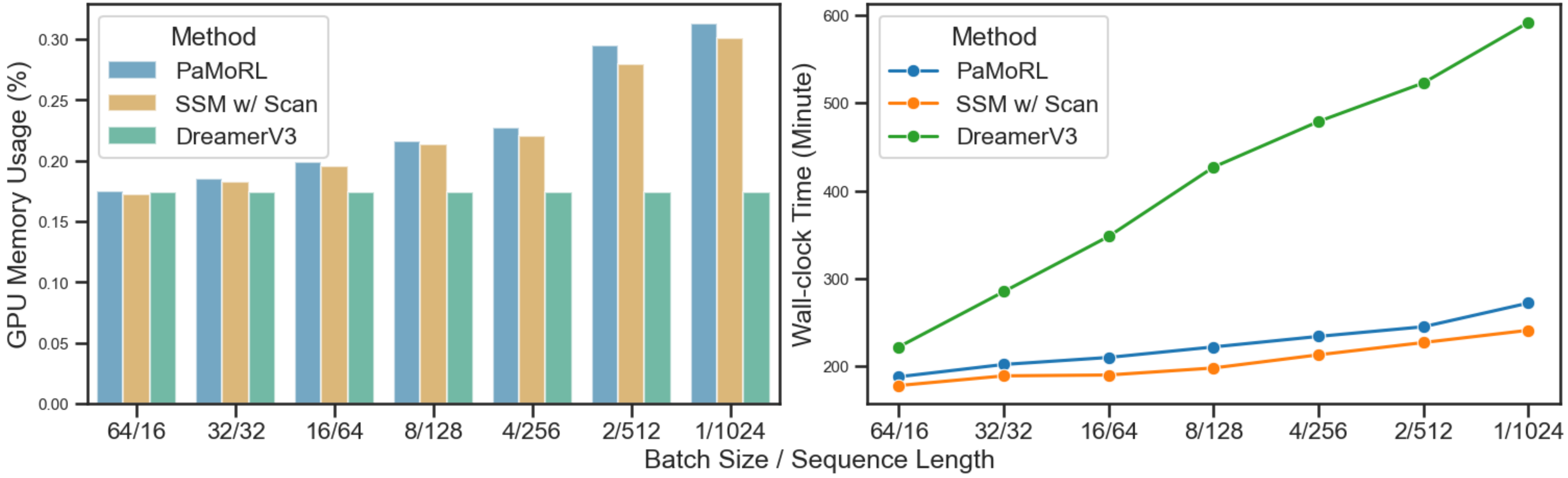




Task	Proprio Control			Vision Control			
	SAC	DreamerV3	PaMoRL (Ours)	CURL	DrQ-v2	DreamerV3	PaMoRL (Ours)
Cartpole Balance	**997.6**	839.6	994.7	963.3	**965.5**	956.4	610.3
Cartpole Balance Sparse	993.1	559	**997.4**	999.4	**1000**	813	996.5
Cartpole Swingup	**861.6**	527.7	773.6	**765.4**	756	374.8	281.9
Cup Catch	949.9	729.6	**957.9**	932.3	468	947.7	**966.3**
Finger Spin	**900**	765.8	835.8	**850.2**	459.4	633.2	765.3
Pendulum Swingup	158.9	**830.4**	707.1	144.1	233.3	**619.3**	26.6
Reacher Easy	744	693.4	**761.6**	467.9	722.1	441.4	**950.2**
Reacher Hard	646.5	**768**	645.9	112.7	**202.9**	120.4	103.7
Cartpole Swingup Sparse	256.6	172.7	**542.3**	8.8	81.2	**392.4**	263.6
Cheetah Run	**680.9**	400.8	313.2	405.1	418.4	587.3	**935.6**
Finger Turn Easy	**630.8**	560.5	617.1	371.5	286.8	366.6	**886.2**
Finger Turn Hard	414	**474.2**	389.7	236.3	268.4	258.5	**500.1**
Hopper Hop	0.1	9.7	**387.5**	84.5	26.3	76.3	**426.9**
Hopper Stand	3.8	**296.1**	151.5	627.7	290.2	**652.5**	189.7
Quadruped Run	139.7	**289**	246.7	170.9	339.4	168	**344.8**
Quadruped Walk	237.5	256.2	**457.9**	131.8	311.6	122.6	**371.6**
Mean	538.4	510.8	**611.2**	454.5	426.8	470.7	**538.7**
Median	**638.7**	543.4	631.5	388.3	325.5	416.9	**463.5**

Results

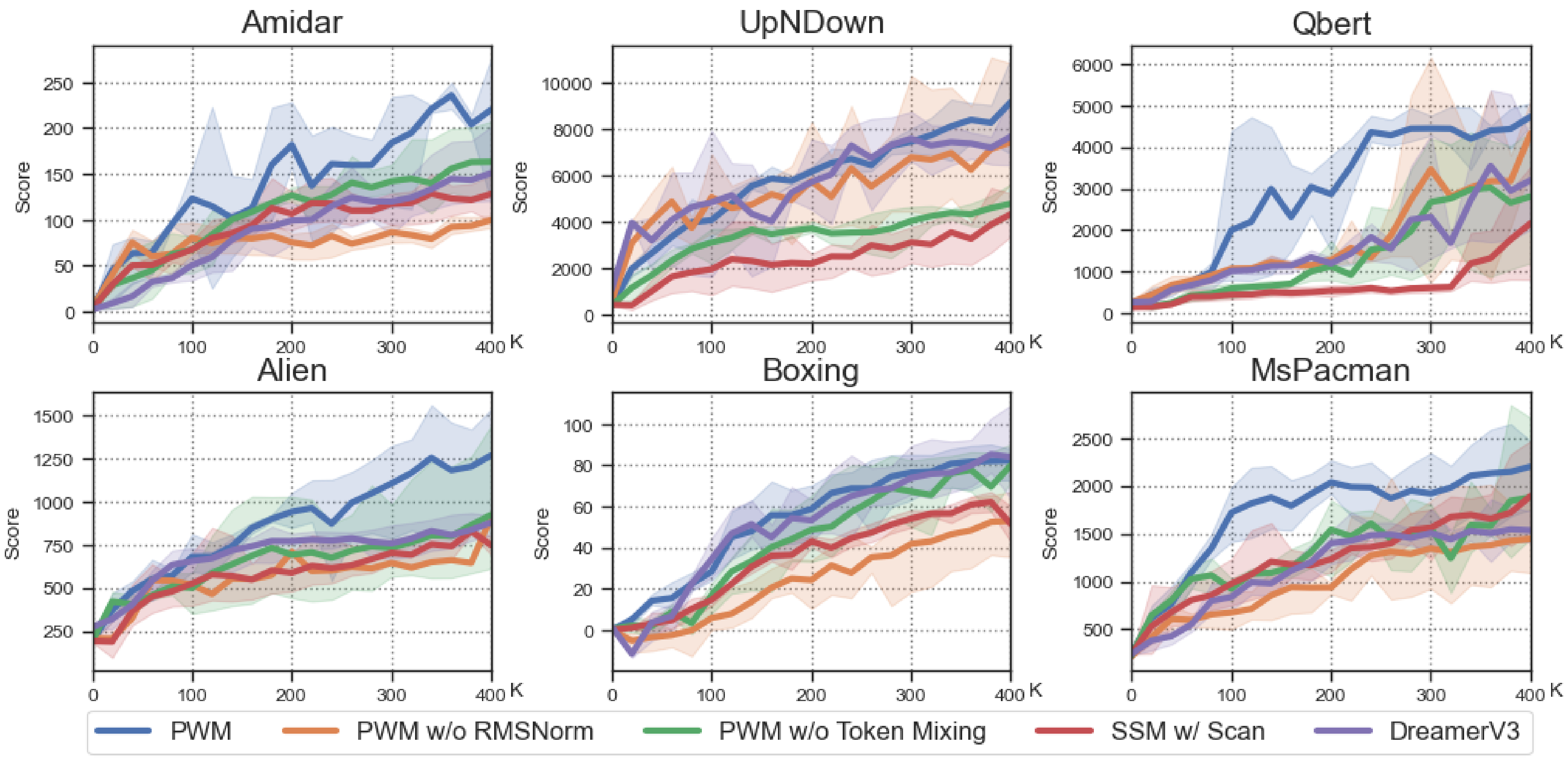




Significant speed-ups with acceptable extra memory overheads

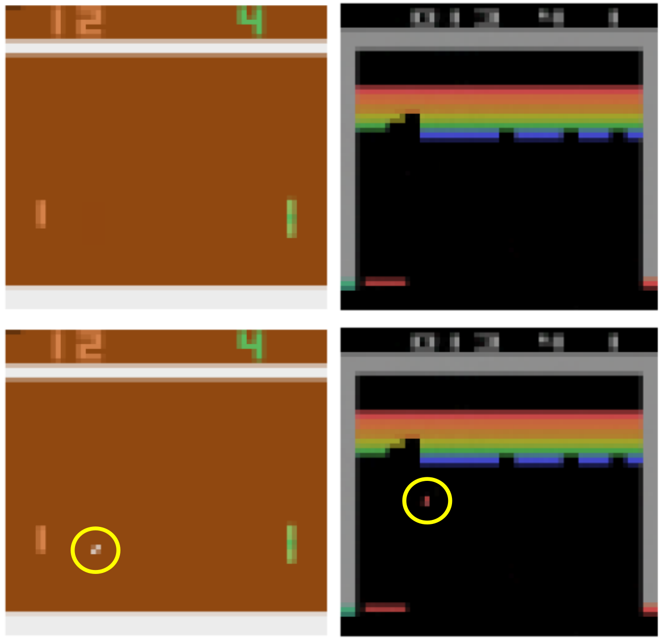
Results

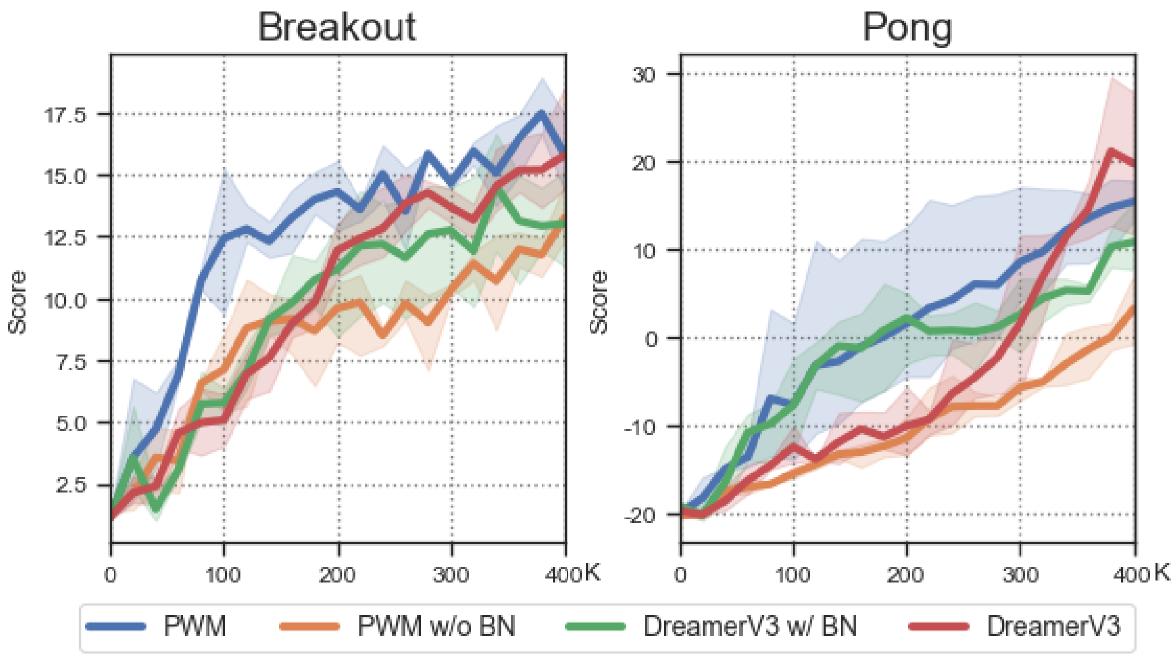
浙江大学 ZHEJIANG UNIVERSITY



RMSNorm, **Token Mixing**, and **Data-dependent Decay Rate** matters

Results





BatchNorm Trick is crucial for extracting the details of pixel observations

Parallelizing Model-based Reinforcement Learning Over the Sequence Length

Zirui Wang, Yue Deng, Junfeng Long, Yin Zhang

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Thanks!

ziseoiwong@zju.edu.cn

https://github.com/Wongziseoi/PaMoRL