Lecture 13: RL with Sequence Modeling
CS885 Reinforcement Learning

2025-02-25

Complementary readings:
Esslinger, Platt & Amato (2022). Deep Transformer Q-Networks for Partially Observable Reinforcement Learning. arXiv.
Chen et al.. (2021). Decision transformer: Reinforcement learning via sequence modeling. NeurIPS, 34, 15084-15097.
Gu, Goel, & Ré (2022). Efficiently modeling long sequences with structured state spaces. ICLR.
Gu, Dao, Ermon, Rudra & Ré (2020). Hippo: Recurrent memory with optimal polynomial projections. NeurIPS, 33, 1474-1487.
Gu & Dao (2023) Mamba: Linear-Time Sequence Modeling with Selective State Spaces, First Conference on Language Modeling.
Cao et al. (2024). Mamba as Decision Maker: Exploring Multi-scale Sequence Modeling in Offline Reinforcement Learning. CoRR.

Pascal Poupart
David R. Cheriton School of Computer Science

UNIVERSITY OF WATERLOO

Outline

- Transformers
  - Deep Transformer Q-Networks
  - Decision Transformers
- Structured State Space Sequence (S4) Model and Mamba
  - MambaDM: Mamba as Decision Maker

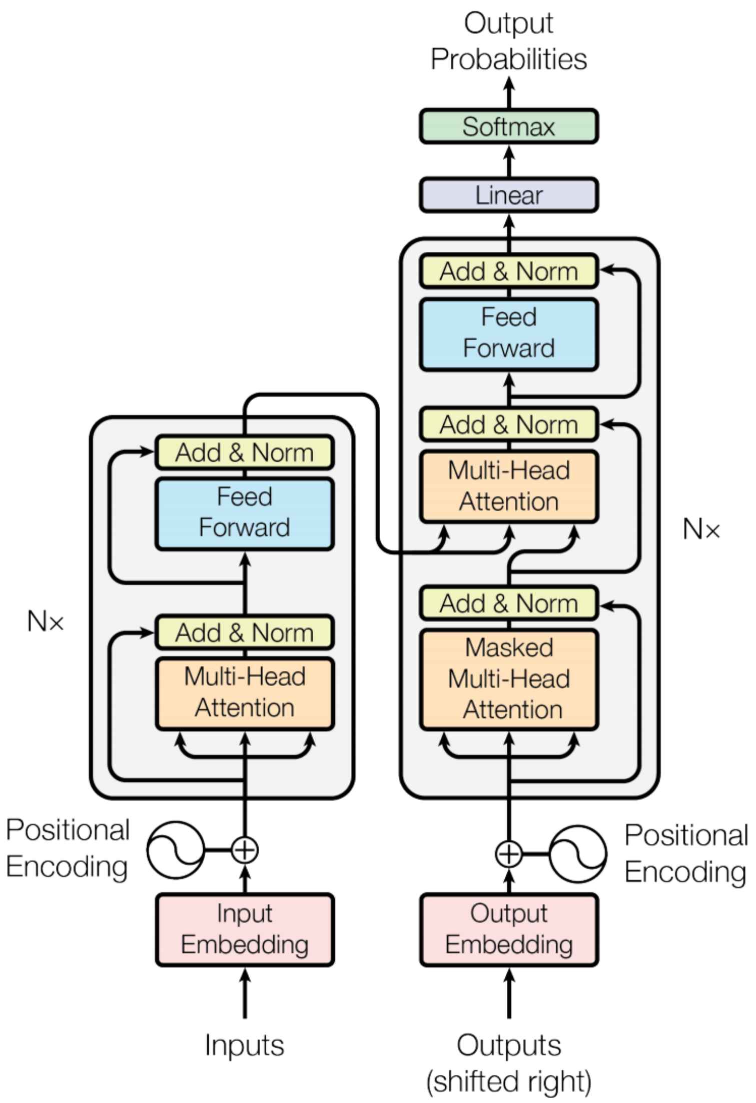
Sequence Models

- Hidden Markov Models
- Recurrent Neural Networks
- Transformers
- Structured State Space Sequence (S4) Models
- MAMBA

Transformers and Attention

- Viswani et al. (2017)
  Attention is all you need

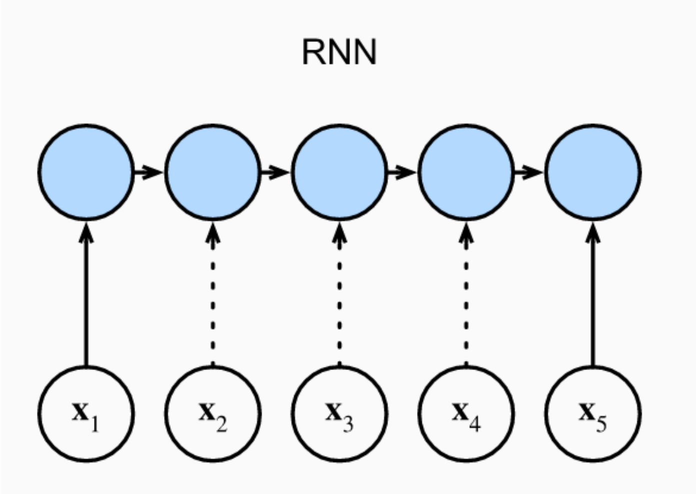
$$attention(Q, K, V) = softmax\left(\frac{Q^T K}{\sqrt{d_k}}\right) V$$

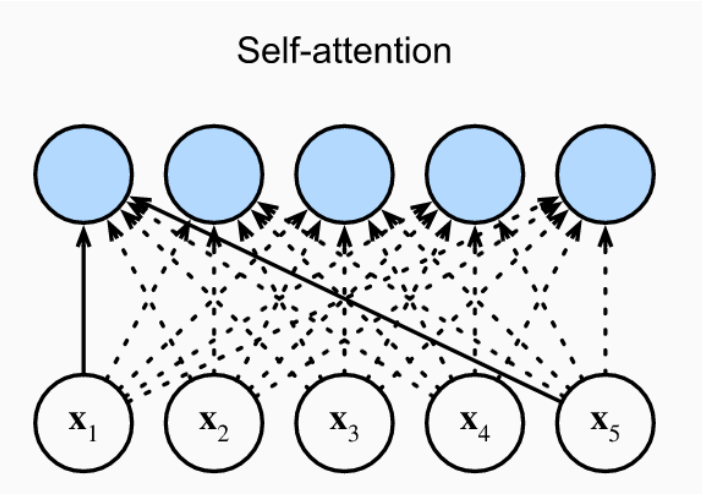


Transformers and Attention

- Advantages over RNNs:
  - Enable long range dependencies
  - Parallel inference

- Disadvantage:
  - Quadratic complexity in sequence length and hidden space dimensionality





from d2l.ai

Transformers vs RNNs

- Transformers have displaced RNNs in NLP

- Since RNNs are also used in RL, how can we leverage transformers?

Transformer in Partially Observable RL

- Replace RNN by Transformer in partially observable RL
- DTQN: Deep Transformer Q-Network (Esslinger et al., 2022)

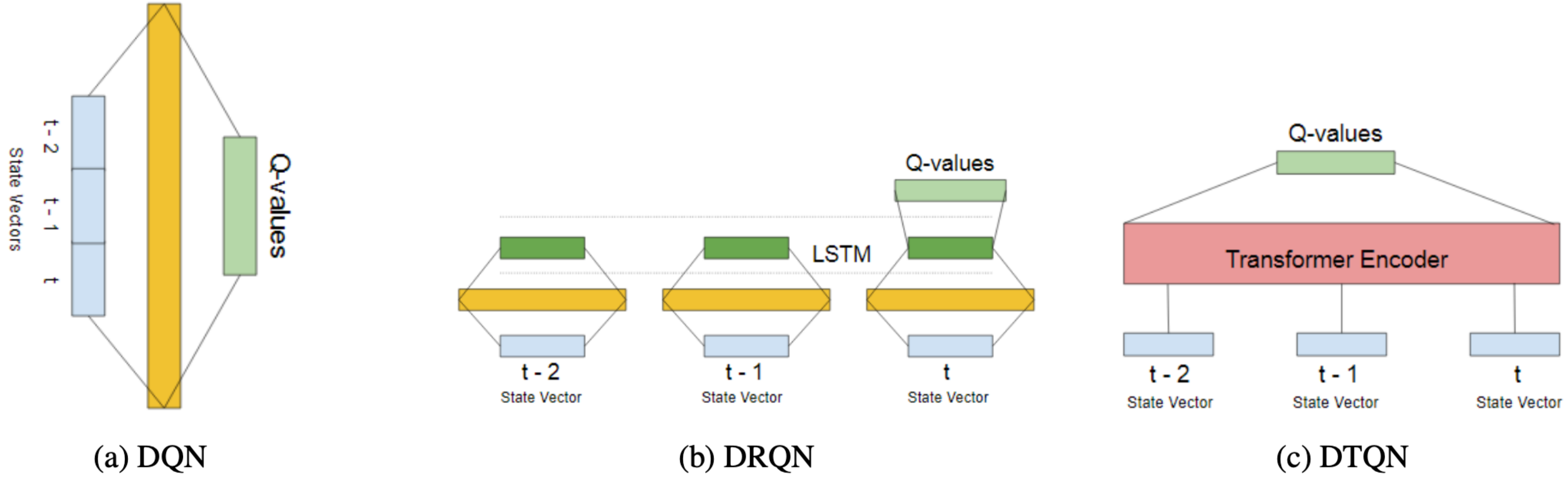


Fig. 2: Different representative architectures. (a) DQN, (b) DRQN, (c) DTQN.

DTQN Architecture

from Esslinger et al., 2022

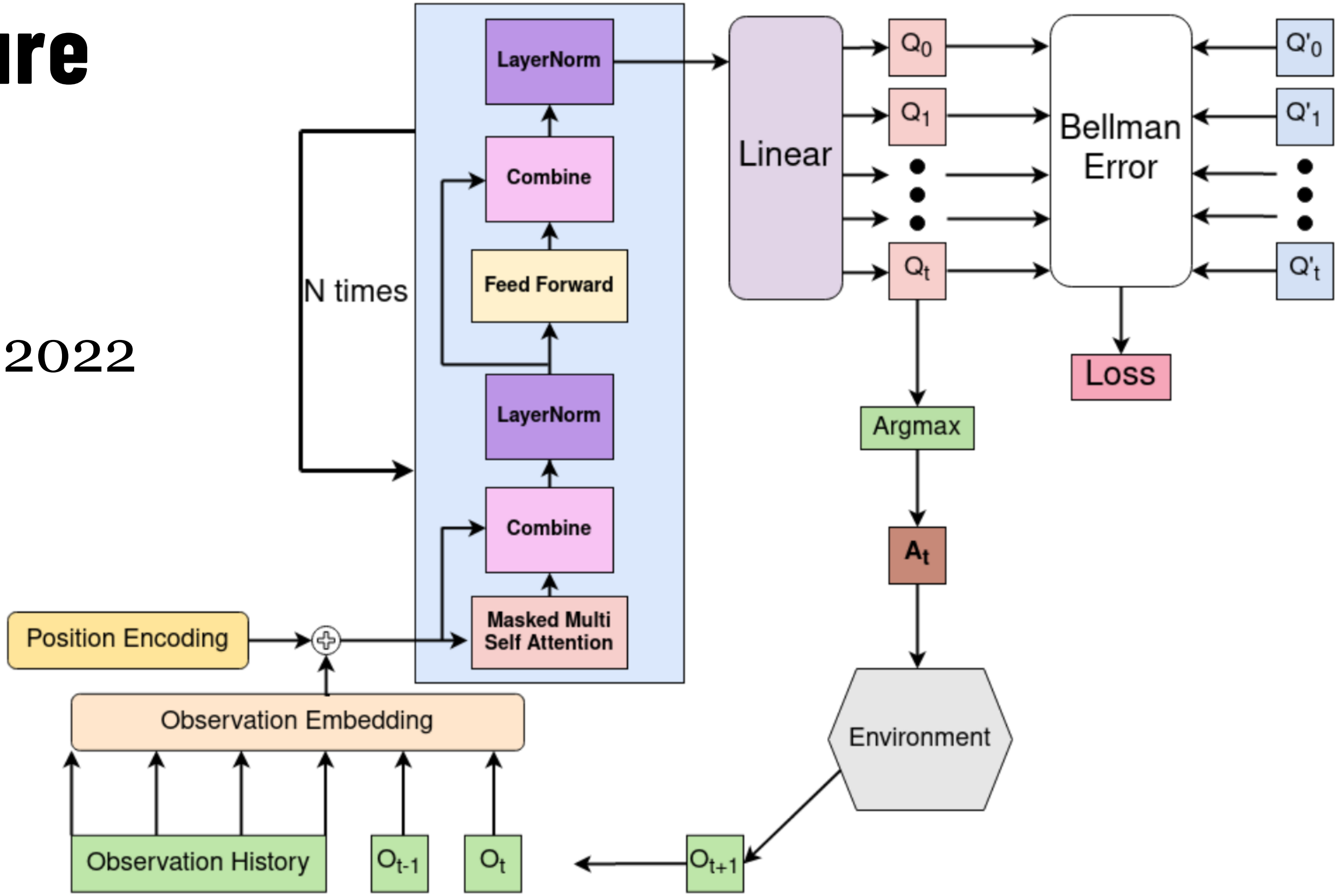
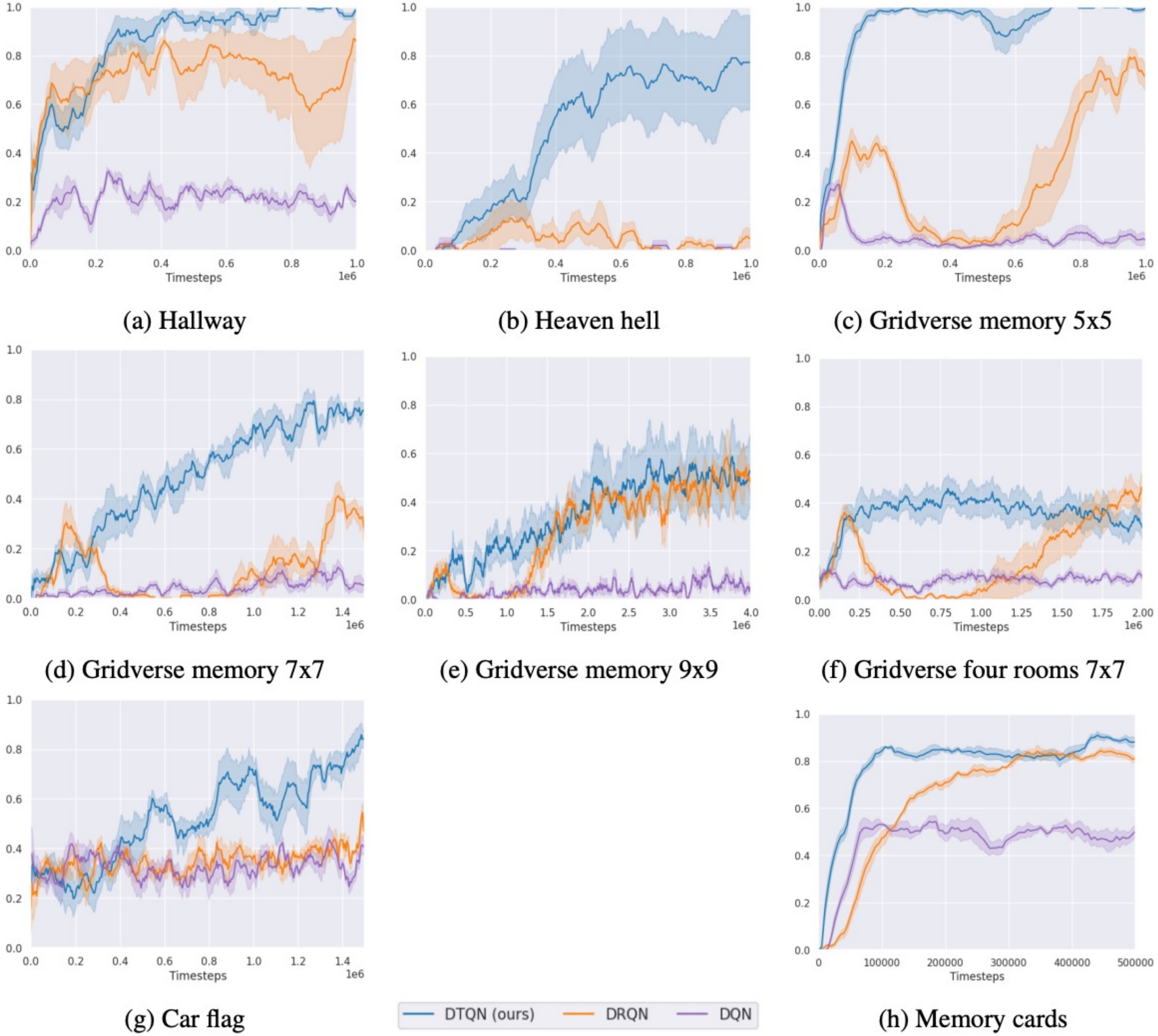


Figure 1: Architectural diagram of DTQN. Each observation in the history is embedded independently, and Q-values are generated for each observation sub-history. Only the last set of Q-values are used to select the next action, but the other Q-values can be utilized for training.

DTQN Results

from Esslinger et al., 2022

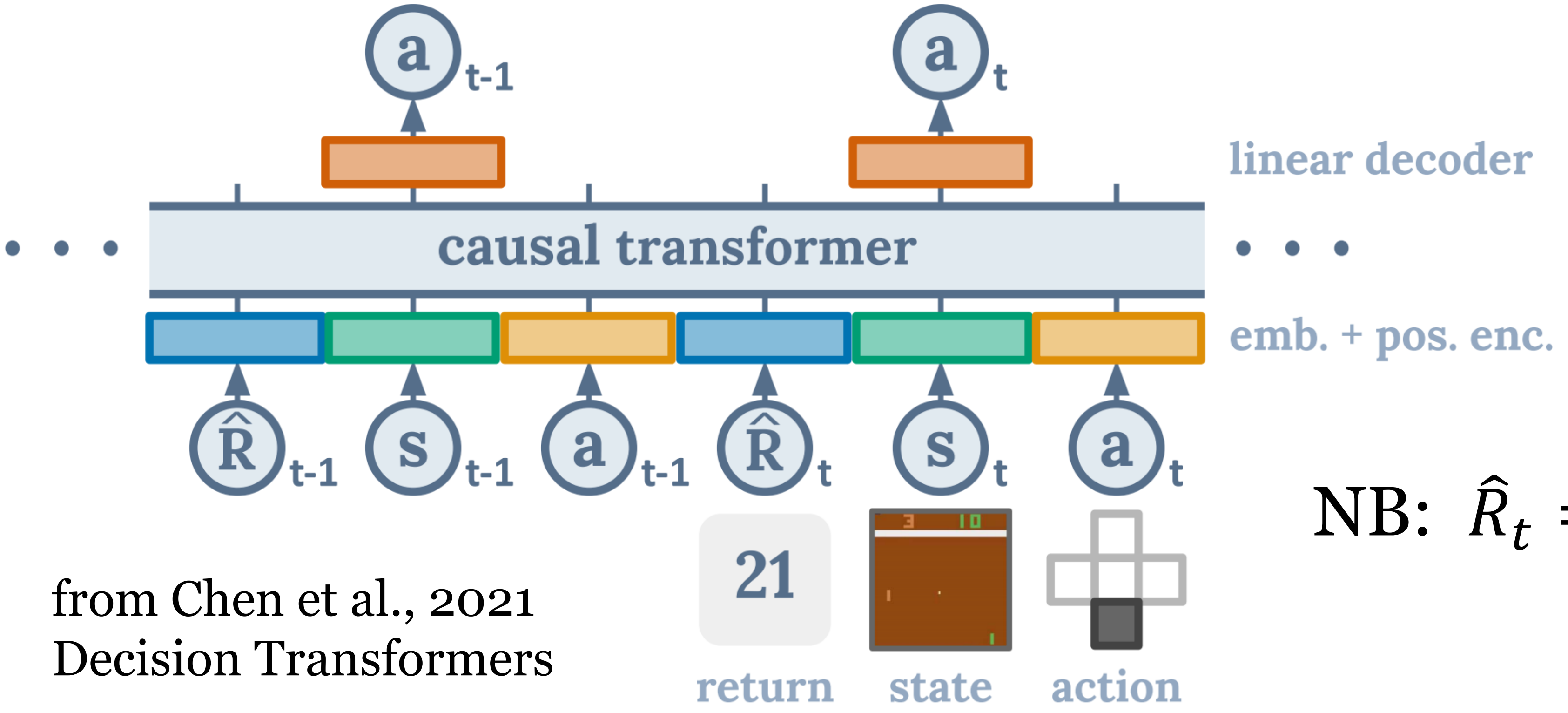


(a) Hallway (b) Heaven hell (c) Gridverse memory 5x5 (d) Gridverse memory 7x7 (e) Gridverse memory 9x9 (f) Gridverse four rooms 7x7 (g) Car flag (h) Memory cards

UNIVERSITY OF WATERLOO

New Paradigm: RL by Sequence Modeling

- Replace everything (i.e., actor and critic) in RL by a Transformer
- In other words: transformers are all you need!

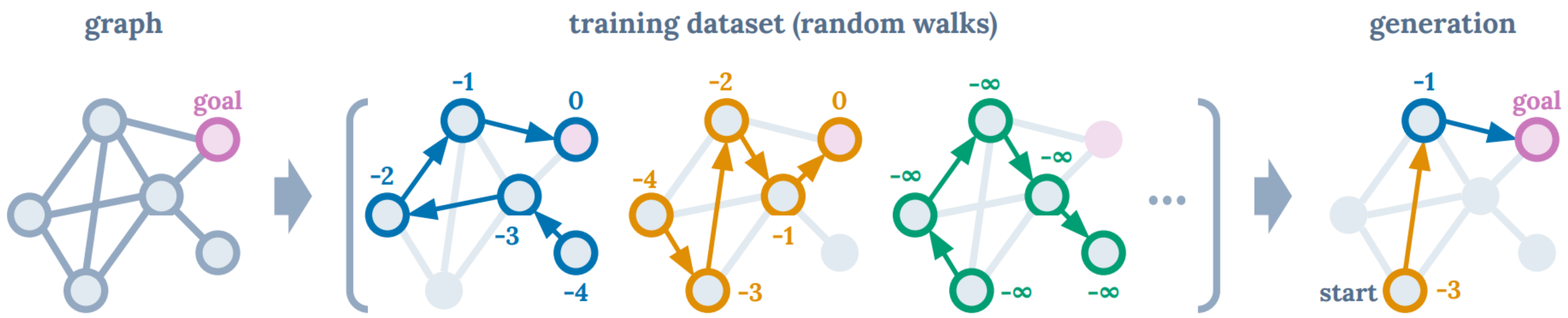


from Chen et al., 2021
Decision Transformers

NB: $\hat{R}_t = \sum_{t'=t} \gamma^{t'} r_{t'}$

Decision Transformers

- Offline RL
- Fixed dataset of trajectories (no exploration)
- Trajectories may include random walks and expert trajectories



Training (Offline RL)

- Given a history of $\langle \hat{R}_1, s_1, a_1, \hat{R}_2, s_2, a_2, \dots, \hat{R}_n, s_n \rangle$
  - Predict a_n
  - Minimize
    - Mean squared error for continuous actions
    - Cross-entropy for discrete actions

Policy execution (Online Execution)

- Select a desired total return $\hat{R}_1$
- Predict next action $\langle \hat{R}_1, s_1 \rangle \rightarrow a_1$ and execute it
- Receive reward r_1 and next state s_2
- Decrement total return $\hat{R}_2 = \hat{R}_1 - r_1$
- Predict next action $\langle \hat{R}_1, s_1, a_1, \hat{R}_2, s_2 \rangle \rightarrow a_2$ and execute it
- …

Results: Expected Rewards

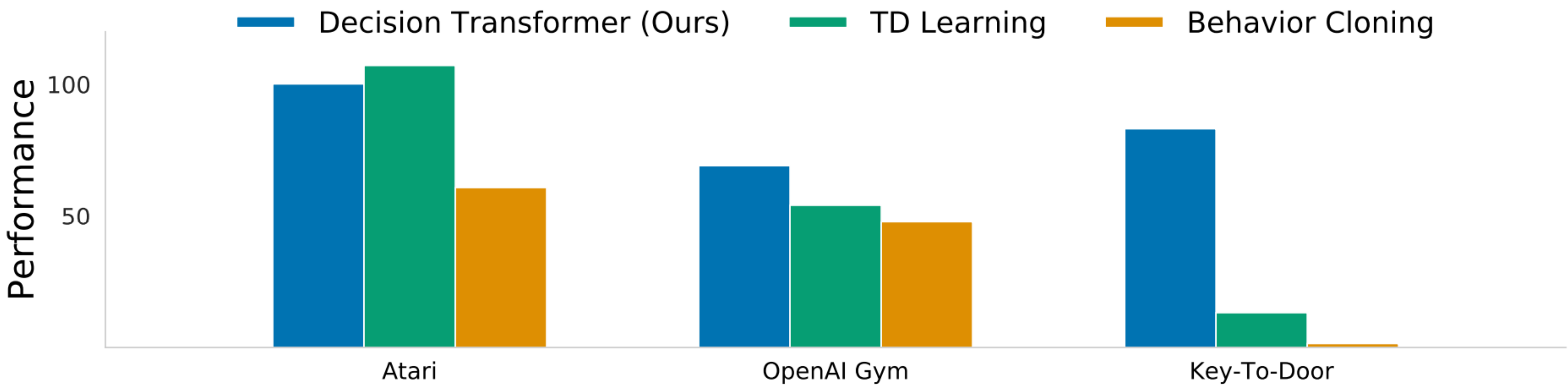


Figure 3: Results comparing Decision Transformer (ours) to TD learning (CQL) and behavior cloning across Atari, OpenAI Gym, and Minigrid. On a diverse set of tasks, Decision Transformer performs comparably or better than traditional approaches. Performance is measured by normalized episode return (see text for details).

Results: modeling the distribution of returns

- How well does Decision Transformer model the distribution of returns?

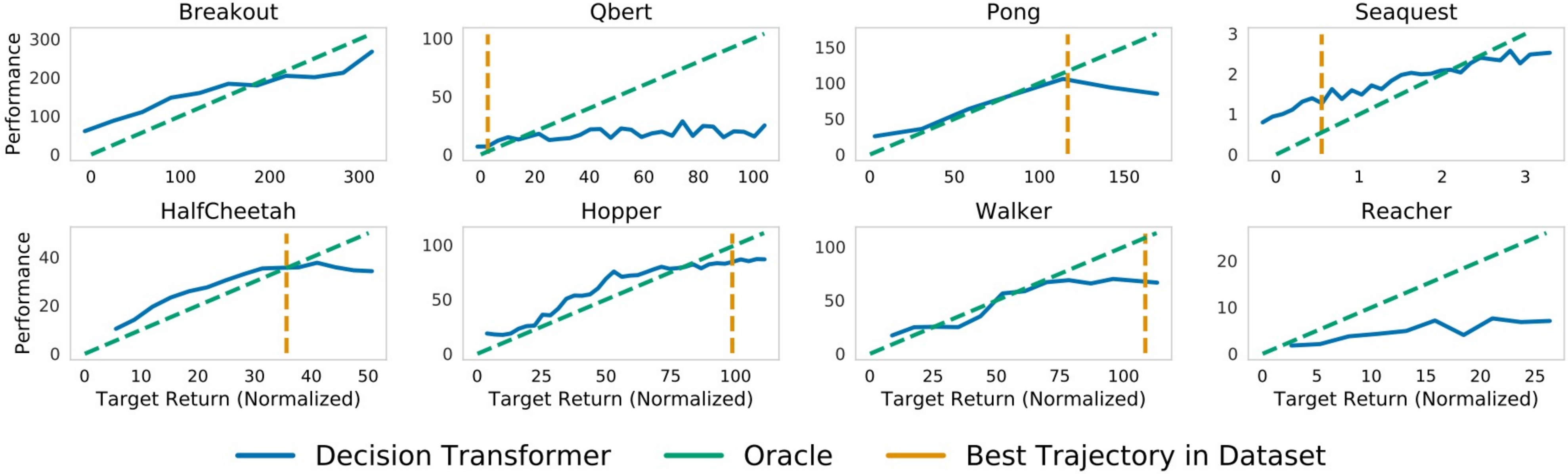


Figure 4: Sampled (evaluation) returns accumulated by Decision Transformer when conditioned on the specified target (desired) returns. **Top:** Atari. **Bottom:** D4RL medium-replay datasets.

UNIVERSITY OF WATERLOO

Results: impact of context length

- What is the benefit of using a longer context length?

Game	DT (Ours)	DT with no context ($K = 1$)
Breakout	$\mathbf{267.5 \pm 97.5}$	73.9 ± 10
Qbert	$\mathbf{25.1 \pm 18.1}$	13.7 ± 6.5
Pong	$\mathbf{106.1 \pm 8.1}$	2.5 ± 0.2
Seaquest	$\mathbf{2.4 \pm 0.7}$	0.5 ± 0.0

Table 5: Ablation on context length. Decision Transformer (DT) performs better when using a longer context length ($K = 50$ for Pong, $K = 30$ for others).

Results: sparse rewards

- How does Decision Transformer perform with sparse rewards?

Dataset	Environment	Delayed (Sparse) DT (Ours)	Delayed (Sparse) CQL	Agnostic BC	Agnostic %BC	Original (Dense) DT (Ours)	Original (Dense) CQL
Medium-Expert	Hopper	**107.3 ± 3.5**	9.0	59.9	102.6	107.6	111.0
Medium	Hopper	60.7 ± 4.5	5.2	63.9	**65.9**	67.6	58.0
Medium-Replay	Hopper	**78.5 ± 3.7**	2.0	27.6	70.6	82.7	48.6

Table 7: Results for D4RL datasets with delayed (sparse) reward. Decision Transformer (DT) and imitation learning are minimally affected by the removal of dense rewards, while CQL fails.

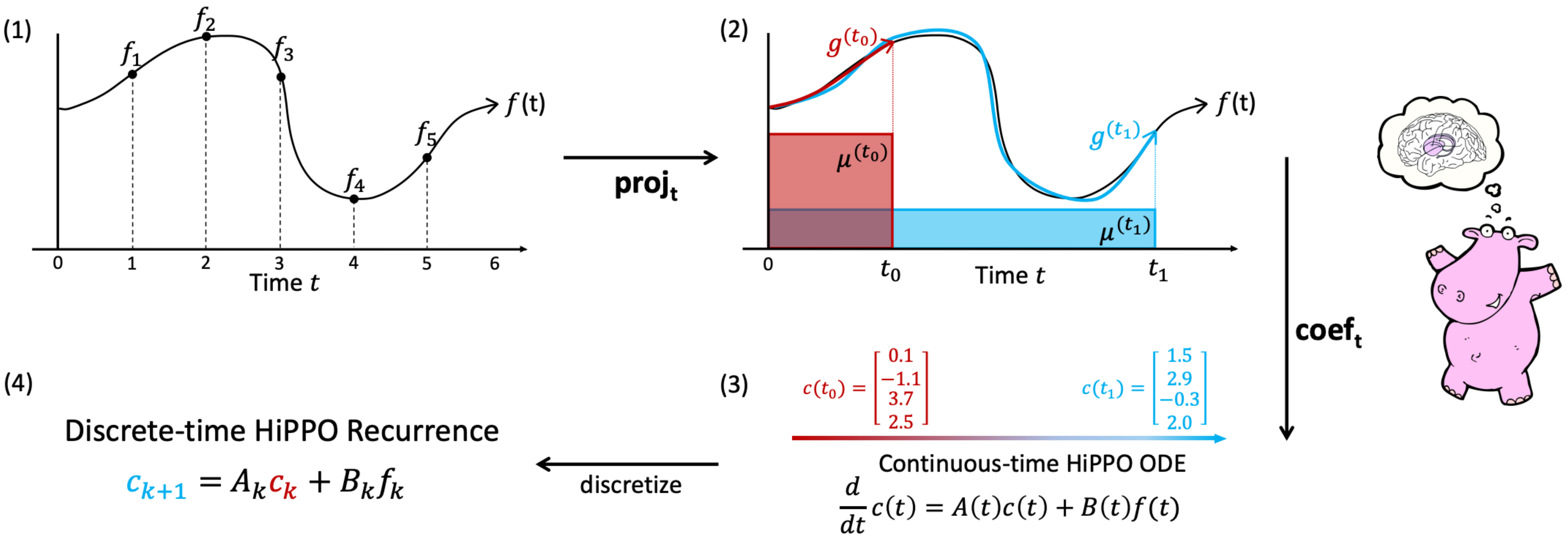
How can we handle long horizons?

- Structured State Space Sequence (S4) Model
  - Very recent approach (Gu, Goel & Re, ICLR 2022)

- Recent competitor to transformers
  - S4 achieved state of the art on Long Range Arena benchmark
  - Scales linearly with sequence length

Structured State Space Sequence (S4) Model

- HiPPO: high-order polynomial projection operators (Gu et al., 2020)



Measures (importance given to past history)

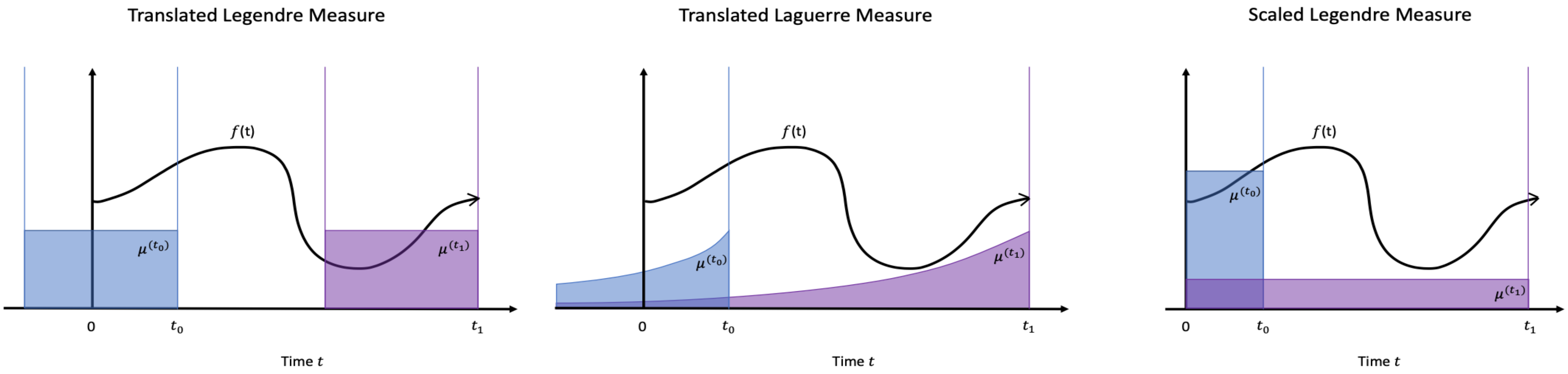
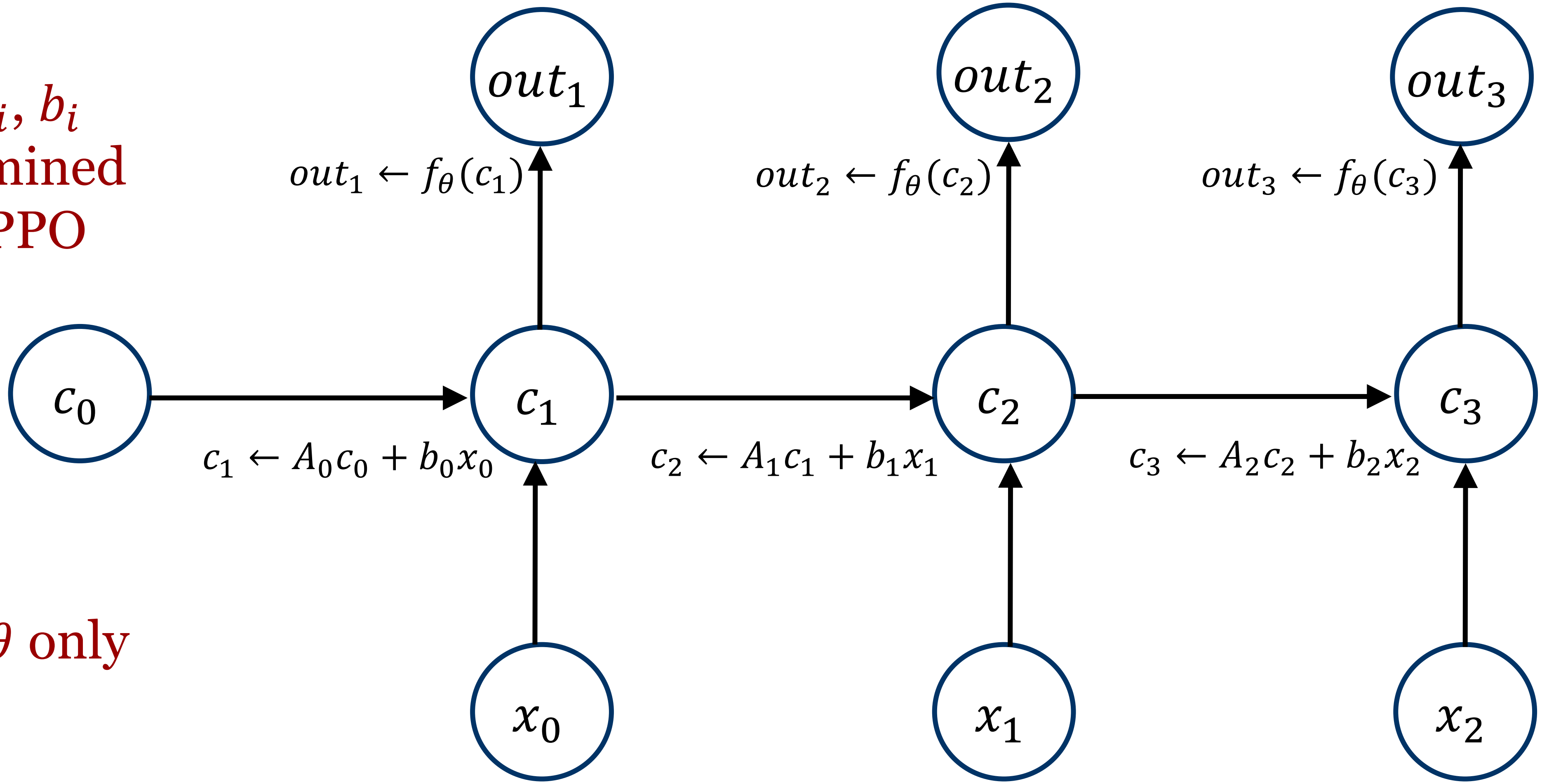


Figure 5: **Illustration of HiPPO measures.** At time t_0, the history of a function $f(x)_{x \le t_0}$ is summarized by polynomial approximation with respect to the measure $\mu^{(t_0)}$ (blue), and similarly for time t_1 (purple). (Left) The Translated Legendre measure (**LegT**) assigns weight in the window $[t - \theta, t]$. For small t, $\mu^{(t)}$ is supported on a region $x < 0$ where f is not defined. When t is large, the measure is not supported near 0, causing the projection of f to forget the beginning of the function. (Middle) The Translated Laguerre (**LagT**) measure decays the past exponentially. It does not forget, but also assigns weight on $x < 0$. (Right) The Scaled Legendre measure (**LegS**) weights the entire history $[0, t]$ uniformly.

RNN with HiPPO

NB: A_i, b_i determined by HiPPO



Train θ only

Computational Complexity

- S4 scales better than CNNs, RNNs and Transformers

Table 1: Complexity of various sequence models in terms of sequence length ($\boldsymbol{L}$), batch size ($\boldsymbol{B}$), and hidden dimension ($\boldsymbol{H}$); tildes denote log factors. Metrics are parameter count, training computation, training space requirement, training parallelizability, and inference computation (for 1 sample and time-step). For simplicity, the state size N of S4 is tied to H. Bold denotes model is theoretically best for that metric. Convolutions are efficient for training while recurrence is efficient for inference, while SSMs combine the strengths of both.

	Convolution[3]	Recurrence	Attention	S4
Parameters	LH	$\boldsymbol{H^2}$	$\boldsymbol{H^2}$	$\boldsymbol{H^2}$
Training	$\boldsymbol{\tilde{L}H(B+H)}$	BLH^2	$B(L^2H+LH^2)$	$\boldsymbol{BH(\tilde{H}+\tilde{L})+B\tilde{L}H}$
Space	$\boldsymbol{BLH}$	$\boldsymbol{BLH}$	$B(L^2+HL)$	$\boldsymbol{BLH}$
Parallel	**Yes**	No	**Yes**	**Yes**
Inference	LH^2	$\boldsymbol{H^2}$	L^2H+H^2L	$\boldsymbol{H^2}$

From Gu, Goel & Re (2022)

Results: Long Range Arena

Model	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Reformer	<u>37.27</u>	56.10	53.40	38.07	68.50	✗	50.56
BigBird	36.05	64.02	59.29	40.83	74.87	✗	54.17
Linear Trans.	16.13	<u>65.90</u>	53.09	42.34	75.30	✗	50.46
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.18
FNet	35.33	65.11	59.61	38.67	<u>77.80</u>	✗	54.42
Nyströmformer	37.15	65.52	<u>79.56</u>	41.58	70.94	✗	57.46
Luna-256	37.25	64.57	79.29	<u>47.38</u>	77.72	✗	<u>59.37</u>
S4	**59.60**	**86.82**	**90.90**	**88.65**	**94.20**	**96.35**	**86.09**

From Gu, Goel & Re (2022)

Results: Speech and Images

Table 5: (**SC10 classification**) Transformer, CTM, RNN, CNN, and SSM models. (*MFCC*) Standard pre-processed MFCC features (length 161). (*Raw*) Unprocessed signals (length 16000). (*0.5×*) Frequency change at test time. ✗ denotes not applicable or computationally infeasible on single GPU. *Please read Appendix D.5 before citing this table.*

	MFCC	RAW	0.5×
Transformer	90.75	✗	✗
Performer	80.85	30.77	30.68
ODE-RNN	65.9	✗	✗
NRDE	89.8	16.49	15.12
ExpRNN	82.13	11.6	10.8
LipschitzRNN	88.38	✗	✗
CKConv	**95.3**	71.66	<u>65.96</u>
WaveGAN-D	✗	<u>96.25</u>	✗
LSSL	93.58	✗	✗
S4	<u>93.96</u>	**98.32**	**96.30**

From Gu, Goel & Re (2022)

Table 6: (**Pixel-level 1-D image classification**) Comparison against reported test accuracies from prior works (Transformer, RNN, CNN, and SSM models). Extended results and citations in Appendix D.

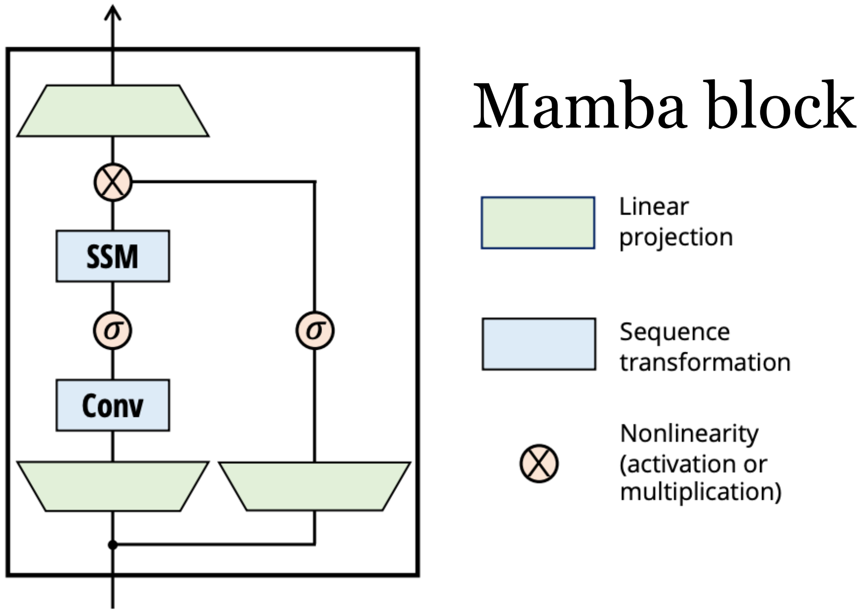
	sMNIST	pMNIST	sCIFAR
Transformer	98.9	97.9	62.2
LSTM	98.9	95.11	63.01
r-LSTM	98.4	95.2	72.2
UR-LSTM	99.28	96.96	71.00
UR-GRU	99.27	96.51	74.4
HiPPO-RNN	98.9	98.3	61.1
LMU-FFT	-	98.49	-
LipschitzRNN	99.4	96.3	64.2
TCN	99.0	97.2	-
TrellisNet	99.20	98.13	73.42
CKConv	99.32	98.54	63.74
LSSL	<u>99.53</u>	**98.76**	<u>84.65</u>
S4	**99.63**	<u>98.70</u>	**91.13**

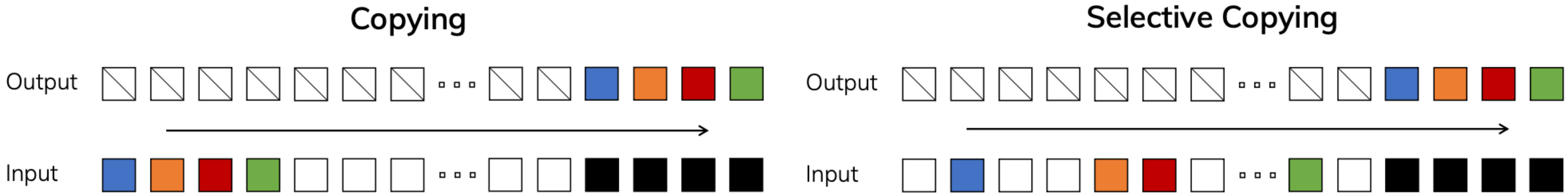
Mamba

- Mamba (Improved S4):
  - Gu & Dao (2023) Mamba: Linear-Time Sequence Modeling with Selective State Spaces, First Conference on Language Modeling.
  - Improved modeling: selective mechanism
  - Improved efficiency: hardware-aware algorithm
  - Simplified architecture: no multilayer perceptron block
- Mamba-2 (simplified Mamba)
  - Dao & Gu (2024) Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality
  - Simpler operators and larger state space (improved efficiency)

Selective Mechanism

- S4: fix operators
  - $c \leftarrow Ac + bx$
- Mamba: input dependent operators
  - $c \leftarrow A(x)c + b(x)x$

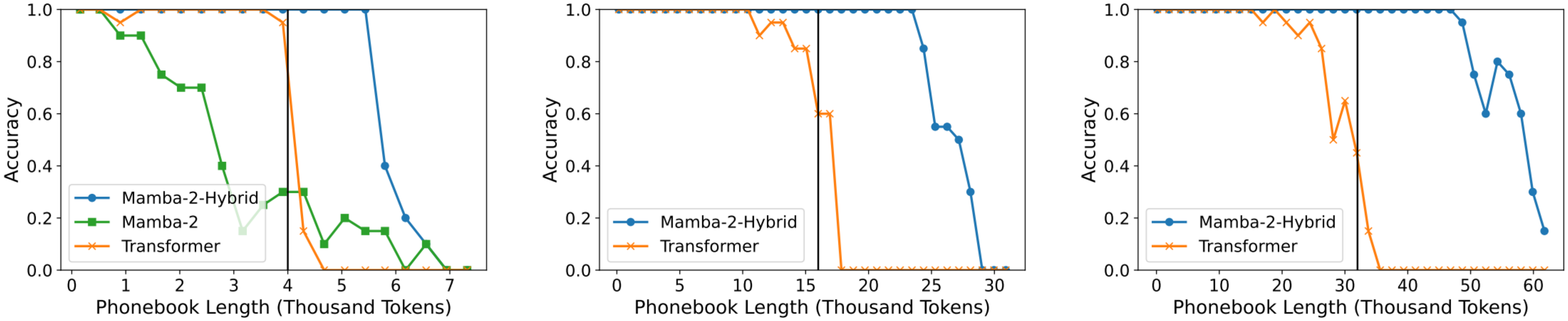




Language Modeling Results

- Wallefe et al. (2024)

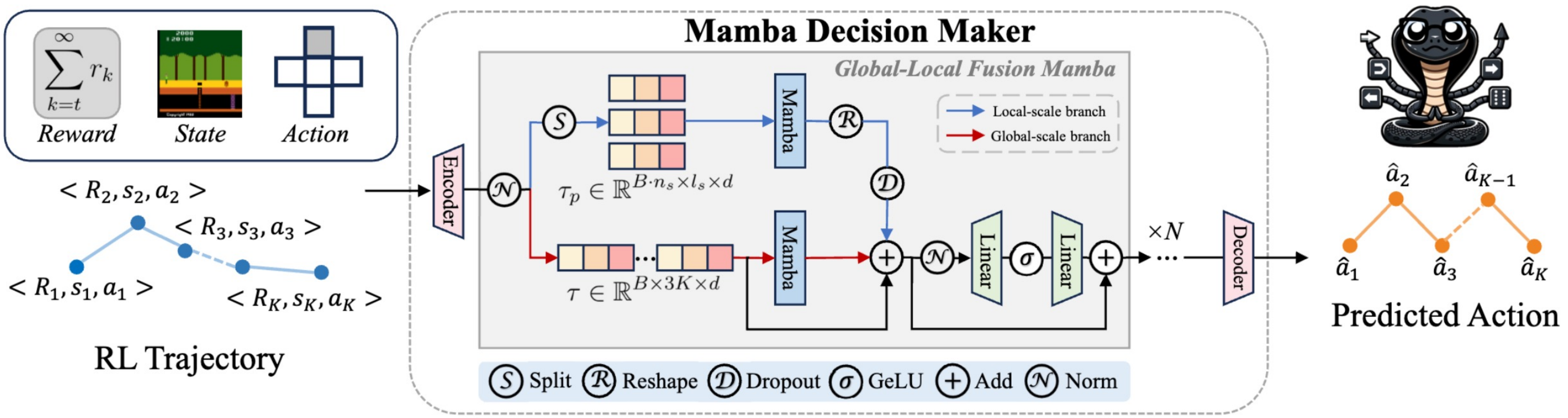
Model	WG	PIQA	HellaSwag	ARC-E	ARC-C	MMLU 0-Shot	MMLU 5-Shot	OpenBook	TruthFul	PubMed	RACE	NQ	SquadV2	Avg
Transformer	69.14	78.62	75.89	73.27	43.77	45.69	50.07	42.00	35.48	69.20	39.52	15.15	53.4	53.17
Mamba-2	**71.59**	**79.82**	**77.69**	75.93	**48.12**	47.25	48.7	**44.2**	35.66	**75.2**	37.7	17.17	51.9	54.69
Mamba-2-Hybrid	71.27	79.65	**77.68**	**77.23**	47.7	**51.46**	**53.60**	42.80	**38.72**	69.80	**39.71**	**17.34**	**58.67**	**55.82**



(a) 4K base models (b) 16K models (c) 32K models

MambaDM: Mamba as Decision Maker

- Cao et al. (2024) Mamba as Decision Maker: Exploring Multi-scale Sequence Modeling in Offline Reinforcement Learning, arxiv.
  - Replace transformer by Mamba in the Decision Transformer



UNIVERSITY OF WATERLOO

Offline RL Results

- Cao et al. (2024)

Atari games

Game	CQL	BC	DT	DC	DC^{hybrid}	DMamba†	**MambaDM**
Breakout	211.1	142.7	242.4 ±31.8	352.7 ±44.7	**416.0** ±105.4	239.2 ±26.4	**365.4** ±20.0
Qbert	104.2	20.3	28.8 ±10.3	**67.0** ±14.7	62.6 ±9.4	42.3 ±8.5	**74.4** ±8.4
Pong	111.9	76.9	105.6 ±2.9	106.5 ±2.0	**111.1** ±1.7	63.2 ±102.1	**110.8** ±2.3
Seaquest	1.7	2.2	**2.7** ±0.7	2.6 ±0.3	**2.7** ±0.04	2.2 ±0.03	**2.9** ±0.1
Asterix	4.6	4.7	5.2 ±1.2	**6.5**±1.0	6.3 ±1.8	5.5±0.3	**7.5**±1.4
Frostbite	9.4	16.1	25.6 ±2.1	27.8±3.7	**28.0**±1.8	25.3±1.5	**33.7**±4.4
Assault	73.2	62.1	52.1±36.2	73.8 ±20.3	**79.0**±13.1	67.2±6.9	**81.4**±3.1
Gopher	2.8	33.8	34.8 ±10.0	**52.5**±9.3	51.6±10.7	27.0±3.9	**54.4**±11.1

Mujoco (robotics)

Dataset	Env.	TD3+BC	IQL	CQL	RvS	DT	DS4	DMamba	**MambaDM**
M	halfcheetah	48.3	47.4	44.0	41.6	**42.6**	42.5	**42.8**	**42.8** ±0.1
	hopper	59.3	63.8	58.5	60.2	68.4	54.2	**83.5**	**85.7** ±7.8
	walker2d	83.7	79.9	72.5	71.7	75.5	**78.0**	**78.2**	**78.2**±0.6
M-R	halfcheetah	44.6	44.1	45.5	38.0	37.0	15.2	**39.6**	**39.1** ±0.1
	hopper	60.9	92.1	95.0	73.5	**85.6**	49.6	82.6	**86.1** ±2.5
	walker2d	81.8	73.7	77.2	60.6	**71.2**	69.0	70.9	**73.4** ±2.6
M-E	halfcheetah	90.7	86.7	91.6	92.2	88.8	**92.7**	**91.9**	86.5 ±1.2
	hopper	98.0	91.5	105.4	101.7	109.6	**110.8**	**111.1**	110.5 ±0.3
	walker2d	110.1	109.6	108.8	106.0	**109.3**	105.7	108.3	**108.8** ±0.1

UNIVERSITY OF WATERLOO