StARformer: Transformer with State-Action-Reward Representations for Visual Reinforcement Learning – Supplementary Material

Jinghuan Shang, Kumara Kahatapitiya, Xiang Li, and Michael S. Ryoo

Stony Brook University, NY 11794, USA
{jishang, kkahatapitiy, xiangli8, mryoo}@cs.stonybrook.edu

1 Appendix

1.1 Detailed Results

In the following Table 1, we give full evaluation results in exact episode returns. In More on the next page: in Fig. 1, we show how each environment changes when scaling from short to relative longer sequences. Table 2 gives per-task ablation results of embedding methods. Table 3 gives per-task ablation results of our Transformer connections.

Table 1. Evaluation of episodic returns (↑) in our proposed StARformer (StAR) and the baseline Decision-Transformer (DT) [2] in Atari and DMC. We also compare with non-Transformer offline-RL methods: CQL [5], QR-DQN [3], REM [1], and BEAR [4], in their applicable tasks (due to action space), and a behavior cloning baseline using ViT as the visual encoder (BC-ViT). The highest scores in each setting and environment are highlighted in **bold**. We use T-test to show the significance ($p < 0.05$) of improvement over the baseline. We also present the max. reward of the training datasets for reference.

Setting	Method	Atari							DMC				
		Assault	Boxing	Breakout	Pong	Pong(50)	Qbert	Seaquest	Ball_in_cup	Cheetah	Finger	Reacher	Walker
offline RL	DT	462 ± 139	78.3 ± 4.6	76.9 ± 17.1	12.8 ± 3.2	17.1 ± 1.8	3488 ± 631	**1131 ± 168**	207.7 ± 123.2	27.9 ± 44.5	312.4 ± 94.4	151.4 ± 29.9	232.5 ± 64.8
	StAR	**761 ± 127**	**81.2 ± 3.9**	**124.1 ± 19.8**	**16.4 ± 2.1**	**18.9 ± 0.7**	6968 ± 1698	781 ± 212	**648.4 ± 75.3**	**275.3 ± 47.9**	**401.1 ± 34.2**	**383.2 ± 59.6**	**343.1 ± 43.8**
	p-value	0.0005	0.1904	0.0005	0.0089	0.0461	0.0011	0.0054	8.4e-8	5.7e-10	0.0170	4.8e-8	0.0004
	CQL	432	56.2	55.8	13.5	-	**14012**	685	176.3	20.3	264.4	142.6	78
	QR-DQN	142	14.3	4.5	2.2	-	0.0	161	-	-	-	-	-
	REM	350	12.7	2.4	0.0	-	0.0	282	-	-	-	-	-
	BEAR	-	-	-	-	-	-	-	160.8	6.3	223.2	102.3	44
Imitation	DT	595 ± 89	72.0 ± 2.6	54.3 ± 1.2	7.7 ± 2.1	9.7 ± 4.2	2099 ± 1075	826 ± 118	319.5 ± 195.7	0.5 ± 0.3	285.2 ± 122.9	127.5 ± 53.5	230.7 ± 86.6
	StAR	**788 ± 146**	**76.2 ± 3.6**	**103.1 ± 21.3**	**15.6 ± 2.6**	**17.7 ± 2.4**	**5709 ± 1002**	**939 ± 97**	**607.7 ± 59.9**	**231.9 ± 46.2**	**400.1 ± 52.8**	**356.3 ± 76.7**	**329.9 ± 34.9**
	p-value	0.0014	0.03233	0.0004	0.0002	0.0029	2.8e-5	0.0430	0.0011	7.0e-8	0.0185	8.3e-7	0.0058
	BC-ViT	442	58.0	4.9	-13.7	-	554	275	125.1	1.2	74.7	107.0	97.6
Dataset		153	98	92	21	21	600	290	541	354	771	348	422

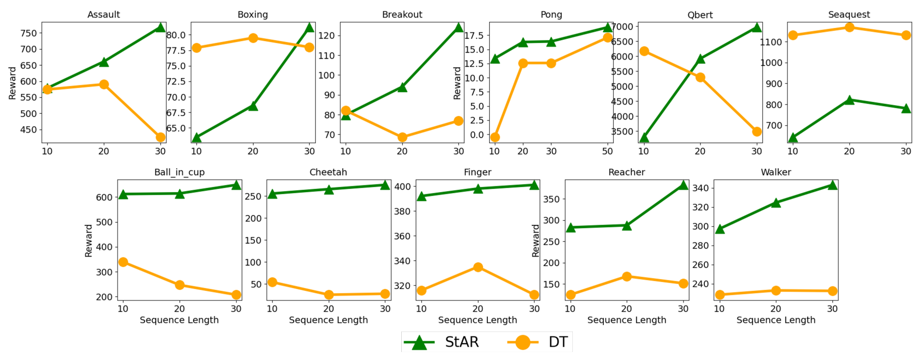

Fig. 1. Performance (offline RL) under trajectory length $T \in \{10, 20, 30\}$. We also include result of $T = 50$ in Pong. In most of the cases, StARformer (green) shows a better performance than DT (yellow) when increasing the trajectory length, and surpasses that of the baseline, validating that our method can effectively model long sequences.

Table 2. Ablation results on StAR-representations in Step Transformer and pure state representation h_t in Sequence Transformer (offline RL, Atari). StAR-rep stands for having StAR-representation in the model. Emb. of s_t is the method used for learning StAR-representation. Emb. of s_t is the pure state representation, which is only applicable to our method. We notice the combination of patch embeddings and CNN features works the best than other methods. Simply replace CNN features to ViT-like patch embeddings in DT will not improve the performance.

Method	StAR-rep.	Emb. of s_t	Emb. of h_t	Assault	Boxing	Breakout	Pong	Qbert	Seaquest
StAR (P+C)	✓	patch	✓, conv	**761 ± 127**	**81.2 ± 3.9**	**124.1 ± 19.8**	**16.4 ± 2.1**	**6968 ± 1698**	781 ± 212
P+P	✓	patch	✓, patch	548 ± 136	49.4 ± 4.7	38.0 ± 14.1	13.0 ± 5.7	1724 ± 472	565 ± 191
P+__	✓	patch	✗	583 ± 107	78.8 ± 2.4	38.7 ± 4.7	13.9 ± 3.3	4170 ± 942	920 ± 130
C+C	✓	conv	✓, conv	694 ± 31	52.5 ± 5.0	55.6 ± 12.6	11.0 ± 4.0	3505 ± 2132	844 ± 274
C+P	✓	conv	✓, patch	285 ± 35	65.0 ± 3.6	46.3 ± 17.9	14.3 ± 2.3	3529 ± 2545	568 ± 14
C+__	✓	conv	✗	509 ± 74	65.3 ± 4.5	52.7 ± 11.5	10.7 ± 3.5	1100 ± 554	853 ± 71
DT with ViT.	✗	-	patch	608 ± 85	74.3 ± 13.8	47.3 ± 7.4	2.7 ± 16.5	1135 ± 585	885 ± 93
DT [2]	✗	-	conv	504 ± 54	78.3 ± 4.6	70.7 ± 8.1	12.8 ± 3.2	3782 ± 695	**1007 ± 170**

Table 3. Ablation results on Transformer connectivity (offline RL). We observe that our original structure (StAR), which is a layer-wise manner, fits more than StAR Fusion and StAR Stack connections shown by higher rewards.

Method	Assault	Boxing	Breakout	Pong	Qbert	Seaquest
StAR	761 ± 127	**81.2 ± 3.9**	**124.1 ± 19.8**	**16.4 ± 2.1**	**6968 ± 1698**	**781 ± 212**
StAR Fusion	756 ± 116	69.4 ± 2.8	29.2 ± 15.9	8.7 ± 5.1	4053 ± 1239	608 ± 174
StAR Stack	**939 ± 157**	64.9 ± 4.9	30.9 ± 5.5	13.7 ± 2.5	575 ± 124	361 ± 261

1.2 More Visualizations

We visualize a segment of real trajectory in Breakout in Fig. 2, with annotated actions and highlighted labels for easier understanding. In general, we observe higher attention scores at the areas where the paddle and the ball locate in different attention heads. We also find relatively consistent semantic meanings in attention heads #1 and #2, with focus on pad and ball, respectively.

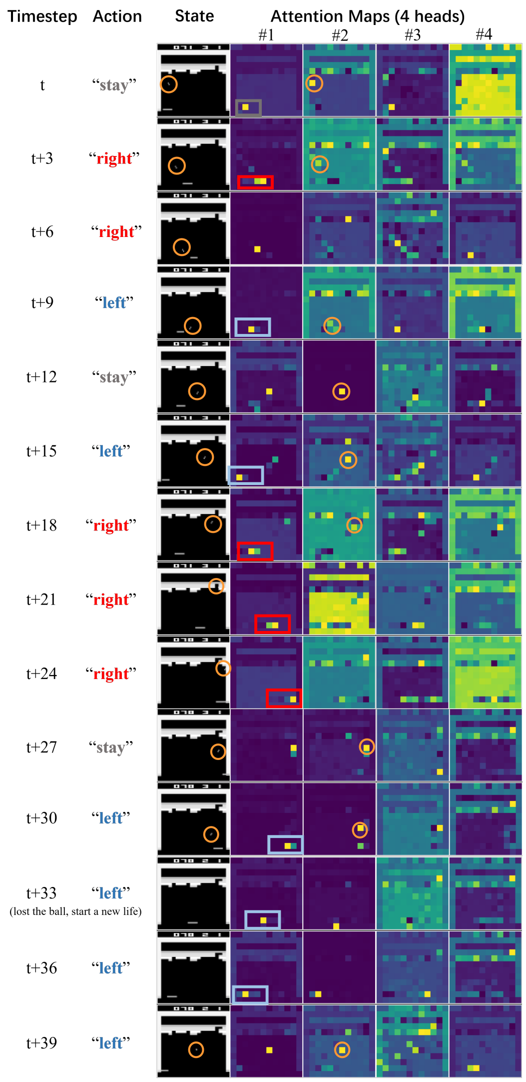

Fig. 2. More attention maps visualization in our Step Transformer.

1.3 Hyper-parameters

The complete list of hyper-parameters are given in Table 4 and 5, for Atari and DMC respectively. We keep most of the hyper-parameters similar to those provided by Decision-Transformer [2] for a fair comparison, including the number of Transformer layers, MSA heads and embedding dimensions in our Sequence Transformer, learning rate and optimizer configurations. Since DT does not conduct experiments in DMC environment with visual input, we tune DT and set learning rate to be 1×10^{-3} in DMC. For the frame skipping in DMC, we use the setting from [6], for both DT and our method.

Table 4. Our hyper-parameter settings in Atari. Underlined parameters are unique/different from DT [2]

Hyper-parameter	Value
Input sequence length (T)	10, 20, 30
Input image size	84×84, gray
Frame stack	4
Frame skip	2
Layers	6
MSA heads (Sequence Transformer)	8
Embedding dimension (Sequence Transformer)	192
Image patch size	7
MSA heads (Step Transformer)	4
Embedding dimension (Step Transformer)	64
Nonlinearity	GeLU for self-attention; ReLU for convolution
Dropout	0.1
Learning rate	6×10^{-4}
Adam betas	(0.9, 0.95)
Grad norm clip	1.0
Weight decay	0.1
Learning rate decay	Linear warmup and cosine decay (see [2])
Warmup tokens	512×20
Final tokens	$2 \times 500000 \times T$

Table 5. Our hyper-parameter settings in DMC. Underlined parameters are unique/different from DT [2]

Hyper-parameter	Value
Input sequence length (T)	10, 20, 30
Input image size	84×84, gray
Frame stack	3
Frame skip	4 for Cheetah and Reacher, 2 for Walker
Layers	6
MSA heads (Sequence Transformer)	8
Embedding dimension (Sequence Transformer)	192
<u>Image patch size</u>	12
<u>MSA heads (Step Transformer)</u>	4
<u>Embedding dimension (Step Transformer)</u>	64
Nonlinearity	GeLU for self-attention; ReLU for convolution
Dropout	0.1
Learning rate	1×10^{-3}
Adam betas	(0.9, 0.95)
Grad norm clip	1.0
Weight decay	0.1
Learning rate decay	Linear warmup and cosine decay (see [2])
Warmup tokens	512×20
Final tokens	$2 \times 100000 \times T$

References

1. Agarwal, R., Schuurmans, D., Norouzi, M.: An optimistic perspective on offline reinforcement learning. In: International Conference on Machine Learning. pp. 104–114. PMLR (2020)
2. Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., Mordatch, I.: Decision transformer: Reinforcement learning via sequence modeling. In: Proc. Adv. Neural Inform. Process. Syst. (Dec 2021)
3. Dabney, W., Rowland, M., Bellemare, M., Munos, R.: Distributional reinforcement learning with quantile regression. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
4. Kumar, A., Fu, J., Soh, M., Tucker, G., Levine, S.: Stabilizing off-policy q-learning via bootstrapping error reduction. Advances in Neural Information Processing Systems **32** (2019)
5. Kumar, A., Zhou, A., Tucker, G., Levine, S.: Conservative q-learning for offline reinforcement learning. Advances in Neural Information Processing Systems **33**, 1179–1191 (2020)
6. Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., Fergus, R.: Improving sample efficiency in model-free reinforcement learning from images. In: Proc. AAAI Conf. Artif. Intell. pp. 10674–10681 (May 2021)