Future-conditioned Unsupervised Pretraining for Decision Transformer

Zhihui Xie, Zichuan Lin, Deheng Ye, Qiang Fu, Wei Yang, Shuai Li

The Power of Pretraining

Does sequential decision making enjoy this formula?

+

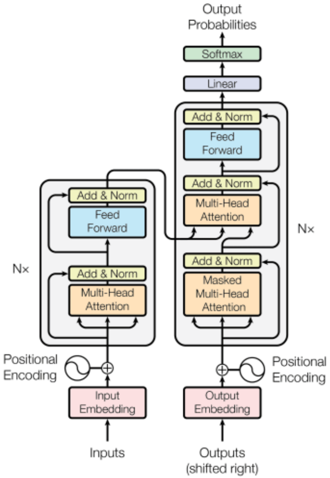
Next-token-prediction

The model is given a sequence of words with the goal of predicting the next word.

Example:
Hannah is a ___

Hannah is a *sister*
Hannah is a *friend*
Hannah is a *marketer*
Hannah is a *comedian*

+



≈

Ingredient 1:
Internet-scale knowledge

Ingredient 2:
Open-ended objectives

Ingredient 3:
Nets w/ weak inductive biases

"Sparks of AGI"[1]

[1]Sparks of Artificial General Intelligence: Early experiments with GPT-4 (Bubeck et al., 2023)

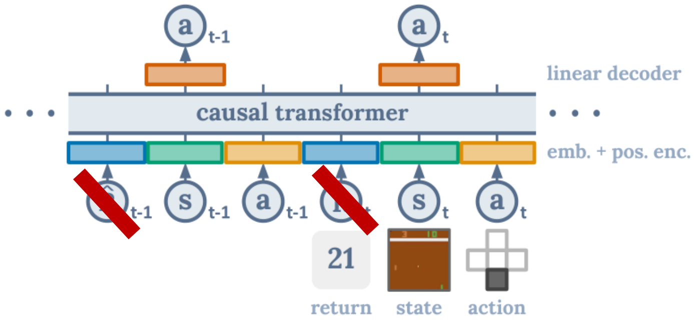
TL;DR

Unsupervised pretraining = **generative modeling**

Supervised finetuning = **controllable generation**

From Supervised to Unsupervised Pretraining

- Offline RL as sequence modeling[1]
  - Learning a policy $\pi_\theta(\cdot \mid \hat{\tau}_{1:t-1}, s_t, \hat{R}_t)$
  - Return-conditioned supervised learning (RCSL)
- Return-conditioned methods are good, but…
  - It cannot handle unsupervised pretraining
  - Scalar reward values can lead to inconsistent policies[2]
- This work: Unsupervised pretraining on diverse, reward-free data
  - Learning a **future-conditioned** policy $\pi_\theta(\cdot \mid \tau_{1:t-1}, s_t, \boldsymbol{z})$
  - z encodes the future trajectory, without rewards!



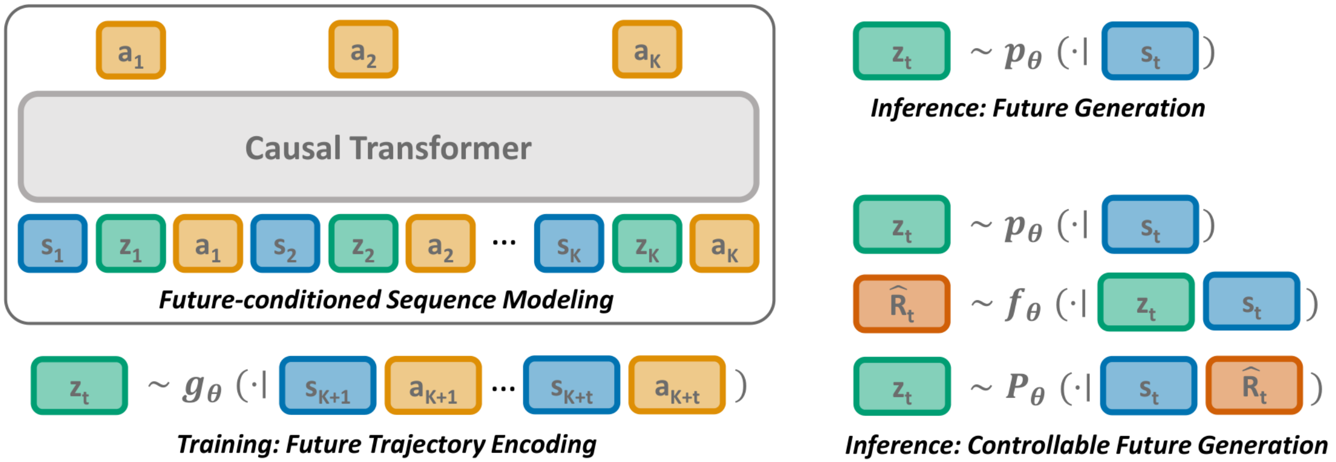
$\hat{\tau}$: Reward-labeled trajectory
τ: Reward-free trajectory

$(s_t, a_t) \rightarrow \hat{R}_t$

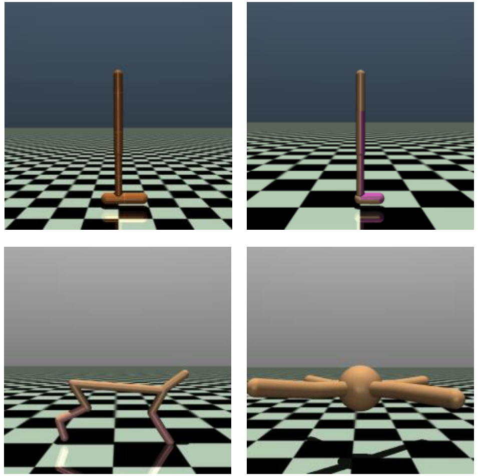
$(s_t, a_t) \rightarrow \boldsymbol{\tau_{future}} = (\boldsymbol{s_{t+1}}, \boldsymbol{a_{t+1}}, \ldots, \boldsymbol{s_T}, \boldsymbol{a_T}) \rightarrow \hat{R}_t$

[1]Decision Transformer: Reinforcement Learning via Sequence Modeling (Chen et al., 2021)
[2]Dichotomy of Control: Separating What You Can Control from What You Cannot (Yang et al., 2022)

Our Proposed Approach: PDT



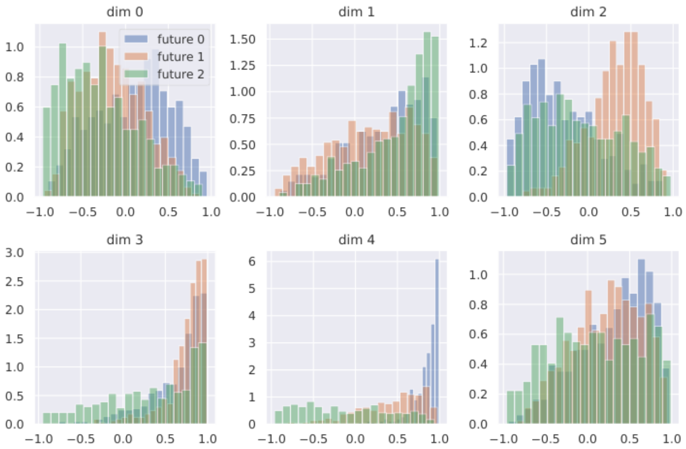
Results on D4RL Datasets



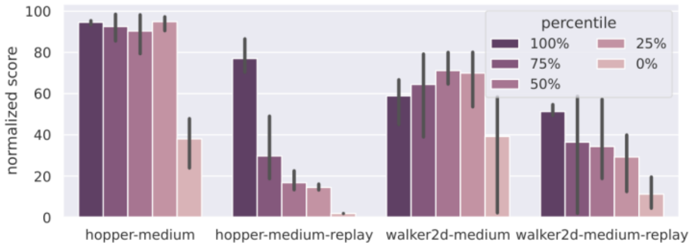
dataset	Mean	Min	Max	SAC	ACL	PDT-0	PDT	δ_{PDT}	ODT-0	ODT	δ_{ODT}
hopper-m	44.32	10.33	99.63	24.22 ± 10.55	57.66 ± 6.23	53.74	**95.26** ± 1.77	**41.52**	66.01	87.22 ± 6.85	21.22
hopper-m-r	14.98	0.58	98.73		51.68 ± 48.74	28.56	**84.96** ± 5.49	**56.40**	74.36	75.31 ± 6.22	0.95
walker2d-m	62.09	-0.18	92.04	35.26 ± 23.51	60.21 ± 27.08	73.70	**75.24** ± 4.60	**1.53**	72.80	72.62 ± 5.51	-0.18
walker2d-m-r	14.84	-1.13	89.97		**87.54** ± 7.31	15.64	58.58 ± 14.78	**42.94**	73.27	70.54 ± 2.89	-2.73
halfcheetah-m	40.68	-0.24	45.02	**57.05** ± 3.89	46.59 ± 2.71	42.86	37.93 ± 1.82	**-4.93**	42.69	35.07 ± 10.40	-7.62
halfcheetah-m-r	27.17	-2.89	42.41		50.56 ± 3.74	24.83	29.70 ± 4.97	**4.88**	40.95	35.60 ± 1.68	-5.35
ant-m	80.30	-4.85	107.31	33.30 ± 12.10	28.44 ± 10.78	**93.86**	89.10 ± 6.49	**-4.77**	93.08	73.80 ± 16.77	-19.28
ant-m-r	30.95	-8.87	96.56		9.53 ± 1.80	53.78	48.18 ± 9.59	**-5.60**	**90.37**	60.48 ± 6.23	-29.89
sum	315.33	-7.25	671.67		392.22	386.96	518.93	**123.94**	**553.53**	510.65	-42.87

- PDT can reuse pretrained behaviors for fast task adaptation
- PDT performs on par with its supervised pretraining counterpart

Analysis



Future conditioning: Different futures lead to different behaviors



Controllable generation: Binding rewards to futures

task	ODT	PDT
halfcheetah-forward-jump	**87.27** ± 14.41	83.80 ± 2.28
halfcheetah-jump	-31.00 ± 49.08	**70.39** ± 16.56
walker2d-forward-jump	29.36 ± 4.55	**45.31** ± 36.81
walker2d-jump	15.81 ± 14.75	**68.70** ± 2.90
sum	101.45	**268.21**

Generalization performance

Takeaways

- RCSL can be easily retrofitted for unsupervised pretraining
  - Target returns → target future trajectories
  - Downstream finetuning → controllable generation
- Experimental results show that our proposed PDT
  - can extract diverse behaviors from unlabeled offline data
  - can selectively high-return behaviors through online finetuning
- Open problems…
  - Better generative modeling?
  - Better strategy to fuse future information?