Dual RL: Unification and New Methods for Reinforcement and Imitation Learning

Harshit Sikchi[1], Qinqing Zheng[2], Amy Zhang[1,2], Scott Niekum[3]
[1]The University of Texas at Austin
[2]FAIR, Meta AI
[3]University of Massachusetts Amherst
ICLR 2024 (spotlight)

2025.10.23.
김동민

https://arxiv.org/abs/2302.08560
https://hari-sikchi.github.io/dual-rl/

Motivation

- on-policy policy gradient는 학습하고 있는 policy를 이용해서 rollout을 하여 활용하므로 데이터만 얻을 수 있다면 안정적인 학습이 가능함
- off-policy, offline-RL은 replay buffer 등의 기존 데이터를 이용해서 학습을 수행
  - SAC, TD3 등은 분포 불일치(distribution mismatch) 로 인해 value function 과대추정과 학습 불안정을 겪음

- offline 데이터를 활용하여 on-policy policy gradient처럼 학습할 수 있을까?
  - Dual RL은 state-action visitation distribution 기반 regularized RL 문제를 convex optimization 으로 표현하고, 이를 Lagrangian dual 로 변환하여 unconstrained 형태로 최적화함
  - 이 dual 문제는 policy gradient 를 off-policy 데이터로 계산하지만 on-policy gradient와 동일한 방향을 가지도록 수정되기 때문에 수렴성 및 안정성을 동시에 확보할 수 있음

Preliminaries

- state-action visitation distribution

$$d^{\pi}(s,a) = (1-\gamma)\pi(a|s)\sum_{t=0}^{\infty}\gamma^{t}P(s_t = s|\pi)$$

- Bellman operators
  - Q-backup

$$\mathcal{T}_r^{\pi}Q(s,a) = r(s,a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a),a'\sim\pi(\cdot|s')}[Q(s',a')$$

  - V-backup

$$\mathcal{T}_r V(s,a) = r(s,a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}[V(s')]$$

- f-divergence

$$D_f(d^{\pi}(s,a)\,||\,d^{O}(s,a))$$

- convex conjugate of f^{*} of function f

$$f^{*}(y) = \sup_{x\in\mathbb{R}}[< x\cdot y > -f(x)]$$

  - convex conjugate with positivity constraint

$$f_p^{*}(y) = \sup_{x\in\mathbb{R}}[< x\cdot y > -f(x)] \quad s.t. \quad x > 0$$

Overview

Regularized RL
- Convex Primal
- f-divergence regularization
- linear Bellman constraints

Lagrangian duality

Dual RL (unconstrained dual)
- optimize over {Q} or {V}
- uses f* (convex conjugate)

On-policy policy gradient computed from off-policy data (stability + convergence)

Dual-Q (2-player game)

$$\max_\pi \min_Q (1-\gamma)\mathbb{E}_{s\sim d_0, a\sim\pi(s)}[Q(s,a)] + \alpha\mathbb{E}_{(s,a)\sim d^O}\left[f^*\left(\left[\mathcal{T}_r^\pi Q(s,a) - Q(s,a)\right]/\alpha\right)\right]$$

Dual-V (1-player optimization)

$$\min_V (1-\gamma)\mathbb{E}_{s\sim d_0}[V(s)] + \alpha\mathbb{E}_{(s,a)\sim d^O}\left[f_p^*\left(\left[\mathcal{T}V(s,a) - V(s)\right]/\alpha\right)\right]$$

Special case: IL
- IL as RL with reward=0
- $d^O = d^E$

Special case: ReCOIL (proposed)
- $d^S_{\text{mix}} := \beta d(s,a) + (1-\beta)d^S(s,a)$
- $d^{E,S}_{\text{mix}} := \beta d^E(s,a) + (1-\beta)d^S(s,a)$

Special case: RL
Pessimistic value learning
CQL, ATAC

Special case: RL
Implicit-way
XQL, f-DVL(proposed)

Reinforcment Learning as a Convex Program (1)

- Reinforcement learning problem with regularized optimization objective is a convex optimization

$$\max_{\pi} J(\pi) = \mathbb{E}_{d^{\pi}(s,a)}[r(s,a)] - \alpha D_f(d^{\pi}(s,a) \,||\, d^{O}(s,a))$$

  - rewriting as a convex problem that searches for a visitation distribution that satisfies the Bellman-flow constraints

$$\max_{\pi} J(\pi) = \max_{\pi} \Big[\max_{d} \mathbb{E}_{d(s,a)}[r(s,a)] - \alpha D_f(d(s,a) \,||\, d^{O}(s,a))$$
$$\text{s.t}\, d(s,a) = (1-\gamma) d_0(s).\pi(a|s) + \gamma \textstyle\sum_{s',a'} d(s',a') p(s|s',a') \pi(a|s),\ \forall s \in \mathcal{S}, a \in \mathcal{A}$$

  - constraint is called Bellman Flow constraint
  - it is required to make visitation distribution d to be valid in RL framework
  - the above problem is termed as **primal-Q**

- Lagrangian Dual without Constraints

$$\max_{\pi} \min_{Q} (1-\gamma)\mathbb{E}_{s\sim d_0, a\sim\pi(s)}[Q(s,a)] + \alpha \mathbb{E}_{(s,a)\sim d^{O}}[f^{*}([\mathcal{T}_r^{\pi} Q(s,a) - Q(s,a)]/\alpha)]$$

  - Q(s,a) is dual variable
  - f^{*} is the convex conjugate of f
  - exact same solution with primal-Q can be obtained because strong duality holds
  - GAN-like adversarial optimization should be performed, and it makes the problem be difficult
  - what if play around with V instead of Q?

Reinforcment Learning as a Convex Program (2)

- what if play around with V instead of Q?

$$\text{s.t } d(s,a) = (1-\gamma)d_0(s).\pi(a|s) + \gamma \sum_{s',a'} d(s',a')p(s|s',a')\pi(a|s), \ \forall s \in \mathcal{S}, a \in \mathcal{A}$$

  - constraints of primal-Q are overconstrained: the constraints already determine the unique solution d^π
  - rendering the inner maximization w.r.t d unnecessary

- **Primal-V**

$$\max_{d \geqslant 0} \mathbb{E}_{d(s,a)}[r(s,a)] - \alpha D_f(d(s,a) \mid\mid d^O(s,a))$$
$$\text{s.t } \sum_{a \in \mathcal{A}} d(s,a) = (1-\gamma)d_0(s) + \gamma \sum_{(s',a') \in \mathcal{S} \times \mathcal{A}} d(s',a')p(s|s',a'), \ \forall s \in \mathcal{S}$$

- **Dual-V** (Lagrangian dual of primal-V)

$$\min_V (1-\gamma)\mathbb{E}_{s \sim d_0}[V(s)] + \alpha \mathbb{E}_{(s,a) \sim d^O}\left[f_p^*\left(\left[\mathcal{T}V(s,a) - V(s)\right]/\alpha\right)\right]$$

  - single-player non-adversarial optimization

Imitation Learning Consideration (1)

- Consider the following **primal-Q** form of f-divergence between the mixture distributions

$$\max_{d(s,a)} -D_f(d^S_{\text{mix}}(s,a) \,||\, d^{E,S}_{\text{mix}}(s,a))$$

$$\text{s.t}\ \forall s \in \mathcal{S}, a \in \mathcal{A},\ \ d(s,a) = (1-\gamma)d_0(s)\pi(a|s) + \gamma \textstyle\sum_{(s',a') \in \mathcal{S}\times\mathcal{A}} d(s',a')p(s|s',a')\pi(a|s)$$

$$d^S_{\text{mix}} := \beta d(s,a) + (1-\beta)d^S(s,a)$$

$$d^{E,S}_{\text{mix}} := \beta d^E(s,a) + (1-\beta)d^S(s,a)$$

  - This is a valid imitation learning formulation
  - since the global maximum of the objective is attained at $d = d^E$

  - The above primal formulation deters offline learning, as it requires sampling from d to estimate the f-divergence
  - Thus consider its dual formulation that allows us to derive an off-policy objective that only requires samples from the offline data
  - This formulation is termed ReCOIL (**Re**laxed **C**overage for **O**ff-policy **I**mitation **L**earning)

$$\max_\pi \min_Q \beta(1-\gamma)\mathbb{E}_{d_0,\pi}[Q(s,a)] + \mathbb{E}_{s,a\sim d^{E,S}_{mix}}[f^*(\mathcal{T}^\pi_0 Q(s,a) - Q(s,a))] - (1-\beta)\mathbb{E}_{s,a\sim d^S}[\mathcal{T}^\pi_0 Q(s,a) - Q(s,a)]$$

- Imitation learning, or occupancy matching is a direct consequence of the regularized RL problem
  - when the reward is set to be 0
  - and the regularization distribution and d^0 are set to be the expert visitation distribution d^E

Imitation Learning Consideration (2)

- ReCOIL
  - without discriminator $r^{\text{imit}}(s,a) = -\log \frac{d^S(s,a)}{d^E(s,a)}$
    - estimating pseudo reward could be ill-defined in state-action space with zero/less expert support
  - without coverage assumption
    - suboptimal data visitation covers the expert visitation $d^S > 0$ wherever $d^E > 0$
  - ReCOIL is a Bellman consistent Energy-based Model

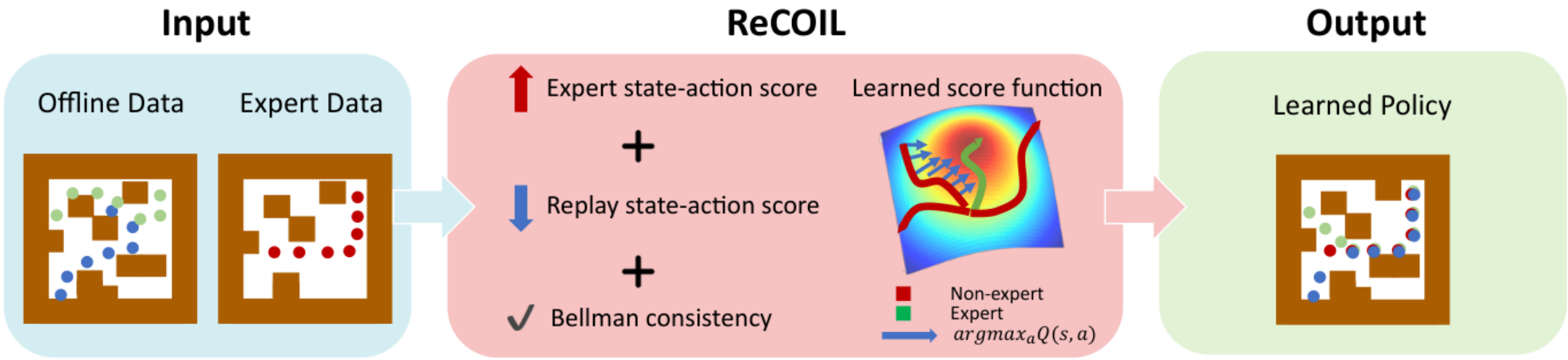


Figure 6: Recipe for ReCOIL: Learn a Bellman consistent EBM - A model which increases the score of expert transitions, and decreases the score of replay transitions while maintaining Bellman consistency throughout.

Implicit maximizers for offline RL

- Dual-V Learning (f-DVL): A new class of offline RL methods
- The dual-V objective gives a new class of methods parameterized by the choice of function f for offline RL

$$\min_V (1-\lambda)\mathbb{E}_{s\sim d^O}[V(s)] + \lambda\mathbb{E}_{(s,a)\sim d^O}\left[\bar{f}_p^*\left(\bar{Q}(s,a) - V(s)\right)\right]$$

- Extreme Q-learning (XQL), a implicit maximization based offline RL method, was derived using the hypothesis that Bellman errors are Gumbel distributed. In face, XQL can simply be seen as a special case of f-DVL when f is chosen to be the reverse-KL divergence. The choice of setting f corresponding to reverse KL divergence leads to a exponential conjugate which makes learning unstable due to exploding gradients. Choosing divergences with low-order f_p^* can lead to stable off-policy algorithms. Our experiments rely on Total-Variation and Chi-square divergences to obtain across the board improvements

Experiments

- Benchmark: D4RL
- Imitation Learning
  - 1) How does ReCOIL perform and compare with previous offline IL methods?
  - 2) Can ReCOIL accurately estimate the policy visitation distribution $d\pi$ and the reward function/intent of the expert?
- Reinforcement Learning
  - 3) How does f-DVL perform and compare with previous offline RL methods?
  - 4) Is the training of f-DVL more stable than XQL?

Experiments:

1) How does ReCOIL perform and compare with previous offline IL methods?

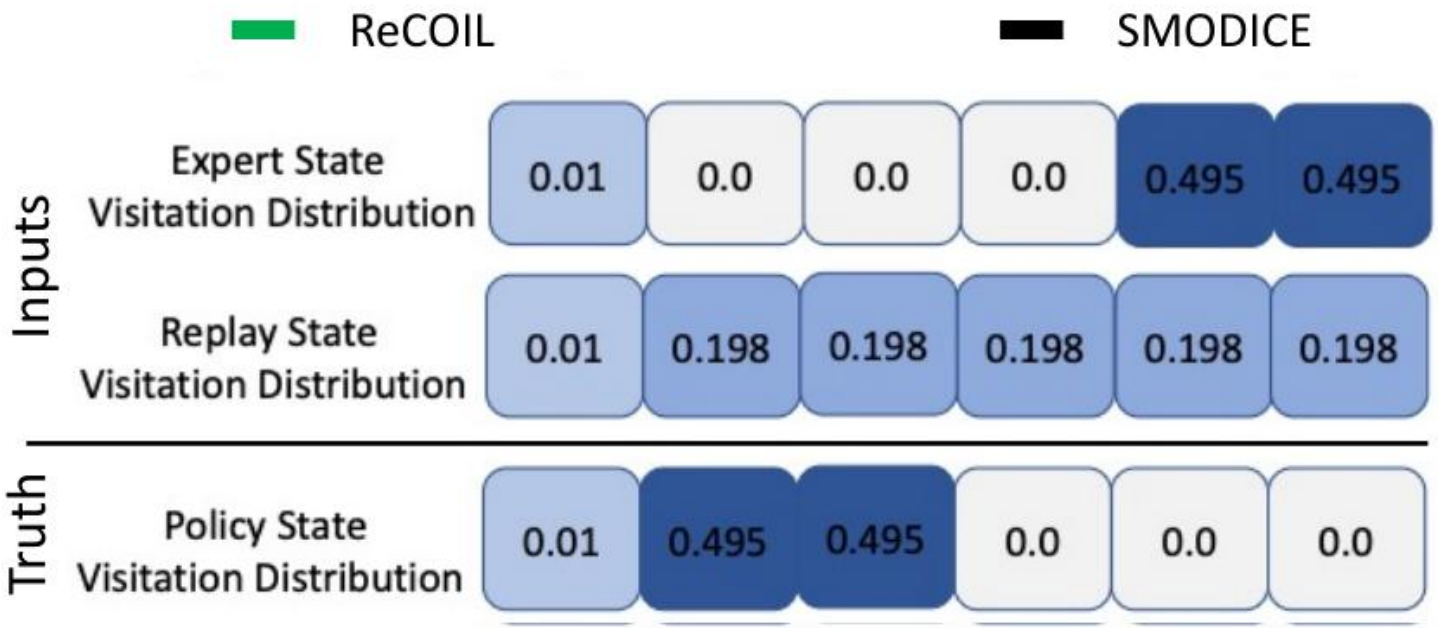
- *random+expert* and *medium+expert*
  - suboptimal dataset: 1 million transitions of the random or medium D4RL datasets
  - expert demonstrations: 200
- *random+few-expert* and *medium+few-expert*
  - suboptimal datasets mixed with only 30 expert demonstrations
  - more difficult setting

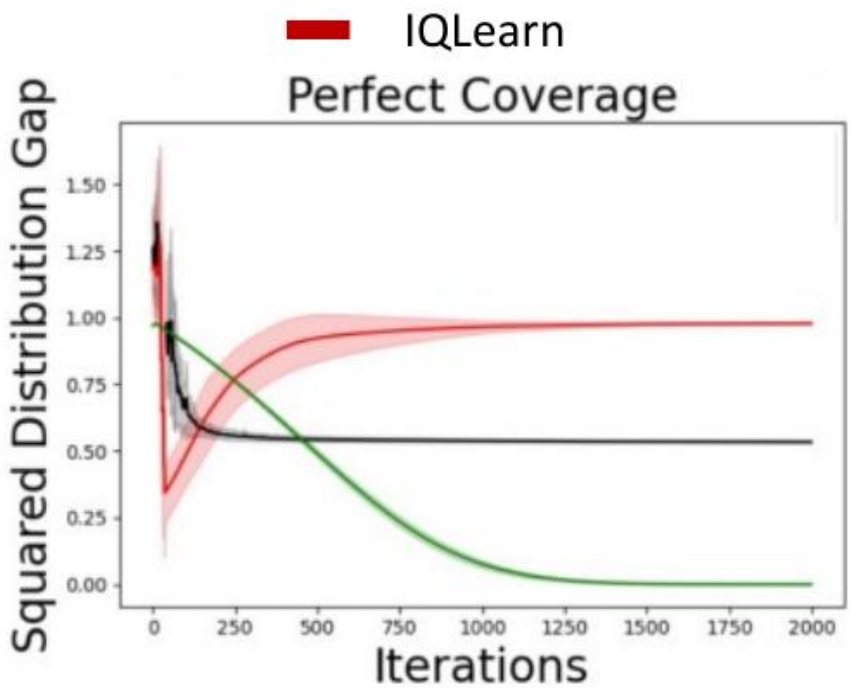
Suboptimal Dataset	Env	RCE	ORIL	SMODICE	BC (only expert data)	BC (full dataset)	IQ-Learn (offline)	ReCOIL	Expert
random+ expert	hopper	51.41±38.63	73.93±11.06	**101.61±7.69**	4.52±1.42	5.64±4.83	1.85 ±2.19	**108.18±3.28**	111.33
	halfcheetah	64.19±11.06	60.49±3.53	**80.16±7.30**	2.2±0.01	2.25±0.00	4.83±7.99	**80.20±6.61**	88.83
	walker2d	20.90±26.80	2.86±3.39	**105.86±3.47**	0.86±0.61	0.91±0.5	0.57±0.09	**102.16±7.19**	106.92
	ant	105.38±14.15	73.67±12.69	**126.78±5.12**	5.17±5.43	30.66±1.35	42.23±20.05	**126.74±4.63**	130.75
random+ few-expert	hopper	25.31±18.97	42.04±13.76	60.11±18.28	4.84±3.83	3.0±0.54	1.37 ±1.23	**97.85±17.89**	111.33
	halfcheetah	2.99±1.07	2.84±5.52	2.28±0.62	-0.93±0.35	2.24±0.01	1.14±1.94	**76.92±7.53**	88.83
	walker2d	40.49±26.52	3.22±3.29	**107.18±1.87**	0.98±0.83	0.74±0.20	0.39±0.27	83.23±19.00	106.92
	ant	**67.62±15.81**	25.41 ± 8.58	-6.10±7.85	0.91±3.93	35.38±2.66	32.99±3.12	**67.14± 8.30**	130.75
medium+ expert	hopper	58.71±34.06	61.68±7.61	49.74±3.62	16.09±12.80	59.25±3.71	12.90±24.00	**88.51±16.73**	111.33
	halfcheetah	65.14±13.82	54.66±0.88	59.50±0.82	-1.79±0.22	42.45± 0.42	25.67±20.82	**81.15±2.84**	88.83
	walker2d	**96.24±14.04**	8.19±7.70	2.62±0.93	2.43±1.82	72.76±3.82	59.37±30.14	**108.54±1.81**	106.92
	ant	**86.14±38.59**	102.74±6.63	104.95±6.43	0.86±7.42	95.47±10.37	37.17±41.15	**120.36±7.67**	130.75
medium few-expert	hopper	**66.15±35.16**	17.40±15.15	47.61±7.08	7.37±1.13	46.87±5.31	11.05±20.59	**50.01±10.36**	111.33
	halfcheetah	**61.14±18.31**	43.24±0.75	46.45±3.12	-1.15±0.06	42.21±0.06	26.27±20.24	**75.96±4.54**	88.83
	walker2d	**85.28±34.90**	6.81±6.76	6.00±6.69	2.02±0.72	70.42±2.86	73.30±2.85	**91.25±17.63**	106.92
	ant	**67.95±36.78**	81.53±8.618	81.53±8.618	-10.45±1.63	81.63±6.67	35.12±50.56	**110.38±10.96**	130.75
cloned+expert	pen	19.60±11.40	-3.10±0.40	-3.36±0.71	13.95±11.04	34.94±11.10	2.18±8.75	**95.04±4.48**	106.42
	door	0.08± 0.15	-0.33±0.01	0.25± 0.54	-0.22±0.05	0.011±0.00	0.07±0.02	**102.75±4.05**	103.94
	hammer	1.95±3.89	0.25± 0.01	0.15± 0.078	2.41±4.48	5.45± 7.84	0.27±0.02	**95.77±17.90**	125.71
	relocate	-0.25±0.04	-0.29±0.01	1.75±3.85	-0.17±0.04	-0.24± 0.01	-0.1±0.12	**67.43±14.60**	118.39
human+expert	pen	17.81±5.91	-3.38±2.29	-2.20±2.40	13.83±10.76	90.76±25.09	14.29±28.82	**103.72±2.90**	106.42
	door	-0.05±0.05	-0.33±0.01	-0.20± 0.11	-0.03±0.05	103.71±1.22	5.6±7.29	**104.70±0.55**	103.94
	hammer	5.00±5.64	1.89±0.70	-0.07±0.39	0.18±0.14	122.61±4.85	5.32±1.38	**125.19±3.29**	125.71
	relocate	0.02±0.10	-0.29±0.01	-0.16±0.04	-0.13±0.11	81.19±7.73	-0.04±0.22	**91.98± 2.89**	118.39
partial+expert	kitchen	6.875±9.24	0.00±0.00	39.16± 1.17	2.5±5.0	45.5±1.87	0.0±0.0	**60.0±5.70**	75.0
mixed+expert	kitchen	1.66±2.35	0.00±0.00	42.5±2.04	2.2±3.8	42.1±1.12	0.0±0.0	**52.0±1.0**	75.0

- Methods based on coverage assumption fail when
  - 1) coverage is low (few expert trajectories in dataset)
  - 2) in high dimensional tasks where the discriminator easily overfits
- ReCOIL outperforms baselines by a large margin

Experiments:

2) Can ReCOIL accurately estimate the policy visitation distribution d^{π} and the reward function/intent of the expert?





- expert와 policy의 visitation distribution이 거의 반대인 상황임에도, ReCOIL이 coverage가 불충분한 오프라인 데이터만으로 정책 π 의 실제 분포 d^{π} 를 가장 잘 복원함

Experiments:

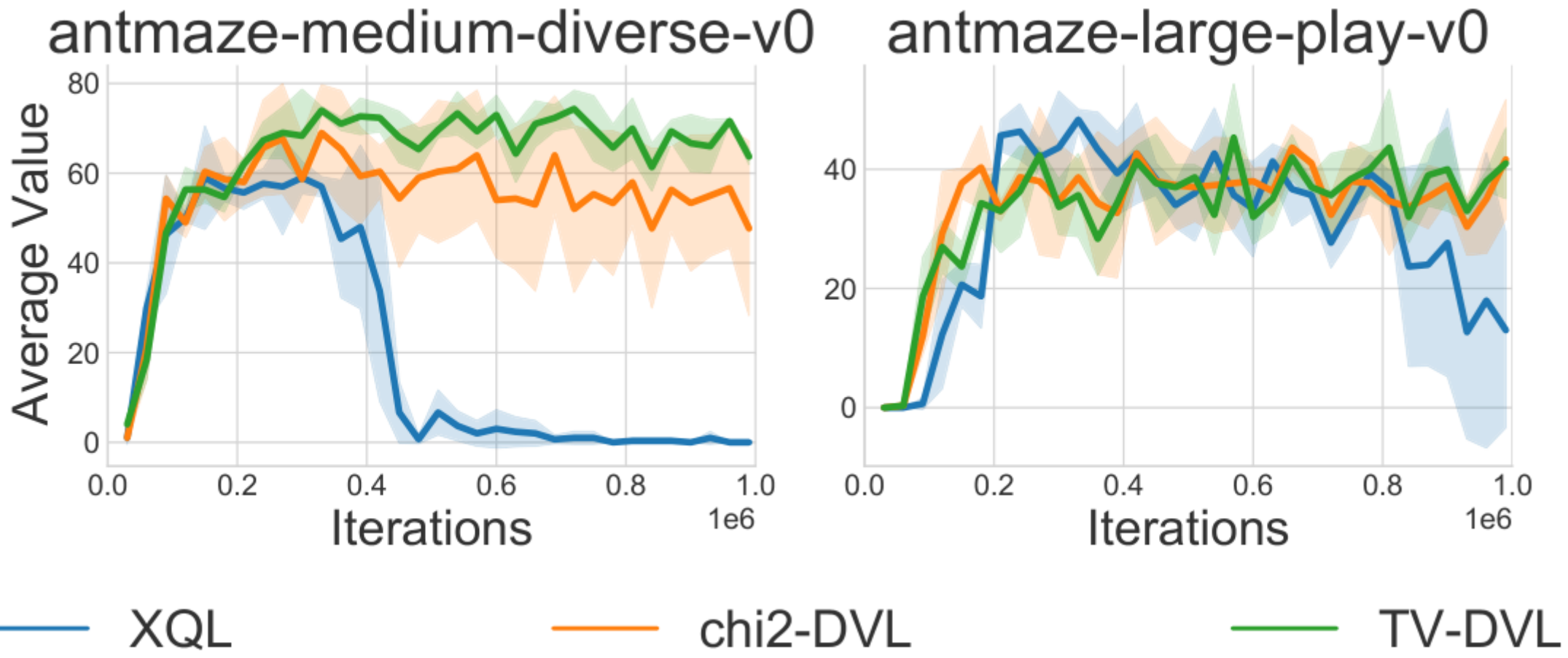
3) How does f-DVL perform and compare with previous offline RL methods?

Dataset	BC	10%BC	DT	TD3+BC	CQL	IQL	XQL(r)	f-DVL (χ^2)	f-DVL (TV)
halfcheetah-medium-v2	42.6	42.5	42.6	**48.3**	44.0	**47.4**	47.4	**47.7**	47.5
hopper-medium-v2	52.9	56.9	67.6	59.3	58.5	66.3	**68.5**	63.0	64.1
walker2d-medium-v2	75.3	75.0	74.0	**83.7**	72.5	78.3	81.4	80.0	81.5
halfcheetah-medium-replay-v2	36.6	40.6	36.6	**44.6**	**45.5**	44.2	44.1	42.9	**44.7**
hopper-medium-replay-v2	18.1	75.9	82.7	60.9	95.0	94.7	95.1	90.7	**98.0**
walker2d-medium-replay-v2	26.0	62.5	66.6	**81.8**	77.2	73.9	58.0	52.1	68.7
halfcheetah-medium-expert-v2	55.2	**92.9**	86.8	90.7	91.6	86.7	90.8	89.3	91.2
hopper-medium-expert-v2	52.5	**110.9**	107.6	98.0	105.4	91.5	94.0	105.8	93.3
walker2d-medium-expert-v2	107.5	109.0	108.1	**110.1**	108.8	**109.6**	**110.1**	**110.1**	**109.6**
antmaze-umaze-v0	54.6	62.8	59.2	78.6	74.0	**87.5**	47.7	83.7	**87.7**
antmaze-umaze-diverse-v0	45.6	50.2	53.0	71.4	**84.0**	62.2	51.7	50.4	48.4
antmaze-medium-play-v0	0.0	5.4	0.0	10.6	61.2	**71.2**	31.2	56.7	**71.0**
antmaze-medium-diverse-v0	0.0	9.8	0.0	3.0	53.7	**70.0**	0.0	48.2	60.2
antmaze-large-play-v0	0.0	0.0	0.0	0.2	15.8	39.6	10.7	36.0	**41.7**
antmaze-large-diverse-v0	0.0	6.0	0.0	0.0	14.9	**47.5**	31.28	44.5	39.3
kitchen-complete-v0	65.0	-	-	-	43.8	62.5	56.7	**67.5**	61.3
kitchen-partial-v0	38.0	-	-	-	49.8	46.3	48.6	58.8	**70.0**
kitchen-mixed-v0	51.5	-	-	-	51.0	51.0	40.4	**53.75**	52.5

- XQL(r)은 원 논문의 실험 프로토콜 오류를 수정하여 결과를 다시 뽑은 것
  - original paper taking the best average return during training as opposed to the standard practice of taking the average of the last iterate performance across different seeds at 1 million gradient steps

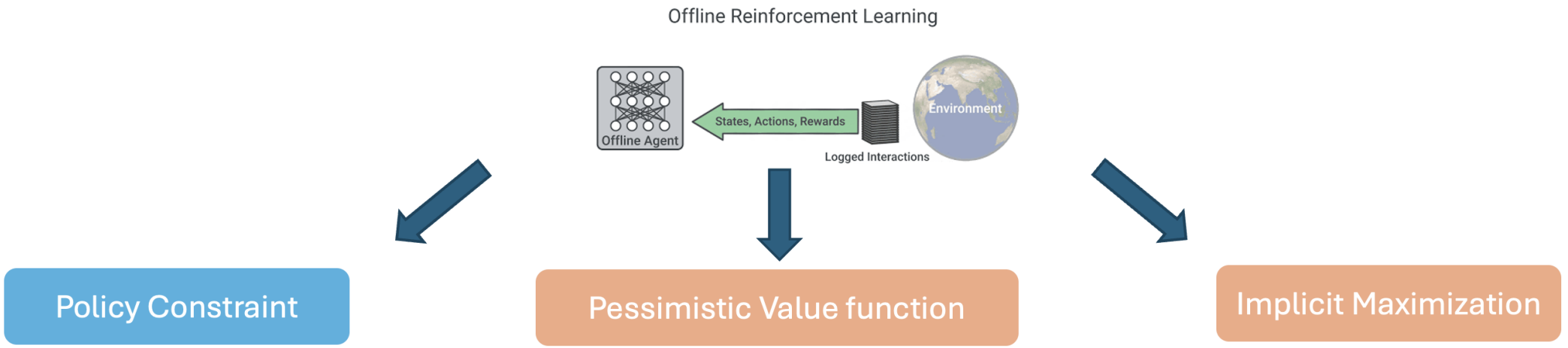
Experiments:

4) Is the training of f-DVL more stable than XQL?



- XQL training diverges due to the numerical instability of its loss function
- f-DVL fixes this problem by using more well-behaved f-divergences

Closing: Unifying existing work with dual RL



	Method	dual-Q/V	Gradient	Objective	Off-Policy Data
RL	AlgaeDICE, GenDICE, *CQL*	Q	semi	regularized RL	Arbitrary
	OptiDICE	V	full	regularized RL	Arbitrary
	XQL, REPS, f-**DVL**	V	semi	regularized RL	Arbitrary
	VIP, GoFAR	V	full	regularized RL	Arbitrary
	Logistic Q-learning	QV[1]	full	regularized RL	✗
IL	*IQLearn, IBC*	Q	semi	$D_f(\rho^\pi \Vert \rho^E)$	Expert-only
	OPOLO, OPIRL	Q	semi	$D_{kl}(\rho^\pi \Vert \rho^E)$	Arbitrary
	SMODICE	V	full	$D_{kl}(\rho^\pi \Vert \rho^E)$	Arbitrary
	DemoDICE, LobsDICE	V	full	$D_{kl}(\rho^\pi \Vert \rho^E) + \alpha D_{kl}(\rho^\pi \Vert \rho^R)$	Arbitrary
	P^2IL	QV[1]	full	$D_C(\rho^\pi \Vert \rho^E)$[1]	✗
	ReCOIL-Q	Q	full	$D_f(\rho^\pi_{mix} \Vert \rho^{E,R}_{mix})$	Arbitrary
	ReCOIL-V	V	full	$D_f(\rho^\pi_{mix} \Vert \rho^{E,R}_{mix})$	Arbitrary

후일담

- arxiv version1:
  - Imitation from Arbitrary Experience: A Dual Unification of Reinforcement and Imitation Learning Methods
  - ICLR Workshop 2023
- arxiv version2-3:
  - Dual RL: Unification and New Methods for Reinforcement and Imitation Learning
  - ICLR 2024
  - 후속 연구가 아닌 버전 업데이트인데 새로운 학회에 발표해도 되는건가?
- visitation distribution 계산을 간소하기 위해 Successor Feature를 도입할 수 있을 것 같은데 저자들의 후속논문이 이와 관련있어보임
  - Proto Successor Measure: Representing the Behavior Space of an RL Agent, ICML 2025
- Qinqing Zheng과 Amy Zhang은 Online Decision Transformer의 저자들임
  - Dual RL framework은 Decision Transformer 계열의 Bellman 방정식을 사용하지 않는 연구들을 포용하지는 못함

Dual RL in a nutshell

Dual RL:
Unification and
New Methods
for Reinforcement
and Imitation Learning

Regularized RL
- Convex Primal
- f-divergence regularization
- linear Bellman constraints

Lagrangian duality

Dual RL (unconstrained dual)
- optimize over {Q} or {V}
- uses f* (convex conjugate)

On-policy policy gradient computed from off-policy data (stability + convergence)

Dual-Q (2-player game)

$$\max_\pi \min_Q (1-\gamma)\mathbb{E}_{s\sim d_0, a\sim\pi(s)}[Q(s,a)] + \alpha\mathbb{E}_{(s,a)\sim d^O}\left[f^*\left(\left[\mathcal{T}_r^\pi Q(s,a) - Q(s,a)\right]/\alpha\right)\right]$$

Dual-V (1-player optimization)

$$\min_V (1-\gamma)\mathbb{E}_{s\sim d_0}[V(s)] + \alpha\mathbb{E}_{(s,a)\sim d^O}\left[f_p^*\left(\left[\mathcal{T}V(s,a) - V(s)\right]/\alpha\right)\right]$$

Special case: IL
- IL as RL with reward=0
- $d^O = d^E$

Special case: ReCOIL
- $d^S_{\text{mix}} := \beta d(s,a) + (1-\beta)d^S(s,a)$
- $d^{E,S}_{\text{mix}} := \beta d^E(s,a) + (1-\beta)d^S(s,a)$

Special case: RL
Pessmistic Value Learning
CQL, ATAC

Special case: RL
Implicit-way
XQL, f-DVL(proposed)