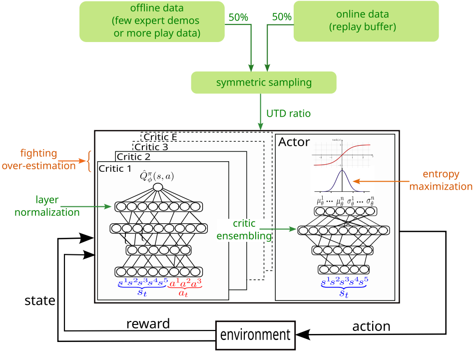
Reinforcement Learning with Prior Data (RLPD)

Olivier Sigaud

Sorbonne Université
http://www.isir.upmc.fr/personnel/sigaud

RLPD: Overview

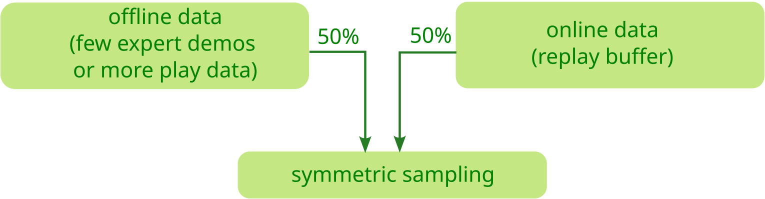


Mechanisms in brown are environment-dependent

- RLPD builds upon SAC and adds several complementary advances:
  - It efficiently combines offline RL with any dataset (expert or play data) with online fine-tuning
  - It uses Layer Normalization
  - It combines it with high UTD ratio
  - Depending on the environment:
    - It uses 1 or 2 critics (TD3 trick) to counteract over-estimation bias
    - It uses entropy maximization or not to favor exploration

Balanced sampling

Offline data + Off-policy learning

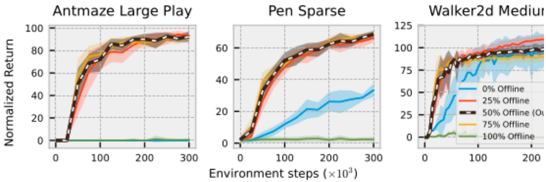


- Inspired from [Ross and Bagnell, 2012]
- Better than offline pre-training then online fine-tuning (see ablations)
- But contradicted by the WSRL paper [Zhou et al., 2024]
- Offline-to-Online is a very active field...

Ross, S. and Bagnell, J. A. (2012) Agnostic system identification for model-based reinforcement learning. arXiv preprint arXiv:1203.1007.

Zhou, Z., Peng, A., Li, Q., Levine, S., and Kumar, A. (2024) Efficient online reinforcement learning fine-tuning need not retain offline data. *arXiv preprint arXiv:2412.07762*

Symmetric sampling vs buffer initialization



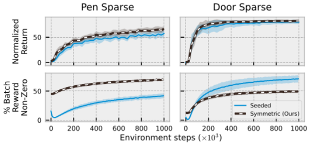
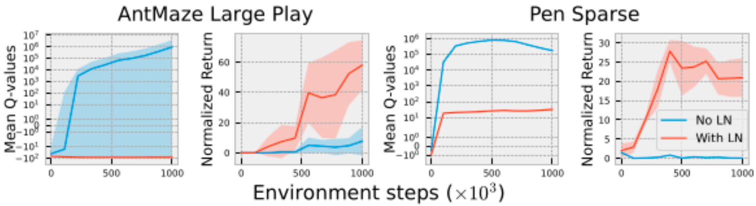


Figure 10. Symmetric sampling improves sample efficiency and reduces variance across seeds, and does not work by simply increasing the reward density in a batch.

- Low sensitivity to the amount of mixing
- 50% offers the best compromise between variance, speed of convergence, and asymptotic performance.
- Another option would be to initialize the buffer with the offline dataset (seeding)
- Initializing the buffer with large amounts of data limits improvement
- Symmetric sampling works better

Layer normalization

Layer Norm



- Offline data + Off-policy learning is not enough to get strong performance
- LayerNorm helps
- Without LayerNorm, Q-values are over-estimated and the policy performs poorly
- Now a common recipe (see also SIMBA, BRO, ...)

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016) Layer normalization. *arXiv preprint arXiv:1607.06450*

Effects of Layer Norm

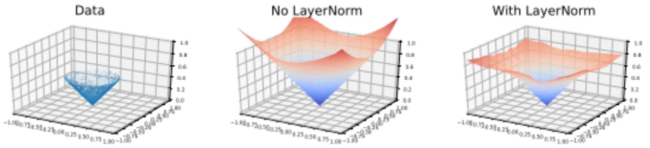


Figure 3. We fit data (left) with a two-layer MLP without LayerNorm (center) and with LayerNorm (right). LayerNorm bounds the values and prevents catastrophic overestimation.

- Prevents catastrophic value extrapolation in OOD data
- See [Kostrikov et al., 2021]: offpolicy methods often prevent exploration to avoid OOD over-estimation
- IQL finds a way to prevent this

Kostrikov, I., Nair, A., and Levine, S. (2021) Offline reinforcement learning with implicit Q-learning. *arXiv preprint arXiv:2110.06169 (ICLR 2023)*

Impact of Layer Norm

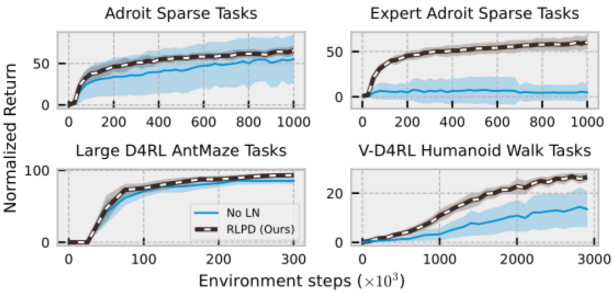
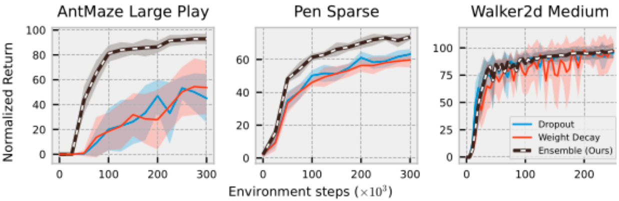


Figure 7. LayerNorm is crucial for strong performance, particularly when data are limited or narrowly distributed.

- Better overall performance

Towards high UTD ratio

High UTD (update-to-data) ratio

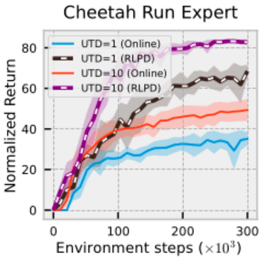


- High UTD ratio: perform many gradient steps from the same data
- High UTD ratio results in statistical overfitting [Li et al., 2023]
- Three techniques:
  1. L2 regularization of parameters [Večerík et al., 2017]
  2. Dropout (DROQ) [Hiraoka et al., 2021]
  3. Random Ensemble Distillation (REDQ) [Chen et al., 2021] → works best
- Use $E = 10$ networks (empirical, not studied)
- Update the actor taking the average over critic gradients

Li, Q., Kumar, A., Kostrikov, I., and Levine, S. (2023) Efficient deep reinforcement learning requires regulating overfitting. *arXiv preprint arXiv:2304.10466*

Chen, X., Wang, C., Zhou, Z., and Ross, K. (2021) Randomized ensembled double Q-learning: Learning fast without a model. *arXiv preprint arXiv:2101.05982*

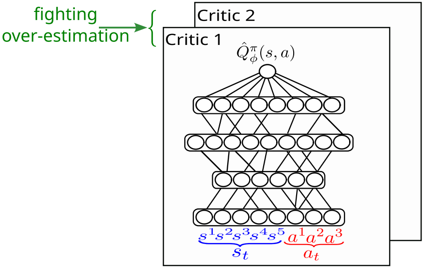
UTD HalfCheetah



- Increasing UTD with RLPD improves sample efficiency from pixels.

Fighting over-estimation bias

Over-estimation bias

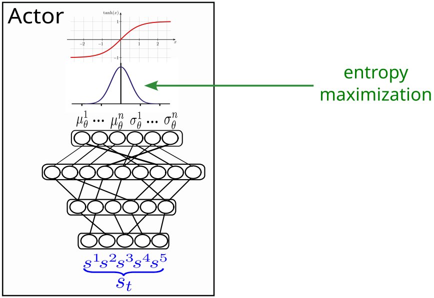


- Using 2 critics as in TD3 and SAC might not be necessary
- This is environment-dependent, choose experimentally
- To combine with ensembling, choose one or two critics among E to perform updates

Fujimoto, S., van Hoof, H., & Meger, D. (2018) Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*

Entropy maximization

Entropy maximization



- Several SOTA RL algos such as SAC explore by maximizing the entropy of the policy (and critic)
- Sometimes, SAC outperforms TD3, sometimes not
- So using entropy maximization should be an environment-dependent decision

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A. Abbeel, P. et al. (2018) Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*

The RLPD algorithm

Algorithm 1 Online RL with Offline Data (RLPD)

1: Select LayerNorm, Large Ensemble Size E, Gradient Steps G, and architecture.
2: Randomly initialize Critic θ_i (set targets $\theta'_i = \theta_i$) for $i = 1, 2, \ldots, E$ and Actor ϕ parameters. Select discount γ, temperature α and critic EMA weight ρ.
3: Determine number of Critic targets to subset $Z \in \{1, 2\}$
4: Initialize empty replay buffer $\mathcal{R}$
5: Initialize buffer $\mathcal{D}$ with offline data
6: **while** True **do**
7: Receive initial observation state s_0
8: **for** t = 0, T **do**
9: Take action $a_t \sim \pi_\phi(\cdot|s_t)$
10: Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{R}$
11: **for** $g = 1, G$ **do**
12: Sample minibatch b_R of $\frac{N}{2}$ from $\mathcal{R}$
13: Sample minibatch b_D of $\frac{N}{2}$ from $\mathcal{D}$
14: Combine b_R and b_D to form batch b of size N
15: Sample set $\mathcal{Z}$ of Z indices from $\{1, 2, \ldots, E\}$
16: With b, set

$$y = r + \gamma \left(\min_{i \in \mathcal{Z}} Q_{\theta'_i}(s', \tilde{a}') \right), \quad \tilde{a}' \sim \pi_\phi(\cdot|s')$$

17: Add entropy term $y = y + \gamma\alpha \log \pi_\phi(\tilde{a}'|s')$
18: **for** $i = 1, E$ **do**
19: Update θ_i minimizing loss:

$$L = \frac{1}{N} \sum_i (y - Q_{\theta_i}(s, a))^2$$

20: **end for**
21: Update target networks $\theta'_i \leftarrow \rho\theta'_i + (1 - \rho)\theta_i$
22: **end for**
23: With b, update ϕ maximizing objective:

$$\frac{1}{E} \sum_{i=1}^{E} Q_{\theta_i}(s, \tilde{a}) - \alpha \log \pi_\phi(\tilde{a}|s), \quad \tilde{a} \sim \pi_\phi(\cdot \mid s)$$

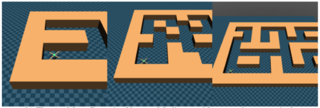
24: **end for**
25: **end while**

- In the paper page 5 (quite clear)
- A mistake line 17: should be $y = y - \gamma\alpha log(\pi_\phi(\bar{a}'|s'))$
- The official implementation is correct

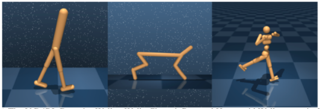
Results

Environments

(a) The Sparse Adroit Domain. Pen, Door and Relocate tasks respectively.



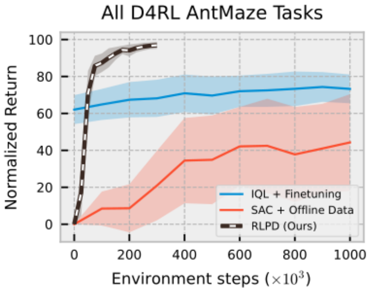
(b) The AntMaze Domain. Umaze, Medium and Large tasks respectively.



(c) The V-D4RL Domain. Walker Walk, Cheetah Run and Humanoid Walk respectively.

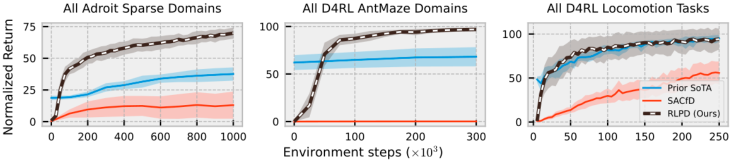
- Adroit, D4RL AntMaze, V-D4RL locomotion

Main AntMaze result (front figure)



- Much better performance and sample efficiency than competitors

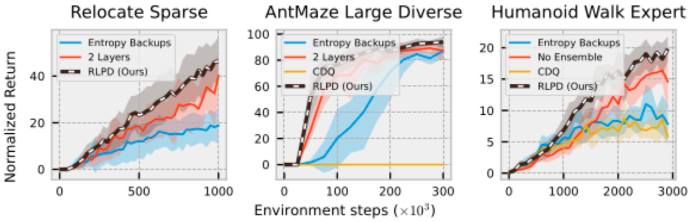
Global results through domains



- 10 seeds, 1 std shaded
- In ADROIT and ANTMAZE, their prior SOTA is IQL + fine-tuning
- In locomotion, the prior SOTA (OFF2ON) is hard to beat

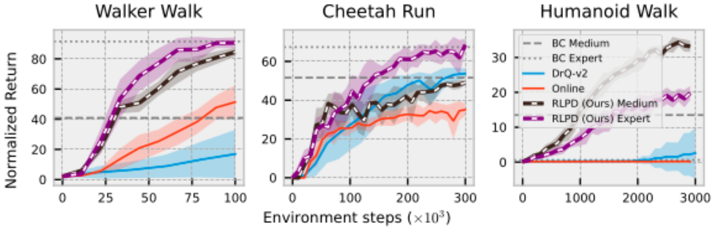
Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J. (2022) Offline-to-online reinforcement learning via balanced replay and pessimistic Q-ensemble. In *Conference on Robot Learning*, pages 1702–1712. PMLR

Ablations: Results on hardest tasks



- With 2 or 3 layers
- With or without entropy maximization
- With or without min from 2 critics
- With or without random ensemble distillation

Training from images



- They use a six layer CNN as input architecture
- LeRobot people also use a pre-trained ResNet 10
- To avoid overfitting, image augmentation (random shift, 4 pixels)

Yarats, D., Kostrikov, I., and Fergus, R. (2021) Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International conference on learning representations*

Hyper-parameters

Table 1. RLPD hyperparameters.

Parameter	Value
Online batch size	128
Offline batch size	128
Discount (γ)	0.99
Optimizer	Adam
Learning rate	3×10^{-4}
Ensemble size (E)	10
Critic EMA weight (ρ)	0.005
Gradient Steps (State Based) (G or UTD)	20
Network Width	256 Units
Initial Entropy Temperature (α)	1.0
Target Entropy	$-\dim(\mathcal{A})/2$
Pixel-Based Hyperparameters	
Action repeat	2
Observation size	[64, 64]
Image shift amount	4

Table 2. Environment specific hyperparameters.

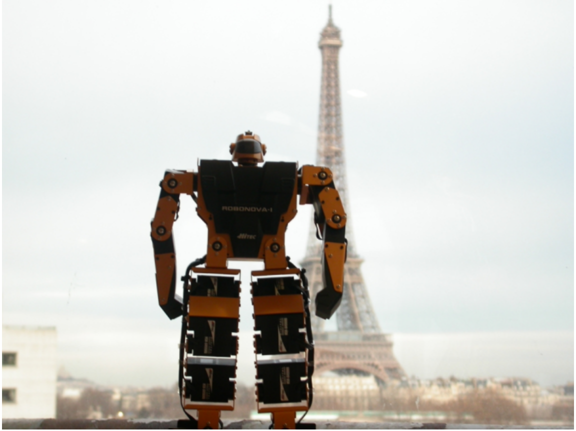
Environment	CDQ	Entropy Backups	MLP Architecture
Locomotion	True	True	2 Layer
AntMaze	False	False	3 Layer
Adroit	True	False	3 Layer
DMC (Pixels)	False	False	2 Layer

- Looks clean, according to LeRobot members, everything is specified

Implementation details

- According to LeRobot members, the following matters:
  - Reward, state and action normalization matters a lot
  - On robots, one should initialize the policy very close to 0 (by dividing last layer weights by ≈ 100)
  - Decoupling data collection and training: use two threads, adjust the rate at which the actor is updated
  - Rather insensitive to the size of the replay buffer
- Mistakes in SERL and HIL-SERL implementations: γ is forgotten:
  - in SERL: https://github.com/rail-berkeley/serl/blob/(...)/agents/continuous/sac.py#L172
  - in HIL-SERL: https://github.com/rail-berkeley/hil-serl/blob/main/serl_launcher/serl_launcher/agents/continuous/sac.py#L187

Any question?



Send mail to: `Olivier.Sigaud@upmc.fr`

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016).
Layer normalization. arxiv.
arXiv preprint arXiv:1607.06450.

Chen, X., Wang, C., Zhou, Z., and Ross, K. (2021).
Randomized ensembled double Q-learning: Learning fast without a model.
arXiv preprint arXiv:2101.05982.

Fujimoto, S., van Hoof, H., and Meger, D. (2018).
Addressing function approximation error in actor-critic methods.
In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1582–1591. PMLR.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018).
Soft actor-critic algorithms and applications.
arXiv preprint arXiv:1812.05905.

Hiraoka, T., Imagawa, T., Hashimoto, T., Onishi, T., and Tsuruoka, Y. (2021).
Dropout Q-functions for doubly efficient reinforcement learning.
arXiv preprint arXiv:2110.02034.

Kostrikov, I., Nair, A., and Levine, S. (2021).
Offline reinforcement learning with implicit Q-learning.
arXiv preprint arXiv:2110.06169 (ICLR 2023).

Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J. (2022).
Offline-to-online reinforcement learning via balanced replay and pessimistic Q-ensemble.
In *Conference on Robot Learning*, pages 1702–1712. PMLR.

Li, Q., Kumar, A., Kostrikov, I., and Levine, S. (2023).
Efficient deep reinforcement learning requires regulating overfitting.
arXiv preprint arXiv:2304.10466.

Ross, S. and Bagnell, J. A. (2012).
Agnostic system identification for model-based reinforcement learning.
arXiv preprint arXiv:1203.1007.

Večerík, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., and Riedmiller, M. (2017).
Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards.
arXiv preprint arXiv:1707.08817.

Yarats, D., Kostrikov, I., and Fergus, R. (2021).
Image augmentation is all you need: Regularizing deep reinforcement learning from pixels.
In *International conference on learning representations.*

Zhou, Z., Peng, A., Li, Q., Levine, S., and Kumar, A. (2024).
Efficient online reinforcement learning fine-tuning need not retain offline data.
arXiv preprint arXiv:2412.07762.