

# 000 001 002 003 004 005 006 007 008 009 010 011 012 OFFLINE REINFORCEMENT LEARNING THROUGH TRAJECTORY CLUSTERING AND LOWER BOUND PENALISATION

013  
014  
015  
016  
017  
018  
019  
020  
021  
022  
023  
024  
025  
026  
027  
028  
029  
030  
**Anonymous authors**

Paper under double-blind review

## 024 025 026 027 028 029 030 ABSTRACT

031  
032  
033  
034  
035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045  
In this paper, we propose a new framework for value regularisation in offline reinforcement learning (RL). While most previous methods evade explicit out-of-distribution (OOD) region identification due to its difficulty, our method explicitly identifies the OOD region, which can be non-convex depending on datasets, via a newly proposed trajectory clustering-based behaviour cloning algorithm. With the obtained explicit OOD region, we then define a Bellman-type operator pushing the value in the OOD region to a tight lower bound while operating normally in the in-distribution region. The value function with this operator can be used for policy acquisition in various ways. Empirical results on multiple offline RL benchmarks show that our method yields the state-of-the-art performance.

## 046 047 048 049 050 1 INTRODUCTION

051  
052  
053  
Offline reinforcement learning (RL) has attracted significant attention due to its sample efficiency and safety. Unlike conventional RL, where an agent learns an optimal policy through interactions with the environment, offline RL disallows any environmental interactions. Instead, the agent is provided with a fixed dataset  $\mathcal{D}$  of trajectories and should derive a competent policy solely from these samples.

054  
055  
056  
057  
058  
059  
060  
Although offline RL also relies on off-policy data, standard off-policy RL algorithms often fail in this setting. The primary cause is the extrapolation error of the critic, which cannot be corrected, as the agent is unable to re-evaluate overestimated out-of-distribution (OOD) actions through environmental interactions (Fujimoto et al., 2019). These errors not only persist but also accumulate through bootstrapping, making careful handling of OOD actions crucial for stable training.

061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
0100  
Value regularisation has emerged as one of the main strategies for addressing the extrapolation issue (Kumar et al., 2020; Lyu et al., 2022; Mao et al., 2023). By penalising the critic values of OOD actions, these methods encourage the agent to prefer in-distribution (ID) actions over OOD ones. A variety of algorithms have been proposed within this paradigm, differing mainly in their choice of regularisation term. However, most rely on indirect proxies to determine the OOD-ness of actions. For example, Mao et al. (2023) approximates the behaviour policy with a Gaussian model to compute importance-sampling ratios. When the dataset contains multiple disparate behaviour policies, such unimodal approximation misrepresent the underlying multimodal structure. They assign spuriously high densities to the inter-modal region, distorting the OOD-ness estimates, eventually degrading the regularisation’s effectiveness.

0101  
0102  
0103  
0104  
0105  
0106  
0107  
0108  
0109  
0110  
0111  
0112  
0113  
0114  
0115  
0116  
0117  
0118  
0119  
0120  
0121  
0122  
0123  
0124  
0125  
0126  
0127  
0128  
0129  
0130  
0131  
0132  
0133  
0134  
0135  
0136  
0137  
0138  
0139  
0140  
0141  
0142  
0143  
0144  
0145  
0146  
0147  
0148  
0149  
0150  
0151  
0152  
0153  
0154  
0155  
0156  
0157  
0158  
0159  
0160  
0161  
0162  
0163  
0164  
0165  
0166  
0167  
0168  
0169  
0170  
0171  
0172  
0173  
0174  
0175  
0176  
0177  
0178  
0179  
0180  
0181  
0182  
0183  
0184  
0185  
0186  
0187  
0188  
0189  
0190  
0191  
0192  
0193  
0194  
0195  
0196  
0197  
0198  
0199  
0200  
0201  
0202  
0203  
0204  
0205  
0206  
0207  
0208  
0209  
0210  
0211  
0212  
0213  
0214  
0215  
0216  
0217  
0218  
0219  
0220  
0221  
0222  
0223  
0224  
0225  
0226  
0227  
0228  
0229  
0230  
0231  
0232  
0233  
0234  
0235  
0236  
0237  
0238  
0239  
0240  
0241  
0242  
0243  
0244  
0245  
0246  
0247  
0248  
0249  
0250  
0251  
0252  
0253  
0254  
0255  
0256  
0257  
0258  
0259  
0260  
0261  
0262  
0263  
0264  
0265  
0266  
0267  
0268  
0269  
0270  
0271  
0272  
0273  
0274  
0275  
0276  
0277  
0278  
0279  
0280  
0281  
0282  
0283  
0284  
0285  
0286  
0287  
0288  
0289  
0290  
0291  
0292  
0293  
0294  
0295  
0296  
0297  
0298  
0299  
0300  
0301  
0302  
0303  
0304  
0305  
0306  
0307  
0308  
0309  
0310  
0311  
0312  
0313  
0314  
0315  
0316  
0317  
0318  
0319  
0320  
0321  
0322  
0323  
0324  
0325  
0326  
0327  
0328  
0329  
0330  
0331  
0332  
0333  
0334  
0335  
0336  
0337  
0338  
0339  
0340  
0341  
0342  
0343  
0344  
0345  
0346  
0347  
0348  
0349  
0350  
0351  
0352  
0353  
0354  
0355  
0356  
0357  
0358  
0359  
0360  
0361  
0362  
0363  
0364  
0365  
0366  
0367  
0368  
0369  
0370  
0371  
0372  
0373  
0374  
0375  
0376  
0377  
0378  
0379  
0380  
0381  
0382  
0383  
0384  
0385  
0386  
0387  
0388  
0389  
0390  
0391  
0392  
0393  
0394  
0395  
0396  
0397  
0398  
0399  
0400  
0401  
0402  
0403  
0404  
0405  
0406  
0407  
0408  
0409  
0410  
0411  
0412  
0413  
0414  
0415  
0416  
0417  
0418  
0419  
0420  
0421  
0422  
0423  
0424  
0425  
0426  
0427  
0428  
0429  
0430  
0431  
0432  
0433  
0434  
0435  
0436  
0437  
0438  
0439  
0440  
0441  
0442  
0443  
0444  
0445  
0446  
0447  
0448  
0449  
0450  
0451  
0452  
0453  
0454  
0455  
0456  
0457  
0458  
0459  
0460  
0461  
0462  
0463  
0464  
0465  
0466  
0467  
0468  
0469  
0470  
0471  
0472  
0473  
0474  
0475  
0476  
0477  
0478  
0479  
0480  
0481  
0482  
0483  
0484  
0485  
0486  
0487  
0488  
0489  
0490  
0491  
0492  
0493  
0494  
0495  
0496  
0497  
0498  
0499  
0500  
0501  
0502  
0503  
0504  
0505  
0506  
0507  
0508  
0509  
0510  
0511  
0512  
0513  
0514  
0515  
0516  
0517  
0518  
0519  
0520  
0521  
0522  
0523  
0524  
0525  
0526  
0527  
0528  
0529  
0530  
0531  
0532  
0533  
0534  
0535  
0536  
0537  
0538  
0539  
0540  
0541  
0542  
0543  
0544  
0545  
0546  
0547  
0548  
0549  
0550  
0551  
0552  
0553  
0554  
0555  
0556  
0557  
0558  
0559  
0560  
0561  
0562  
0563  
0564  
0565  
0566  
0567  
0568  
0569  
0570  
0571  
0572  
0573  
0574  
0575  
0576  
0577  
0578  
0579  
0580  
0581  
0582  
0583  
0584  
0585  
0586  
0587  
0588  
0589  
0590  
0591  
0592  
0593  
0594  
0595  
0596  
0597  
0598  
0599  
0600  
0601  
0602  
0603  
0604  
0605  
0606  
0607  
0608  
0609  
0610  
0611  
0612  
0613  
0614  
0615  
0616  
0617  
0618  
0619  
0620  
0621  
0622  
0623  
0624  
0625  
0626  
0627  
0628  
0629  
0630  
0631  
0632  
0633  
0634  
0635  
0636  
0637  
0638  
0639  
0640  
0641  
0642  
0643  
0644  
0645  
0646  
0647  
0648  
0649  
0650  
0651  
0652  
0653  
0654  
0655  
0656  
0657  
0658  
0659  
0660  
0661  
0662  
0663  
0664  
0665  
0666  
0667  
0668  
0669  
0670  
0671  
0672  
0673  
0674  
0675  
0676  
0677  
0678  
0679  
0680  
0681  
0682  
0683  
0684  
0685  
0686  
0687  
0688  
0689  
0690  
0691  
0692  
0693  
0694  
0695  
0696  
0697  
0698  
0699  
0700  
0701  
0702  
0703  
0704  
0705  
0706  
0707  
0708  
0709  
07010  
07011  
07012  
07013  
07014  
07015  
07016  
07017  
07018  
07019  
07020  
07021  
07022  
07023  
07024  
07025  
07026  
07027  
07028  
07029  
07030  
07031  
07032  
07033  
07034  
07035  
07036  
07037  
07038  
07039  
07040  
07041  
07042  
07043  
07044  
07045  
07046  
07047  
07048  
07049  
07050  
07051  
07052  
07053  
07054  
07055  
07056  
07057  
07058  
07059  
07060  
07061  
07062  
07063  
07064  
07065  
07066  
07067  
07068  
07069  
07070  
07071  
07072  
07073  
07074  
07075  
07076  
07077  
07078  
07079  
07080  
07081  
07082  
07083  
07084  
07085  
07086  
07087  
07088  
07089  
07090  
07091  
07092  
07093  
07094  
07095  
07096  
07097  
07098  
07099  
07100  
07101  
07102  
07103  
07104  
07105  
07106  
07107  
07108  
07109  
07110  
07111  
07112  
07113  
07114  
07115  
07116  
07117  
07118  
07119  
07120  
07121  
07122  
07123  
07124  
07125  
07126  
07127  
07128  
07129  
07130  
07131  
07132  
07133  
07134  
07135  
07136  
07137  
07138  
07139  
07140  
07141  
07142  
07143  
07144  
07145  
07146  
07147  
07148  
07149  
07150  
07151  
07152  
07153  
07154  
07155  
07156  
07157  
07158  
07159  
07160  
07161  
07162  
07163  
07164  
07165  
07166  
07167  
07168  
07169  
07170  
07171  
07172  
07173  
07174  
07175  
07176  
07177  
07178  
07179  
07180  
07181  
07182  
07183  
07184  
07185  
07186  
07187  
07188  
07189  
07190  
07191  
07192  
07193  
07194  
07195  
07196  
07197  
07198  
07199  
07200  
07201  
07202  
07203  
07204  
07205  
07206  
07207  
07208  
07209  
07210  
07211  
07212  
07213  
07214  
07215  
07216  
07217  
07218  
07219  
07220  
07221  
07222  
07223  
07224  
07225  
07226  
07227  
07228  
07229  
07230  
07231  
07232  
07233  
07234  
07235  
07236  
07237  
07238  
07239  
07240  
07241  
07242  
07243  
07244  
07245  
07246  
07247  
07248  
07249  
07250  
07251  
07252  
07253  
07254  
07255  
07256  
07257  
07258  
07259  
07260  
07261  
07262  
07263  
07264  
07265  
07266  
07267  
07268  
07269  
07270  
07271  
07272  
07273  
07274  
07275  
07276  
07277  
07278  
07279  
07280  
07281  
07282  
07283  
07284  
07285  
07286  
07287  
07288  
07289  
07290  
07291  
07292  
07293  
07294  
07295  
07296  
07297  
07298  
07299  
07300  
07301  
07302  
07303  
07304  
07305  
07306  
07307  
07308  
07309  
07310  
07311  
07312  
07313  
07314  
07315  
07316  
07317  
07318  
07319  
07320  
07321  
07322  
07323  
07324  
07325  
07326  
07327  
07328  
07329  
07330  
07331  
07332  
07333  
07334  
07335  
07336  
07337  
07338  
07339  
07340  
07341  
07342  
07343  
07344  
07345  
07346  
07347  
07348  
07349  
07350  
07351  
07352  
07353  
07354  
07355  
07356  
07357  
07358  
07359  
07360  
07361  
07362  
07363  
07364  
07365  
07366  
07367  
07368  
07369  
07370  
07371  
07372  
07373  
07374  
07375  
07376  
07377  
07378  
07379  
07380  
07381  
07382  
07383  
07384  
07385  
07386  
07387  
07388  
07389  
07390  
07391  
07392  
07393  
07394  
07395  
07396  
07397  
07398  
07399  
07400  
07401  
07402  
07403  
07404  
07405  
07406  
07407  
07408  
07409  
07410  
07411  
07412  
07413  
07414  
07415  
07416  
07417  
07418  
07419  
07420  
07421  
07422  
07423  
07424  
07425  
07426  
07427  
07428  
07429  
07430  
07431  
07432  
07433  
07434  
07435  
07436  
07437  
07438  
07439  
07440  
07441  
07442  
07443  
07444  
07445  
07446  
07447  
07448  
07449  
07450  
07451  
07452  
07453  
07454  
07455  
07456  
07457  
07458  
07459  
07460  
07461  
07462  
07463  
07464  
07465  
07466  
07467  
07468  
07469  
07470  
07471  
07472  
07473  
07474  
07475  
07476  
07477  
07478  
07479  
07480  
07481  
07482  
07483  
07484  
07485  
07486  
07487  
07488  
07489  
07490  
07491  
07492  
07493  
07494  
07495  
07496  
07497  
07498  
07499  
07500  
07501  
07502  
07503  
07504  
07505  
07506  
07507  
07508  
07509  
07510  
07511  
07512  
07513  
07514  
07515  
07516  
07517  
07518  
07519  
07520  
07521  
07522  
07523  
07524  
07525  
07526  
07527  
07528  
07529  
07530  
07531  
07532  
07533  
07534  
07535  
07536  
07537  
07538  
07539  
07540  
07541  
07542  
07543  
07544  
07545  
07546  
07547  
07548  
07549  
07550  
07551  
07552  
07553  
07554  
07555  
07556  
07557  
07558  
07559  
07560  
07561  
07562  
07563  
07564  
07565  
07566  
07567  
07568  
07569  
07570  
07571  
07572  
07573  
07574  
07575  
07576  
07577  
07578  
07579  
07580  
07581  
07582  
07583  
07584  
07585  
07586  
07587  
07588  
07589  
07590  
07591  
07592  
07593  
07594  
07595  
07596  
07597  
07598  
07599  
07600  
07601  
07602  
07603  
07604  
07605  
07606  
07607  
07608  
07609  
07610  
07611  
07612  
07613  
07614  
07615  
07616  
07617  
07618  
07619  
07620  
07621  
07622  
07623  
07624  
07625  
07626  
07627  
07628  
07629  
07630  
07631  
07632  
07633  
07634  
07635  
07636  
07637  
07638  
07639  
07640  
07641  
07642  
07643  
07644  
07645  
07646  
07647  
07648  
07649  
07650  
07651  
07652  
07653  
07654  
07655  
07656  
07657  
07658  
07659  
07660  
07661  
07662  
07663  
07664  
07665  
07666  
07667  
07668  
07669  
07670  
07671  
07672  
07673  
07674  
07675  
07676  
07677  
07678  
07679  
07680  
07681  
07682  
07683  
07684  
07685  
07686  
07687  
07688  
07689  
07690  
07691  
07692  
07693  
07694  
07695  
07696  
07697  
07698  
07699  
07700  
07701  
07702  
07703  
07704  
07705  
07706  
07707  
07708  
07709  
07710  
07711  
07712  
07713  
07714  
07715  
07716  
07717  
07718  
07719  
07720  
07721  
07722  
07723  
07724  
07725  
07726  
07727  
07728  
07729  
07730  
07731  
07732  
07733  
07734  
07735  
07736  
07737  
07738  
07739  
07740  
07741  
07742  
07743  
07744  
07745  
07746  
07747  
07748  
07749  
07750  
07751  
07752  
07753  
07754  
07755  
07756  
07757  
07758  
07759  
07760  
07761  
07762  
07763  
07764  
07765  
07766  
07767  
07768  
07769  
07770  
07771  
07772  
07773  
07774  
07775  
07776  
07777  
07778  
07779  
07780  
07781  
07782  
07783  
07784  
07785  
07786  
07787  
07788  
07789  
07790  
07791  
07792  
07793  
07794  
07795  
07796  
07797  
07798  
07799  
07800  
07801  
07802  
07803  
07804  
07805  
07806  
07807  
07808  
07809  
07810  
07811  
07812  
07813  
07814  
07815  
07816  
07817  
07818  
07819  
07820  
07821  
07822  
07823  
07824  
07825  
07826  
07827  
07828  
07829  
07830  
07831  
07832  
07833  
07834  
07835  
07836  
07837  
07838  
07839  
07840  
07841  
07842  
07843  
07844  
07845  
07846  
07847  
07848  
07849  
07850  
07851  
07852  
07853  
07854  
07855  
07856  
07857  
07858  
07859  
07860  
07861  
07862  
07863  
07864  
07865  
07866  
07867  
07868  
07869  
07870  
07871  
07872  
07873  
07874  
07875  
07876  
07877  
07878  
07879  
07880  
07881  
07882  
07883  
07884  
07885  
07886  
07887  
07888  
07889  
07890  
07891  
07892  
07893  
07894  
07895  
07896  
07897  
07898  
07899  
07900  
07901  
07902

054

- **Behaviour policy modelling via trajectory clustering.**  
055 By linking trajectory clustering in offline RL to task identification in meta RL, we propose  
056 an algorithm that learns a Gaussian mixture model of the dataset, enabling accurate density  
057 estimation in multimodal datasets.

058

- **A tight lower-bound-based value regulariser.**  
059 We theoretically derive a tight lower bound on the optimal action-value function and incor-  
060 porate it into a value regulariser.

061

## 062 2 BACKGROUND

063

064 **Notation** For the list of notations used in this paper and their meanings, refer to Appendix Sec. A.

065

066 **Markov Decision Process** An RL problem is formulated as a Markov Decision Process (MDP),  
067 which is defined as a 6-tuple  $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma, \rho_0 \rangle$ , where  $\mathcal{S} \subseteq \mathbb{R}^{d_s}$  is the state space,  $\mathcal{A} \subseteq \mathbb{R}^{d_a}$   
068 is the action space,  $P: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{S})$  is the transition dynamics,  $R: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{P}(\mathbb{R})$  is the  
069 reward function,  $\gamma \in [0, 1]$  is the discount factor, and  $\rho_0 \in \mathcal{P}(\mathcal{S})$  is the initial state distribution. We  
070 will assume that the support of  $R(s, a, s')$  is bounded above by  $r_{\max}$  and bounded below by  $r_{\min}$   
071 for all  $s, s' \in \mathcal{S}$  and  $a \in \mathcal{A}$ .

072 **Value Functions** Given a policy  $\pi$ , the Bellman operator  $\mathcal{T}^\pi$  on  $L^\infty(\mathcal{S} \times \mathcal{A})$  is defined by the  
073 following equation:

$$074 (\mathcal{T}^\pi Q)(s, a) = \mathbb{E}_{s' \sim P(s, a)} [\mathbb{E}_{r \sim R(s, a, s')} [r]] + \mathbb{E}_{s' \sim P(s, a)} [\mathbb{E}_{a' \sim \pi(s')} [Q(s', a')]].$$

075

076 Then, the action-value function (or Q-function)  $Q^\pi: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  is defined as the unique fixed  
077 point of  $\mathcal{T}^\pi$ , and the state-value function  $V^\pi: \mathcal{S} \rightarrow \mathbb{R}$  is given by  $V^\pi(s) = \mathbb{E}_{a \sim \pi(s)} [Q^\pi(s, a)]$ . The  
078 objective of RL is to find an optimal policy  $\pi^*$  such that  $V^{\pi^*} \succeq V^\pi$  for any policy  $\pi$ .

079 **Offline Reinforcement Learning** For offline RL, interactions with the environment is prohibited,  
080 and the agent has to learn a policy from a given dataset  $\mathcal{D}$  of trajectories. Throughout this paper,  
081 we will assume that each trajectory  $\tau \in \mathcal{D}$  is sampled with a uni-modal behaviour policy  $\beta \in$   
082  $\{\beta_0, \beta_1, \beta_2, \dots, \beta_{K-1}\}$ , where the candidate set  $\mathcal{B} = \{\beta_0, \beta_1, \beta_2, \dots, \beta_{K-1}\}$  is fixed but unknown  
083 to the agent.

## 084 3 MOTIVATION

085

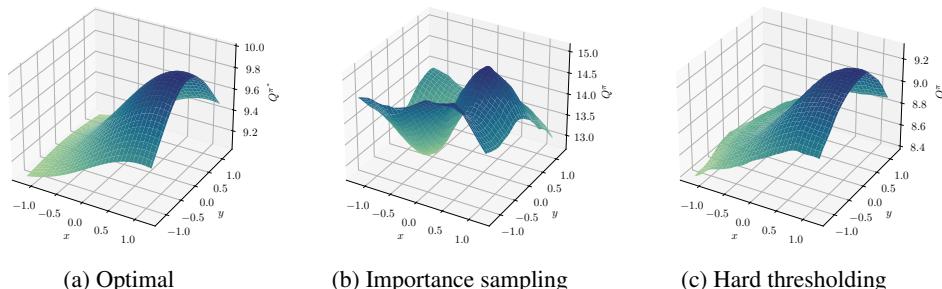
086 Critic penalization or value regularisation penalises the Q-values for OOD actions, while minimizing  
087 the temporal difference error for in-distribution (ID) actions. We may formulate it with the following  
088 equation

$$089 \min_Q \mathbb{E}_{(s, a) \sim \mathcal{D}} [(Q(s, a) - \mathcal{T}^\pi Q(s, a))^2] + \mathfrak{R},$$

090

091 where  $\mathfrak{R}$  is a regularizer. A crucial requirement of the regularizer is that it should be able to discrim-  
092 inate between ID and OOD actions since we only want to penalise the values of OOD actions. One

093



094  
095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107 Figure 1: The Q-values on the plane spanned by  $\mathbf{e}_1$  and  $\mathbf{e}_2$  estimated by each method, i.e.,  $Q(x\mathbf{e}_1 +$   
108  $y\mathbf{e}_2)$ . Due to the high variance of importance sampling ratios, the importance sampling method fails  
109 to approximate the optimal Q-function accurately.

108 of the first approaches was to set the regulariser as (Kumar et al., 2020)  
 109

$$110 \quad \mathfrak{R} = \mathbb{E}_{s \sim \mathcal{D}, a \sim \mu} [Q(s, a)] - \mathbb{E}_{s \sim \mathcal{D}, a \sim \beta} [Q(s, a)],$$

111 where  $\beta$  is the behaviour policy and  $\mu$  is some distribution that satisfies the condition  $\text{supp } \mu \subseteq$   
 112  $\text{supp } \beta$  (Kumar et al., 2020). While minimising the Q values for OOD actions sampled by  $\mu$ , they  
 113 simultaneously maximised the Q values for ID actions sampled from  $\beta$  to compensate for over-  
 114 penalisation. However, as Mao et al. (2023) points out, this approach has two shortcomings: (i)  
 115 the requirement  $\text{supp } \mu \subseteq \text{supp } \beta$  may not hold in general; and (ii) if the dataset contains a large  
 116 portion of suboptimal actions, their Q values would be overestimated. To address these issues, they  
 117 proposed an importance sampling (IS)-based method that utilises the following regulariser:  
 118

$$119 \quad \mathfrak{R}_{\text{IS}} = \mathbb{E}_{s \sim \mathcal{D}, a \sim \mu} [(Q(s, a) - Q_{\text{targ}}(s, a))^2] - \mathbb{E}_{s \sim \mathcal{D}, a \sim \beta} \left[ \frac{\mu(a | s)}{\beta(a | s)} (Q(s, a) - Q_{\text{targ}}(s, a))^2 \right],$$

120 where  $\mu$  is a probability distribution supported on the entire action space and  $Q_{\text{targ}}$  is a regulariser  
 121 target, which they set to  $r_{\min}/(1 - \gamma)$  for all  $s \in \mathcal{S}, a \in \mathcal{A}$ . Since the two terms cancel each other  
 122 on  $\text{supp } \beta$ ,  $\mathfrak{R}_{\text{IS}}$  is equivalent to  $\mathbb{E}_{s \sim \mathcal{D}, a \sim \mu} [\mathbf{1}_{\mathcal{A} \setminus \text{supp } \beta} (Q(s, a) - Q_{\text{targ}}(s, a))^2]$ , which corresponds  
 123 to the goal of penalising the Q values of OOD actions.  
 124

125 A significant drawback of  $\mathfrak{R}_{\text{IS}}$  is that IS ratios are  
 126 known to have high variance, especially for high-  
 127 dimensional spaces. Consider a simple single-state  
 128 infinite-horizon MDP with a six-dimensional action  
 129 space, and an offline RL dataset of size 1 000 000 sam-  
 130 pled from a behaviour policy  $\mathcal{N}(\mathbf{0}, \mathbf{I}_6)$ . Suppose the  
 131 optimal action is  $\mathbf{a}^* = \mathbf{e}_1$ . Then the IS ratio between  
 132  $\mu = \mathcal{N}(\mathbf{a}^*, 0.04\mathbf{I}_6)$  and  $\beta$  of the samples in the dataset  
 133 ranges from  $1.88 \times 10^{-225}$  to  $1.93 \times 10^4$ . As demon-  
 134 strated in Figure 1b and Table 1, the IS method yields an  
 135 inaccurate Q value estimation and a suboptimal policy  
 136 due to this severe fluctuation of IS ratios.

137 To overcome these limitations of the previous value regularization methods, we here propose to  
 138 explicitly identify the set of OOD actions  $\text{OOD}(s)$  for each state  $s \in \mathcal{S}$ , and set the regulariser to  
 139 zero for ID actions. Such hard thresholding (HT) allows a more stable training process, resulting in  
 140 a more accurate Q value estimations and better-performing policies, as seen in Fig. 1c and Table 1.  
 141

## 142 4 PROPOSED METHOD

144 This section is structured as follows. In Section 4.1, we first discuss how we can compute the set  
 145  $\text{OOD}(s)$ . We then propose a new lower bound of  $Q^{\pi^*}$  and show its effectiveness as a penalisation  
 146 target in Section 4.2. Finally, we provide a practical offline RL algorithm in Section 4.3.  
 147

### 148 4.1 IDENTIFYING THE OUT-OF-DISTRIBUTION ACTION SET

150 Likelihood is the most natural way to measure how OOD a particular sample is. However, choosing  
 151 the threshold value is not trivial. For blunt distributions, we should use a lower threshold value,  
 152 whereas for sharp distributions, we can choose a higher value. We propose a systematic method  
 153 of setting the threshold value by adopting the concept of *highest density region* (HDR; Hyndman  
 154 1996), which is basically a generalisation of a confidence interval to multivariate random variables.<sup>1</sup>

155 **Definition 1** (Hyndman 1996). Let  $f(X)$  be the pdf of a random variable  $X$ . Then, the  $100(1 - \alpha)\%$   
 156 *highest density region (HDR)* is the subset  $\mathcal{R}(f_\alpha)$  of the sample space of  $X$  such that  $\mathcal{R}(f_\alpha) = \{x : f(x) \geq f_\alpha\}$ , where  $f_\alpha = \sup\{y : \mathbb{P}(X \in \mathcal{R}(y)) \geq 1 - \alpha\}$ .  
 157

159 In the following subsections, we discuss how to compute the HDR under different assumptions.  
 160

161 <sup>1</sup>We provide a diagram (Figure 10) showing the  $100(1 - \alpha)\%$  HDR of a normal distribution on page 28 to  
 162 aid the understanding of the concept of a HDR.

162 4.1.1 HOMOGENEOUS DATASETS  
163

164 We first discuss the case when the offline dataset  $\mathcal{D}$  is homogeneous, that is, it was generated from a  
165 single uni-modal behaviour policy  $\beta$ . Then, we may obtain a fairly accurate Gaussian approximation  
166  $\hat{\beta}$  of  $\beta$  through behaviour cloning. Let  $\mu: \mathcal{S} \rightarrow \mathbb{R}^{d_a}$  and  $\Sigma: \mathcal{S} \rightarrow \mathbb{R}^{d_a \times d_a}$  be the mean and  
167 covariance matrix functions of  $\hat{\beta}$ , respectively. Assuming  $\Sigma(s)$  is positive definite for all  $s \in \mathcal{S}$ , the  
168  $100(1 - \alpha)\%$  HDR has the following closed-form representation (Proposition 4 in Appendix):

$$169 \mathcal{R}_{\hat{\beta}}(f_{\alpha}; s) = \left\{ \mathbf{x} \in \mathbb{R}^{d_a} : A_{\hat{\beta}}(\mathbf{x}; s) \leq F_{\chi_{d_a}^2}^{-1}(1 - \alpha) \right\},$$

170 where  $F_{\chi_{d_a}^2}^{-1}$  is the inverse cumulative distribution function of a chi-squared random variable with  $d_a$   
171 degrees of freedom and

$$172 A_{\hat{\beta}}(\mathbf{x}; s) = (\mathbf{x} - \mu(s))^{\top} \Sigma(s)^{-1} (\mathbf{x} - \mu(s)).$$

173 Choosing an appropriate value of  $0 < \alpha < 1$ , we can define  $\text{OOD}(s)$  as

$$174 \text{OOD}(s) = \mathcal{A} \setminus \mathcal{R}_{\hat{\beta}}(f_{\alpha}; s). \quad (1)$$

## 175 4.1.2 HETEROGENEOUS DATASETS

176 The definition of  $\text{OOD}(s)$  for a homogeneous dataset given in (1) can be generalised to the hetero-  
177 geneous case as

$$178 \text{OOD}(s) = \mathcal{A} \setminus \left( \bigcup_{\beta \in \mathcal{B}} \mathcal{R}_{\beta}(f_{\alpha}; s) \right)$$

179 for  $\mathcal{B}$ , where  $\mathcal{B}$  is the behaviour policy candidate set. If we could identify and isolate all of the  
180 trajectories in the dataset sampled from a particular behaviour policy  $\beta \in \mathcal{B}$ , then obtaining an  
181 estimation  $\hat{\beta}$  of  $\beta$  is straightforward by applying a behaviour cloning algorithm on those isolated  
182 trajectories. Then, with the estimated  $\hat{\mathcal{B}} = \{\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_{K-1}\}$ , we could compute  $\text{OOD}(s)$  for  
183 each state  $s$  as above. Therefore, in the rest of this section, we will propose how to cluster the  
184 trajectories. Note that the proposed clustering algorithm is useful not only for value regularization  
185 here but also for other offline real-world data analysis.

186 Our key idea is that the trajectory clustering problem closely resembles the task inference problem  
187 in meta RL. For each policy  $\pi$ , there is a corresponding Markov reward process (MRP)  $\mathcal{M}^{\pi} =$   
188  $\langle \mathcal{S}, P^{\pi}, R^{\pi}, \gamma \rangle$ , where for all  $s, s' \in \mathcal{S}$  and  $r \in \mathbb{R}$ , the transition probability function  $P^{\pi}: \mathcal{S} \rightarrow$   
189  $\mathcal{P}(\mathcal{S})$  and the reward probability function  $R^{\pi}: \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{P}(\mathbb{R})$  are defined by the equations

$$190 P^{\pi}(s' | s) = \mathbb{E}_{a \sim \pi(\cdot | s)} [P(s' | s, a)], \quad (2)$$

$$191 R^{\pi}(r | s, s') = \mathbb{E}_{a \sim \pi(\cdot | s)} [R(r | s, a, s')], \quad (3)$$

192 respectively. Since the dataset  $\mathcal{D}$  can then be viewed as a collection of trajectories, where each  
193 trajectory is sampled from one of the MRPs  $\mathcal{M}^{\beta_0}, \mathcal{M}^{\beta_1}, \mathcal{M}^{\beta_2}, \dots, \mathcal{M}^{\beta_{K-1}}$ , trajectory clustering  
194 task can be viewed as an MRP inference problem. As this formulation is almost equivalent to the  
195 MDP inference problem setting in meta RL, we infer the MRP instead of the MDP and apply a  
196 technique similar to variational Bayes-adaptive deep RL (variBAD; Zintgraf et al. 2021).

197 Our goal is to infer the behaviour policy index given a trajectory. To achieve this objective, we  
198 represent the index as a discrete latent variable  $m$  supported on  $[K] = \{0, 1, \dots, K-1\}$  and write

$$199 P^{\beta_m}(s) \approx P(s | m), \quad R^{\beta_m}(s, s') \approx R(s, s' | m), \quad \beta_m(s) \approx \beta(s | m),$$

200 for all  $s, s' \in \mathcal{S}$ , sharing  $P$ ,  $R$ , and  $\beta$  across trajectories. The marginal pdf of a trajectory  $\tau_{:T} =$   
201  $(s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \dots, s_{T-1}, a_{T-1}, r_{T-1}, s_T)$  is

$$202 p(\tau_{:T}) = \rho_0(s_0) \sum_{m=0}^{K-1} p(m) \prod_{t=0}^{T-1} P(s_{t+1} | s_t; m) \beta(a_t | s_t; m) R(r_t | s_t, s_{t+1}; m), \quad (4)$$

203 where  $p(m)$  is the prior distribution on  $m$ . Modelling  $P$ ,  $R$ , and  $\beta$  with neural networks  
204 parametrised by  $\theta$  results in a loss that depends on  $\theta$ . However, the multi-modality of (4) causes

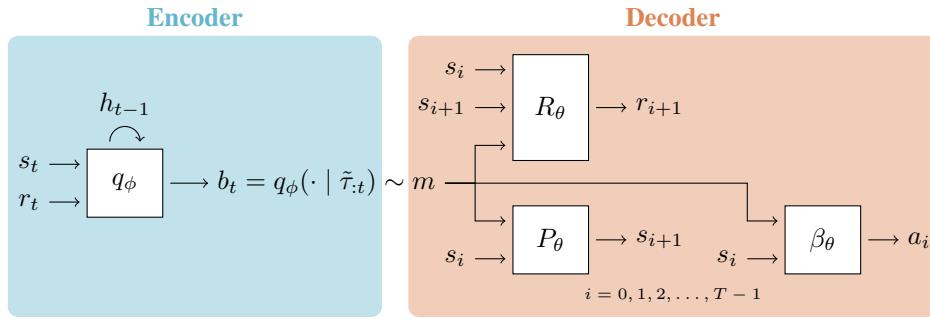


Figure 2: Overview of our architecture. We also provided a diagram of the variBAD architecture in Figure 11 for comparison.

gradient-based optimisation algorithms to produce sub-optimal solutions. We circumvent this issue by introducing amortised inference network  $q_\phi$  that takes a variable-length action-less trajectory  $\tilde{\tau}_{:t} = (s_0, r_0, s_1, r_1, s_2, r_2, \dots, s_{t-1}, r_{t-1}, s_t)$  as an input and outputs a distribution in  $\mathcal{P}_d([K])$ . Instead of maximising (4), we maximise the evidence lower bound (ELBO), which can be written by the following equation (Proposition 5):

$$\begin{aligned} \text{ELBO}_{\theta, \phi}(\tau; t) = & \\ & - D_{\text{KL}}(q_\phi(\tilde{\tau}_{:t}) \parallel p) + \sum_{i=0}^{T-1} \mathbb{E}_{m \sim q_\phi(\tilde{\tau}_{:t})} [\log R_\theta(r_i | s_i, s_{i+1}; m)] \\ & + \sum_{i=0}^{T-1} \mathbb{E}_{m \sim q_\phi(\tilde{\tau}_{:t})} [\log \beta_\theta(a_i | s_i; m)] + \sum_{i=0}^{T-1} \mathbb{E}_{m \sim q_\phi(\tilde{\tau}_{:t})} [\log P_\theta(s_{i+1} | s_i; m)]. \end{aligned} \quad (5)$$

The first term  $\log \rho_0(s_0)$  in (9) can be omitted because it is constant with respect to  $\theta$  and  $\phi$ . The final objective for trajectory clustering is to maximise

$$\mathbb{E}_{\tau \sim \mathcal{D}} \left[ \frac{1}{T_\tau} \sum_{t=0}^{T_\tau-1} \text{ELBO}_{\theta, \phi}(\tau; t) \right], \quad (6)$$

where  $T_\tau$  is the length of the trajectory  $\tau$  sampled from the dataset. An overview of our clustering algorithm is given in Figure 2.

After we finish training, we compute the behaviour policy estimations and cluster assignments according to the equations  $\hat{\beta}_i = \beta_\theta(\cdot; i)$  and  $\mathbb{A}(s) = \arg \max_m q_\phi(m | \tilde{\tau}(s))$ , respectively, for each  $i \in [K]$  and  $s \in \mathcal{D}$ , where  $\tilde{\tau}(s)$  is the action-less trajectory containing the state  $s$ .

## 4.2 LOWER BOUND PENALISATION

In this section, we derive a new lower bound on the value function. As we did in the previous section, we start with the case where the offline dataset  $\mathcal{D}$  is generated from a single behaviour policy  $\beta$ . The ideal penalisation method would be to use  $Q^{\pi^*}(s, a)$  as a target, where  $\pi^*$  is the optimal policy, but the value of  $Q^{\pi^*}$  is inaccessible. So we aim to use a lower bound instead. In order to compute a lower bound, we first need to make some assumptions on the regularity of  $P$  and  $V^\beta$ .

**Assumption 1.** There is  $K_P > 0$  such that for all  $s \in \mathcal{S}$  and  $a, a' \in \mathcal{A}$ ,  $W_1(P(s, a), P(s, a')) < K_P \|a - a'\|$ , where  $W_1(P, Q)$  is the Wasserstein distance of order 1 between two probability distributions  $P, Q \in \mathcal{P}(\mathcal{S})$ .

**Assumption 2.** The value function of the behaviour policy  $\beta$  is  $K_V$ -Lipschitz, that is, for all  $s, s' \in \mathcal{S}$ ,  $|V^\beta(s) - V^\beta(s')| < K_V \|s - s'\|$ .

Then, we can obtain a lower bound of  $Q^{\pi^*}$  with these assumptions.

**Proposition 1.** Define  $Q_\beta^{\text{LB}}: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  by the equation

$$Q_\beta^{\text{LB}}(s, a) = \max \left\{ V^\beta(s) - r_{\max} + r_{\min} - \gamma K_V K_P \mathbb{E}_{a' \sim \beta(s)} [\|a - a'\|], \frac{r_{\min}}{1 - \gamma} \right\}. \quad (7)$$

270 For any policy  $\pi: \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$  such that  $V^\pi \succeq V^\beta, Q^\pi \succeq Q_\beta^{\text{LB}}$ .  
 271

272 *Proof.* See page 15. □  
 273

274 Note that this lower bound is tighter than the previous bound  $r_{\min}/(1 - \gamma)$ . The lower bound allows  
 275 us to define the penalised Bellman optimality operator  $\mathcal{T}_\beta^\pi$  for policy  $\pi$  by the equation  
 276

$$277 \quad (\mathcal{T}_\beta^* Q)(s, a) = \begin{cases} Q_\beta^{\text{LB}}(s, a) & \text{if } a \in \text{OOD}(s), \\ (\mathcal{T}^* Q)(s, a) & \text{otherwise,} \end{cases}$$

279 where  $\mathcal{T}^*$  is the Bellman optimality operator defined as  
 280

$$281 \quad (\mathcal{T}^* Q)(s, a) = \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} \left[ r + \gamma \sup_{a' \in \mathcal{A}} Q(s', a') \right].$$

284 We can show that through repeated application of  $\mathcal{T}_\beta^*$ , it is possible to obtain a deterministic policy  
 285  $\pi_\beta^*: \mathcal{S} \rightarrow \mathcal{A}$  that is optimal among the policies whose action for each state  $s \in \mathcal{S}$  does not lie in  
 286  $\text{OOD}(s)$ .

287 **Theorem 2.** Any initial bounded real-valued function on  $\mathcal{S} \times \mathcal{A}$  can converge to a unique fixed point  
 288  $Q_\beta^*$  by repeatedly applying  $\mathcal{T}_\beta^*$ . Suppose for each  $s \in \mathcal{S}$ ,

$$289 \quad Q^\beta(s, a_s) \geq \mathbb{E}_{a \sim \beta(s)} [Q^\beta(s, a)]$$

291 for some  $a_s \in \mathcal{A} \setminus \text{OOD}(s)$ . If there exists a deterministic policy  $\pi_\beta^*: \mathcal{S} \rightarrow \mathcal{A}$  that is optimal under  
 292 the constraint  $\pi(s) \notin \text{OOD}(s)$  for all  $s \in \mathcal{S}$ , then  $\pi_\beta^*(s) = \arg \max_{a \in \mathcal{A}} Q_\beta^*(s, a)$  for all  $s \in \mathcal{S}$ .  
 293

294 *Proof.* See page 20. □  
 295

296 Now, the penalised Bellman optimality operator can easily be generalised to the heterogeneous  
 297 dataset case with the set  $\mathcal{B}$  of behaviour policy candidates and the set  $\mathcal{V}(s)$  of valid behaviour policies  
 298 given a state  $s \in \mathcal{S}$ .

$$299 \quad (\mathcal{T}_\mathcal{B}^* Q)(s, a) = \begin{cases} Q_\mathcal{B}^{\text{LB}}(s, a) & \text{if } a \in \text{OOD}(s), \\ (\mathcal{T}^* Q)(s, a) & \text{otherwise,} \end{cases}$$

301 where  $Q_\mathcal{B}^{\text{LB}}: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  is defined as  $Q_\mathcal{B}^{\text{LB}}(s, a) = \max_{\beta \in \mathcal{V}(s)} Q_\beta^{\text{LB}}(s, a)$  for each  $s \in \mathcal{S}$  and  
 302  $a \in \mathcal{A}$ . We can prove a similar performance guarantee for the policy obtained by repeatedly applying  
 303  $\mathcal{T}_\mathcal{B}^*$ .  
 304

305 **Theorem 3.** Any initial bounded real-valued function on  $\mathcal{S} \times \mathcal{A}$  can converge to a unique fixed point  
 306  $Q_\mathcal{B}^*$  by repeatedly applying  $\mathcal{T}_\mathcal{B}^*$ . Suppose for each  $\beta \in \mathcal{B}$  and  $s \in \mathcal{S}$ ,

$$307 \quad Q^\beta(s, a_s^\beta) \geq \mathbb{E}_{a \sim \beta(s)} [Q^\beta(s, a)]$$

309 for some  $a_s^\beta \in \mathcal{A} \setminus \text{OOD}(s)$ . If there exists a deterministic policy  $\pi_\mathcal{B}^*: \mathcal{S} \rightarrow \mathcal{A}$  that is optimal under  
 310 the constraint  $\pi(s) \notin \text{OOD}(s)$  for all  $s \in \mathcal{S}$ , then  $\pi_\mathcal{B}^*(s) = \arg \max_{a \in \mathcal{A}} Q_\mathcal{B}^*(s, a)$  for all  $s \in \mathcal{S}$ .  
 311

312 *Proof.* See page 21. □  
 313

### 4.3 PRACTICAL ALGORITHM

315 The overall flow of our algorithm is as follows:  
 316

- 317 **I. Behaviour policy learning.** Run the trajectory clustering algorithm to obtain  $\hat{\mathcal{B}}$  and  $\hat{\mathcal{V}}(s)$ . Or if  
 318 it is known a priori that the dataset is homogeneous, then run a behaviour cloning algorithm to  
 319 obtain  $\hat{\beta}$ .
- 320 **II. Behaviour value learning.** Learn a value function  $\hat{V}^{\hat{\beta}}$  for each  $\hat{\beta} \in \hat{\mathcal{B}}$  through temporal differ-  
 321 ence learning.
- 322 **III. Policy learning.** Obtain and apply  $\mathcal{T}_{\hat{\mathcal{B}}}^*$  repeatedly on a randomly initialised Q-function until  
 323 convergence. Find a policy that maximises the learned Q-function.

324 This section mainly focuses on the trajectory clustering algorithm of Stage I. Additional details of  
 325 our algorithm can be found in Section C of Appendix.

326  
 327 The network architecture used for trajectory clustering consists of three parts: the encoder, the latent  
 328 sampler, and the decoder. The architecture is generally similar to that of variBAD except for a few  
 329 adaptations. In this section, we will first go over how and why we modified each part. Then, we will  
 330 propose a simple technique to adaptively set the number of clusters.

331 The encoder needs to take an action-less trajectory  $\tilde{\tau}$  as an input and output the amortised poste-  
 332 rior. Since the length of the trajectory may vary from one to another, the network should be capable  
 333 of taking variable-length sequence as its input. For that purpose, variBAD utilises gated recurrent  
 334 units (GRU; Cho et al. 2014). GRUs and other recurrent neural network variants suffer from the van-  
 335 ishing gradient problem (Bengio et al., 1994), which hampers their ability to process long sequences.  
 336 Truncated backpropagation through time (Williams & Peng, 1990) can mitigate the phenomena to  
 337 a certain extent, but we instead adopt the state space model architecture that is recently gaining in-  
 338 terest in the area of sequence modelling (Gu et al., 2020; 2021; 2022; Gu & Dao, 2023; Dao & Gu,  
 339 2024). In particular, we use the S5 layer (Smith et al., 2023), which is simple and computationally  
 340 efficient.

341 The second modification was made on the way latents are sampled and ELBOs are computed. As  
 342 the latent variable in the variBAD architecture is continuous, it is impossible to analytically compute  
 343 the expectation, and hence, the reparametrisation trick (Kingma & Welling, 2014) must be used.  
 344 Although the latent variable is discrete in our case, exact computation is still inefficient because it  
 345 requires multiple forward and backward passes through the decoder. We instead utilise the vector  
 346 quantised-variational autoencoder (VQ-VAE; van den Oord et al. 2017) to approximate the ELBO.  
 347 Under the VQ-VAE formulation, the amortised posterior  $q_\phi$  is modelled as

$$q_\phi(m = k \mid \tilde{\tau}_{:t}) = \begin{cases} 1 & \text{if } k = k_\phi(\tilde{\tau}_{:t}) \\ 0 & \text{otherwise,} \end{cases}$$

350 where  $e_0, e_1, \dots, e_{K-1}$  are latent embedding vectors and

$$k_\phi(\tilde{\tau}_{:t}) = \arg \min_{j \in [K]} \|q_\phi(\tilde{\tau}_{:t}) - e_k\|_2.$$

354 Note that for simplicity, we have abused the notation  $q_\phi$  to denote both the posterior and the encoder.  
 355 The gradient flows into the encoder  $q_\phi$  via the loss function

$$\ell_{\text{VQ}}(\phi; \tilde{\tau}_{:t}) = \|q_\phi(\tilde{\tau}_{:t}) - e_{k_\phi(\tilde{\tau}_{:t})}\|_2^2$$

358 and the latent embedding vectors are updated with exponential moving averages (EMA).

359  
 360 For the decoder, we use Gaussian distributions with diagonal covariance matrix to represent  $P_\theta$ ,  
 361  $\beta_\theta$ , and  $R_\theta$ . Most RL environments have a bounded action space, whereas a Gaussian distribution  
 362 has unbounded support. To estimate the behaviour policy more accurately, we first normalize the  
 363 actions between  $-1$  and  $1$  and apply the inverse hyperbolic tangent function on each dimension of  
 364 the actions to map them onto  $\mathbb{R}^{d_a}$ . Note that we use the mapped actions when learning the critic, that  
 365 is, the critic function takes  $\tanh^{-1}(a)$  instead of  $a$  as input. Finally, instead of taking the summation  
 366 over the entire trajectory in (5) and (6), we adopt the implementation trick of variBAD and randomly  
 367 subsample  $N_d$  transition steps in (5) and  $N_e$  ELBO terms in (6). To conclude, the loss function for  
 368 the trajectory clustering algorithm is

$$\ell_{\text{TC}}(\theta, \phi; \tau) = \frac{1}{N_e N_d} \sum_{t \in \mathcal{I}_e} \sum_{i \in \mathcal{I}_d} A_\theta(s_i, a_i, r_i, s_{i+1}; e_{k_\phi(\tilde{\tau}_{:t})}) + \lambda_{\text{VQ}} \ell_{\text{VQ}}(\phi; \tilde{\tau}_{:t}),$$

371 where

$$A_\theta(s, a, r, s'; m) = \log \beta_\theta(a \mid s; m) + \lambda_T \log P_\theta(s' \mid s; m) + \lambda_R \log R_\theta(r \mid s, s'; m), \quad (8)$$

373  
 374  $\mathcal{I}_e$  and  $\mathcal{I}_d$  are sets of indices sampled uniformly at random with replacement from  $[T_\tau]$  with sizes  
 375  $N_e$  and  $N_d$ , respectively, and  $\lambda_{\text{VQ}}$ ,  $\lambda_T$ ,  $\lambda_R$  are tunable hyperparameters.

377 Choosing the right number of clusters is crucial for high performance in most clustering algorithms.  
 To alleviate the burden of hyperparameter tuning, we adopt a two-phase training paradigm. During

378 Table 2: Average normalised scores on the D4RL benchmark. Note that “ha” means halfcheetah,  
 379 “ho” means hopper, “wa” means walker2d, “m” means medium, “r” means replay, “ra” means  
 380 random, and “e” means expert.

Dataset	BC	TD3BC	BCQ	BEAR	CQL	IQL	MCQ	SVR	Ours
ha-ra	2.6	11.0	2.2	2.3	17.5	13.1	28.5	27.2	27.0 ± 1.1
ho-ra	4.1	8.5	7.8	3.9	7.9	7.9	31.8	31.0	31.5 ± 0.2
wa-ra	1.2	1.6	4.9	12.8	5.1	5.4	17.0	2.2	16.6 ± 7.9
ha-m	42.0	48.3	46.6	43.0	47.0	47.4	64.3	60.5	63.5 ± 1.2
ho-m	56.2	59.3	59.4	51.8	53.0	66.2	78.4	103.5	103.4 ± 0.9
wa-m	71.0	83.7	71.8	-0.2	73.3	78.3	91.0	92.4	96.5 ± 13.9
ha-m-r	36.4	44.6	42.2	36.3	45.5	44.2	56.8	52.5	52.2 ± 0.8
ho-m-r	21.8	60.9	60.9	52.2	88.7	94.7	101.6	103.7	102.2 ± 1.1
wa-m-r	24.9	81.8	57.0	7.0	81.8	73.8	91.3	95.6	95.4 ± 19.2
ha-m-e	59.6	90.7	95.4	46.0	75.6	86.7	87.5	94.2	90.9 ± 4.2
ho-m-e	51.7	98.0	106.9	50.6	105.6	91.5	111.2	111.2	112.4 ± 1.1
wa-m-e	101.2	110.1	107.7	22.1	107.9	109.6	114.2	109.3	108.3 ± 0.7
ha-e	88.2	81.7	92.7	92.9	96.3	95.0	96.2	96.1	96.6 ± 0.9
ho-e	110.9	107.8	109.0	54.6	96.5	109.4	111.4	111.1	112.7 ± 0.9
wa-e	107.7	110.2	106.3	106.6	108.5	109.9	107.2	110.0	113.4 ± 0.5
Average	52.3	67.5	64.5	38.8	67.3	68.9	79.2	80.0	81.5

402  
 403 the first phase of the paradigm, we set the codebook size to be sufficiently large. After completing the  
 404 first phase, we compute the cluster assignments for each state in the dataset. If the number of states  
 405 assigned to a particular cluster does not exceed a certain threshold, we remove the corresponding  
 406 code from the VQ-VAE codebook. The training is resumed with the remaining codebook. This way,  
 407 we could adaptively determine the number of clusters without needing to perform an exhaustive  
 408 hyperparameter search.  
 409

## 410 5 EXPERIMENTS

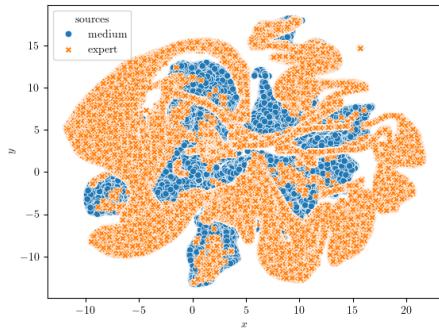
### 413 5.1 RESULTS ON THE D4RL BENCHMARK

414 In order to evaluate how well our algorithm perform on various offline RL tasks, we tested our  
 415 method on the D4RL (Fu et al., 2020) benchmark. We compared it with existing offline RL methods  
 416 such as BC (Pomerleau, 1988), TD3+BC (Fujimoto & Gu, 2021), BCQ (Fujimoto et al., 2019),  
 417 CQL (Kumar et al., 2020), BEAR (Kumar et al., 2019), IQL (Kostrikov et al., 2022), MCQ (Lyu  
 418 et al., 2022), and SVR (Mao et al., 2023). We trained our method with five different seeds to obtain  
 419 five different policies and sampled ten trajectories with each of them. We report the average and  
 420 standard deviation of the fifty normalized scores in Table 2. The results show that our algorithm  
 421 can successfully learn high-performing policies from most datasets, while attaining state-of-the-art  
 422 scores on some of them.

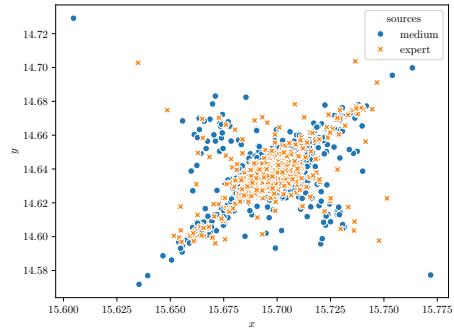
### 424 5.2 EXPERIMENTS ON A HETEROGENEOUS DATASET

425 Although D4RL datasets such as “hopper-medium-expert-v2” were sampled with more than one  
 426 behaviour policies, the action distributions are actually unimodal on most states due to the state dis-  
 427 tribution being so different between the trajectories of the two behaviour policies. Figure 3 presents  
 428 a visualisation of the entire and initial state distributions of the “hopper-medium-expert-v2” dataset  
 429 where we have used the uniform manifold approximation and projection (UMAP; McInnes & Healy  
 430 2018) technique for dimension reduction. We can see that expert and medium states are clearly  
 431 separated, except for the initial states.

432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443



(a) Entire state distribution



(b) Initial state distribution

444  
445  
446  
447  
448  
Figure 3: The UMAP of the states in the “hopper-medium-expert-v2” dataset.

449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
To demonstrate the effectiveness of our trajectory clustering algorithm, we created a custom dataset with drastically different initial state behaviours using the “Hopper-v5” environment provided by the Gymnasium library (Towers et al., 2024). Half of the samples in the dataset were sampled from an expert policy, and the other half was sampled from a policy that tripped over within eight timesteps. Table 3 demonstrates that our method can effectively classify the two datasets and learn an optimal policy from a truly heterogeneous dataset.

460

461  
5.3 ANALYSIS ON THE TRAJECTORY CLUSTERING ALGORITHM

462

463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
In meta reinforcement learning settings, each MDP has independent transition and reward dynamics, so they must be modelled in order to infer the MDP from trajectories. Under our formulation, on the other hand, transition and reward dynamics of each MRP are correlated with each other through the policy as we can see from (2) and (3). Although this implies that we may identify the MRP solely through modelling the behaviour policy, we hypothesized that modelling transition and reward dynamics can provide meaningful auxiliary information leading to better clustering performance. Therefore, we compared the performance of our algorithm under four different configurations  $(\lambda_T, \lambda_R) \in \{(1, 1), (1, 0), (0, 1), (0, 0)\}$ , where  $\lambda_T$  and  $\lambda_R$  are the weights for transition and reward models defined in (8). To evaluate the accuracy of our trajectory clustering algorithm, we created custom D4RL datasets by concatenating random, medium, and expert datasets. The mean and standard deviation of adjusted rand indices (ARI; Hubert & Arabie 1985) and normalised mutual information scores (NMI) for each configuration over 5 different seeds are reported in Table 4. We can see that the configuration  $(\lambda_T, \lambda_R) = (1, 0)$  performs the best on average. Unlike  $s_{i+1}$ , which is in the vicinity of  $s_i$  regardless of the  $a_i$ ,  $r_i$  can vary drastically between policies, making it difficult to model rewards from different policies with a single neural network. We speculate this to be the reason why training a reward model negatively affects the performance of our trajectory clustering algorithm. For experiments on other datasets, refer to Section D.2.

449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
Table 3: The performance of SVR and our method on the custom heterogeneous dataset.

Algorithm	Length	Return
SVR	$8.00 \pm 0.00$	$4.05 \pm 0.01$
Ours	$436.3 \pm 32.1$	$4062.0 \pm 24.5$

460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
Table 4: The impact of hyperparameters  $\lambda_T$  and  $\lambda_R$  on the average performance of our trajectory clustering algorithm evaluated on six custom D4RL datasets. The performance is measured in terms of adjusted rand index (ARI) and normalised mutual information score (NMI).

$\lambda_R$	$\lambda_T$	ARI	NMI
0	0	$0.98 \pm 0.07$	$0.98 \pm 0.06$
0	1	$0.91 \pm 0.21$	$0.92 \pm 0.17$
1	0	$0.99 \pm 0.02$	$0.98 \pm 0.02$
1	1	$0.86 \pm 0.27$	$0.87 \pm 0.24$

486  
 487 Table 5: The impact of the hyperparameter  $\alpha$  on the average normalised score on three different  
 488 datasets in the D4RL benchmark. Note that “ha” means halfcheetah, “ho” means hopper, “wa”  
 489 means walker2d, and “m” means medium.

Dataset	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$
ha-m	63.2 $\pm$ 1.1	62.4 $\pm$ 1.4	63.5 $\pm$ 1.2	63.3 $\pm$ 1.0	62.5 $\pm$ 1.2
ho-m	102.2 $\pm$ 0.5	103.3 $\pm$ 0.9	103.4 $\pm$ 0.9	102.4 $\pm$ 0.7	103.1 $\pm$ 0.5
wa-m	93.0 $\pm$ 22.9	91.3 $\pm$ 22.4	96.1 $\pm$ 13.9	94.2 $\pm$ 20.4	97.0 $\pm$ 15.2

496  
 497 Table 6: The impact of the hyperparameter  $K$  on the average normalised score on three different  
 498 datasets in the D4RL benchmark. Note that “ha” means halfcheetah, “ho” means hopper, “wa”  
 499 means walker2d, and “m” means medium.

Dataset	$K = 0.1$	$K = 0.2$	$K = 0.5$	$K = 1.0$	$K = 2.0$
ha-m	62.5 $\pm$ 0.7	63.1 $\pm$ 1.4	63.5 $\pm$ 1.2	63.4 $\pm$ 0.9	63.1 $\pm$ 1.1
ho-m	102.1 $\pm$ 2.8	102.6 $\pm$ 0.7	102.2 $\pm$ 0.8	102.0 $\pm$ 5.7	103.4 $\pm$ 0.9
wa-m	86.9 $\pm$ 27.6	90.6 $\pm$ 25.0	93.1 $\pm$ 26.7	96.1 $\pm$ 13.9	93.9 $\pm$ 15.6

Dataset	$K = 5.0$	$K = 10.0$	$K = 20.0$	$K = 50.0$	$K = 100.0$
ha-m	63.3 $\pm$ 1.3	62.8 $\pm$ 0.8	63.2 $\pm$ 1.2	62.6 $\pm$ 1.0	62.1 $\pm$ 1.2
ho-m	102.3 $\pm$ 3.9	102.5 $\pm$ 1.5	100.0 $\pm$ 6.8	84.1 $\pm$ 20.7	92.5 $\pm$ 21.1
wa-m	92.3 $\pm$ 1.6	89.5 $\pm$ 1.5	88.0 $\pm$ 1.6	81.7 $\pm$ 9.1	78.4 $\pm$ 13.7

#### 518 5.4 ABLATION STUDY

520 We investigate the impact of the choice of hyperparameters  $\alpha$  and  $K = K_V K_P$  on the performance  
 521 of our method on three different datasets: halfcheetah-medium-v2, hopper-medium-v2, and  
 522 walker2d-medium-v2. As shown in Table 5, the performance is robust to a wide range of  $\alpha$  values.  
 523 Similarly, we can see from Table 6 that the performance remains stable for moderate choices of  
 524  $K$ . While large values ( $K \geq 50$ ) lead to degradation, particularly for hopper-medium-v2. Overall,  
 525 the result indicate that our method does not require precise tuning on  $\alpha$  and  $K$  to achieve strong  
 526 performance.

## 530 6 CONCLUSION

533 In this paper, we propose a new value regularisation algorithm for offline RL penalizing their critic  
 534 values, based on the OOD action set that we were able to explicitly identify. We determine how OOD  
 535 an action is based on its likelihood, where the threshold is set adaptively according to the shape of  
 536 the behaviour policy. To enable likelihood analysis for heterogeneous datasets where simple be-  
 537 haviour cloning fails, we introduce a novel trajectory clustering technique based on a meta-learning  
 538 formulation of the clustering problem. Our method of penalising the critic values for OOD actions  
 539 by regressing them towards a lower bound of the optimal Q-value function is proven to be effective  
 both theoretically and empirically.

540 REFERENCES  
541

542 Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. Learning long-term dependencies with  
543 gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994.

544 James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal  
545 Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao  
546 Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

547 Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülcehre, Dzmitry Bahdanau, Fethi Bougares, Hol-  
548 ger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder  
549 for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods  
550 in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of  
551 SIGDAT, a Special Interest Group of the ACL*, pp. 1724–1734. ACL, 2014.

552 Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through  
553 structured state space duality. In *Forty-first International Conference on Machine Learning, ICML  
554 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

555 Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep  
556 data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020.

557 Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In  
558 *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Infor-  
559 mation Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 20132–20145,  
560 2021.

561 Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without  
562 exploration. In *Proceedings of the 36th International Conference on Machine Learning, ICML  
563 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine  
564 Learning Research*, pp. 2052–2062. PMLR, 2019.

565 Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*,  
566 abs/2312.00752, 2023.

567 Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory  
568 with optimal polynomial projections. In *Advances in Neural Information Processing Systems 33:  
569 Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December  
570 6-12, 2020, virtual*, 2020.

571 Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Com-  
572 bining recurrent, convolutional, and continuous-time models with linear state space layers. In  
573 *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Infor-  
574 mation Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 572–585, 2021.

575 Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured  
576 state spaces. In *The Tenth International Conference on Learning Representations, ICLR 2022,  
577 Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

578 Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas  
579 Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2024.

580 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances  
581 in Neural Information Processing Systems 33: Annual Conference on Neural Information Pro-  
582 cessing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

583 Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218,  
584 Dec 1985. ISSN 1432-1343.

585 Rob J. Hyndman. Computing and graphing highest density regions. *The American Statistician*, 50  
586 (2):120–126, 1996.

587 Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International  
588 Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014,  
589 Conference Track Proceedings*, 2014.

594 Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-  
 595 learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual*  
 596 *Event, April 25-29, 2022*. OpenReview.net, 2022.

597

598 Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy  
 599 q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Sys-*  
 600 *tems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019,*  
 601 *December 8-14, 2019, Vancouver, BC, Canada*, pp. 11761–11771, 2019.

602 Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline  
 603 reinforcement learning. In *Advances in Neural Information Processing Systems 33: Annual Con-*  
 604 *ference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020,*  
 605 *virtual*, 2020.

606

607 Jiachen Li, Edwin Zhang, Ming Yin, Qinxun Bai, Yu-Xiang Wang, and William Yang Wang. Offline  
 608 reinforcement learning with closed-form policy improvement operators. In *International Confer-*  
 609 *ence on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202  
 610 *of Proceedings of Machine Learning Research*, pp. 20485–20528. PMLR, 2023.

611

612 Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa,  
 613 David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *4th*  
 614 *International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May*  
 615 *2-4, 2016, Conference Track Proceedings*, 2016.

616

617 Jianlan Luo, Perry Dong, Jeffrey Wu, Aviral Kumar, Xinyang Geng, and Sergey Levine. Action-  
 618 quantized offline reinforcement learning for robotic skill learning. In *Conference on Robot Learn-*  
 619 *ing, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine*  
 620 *Learning Research*, pp. 1348–1361. PMLR, 2023.

621

622 Jiafei Lyu, Xiaoteng Ma, Xiu Li, and Zongqing Lu. Mildly conservative q-learning for offline  
 623 reinforcement learning. In *Advances in Neural Information Processing Systems 35: Annual Con-*  
 624 *ference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA,*  
 625 *November 28 - December 9, 2022*, 2022.

626

627 Yihuan Mao, Chengjie Wu, Xi Chen, Hao Hu, Ji Jiang, Tianze Zhou, Tangjie Lv, Changjie Fan,  
 628 Zhipeng Hu, Yi Wu, Yujing Hu, and Chongjie Zhang. Stylized offline reinforcement learning:  
 629 Extracting diverse high-quality behaviors from heterogeneous datasets. In *The Twelfth Interna-*  
 630 *tional Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.  
 631 OpenReview.net, 2024.

632

633 Yixiu Mao, Hongchang Zhang, Chen Chen, Yi Xu, and Xiangyang Ji. Supported value regulariza-  
 634 tion for offline reinforcement learning. In *Advances in Neural Information Processing Systems*  
 635 *36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New*  
 636 *Orleans, LA, USA, December 10 - 16, 2023*, 2023.

637

638 Leland McInnes and John Healy. UMAP: uniform manifold approximation and projection for di-  
 639 *mension reduction*. *CoRR*, abs/1802.03426, 2018.

640

641 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Pretten-  
 642 hofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and  
 643 E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*,  
 644 12:2825–2830, 2011.

645

646 Dean Pomerleau. ALVINN: an autonomous land vehicle in a neural network. In *Advances in Neural*  
 647 *Information Processing Systems 1, [NIPS Conference, Denver, Colorado, USA, 1988]*, pp. 305–  
 648 313. Morgan Kaufmann, 1988.

649

650 Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers  
 651 for sequence modeling. In *The Eleventh International Conference on Learning Representations,*  
 652 *ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

648 Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 2256–2265. JMLR.org, 2015.

652 Mark Towers, Ariel Kwiatkowski, Jordan K. Terry, John U. Balis, Gianluca De Cola, Tristan Deleu,  
 653 Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, An-  
 654 drea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A  
 655 standard interface for reinforcement learning environments. *CoRR*, abs/2407.17032, 2024.

657 Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learn-  
 658 ing. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural*  
 659 *Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6306–  
 660 6315, 2017.

661 Cédric Villani. *Cyclical monotonicity and Kantorovich duality*, pp. 51–92. Springer Berlin Hei-  
 662 delberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-71050-9.

664 Qiang Wang, Yixin Deng, Francisco Roldan Sanchez, Keru Wang, Kevin McGuinness, Noel E.  
 665 O’Connor, and Stephen J. Redmond. Dataset clustering for improved offline policy learning.  
 666 *CoRR*, abs/2402.09550, 2024.

667 Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy  
 668 class for offline reinforcement learning. In *The Eleventh International Conference on Learning*  
 669 *Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

671 Ronald J. Williams and Jing Peng. An efficient gradient-based algorithm for on-line training of  
 672 recurrent network trajectories. *Neural Comput.*, 2(4):490–501, 1990.

673 Luisa M. Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin  
 674 Gal, Katja Hofmann, and Shimon Whiteson. Varibad: Variational bayes-adaptive deep RL via  
 675 meta-learning. *J. Mach. Learn. Res.*, 22:289:1–289:39, 2021.

## 678 A NOTATIONS

- 680 •  $\mathbf{0}$ : a zero vector with dimensionality implied by context
- 681 •  $D_{\text{KL}}(P_1 \parallel P_2)$ : the Kullback–Leibler (KL) divergence from a probability distribution  $P_1$   
 682 to another probability distribution  $P_2$
- 683 •  $\mathbf{e}_i$ : the  $i$ -th standard basis of a Euclidean space
- 684 •  $f(y \mid x)$ : the value of the pdf (or pmf) of the distribution  $f(x)$  at  $y$ , where  $Y$  is a set and  
 685  $f: X \rightarrow \mathcal{P}(Y)$
- 686 •  $f \succeq g$ :  $f(x) \geq g(x)$  for all  $x \in X$ , where  $f$  and  $g$  are real-valued functions defined on a  
 687 set  $X$
- 688 •  $f \equiv g$ :  $f(x) = g(x)$  for all  $x \in X$ , where  $f$  and  $g$  are real-valued functions defined on a  
 689 set  $X$
- 690 •  $\mathbf{I}_d$ : an identity matrix with  $d$  rows and  $d$  columns
- 691 •  $L^\infty(X)$ : the space of bounded real value functions on a set  $X$  endowed with the supremum  
 692 norm
- 693 •  $[N]$ : the set  $\{0, 1, \dots, N - 1\}$ , where  $N$  is an integer
- 694 •  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ : a multi-variate Gaussian distribution with mean vector  $\boldsymbol{\mu}$  and covariance matrix  
 695  $\boldsymbol{\Sigma}$
- 696 •  $\mathbb{P}(E)$ : probability of an event  $E$
- 697 •  $\mathcal{P}(X)$ : family of absolutely continuous probability distributions with finite first moments  
 698 supported on a subset of  $X$ , where  $X \subseteq \mathbb{R}^d$
- 699 •  $\mathcal{P}_d(X)$ : the family of discrete distributions supported on a subset of  $X$ , where  $X \subseteq \mathbb{R}^d$

702     •  $\text{supp } \mu$ : the support of a probability distribution  $\mu$   
 703     •  $W_1(P_1, P_2)$ : the Wasserstein distance of order 1 between two probability distributions  
 704      $P_1, P_2 \in \mathcal{P}(X)$   
 705

706     B PROOFS  
 707

709     **Proposition 4.** *Let  $\mathbf{X}$  be a multivariate Gaussian random variable with mean vector  $\boldsymbol{\mu} \in \mathbb{R}^d$  and  
 710     positive definite covariance matrix  $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ . The  $100(1 - \alpha)\%$  HDR is*

711     
$$\mathcal{R}(f_\alpha) = \left\{ \mathbf{x} \in \mathbb{R}^d : (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq F_{\chi_d^2}^{-1}(1 - \alpha) \right\},$$
  
 712

713     *where  $F_{\chi_d^2}$  is the cumulative distribution function of a chi-squared random variable with  $d$  degrees  
 714     of freedom.*

716     *Proof.* Let  $\mathbf{Z} = (Z_1, Z_2, \dots, Z_d) = \sqrt{\boldsymbol{\Sigma}^{-1}}(\mathbf{X} - \boldsymbol{\mu})$ . By the change of variables formula,

717     
$$\begin{aligned} p_{\mathbf{Z}}(\mathbf{z}) &= \left| \det(\sqrt{\boldsymbol{\Sigma}}) \right| p_{\mathbf{X}}\left(\boldsymbol{\mu} + \sqrt{\boldsymbol{\Sigma}}\mathbf{z}\right) \\ 718 &= \det(\boldsymbol{\Sigma})^{1/2} (2\pi)^{-d/2} \det(\boldsymbol{\Sigma})^{-1/2} \exp\left(-\frac{1}{2}\mathbf{z}^\top \mathbf{z}\right) \\ 719 &= (2\pi)^{-d/2} \exp\left(-\frac{1}{2}\mathbf{z}^\top \mathbf{z}\right), \end{aligned}$$
  
 720  
 721  
 722  
 723

724     *where  $p_{\mathbf{X}}$  and  $p_{\mathbf{Z}}$  are the pdfs of random vectors  $\mathbf{X}$  and  $\mathbf{Z}$ , respectively. We can see that  $\mathbf{Z}$  is a  
 725     standard normal random vector. Since*

726     
$$\begin{aligned} \mathcal{R}(y) &= \left\{ \mathbf{x} \in \mathbb{R}^d : (2\pi)^{-d/2} \det(\boldsymbol{\Sigma})^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \geq y \right\} \\ 727 &= \left\{ \mathbf{x} \in \mathbb{R}^d : (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \leq -2 \log y + d \log(2\pi) + \log \det(\boldsymbol{\Sigma}) \right\}, \end{aligned}$$
  
 728  
 729

730     *we have*

731     
$$\begin{aligned} \mathbb{P}(\mathbf{X} \in \mathcal{R}(y)) &= \mathbb{P}(\mathbf{Z}^\top \mathbf{Z} \leq -2 \log y + d \log(2\pi) + \log \det(\boldsymbol{\Sigma})) \\ 732 &= \mathbb{P}\left(\sum_{i=1}^d Z_i^2 \leq -2 \log y + d \log(2\pi) + \log \det(\boldsymbol{\Sigma})\right). \end{aligned}$$
  
 733  
 734

735      *$Z_1, Z_2, \dots, Z_d$  are independent, so  $\sum_{i=1}^d Z_i^2$  is a chi-squared random variable. This implies*

736     
$$\mathbb{P}(\mathbf{X} \in \mathcal{R}(y)) = F_{\chi_d^2}(-2 \log y + d \log(2\pi) + \log \det(\boldsymbol{\Sigma})).$$

737      *$\mathbb{P}(\mathbf{X} \in \mathcal{R}(y)) \geq 1 - \alpha$  if and only if*

738     
$$-2 \log y + d \log(2\pi) + \log \det(\boldsymbol{\Sigma}) \geq F_{\chi_d^2}^{-1}(1 - \alpha).$$

739     *Therefore,*

740     
$$f_\alpha = (2\pi)^{d/2} \det(\boldsymbol{\Sigma})^{1/2} \exp\left(-\frac{1}{2}F_{\chi_d^2}^{-1}(1 - \alpha)\right),$$

741     *which means*

742     
$$\mathcal{R}(f_\alpha) = \left\{ \mathbf{x} \in \mathbb{R}^d : (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \leq F_{\chi_d^2}^{-1}(1 - \alpha) \right\}.$$
  
 743  
 744

745     □

746     **Proposition 5.** *Let  $m$  be a discrete latent variable supported on  $[K]$  and*

747     
$$\tau_{:T} = (s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \dots, s_{T-1}, a_{T-1}, r_{T-1}, s_T)$$

748     *be a trajectory rolled-out according to the following sampling process:  $s_0 \sim \rho_0$ ,  $m \sim p$ , and for  
 749     each  $t \in [T]$ ,  $s_{t+1} \sim P(s_t ; m)$ ,  $a_t \sim \beta(s_t ; m)$ , and  $r_t \sim R(s_t, s_{t+1} ; m)$ . The marginal pdf can be  
 750     written as*

751     
$$p(\tau_{:T}) = \rho_0(s_0) \sum_{m=0}^{K-1} p(m) \prod_{t=0}^{T-1} P(s_{t+1} \mid s_t ; m) \beta(a_t \mid s_t ; m) R(r_t \mid s_t, s_{t+1} ; m)$$
  
 752  
 753

756 and for any distribution  $q$  on  $[K]$ ,

757

$$758 \log p(\tau_{:T}) \geq \log \rho_0(s_0) - D_{\text{KL}}(q \parallel p) \quad (9)$$

759

$$760 + \sum_{t=0}^{T-1} \mathbb{E}_{m \sim q} [\log P(s_{t+1} \mid s_t; m) + \log \beta(a_t \mid s_t; m) + \log R(r_t \mid s_t, s_{t+1}; m)].$$

761

762 *Proof.* Let us denote the action-less trajectory by  $\tilde{\tau}_{:T}$ , that is,

763

$$\tilde{\tau}_{:T} = (s_0, r_0, s_1, r_1, s_2, r_2, \dots, s_{T-1}, r_{T-1}, s_T).$$

764

765 By Jensen's inequality,

766

$$\log p(\tau_{:T})$$

767

$$768 = \log \rho_0(s_0) + \log \sum_{m=0}^{K-1} p(m) \prod_{t=0}^{T-1} [P(s_{t+1} \mid s_t; m) \beta(a_t \mid s_t; m) R(r_t \mid s_t, s_{t+1}; m)]$$

769

$$770 = \log \rho_0(s_0) + \log \sum_{m=0}^{K-1} q(m) \cdot \frac{p(m)}{q(m)} \prod_{t=0}^{T-1} [P(s_{t+1} \mid s_t; m) \beta(a_t \mid s_t; m) R(r_t \mid s_t, s_{t+1}; m)]$$

771

$$772 \geq \log \rho_0(s_0) + \mathbb{E}_{m \sim q} \left[ \log \frac{p(m)}{q(m)} + \sum_{t=0}^{T-1} A(s_t, a_t, r_t, s_{t+1}; m) \right]$$

773

$$774 = \log \rho_0(s_0) - D_{\text{KL}}(q \parallel p) + \sum_{t=0}^{T-1} \mathbb{E}_{m \sim q} [A(s_t, a_t, r_t, s_{t+1}; m)],$$

775

776 where

777

$$A(s_t, a_t, r_t, s_{t+1}; m) = \log P(s_{t+1} \mid s_t; m) + \log \beta(a_t \mid s_t; m) + \log R(r_t \mid s_t, s_{t+1}; m).$$

778

□

780 We restate the two assumptions we made in Section 4.2 for the reader's convenience.

781 **Assumption 3.** There is  $K_P > 0$  such that for all  $s \in \mathcal{S}$  and  $a_1, a_2 \in \mathcal{A}$ ,  $W_1(P(s, a_1), P(s, a_2)) < K_P \|a_1 - a_2\|$ .

782 **Assumption 4.** The value function of the behaviour policy  $\beta$  is  $K_V$ -Lipschitz.

783 **Lemma 6.** For any policy  $\pi$  and  $s \in \mathcal{S}$ ,

784

$$\frac{r_{\min}}{1-\gamma} \leq V^\beta(s) \leq \frac{r_{\max}}{1-\gamma}.$$

785

786 *Proof.* By the definition of  $V^\pi$ , for all  $s \in \mathcal{S}$ ,

787

$$V^\pi(s) = \mathbb{E}_{\tau \sim \pi|s} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right] \geq \mathbb{E}_{\tau \sim \pi|s} \left[ \sum_{t=0}^{\infty} \gamma^t r_{\min} \right] = \frac{r_{\min}}{1-\gamma},$$

788

789 and

790

$$V^\pi(s) = \mathbb{E}_{\tau \sim \pi|s} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right] \leq \mathbb{E}_{\tau \sim \pi|s} \left[ \sum_{t=0}^{\infty} \gamma^t r_{\max} \right] = \frac{r_{\max}}{1-\gamma}.$$

791

□

792 **Proposition 7.** Define  $Q_\beta^{\text{LB}}: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  by the equation

793

$$Q_\beta^{\text{LB}}(s, a) = \max \left\{ V^\beta(s) - r_{\max} + r_{\min} - \gamma K_V K_P \mathbb{E}_{a' \sim \beta(s)} [\|a - a'\|], \frac{r_{\min}}{1-\gamma} \right\}.$$

794

795 For any policy  $\pi: \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$  such that  $V^\pi \succeq V^\beta$ ,  $Q^\pi \succeq Q_\beta^{\text{LB}}$ .

810 *Proof.* By Lemma 6 and the definition of  $Q^\pi$ , for all  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ ,

$$\begin{aligned} 812 \quad Q^\pi(s, a) &= \mathbb{E}_{s' \sim P(s, a, s'), r \sim R(s, a, s')} [r + \gamma V^\pi(s')] \\ 813 \quad &\geq \mathbb{E}_{s' \sim P(s, a, s'), r \sim R(s, a, s')} \left[ r_{\min} + \gamma \frac{r_{\min}}{1 - \gamma} \right] \\ 814 \quad &= \frac{r_{\min}}{1 - \gamma}. \\ 815 \end{aligned}$$

816 So we only need to show that for all  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ ,

$$817 \quad Q^\pi(s, a) \geq V^\beta(s) - r_{\max} + r_{\min} - \gamma K_V K_P \mathbb{E}_{a' \sim \beta(s)} [\|a - a'\|]. \\ 818$$

819 Let  $a_1, a_2 \in \mathcal{A}$ . By the Kantorovich–Rubinstein formula (Villani, 2009),

$$\begin{aligned} 820 \quad &|\mathbb{E}_{s' \sim P(\cdot | s, a_1)} [V^\beta(s')] - \mathbb{E}_{s' \sim P(\cdot | s, a_2)} [V^\beta(s')]| \leq K_V W_1(P(\cdot | s, a_1), P(\cdot | s, a_2)) \\ 821 \quad &\leq K_V K_P \|a_1 - a_2\|. \\ 822 \end{aligned}$$

823 Therefore,

$$\begin{aligned} 824 \quad Q^\pi(s, a) &= \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} [r + \gamma V^\pi(s')] \\ 825 \quad &\geq \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} [r + \gamma V^\beta(s')] \\ 826 \quad &= V^\beta(s) - \mathbb{E}_{a' \sim \beta(\cdot | s)} [\mathbb{E}_{s' \sim P(s, a'), r \sim R(s, a', s')} [r + \gamma V^\beta(s')]] \\ 827 \quad &\quad + \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} [r + \gamma V^\beta(s')] \\ 828 \quad &\geq V^\beta(s) - r_{\max} + r_{\min} + \gamma \mathbb{E}_{a' \sim \beta(s)} [\mathbb{E}_{s' \sim P(s, a)} [V^\beta(s')] - \mathbb{E}_{s' \sim P(s, a')} [V^\beta(s')]] \\ 829 \quad &\geq V^\beta(s) - r_{\max} + r_{\min} - \gamma K_V K_P \mathbb{E}_{a' \sim \beta(\cdot | s)} [\|a - a'\|]. \\ 830 \end{aligned}$$

831 Note that we have used the fact that

$$832 \quad V^\beta(s) = \mathbb{E}_{a' \sim \beta(s), s' \sim P(s, a'), r \sim R(s, a', s')} [r + \gamma V^\beta(s')]. \\ 833$$

834  $\square$

835 **Theorem 8.** Let  $\{A_s\}_{s \in \mathcal{S}}$  be a family of subsets of  $\mathcal{A}$ ,  $\tilde{Q} \in L^\infty(\mathcal{S} \times \mathcal{A})$ , and  $\mathcal{T}_A$  be an operator on  
836 the space of real-valued functions on  $\mathcal{S} \times \mathcal{A}$  defined by the equation

$$837 \quad (\mathcal{T}_A Q)(s, a) = \begin{cases} (\mathcal{T}^* Q)(s, a) & \text{if } a \in A_s, \\ \tilde{Q}(s, a) & \text{otherwise,} \end{cases}$$

838 for each  $Q \in L^\infty(\mathcal{S} \times \mathcal{A})$ . Then any bounded real-valued function on  $\mathcal{S} \times \mathcal{A}$  converges to a unique  
839 fixed point  $Q_A$  by repeatedly applying  $\mathcal{T}_A$ .

840 *Proof.* Fix  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}$ , and  $Q \in L^\infty(\mathcal{S} \times \mathcal{A})$ . If  $a \in A_s$ ,

$$\begin{aligned} 841 \quad |(\mathcal{T}_A Q)(s, a)| &= |(\mathcal{T}^* Q)(s, a)| \\ 842 \quad &= \left| \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} \left[ r + \gamma \sup_{a' \in \mathcal{A}} Q(s', a') \right] \right| \\ 843 \quad &\leq \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} \left[ \left| r + \gamma \sup_{a' \in \mathcal{A}} Q(s', a') \right| \right] \\ 844 \quad &\leq \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} \left[ |r| + \gamma \left| \sup_{a' \in \mathcal{A}} Q(s', a') \right| \right] \\ 845 \quad &\leq \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} [\max \{|r_{\max}|, |r_{\min}|\} + \gamma \|Q\|_\infty], \\ 846 \quad &= \max \{|r_{\max}|, |r_{\min}|\} + \gamma \|Q\|_\infty. \\ 847 \end{aligned}$$

848 Otherwise,

$$849 \quad |(\mathcal{T}_A Q)(s, a)| = |\tilde{Q}(s, a)| \leq \|\tilde{Q}\|_\infty.$$

850 So

$$851 \quad \|\mathcal{T}_A Q\|_\infty \leq \max \left\{ \|\tilde{Q}\|_\infty, \max \{|r_{\max}|, |r_{\min}|\} + \gamma \|Q\|_\infty \right\} < \infty,$$

864 that is,  $\mathcal{T}_A Q \in L^\infty(\mathcal{S} \times \mathcal{A})$ . So the restriction of  $\mathcal{T}_A$  onto  $L^\infty(\mathcal{S} \times \mathcal{A})$  is an operator on  $L^\infty(\mathcal{S} \times \mathcal{A})$ .  
 865 With a slight abuse of notation, we will just denote the restriction by  $\mathcal{T}_A$  from now on.  
 866

867 Now we go on and prove that  $\mathcal{T}_A$  is a contraction operator. Fix  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}$  and  $Q_1, Q_2 \in$   
 868  $L^\infty(\mathcal{S} \times \mathcal{A})$ . If  $a \in A_s$ ,

$$\begin{aligned} |(\mathcal{T}_A Q_1)(s, a) - (\mathcal{T}_A Q_2)(s, a)| &= |(\mathcal{T}^* Q_1)(s, a) - (\mathcal{T}^* Q_2)(s, a)| \\ &= \gamma \left| \mathbb{E}_{s' \sim P(s, a)} \left[ \sup_{a' \in \mathcal{A}} Q_1(s', a') - \sup_{a'' \in \mathcal{A}} Q_2(s', a'') \right] \right| \\ &\leq \gamma \mathbb{E}_{s' \sim P(s, a)} \left[ \left| \sup_{a' \in \mathcal{A}} Q_1(s', a') - \sup_{a'' \in \mathcal{A}} Q_2(s', a'') \right| \right] \\ &\leq \gamma \mathbb{E}_{s' \sim P(s, a)} \left[ \sup_{a' \in \mathcal{A}} |Q_1(s', a') - Q_2(s', a')| \right] \\ &\leq \gamma \|Q_1 - Q_2\|_\infty. \end{aligned}$$

879 Otherwise,

$$880 |(\mathcal{T}_A Q_1)(s, a) - (\mathcal{T}_A Q_2)(s, a)| = \left| \tilde{Q}(s, a) - \tilde{Q}(s, a) \right| = 0 \leq \gamma \|Q_1 - Q_2\|_\infty.$$

882 Therefore, and  $\mathcal{T}_A$  is a contraction mapping on  $L^\infty(\mathcal{S} \times \mathcal{A})$ . By the contraction mapping theorem,  
 883 any initial-bounded Q-function would converge to a unique fixed point  $Q_A$ .  $\square$   
 884

885 **Lemma 9.** *Let  $\pi_1$  and  $\pi_2$  be two policies. If  $\mathbb{E}_{a \sim \pi_1(s)}[Q^{\pi_2}(s, a)] \geq V^{\pi_2}(s)$  for all  $s \in \mathcal{S}$ , then*  
 886  $V^{\pi_1} \succeq V^{\pi_2}$ .

887 *Proof.* We define a sequence  $(Q_n)$  of bounded real-valued functions on  $\mathcal{S} \times \mathcal{A}$  by the recurrence  
 888 relation

$$889 Q_n = \begin{cases} Q^{\pi_2} & \text{if } n = 0, \\ 890 \mathcal{T}^{\pi_1} Q_{n-1} & \text{otherwise.} \end{cases}$$

892 We first show that  $Q_n \succeq Q^{\pi_2}$  by mathematical induction. The base case is trivial because  $Q_0 \equiv$   
 893  $Q^{\pi_2}$ . Suppose  $Q_{n-1} \succeq Q^{\pi_2}$ . Then for each  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ ,

$$\begin{aligned} 894 Q_n(s, a) &= (\mathcal{T}^{\pi_1} Q_{n-1})(s, a) \\ 895 &= \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} [r + \gamma \mathbb{E}_{a' \sim \pi_1(s')} [Q_{n-1}(s', a')]] \\ 896 &\geq \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} [r + \gamma \mathbb{E}_{a' \sim \pi_1(s')} [Q^{\pi_2}(s', a')]] \\ 897 &\geq \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} [r + \gamma \mathbb{E}_{a' \sim \pi_2(s')} [Q^{\pi_2}(s', a')]] \\ 898 &= (\mathcal{T}^{\pi_2} Q^{\pi_2})(s, a) \\ 899 &= Q^{\pi_2}(s, a). \end{aligned}$$

900 So  $Q_n \succeq Q^{\pi_2}$ . By mathematical induction,  $Q_n \succeq Q^{\pi_2}$  for all  $n$ . For all  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ ,

$$901 Q^{\pi_1}(s, a) = \lim_{n \rightarrow \infty} Q_n(s, a) \geq Q^{\pi_2}(s, a).$$

902 Therefore, for all  $s \in \mathcal{S}$ ,

$$903 V^{\pi_1}(s) = \mathbb{E}_{a \sim \pi_1(s)} [Q^{\pi_1}(s, a)] \geq \mathbb{E}_{a \sim \pi_1(s)} [Q^{\pi_2}(s, a)] \geq V^{\pi_2}(s),$$

904 that is,  $V^{\pi_1} \succeq V^{\pi_2}$ .  $\square$

905 **Theorem 10.** *Let  $\{A_s\}_{s \in \mathcal{S}}$  be a family of subsets of  $\mathcal{A}$ ,  $\tilde{Q} \in L^\infty(\mathcal{S} \times \mathcal{A})$ , and  $Q_A$  be a bounded  
 906 real-valued function that satisfies the relation*

$$907 Q_A(s, a) = \begin{cases} (\mathcal{T}^* Q_A)(s, a) & \text{if } a \in A_s, \\ 908 \tilde{Q}(s, a) & \text{otherwise,} \end{cases} \quad (10)$$

909 for all  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ . Suppose there is a policy  $\pi$  such that for all  $s \in \mathcal{S}$ ,

$$910 V^\pi(s) \geq \sup_{a \in \mathcal{A}} \tilde{Q}(s, a)$$

918 and

919 
$$Q^\pi(s, a_s) \geq V^\pi(s)$$

920 for some  $a_s \in A_s$ . If there exists a deterministic policy  $\pi_A^* : \mathcal{S} \rightarrow \mathcal{A}$  that is optimal under the  
921 constraint  $\pi_A(s) \in A_s$  for all  $s \in \mathcal{S}$ , then

922 
$$\pi_A^*(s) = \arg \max_{a \in \mathcal{A}} Q_A(s, a).$$

924

925 *Proof.* Define  $\mathcal{T}_A$  as in Theorem 8. We can see that there is a unique bounded real-valued function  
926  $Q_A$  that satisfies (10), because by Theorem 8,  $\mathcal{T}_A$  has unique fixed point  $Q_A$ .  
927

928 We proceed to prove that for each  $s \in \mathcal{S}$ ,  $Q_A(s, \pi_A^*(s)) \geq V^{\pi_A^*}(s)$ . Define a sequence  $(Q_n)$  of  
929 bounded real-valued functions on  $\mathcal{S} \times \mathcal{A}$  by the recurrence relation

930 
$$Q_n = \begin{cases} Q_0 & \text{if } n = 0, \\ \mathcal{T}_A Q_{n-1} & \text{otherwise,} \end{cases} \quad (11)$$

932

933 where  $Q_0 : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  is defined as

934 
$$Q_0(s, a) = \begin{cases} Q_A^{\pi_A^*}(s, a) & \text{if } a \in A_s, \\ \tilde{Q}(s, a) & \text{otherwise.} \end{cases}$$

936

937 When  $n = 0$ , for all  $s \in \mathcal{S}$

938 
$$Q_0(s, \pi_A^*(s)) = Q^{\pi_A^*}(s, \pi_A^*(s)) = V^{\pi_A^*}(s),$$

939

940 because  $\pi_A^*(s) \in A_s$ . Assume  $Q_{n-1}(s, \pi_A^*(s)) \geq V^{\pi_A^*}(s)$  for all  $s \in \mathcal{S}$ . Then for all  $s \in \mathcal{S}$ ,

941 
$$\begin{aligned} Q_n(s, \pi_A^*(s)) &= \mathbb{E}_{s' \sim P(s, \pi_A^*(s)), r \sim R(s, \pi_A^*(s), s')} \left[ r + \gamma \sup_{a' \in \mathcal{A}} Q_{n-1}(s', a') \right] \\ &\geq \mathbb{E}_{s' \sim P(s, \pi_A^*(s)), r \sim R(s, \pi_A^*(s), s')} [r + \gamma Q_{n-1}(s', \pi_A^*(s'))] \\ &\geq \mathbb{E}_{s' \sim P(s, \pi_A^*(s)), r \sim R(s, \pi_A^*(s), s')} \left[ r + \gamma V^{\pi_A^*}(s') \right] \\ &= \mathbb{E}_{s' \sim P(s, \pi_A^*(s)), r \sim R(s, \pi_A^*(s), s')} \left[ r + \gamma \mathbb{E}_{a' \sim \pi_A^*(s')} \left[ Q^{\pi_A^*}(s', a') \right] \right] \\ &= (\mathcal{T}^{\pi_A^*} Q^{\pi_A^*})(s, \pi_A^*(s)) \\ &= Q^{\pi_A^*}(s, \pi_A^*(s)) \\ &= V^{\pi_A^*}(s). \end{aligned}$$

952

953 So by mathematical induction,  $Q_n(s, \pi_A^*(s)) \geq V^{\pi_A^*}(s)$  for all  $s \in \mathcal{S}$  and  $n \geq 0$ . Therefore,  
954

955 
$$Q_A(s, \pi_A^*(s)) = \lim_{n \rightarrow \infty} Q_n(s, \pi_A^*(s)) \geq V^{\pi_A^*}(s).$$

956

957 Since for all  $s \in \mathcal{S}$  and  $a \in A_s$ ,

958 
$$Q_A(s, a) = (\mathcal{T}_A Q_A)(s, a) = \tilde{Q}(s, a) \leq V^\pi(s).$$

959

960 We can define a deterministic policy  $\pi_A : \mathcal{S} \rightarrow \mathcal{A}$  that maps  $s \in \mathcal{S}$  to  $a_s$ . Since  $\pi_A(s) = a_s \in A_s$   
961 for all  $s \in \mathcal{S}$  and  $\pi_A^*$  is the optimal policy among the policies that satisfy this constraint, we have  
962  $V^{\pi_A^*} \succeq V^{\pi_A}$ . So we may conclude that for all  $s \in \mathcal{S}$ ,

963 
$$\sup_{a \in \mathcal{A} \setminus A_s} Q_A(s, a) \leq V^\pi(s) \leq V^{\pi_A}(s) \leq V^{\pi_A^*}(s) = Q_A(s, \pi_A^*(s)). \quad (12)$$

965

966 We finish the proof by showing that for all  $s \in \mathcal{S}$ ,

967 
$$Q_A(s, \pi_A^*(s)) = \max_{a \in \mathcal{A}} Q_A(s, a).$$

968

969 Recall the sequence  $(Q_n)$  we previously defined by the recurrence relation (11). We will prove that  
970 for every  $n, s \in \mathcal{S}$ , and  $a \in \mathcal{A}$ ,

971 
$$Q_n(s, a) \leq V^{\pi_A^*}(s) = Q_n(s, \pi_A^*(s)).$$

972

18

972 Assume  $n = 0$ . Fix  $s^\dagger \in \mathcal{S}$  and  $a^\dagger \in \mathcal{A}$ . If  $a^\dagger \notin A_{s^\dagger}$ , then by the observation we made in (12),  
 973

$$974 Q_0(s^\dagger, a^\dagger) = \tilde{Q}(s^\dagger, a^\dagger) \leq V^{\pi_A^*}(s^\dagger).$$

975 If  $a^\dagger \in A_{s^\dagger}$ , consider a policy  $\pi^\dagger: \mathcal{S} \rightarrow \mathcal{A}$  defined as  
 976

$$977 \pi^\dagger(s) = \begin{cases} a^\dagger & \text{if } s = s^\dagger, \\ 978 \pi_A^*(s) & \text{otherwise.} \end{cases}$$

979 For all  $s \in \mathcal{S}$ ,  $\pi^\dagger(s) \in A_s$ , so  $V^{\pi_A^*} \succeq V^{\pi^\dagger}$ . If  $Q_0(s^\dagger, a^\dagger) < V^{\pi_A^*}(s^\dagger)$ , it satisfies our hypothesis.  
 980 Otherwise,

$$981 Q^{\pi_A^*}(s^\dagger, \pi^\dagger(s^\dagger)) = Q_0(s^\dagger, a^\dagger) \geq V^{\pi_A^*}(s^\dagger) = Q^{\pi_A^*}(s^\dagger, \pi_A^*(s^\dagger))$$

982 and for  $s \neq s^\dagger$ ,

$$983 Q^{\pi_A^*}(s, \pi^\dagger(s)) = Q^{\pi_A^*}(s, \pi_A^*(s)),$$

984 so by Lemma 9,  $V^{\pi^\dagger} \succeq V^{\pi_A^*}$ , which means  $V^{\pi^\dagger} \equiv V^{\pi_A^*}$ . Then

$$\begin{aligned} 985 Q^{\pi_A^*}(s^\dagger, a^\dagger) &= (\mathcal{T}^{\pi_A^*} Q^{\pi_A^*})(s^\dagger, a^\dagger) \\ 986 &= \mathbb{E}_{s' \sim P(s^\dagger, a^\dagger), r \sim R(s^\dagger, a^\dagger, s')} \left[ r + \gamma Q^{\pi_A^*}(s', \pi_A^*(s')) \right] \\ 987 &= \mathbb{E}_{s' \sim P(s^\dagger, a^\dagger), r \sim R(s^\dagger, a^\dagger, s')} \left[ r + \gamma V^{\pi_A^*}(s') \right] \\ 988 &= \mathbb{E}_{s' \sim P(s^\dagger, a^\dagger), r \sim R(s^\dagger, a^\dagger, s')} \left[ r + \gamma V^{\pi^\dagger}(s') \right] \\ 989 &= \mathbb{E}_{s' \sim P(s^\dagger, a^\dagger), r \sim R(s^\dagger, a^\dagger, s')} \left[ r + \gamma Q^{\pi^\dagger}(s', \pi^\dagger(s')) \right] \\ 990 &= (\mathcal{T}^{\pi^\dagger} Q^{\pi^\dagger})(s^\dagger, a^\dagger) \\ 991 &= Q^{\pi^\dagger}(s^\dagger, \pi^\dagger(s^\dagger)) \\ 992 &= V^{\pi^\dagger}(s^\dagger) \\ 993 &= V^{\pi_A^*}(s^\dagger). \end{aligned}$$

994 So  $Q_0(s^\dagger, a^\dagger) \leq V^{\pi_A^*}(s^\dagger)$  in both cases. Since it is obvious that  
 995

$$996 Q_0(s^\dagger, \pi_A^*(s^\dagger)) = Q^{\pi_A^*}(s^\dagger, \pi_A^*(s^\dagger)) = V^{\pi_A^*}(s^\dagger),$$

997 our hypothesis holds for  $n = 0$ .

998 Assume the hypothesis holds for  $n - 1$ . Fix  $s^\dagger \in \mathcal{S}$  and  $a^\dagger \in \mathcal{A}$ . If  $a^\dagger \notin A_{s^\dagger}$ ,  
 999

$$1000 Q_n(s^\dagger, a^\dagger) = (\mathcal{T}_\beta^* Q_{n-1})(s^\dagger, a^\dagger) = Q_\beta^{\text{LB}}(s^\dagger, a^\dagger) \leq V^{\pi_\beta^*}(s^\dagger)$$

1001 by (12). Otherwise,

$$\begin{aligned} 1002 Q_n(s^\dagger, a^\dagger) &= (\mathcal{T}_A Q_{n-1})(s^\dagger, a^\dagger) \\ 1003 &= (\mathcal{T}^* Q_{n-1})(s^\dagger, a^\dagger) \\ 1004 &= \mathbb{E}_{s' \sim P(s^\dagger, a^\dagger), r \sim R(s^\dagger, a^\dagger, s')} \left[ r + \gamma \sup_{a' \in \mathcal{A}} Q_{n-1}(s', a') \right] \\ 1005 &= \mathbb{E}_{s' \sim P(s^\dagger, a^\dagger), r \sim R(s^\dagger, s^\dagger, s')} \left[ r + \gamma \max_{a' \in \mathcal{A}} Q_{n-1}(s', a') \right] \\ 1006 &= \mathbb{E}_{s' \sim P(s^\dagger, a^\dagger), r \sim R(s^\dagger, s^\dagger, s')} \left[ r + \gamma V^{\pi_A^*}(s') \right] \\ 1007 &= \mathbb{E}_{s' \sim P(s^\dagger, a^\dagger), r \sim R(s^\dagger, s^\dagger, s')} \left[ r + \gamma \mathbb{E}_{a' \sim \pi_A^*(s')} \left[ Q^{\pi_A^*}(s', a') \right] \right] \\ 1008 &= (\mathcal{T}^{\pi_A^*} Q^{\pi_A^*})(s^\dagger, a^\dagger) \\ 1009 &= Q^{\pi_A^*}(s^\dagger, a^\dagger) \\ 1010 &= Q_0(s^\dagger, a^\dagger) \\ 1011 &\leq V^{\pi_A^*}(s^\dagger). \end{aligned}$$

1026 When  $a^\dagger = \pi_A^*(s^\dagger)$ , the inequality becomes equality. So by mathematical induction, for every  $n$ ,  
 1027  $s \in \mathcal{S}$ , and  $a \in \mathcal{A}$ ,

$$1028 \quad Q_n(s, a) \leq V^{\pi_A^*}(s) = Q_n(s, \pi_A^*(s)).$$

1030 Sending  $n$  to infinity, we can see that for all  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ ,

$$1031 \quad Q_A(s, a) = \lim_{n \rightarrow \infty} Q_n(s, a) \leq V^{\pi_A^*}(s) = \lim_{n \rightarrow \infty} Q_n(s, \pi_A^*(s)) = Q_A(s, \pi_A^*(s)).$$

1033 Therefore,

$$1035 \quad Q_A(s, \pi_A^*(s)) = \max_{a \in \mathcal{A}} Q_A(s, a).$$

1037  $\square$

1038 **Theorem 11.** Any initial bounded real-valued function on  $\mathcal{S} \times \mathcal{A}$  can converge to a unique fixed  
 1039 point  $Q_\beta^*$  by repeatedly applying  $\mathcal{T}_\beta^*$ . Suppose for each  $s \in \mathcal{S}$ ,

$$1041 \quad Q^\beta(s, a_s) \geq \mathbb{E}_{a \sim \beta(s)} [Q^\beta(s, a)]$$

1043 for some  $a_s \in \mathcal{A} \setminus \text{OOD}(s)$ . If there exists a deterministic policy  $\pi_\beta^* : \mathcal{S} \rightarrow \mathcal{A}$  that is optimal under  
 1044 the constraint  $\pi(s) \notin \text{OOD}(s)$  for all  $s \in \mathcal{S}$ , then  $\pi_\beta^*(s) = \arg \max_{a \in \mathcal{A}} Q_\beta^*(s, a)$  for all  $s \in \mathcal{S}$ .

1046 *Proof.* First observe that for all  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ ,

$$1048 \quad Q_\beta^{\text{LB}}(s, a) \geq \frac{r_{\min}}{1 - \gamma},$$

1050 and

$$1052 \quad Q_\beta^{\text{LB}}(s, a) \leq V^\beta(s) - r_{\max} + r_{\min} \leq \frac{r_{\max}}{1 - \gamma} - r_{\max} + r_{\min},$$

1054 by Lemma 6. This implies  $Q_\beta^{\text{LB}} \in L^\infty(\mathcal{S} \times \mathcal{A})$ . Since  $\tilde{Q} = Q_\beta^{\text{LB}}$  and  $A_s = \mathcal{A} \setminus \text{OOD}(s)$  satisfies the  
 1055 conditions of Lemma 8, any initially bounded real-valued function on  $\mathcal{S} \times \mathcal{A}$  converges to a unique  
 1056 fixed point, which we denote by  $Q_\beta^*$ , through repeated application of  $\mathcal{T}_A$ , which is in fact,  $\mathcal{T}_\beta^*$ .

1057 For all  $s \in \mathcal{S}$ ,

$$1058 \quad Q_\beta^{\text{LB}}(s, a) \leq V^\beta(s) - r_{\max} + r_{\min} \leq V^\beta(s),$$

1060 which means

$$1062 \quad \sup_{a \in \mathcal{A}} Q_\beta^{\text{LB}}(s, a) \leq V^\beta(s) = \mathbb{E}_{a \sim \beta(s)} [Q^\beta(s, a)] < \sup_{a \in \mathcal{A} \setminus \text{OOD}(s)} Q^\beta(s, a).$$

1064 Now we can see that the second part of the theorem is a special case of Theorem 10, where  $A_s =$   
 1065  $\mathcal{A} \setminus \text{OOD}(s)$ ,  $\tilde{Q} = Q_\beta^{\text{LB}}$ ,  $Q_A = Q_\beta^*$ ,  $\pi = \beta$ , and  $\pi_A^* = \pi_\beta^*$ .  $\square$

1067 **Lemma 12.** Let  $\Pi = \{\pi_0, \pi_1, \pi_2, \dots, \pi_{N-1}\}$  be a finite set of policies. If  $\pi^*$  is a policy such that  
 1068 for each  $s \in \mathcal{S}$ , there is  $i \in [N]$  such that  $\pi^*(s) = \pi_i(s)$  and  $V^{\pi_i}(s) = \max_{\pi \in \Pi} V^\pi(s)$ , then  
 1069  $V^{\pi^*} \succeq V^\pi$  for every  $\pi \in \Pi$ .

1071 *Proof.* Define a sequence  $(Q_n)$  of bounded real-valued functions by the recurrence relation  
 1072

$$1073 \quad Q_n = \begin{cases} \max_{\pi \in \Pi} Q^\pi & \text{if } n = 0, \\ \mathcal{T}^{\pi^*} Q_{n-1}, & \text{otherwise.} \end{cases}$$

1076 We want to show that  $Q_n \succeq \max_{\pi \in \Pi} Q^\pi$  for all  $n \geq 0$ . The base case is trivial. Assume  
 1077  $Q_{n-1} \succeq \max_{\pi \in \Pi} Q^\pi$ . For each  $s \in \mathcal{S}$ , there is  $i \in [N]$  such that  $\pi^*(s) = \pi_i(s)$  and  
 1078  $V^{\pi_i}(s) = \max_{\pi \in \Pi} V^\pi(s)$ , which implies

$$1079 \quad \mathbb{E}_{a \sim \pi^*(s)} [Q_{n-1}(s, a)] \geq \mathbb{E}_{a \sim \pi_i(s)} [Q^{\pi_i}(s)] = V^{\pi_i}(s) = \max_{\pi \in \Pi} V^\pi(s).$$

1080 Now for all  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ ,

$$\begin{aligned}
 Q_n(s, a) &= (\mathcal{T}^{\pi^*} Q_{n-1})(s, a) \\
 &= \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} [r + \gamma \mathbb{E}_{a' \sim \pi^*(s')} [Q_{n-1}(s', a')]] \\
 &\geq \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} \left[ r + \gamma \max_{\pi \in \Pi} V^{\pi}(s') \right] \\
 &= \max_{\pi \in \Pi} \mathbb{E}_{s' \sim P(s, a), r \sim R(s, a, s')} [r + \gamma \mathbb{E}_{a' \sim \pi(s')} [Q^{\pi}(s', a')]] \\
 &= \max_{\pi \in \Pi} (\mathcal{T}^{\pi} Q^{\pi})(s, a) \\
 &= \max_{\pi \in \Pi} Q^{\pi}(s, a).
 \end{aligned}$$

1092 By mathematical induction,  $Q_n \succeq \max_{\pi \in \Pi} Q^{\pi}$  for all  $n \geq 0$ . Therefore,

$$Q^{\pi^*}(s, a) = \lim_{n \rightarrow \infty} Q_n(s, a) \geq \max_{\pi \in \Pi} Q^{\pi}(s, a)$$

1095 for all  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ .

1097 Fix  $s \in \mathcal{S}$ . There is  $i \in [N]$  such that  $\pi^*(s) = \pi_i(s)$  and  $V^{\pi_i}(s) = \max_{\pi \in \Pi} V^{\pi}(s)$ . Then

$$V^{\pi^*}(s) = \mathbb{E}_{a \sim \pi^*(s)} [Q^{\pi^*}(s, a)] \geq \mathbb{E}_{a \sim \pi_i(s)} [Q^{\pi_i}(s, a)] = V^{\pi_i}(s) = \max_{\pi \in \Pi} V^{\pi}(s).$$

1100 Our choice of  $s$  was arbitrary, so  $V^{\pi^*} \succeq \max_{\pi \in \Pi} V^{\pi}$ .  $\square$

1102 **Theorem 13.** Any initial bounded real-valued function on  $\mathcal{S} \times \mathcal{A}$  can converge to a unique fixed  
1103 point  $Q_{\mathcal{B}}^*$  by repeatedly applying  $\mathcal{T}_{\mathcal{B}}^*$ . Suppose for each  $\beta \in \mathcal{B}$  and  $s \in \mathcal{S}$ ,

$$Q^{\beta}(s, a_s^{\beta}) \geq \mathbb{E}_{a \sim \beta(s)} [Q^{\beta}(s, a)]$$

1106 for some  $a_s^{\beta} \in \mathcal{A} \setminus \text{OOD}(s)$ . If there exists a deterministic policy  $\pi_{\mathcal{B}}^* : \mathcal{S} \rightarrow \mathcal{A}$  that is optimal under  
1107 the constraint  $\pi(s) \notin \text{OOD}(s)$  for all  $s \in \mathcal{S}$ , then  $\pi_{\mathcal{B}}^*(s) = \arg \max_{a \in \mathcal{A}} Q_{\mathcal{B}}^*(s, a)$  for all  $s \in \mathcal{S}$ .

1109 *Proof.* First observe that for all  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}$ , and  $\beta \in \mathcal{B}$ ,

$$Q_{\beta}^{\text{LB}}(s, a) \geq \frac{r_{\min}}{1 - \gamma},$$

1113 and

$$Q_{\beta}^{\text{LB}}(s, a) \leq V^{\beta}(s) - r_{\max} + r_{\min} \leq \frac{r_{\max}}{1 - \gamma} - r_{\max} + r_{\min},$$

1116 by Lemma 6. So obviously,

$$Q_{\mathcal{B}}^{\text{LB}}(s, a) = \max_{\beta \in \mathcal{B}} Q_{\beta}^{\text{LB}}(s, a) \geq \frac{r_{\min}}{1 - \gamma},$$

1119 and

$$Q_{\mathcal{B}}^{\text{LB}}(s, a) = \max_{\beta \in \mathcal{B}} Q_{\beta}^{\text{LB}}(s, a) \leq \frac{r_{\max}}{1 - \gamma} - r_{\max} + r_{\min},$$

1122 for all  $s \in \mathcal{S}$  and  $\mathcal{A}$ . This implies  $Q_{\mathcal{B}}^{\text{LB}} \in L^{\infty}(\mathcal{S} \times \mathcal{A})$ . Since  $\tilde{Q} = Q_{\mathcal{B}}^{\text{LB}}$  and  $A_s = \mathcal{A} \setminus \text{OOD}(s)$   
1123 satisfies the conditions of Lemma 8, any initially bounded real-valued function on  $\mathcal{S} \times \mathcal{A}$  converges  
1124 to a unique fixed point, which we denote by  $Q_{\mathcal{B}}^*$ , through repeated application of  $\mathcal{T}_{\mathcal{A}}$ , which is in  
1125 fact,  $\mathcal{T}_{\mathcal{B}}^*$ .

1126 For each  $\beta \in \mathcal{B}$  define a deterministic policy  $\beta' : \mathcal{S} \rightarrow \mathcal{A}$  so that  $\beta'(s) = a_s^{\beta}$  for each  $s \in \mathcal{S}$ . Then

$$Q^{\beta}(s, \beta'(s)) = Q^{\beta}(s, a_s^{\beta}) \geq \mathbb{E}_{a \sim \beta(s)} [Q^{\beta}(s, a)]$$

1130 for all  $s \in \mathcal{S}$ , so  $V^{\beta'} \succeq V^{\beta}$  by Lemma 9. We will denote the set  $\{\beta' : \beta \in \mathcal{B}\}$  by  $\mathcal{B}'$ . Consider a  
1131 policy  $\beta^* : \mathcal{S} \rightarrow \mathcal{A}$  defined as

$$\beta^*(s) = \left( \arg \max_{\beta' \in \mathcal{B}'} V^{\beta'}(s) \right) (s),$$

1134 that is, for each state, we follow the  $\beta'$  with the highest value. Obviously,  $\beta^*(s) \in \mathcal{A} \setminus \text{OOD}(s)$  for  
 1135 all  $s \in \mathcal{S}$ , and by Lemma 12, for all  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}$ , and  $\beta \in \mathcal{B}$ ,

$$1137 Q_{\beta}^{\text{LB}}(s, a) \leq V^{\beta}(s) - r_{\max} + r_{\min} \leq V^{\beta}(s) \leq V^{\beta'}(s) \leq V^{\beta^*}(s),$$

1138 which means

$$1139 V^{\beta^*}(s) \geq \sup_{a \in \mathcal{A}} \max_{\beta \in \mathcal{B}} Q_{\beta}^{\text{LB}}(s, a) = \sup_{a \in \mathcal{A}} Q_{\mathcal{B}}^{\text{LB}}(s, a).$$

1140 Now we can see that the second part of the theorem is a special case of Theorem 10, where  $A_s =$   
 1141  $\mathcal{A} \setminus \text{OOD}(s)$ ,  $\tilde{Q} = Q_{\mathcal{B}}^{\text{LB}}$ ,  $Q_A = Q_{\mathcal{B}}^*$ ,  $\pi = \beta^*$ , and  $\pi_A^* = \pi_{\beta}^*$ .  $\square$

1143 **Proposition 14.** *Let  $\mathbf{X}$  be a non-degenerate multivariate Gaussian random vector with mean  $\mu \in$   
 1144  $\mathbb{R}^d$  and a diagonal covariance matrix  $\text{diag}(\sigma)^2 \in \mathbb{R}^{d \times d}$ . For  $\mathbf{y} \in \mathbb{R}^d$ ,*

$$1146 \mathbb{E} [\|\mathbf{X} - \mathbf{y}\|_1] = \sum_{i=1}^d \left[ (y_i - \mu_i) \text{erf} \left( \frac{y_i - \mu_i}{\sigma_i \sqrt{2}} \right) + \sqrt{\frac{2}{\pi}} \sigma_i \exp \left( -\frac{(y_i - \mu_i)^2}{2\sigma_i^2} \right) \right].$$

1149 *Proof.* Let  $\mathbf{X} = (X_1, X_2, \dots, X_d)$ ,  $\mathbf{y} = (y_1, y_2, \dots, y_d)$ ,  $\mu = (\mu_1, \mu_2, \dots, \mu_d)$ , and  $\sigma =$   
 1150  $(\sigma_1, \sigma_2, \dots, \sigma_d)$ . We may assume that  $\sigma_1, \sigma_2, \dots, \sigma_d > 0$ . Then

$$1152 \mathbb{E} [\|\mathbf{X} - \mathbf{y}\|_1] = \mathbb{E} \left[ \sum_{i=1}^d |X_i - y_i| \right] = \sum_{i=1}^d \mathbb{E} [|X_i - y_i|].$$

1154 Define  $g_i(y) = \mathbb{E} [|X_i - y|]$ .

$$1156 g'_i(y) = \mathbb{E} \left[ \frac{d}{dy} |X_i - y| \right] = \mathbb{E} [\mathbf{1}_{X_i < y} - \mathbf{1}_{X_i > y}] = F_{X_i}(y) - (1 - F_{X_i}(y)) = 2F_{X_i}(y) - 1,$$

1158 where  $F_{X_i}$  is the cumulative distribution function of  $X_i$ . So

$$1160 g'_i(y) = \text{erf} \left( \frac{y - \mu_i}{\sigma_i \sqrt{2}} \right).$$

1162 Observe that

$$1163 g_i(\mu_i) = \mathbb{E} [|X_i - \mu_i|]$$

$$1164 = \frac{1}{\sqrt{2\pi}\sigma_i} \int_{\mu_i}^{\infty} (x_i - \mu_i) \exp \left( -\frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right) dx_i$$

$$1167 - \frac{1}{\sqrt{2\pi}\sigma_i} \int_{-\infty}^{\mu_i} (x_i - \mu_i) \exp \left( -\frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right) dx_i.$$

1169 Substituting  $u_i = (x_i - \mu_i)/\sigma_i$ ,

$$1171 g_i(\mu_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \left[ \int_0^{\infty} \sigma_i u_i e^{-\frac{1}{2}u_i^2} \sigma_i du_i - \int_{-\infty}^0 \sigma_i u_i e^{-\frac{1}{2}u_i^2} \sigma_i du_i \right]$$

$$1174 = \frac{\sigma_i}{\sqrt{2\pi}} \left[ \int_0^{\infty} u_i e^{-\frac{1}{2}u_i^2} du_i - \int_{-\infty}^0 u_i e^{-\frac{1}{2}u_i^2} du_i \right]$$

$$1177 = \sqrt{\frac{2}{\pi}} \sigma_i.$$

1178 By the fundamental theorem of calculus,

$$1180 g_i(y) = g_i(\mu_i) + \int_{\mu_i}^y g'_i(v) dv = \sqrt{\frac{2}{\pi}} \sigma_i + \int_{\mu_i}^y \text{erf} \left( \frac{v - \mu_i}{\sigma_i \sqrt{2}} \right) dv.$$

1182 Substituting  $z = (v - \mu_i)/(\sigma_i \sqrt{2})$ ,

$$1184 \int_{\mu_i}^y \text{erf} \left( \frac{v - \mu_i}{\sigma_i \sqrt{2}} \right) dv = \int_0^{(y - \mu_i)/(\sigma_i \sqrt{2})} \text{erf}(z) \sigma_i \sqrt{2} dz$$

$$1186 = \sqrt{2} \sigma_i \left[ \left( \frac{y - \mu_i}{\sigma_i \sqrt{2}} \right) \text{erf} \left( \frac{y - \mu_i}{\sigma_i \sqrt{2}} \right) + \frac{1}{\sqrt{\pi}} \exp \left( -\frac{(y - \mu_i)^2}{2\sigma_i^2} \right) - \frac{1}{\sqrt{\pi}} \right].$$

1188 Therefore,

$$1189 \quad g_i(y) = (y - \mu_i) \operatorname{erf} \left( \frac{y - \mu_i}{\sigma_i \sqrt{2}} \right) + \sqrt{\frac{2}{\pi}} \sigma_i \exp \left( -\frac{(y - \mu_i)^2}{2\sigma_i^2} \right),$$

1190 which implies

$$1191 \quad \mathbb{E} [\|\mathbf{X} - \mathbf{y}\|_1] = \sum_{i=1}^d g_i(y_i) = \sum_{i=1}^d \left[ (y_i - \mu_i) \operatorname{erf} \left( \frac{y_i - \mu_i}{\sigma_i \sqrt{2}} \right) + \sqrt{\frac{2}{\pi}} \sigma_i \exp \left( -\frac{(y_i - \mu_i)^2}{2\sigma_i^2} \right) \right].$$

1192  $\square$

## 1193 C PRACTICAL ALGORITHM

### 1194 C.1 STAGE I: BEHAVIOUR POLICY LEARNING

1195 In theory, each behaviour policy is well-defined on every state  $s \in \mathcal{S}$ . However, in practice, we can  
 1196 trust our estimations only in the vicinity of the states they were trained on. The problem is, we train  
 1197 each  $\hat{\beta}$  only on the states they are assigned to. Therefore, we need a mechanism to determine which  
 1198 behaviour policy estimates we can trust given a state  $s \in \mathcal{S}$ . For this purpose, we additionally train a  
 1199 classifier  $f_\psi: \mathcal{S} \rightarrow \mathcal{P}_d([K])$  using the computed assignments and determine the credible set by the  
 1200 equation

$$1201 \quad \hat{\mathcal{V}}(s) = \left\{ \hat{\beta}_i \in \hat{\mathcal{B}} : f_\psi(i \mid s) \geq \frac{1}{b} \max_{j \in [K]} f_\psi(j \mid s) \right\},$$

1202 where  $b > 0$  is a hyperparameter. We accordingly modify the definition of  $Q_{\hat{\mathcal{B}}}^{\text{LB}}$  to

$$1203 \quad Q_{\hat{\mathcal{B}}}^{\text{LB}}(s, a) = \max_{\hat{\beta} \in \hat{\mathcal{V}}(s)} Q_{\hat{\beta}}^{\text{LB}}(s, a),$$

1204 for all  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ .

### 1205 C.2 STAGE II: BEHAVIOUR VALUE LEARNING

1206 To learn the value functions of all  $K$  behaviour policies in parallel, we leverage a network  $V_\zeta: \mathcal{S} \rightarrow \mathbb{R}^K$  with  $K$  outputs. The per sample temporal difference (TD) loss function can be written by the  
 1207 equation

$$1208 \quad \ell_V(\zeta; s, r, s') = (V_\zeta(s)[\mathbb{A}(s)] - r - \gamma V_{\zeta'}(s')[\mathbb{A}(s')])^2,$$

1209 where  $s$ ,  $r$ , and  $s'$  are the state, reward, and next state sampled from the dataset, respectively,  $\zeta'$  is  
 1210 the target network parameter that is updated by polyak averaging as in Lillicrap et al. (2016), and  
 1211  $V_\zeta(s)[i]$  is the  $i$ -th coordinate of  $V_\zeta(s)$ . Note that  $\mathbb{A}(s)$ , the cluster assignment of  $s$ , is equal to  $\mathbb{A}(s')$ ,  
 1212 because we assign each trajectory to the same cluster.

### 1213 C.3 STAGE III: POLICY LEARNING

1214 In practice, it is infeasible to compute the supremum term in the penalised Bellman operator  $\mathcal{T}_{\hat{\mathcal{B}}}^*$ . We  
 1215 instead adopt the actor-critic formulation that alternates between the policy improvement step and  
 1216 the critic learning step. The goal of the policy improvement step is to find an action that maximises  
 1217 the critic for each state. The challenge is that the critic is highly non-convex due to the penalisation  
 1218 of critic values for actions between the means of the behaviour policies, causing gradient methods to  
 1219 yield suboptimal solutions. Hence, we search for the optimal action in the vicinity of each behaviour  
 1220 policy's mean simultaneously. This is done by training a network to output not the optimal action  
 1221 itself but the difference between the optimal action and one of the behaviour policy means. Utilising  
 1222 an ensemble of  $K$  neural networks  $g_{\psi_\pi^0}, g_{\psi_\pi^1}, \dots, g_{\psi_\pi^{K-1}}$ , we can compute  $K$  action candidates  
 1223  $a_0, a_1, \dots, a_{K-1}$ , where

$$1224 \quad a_i = \boldsymbol{\mu}_{\hat{\beta}_i}(s) + g_{\psi_\pi^i}(s),$$

1225 and  $\boldsymbol{\mu}_{\hat{\beta}}$  is the mean vector of the behaviour policy estimate  $\hat{\beta}$ . Then we choose the best action  $a_{i_s^*}$   
 1226 with respect to the current critic function  $Q_{\psi_Q}$ , that is,

$$1227 \quad i_s^* = \arg \max_{i \in \hat{\mathcal{I}}(s)} Q_{\psi_Q}(s, a_i),$$

1242 where

$$\hat{\mathcal{I}}(s) = \left\{ i \in [K] : \hat{\beta}_i \in \hat{\mathcal{V}}(s) \right\}.$$

1245 The loss function for the policy improvement step can be written by the following equation:

$$\ell_\pi(\psi_\pi; s) = -\max_{i \in \hat{\mathcal{I}}^*} Q_{\psi_Q} \left( s, \boldsymbol{\mu}_{\hat{\beta}_i}(s) + g_{\psi_\pi^i}(s) \right).$$

1249 The critic learning step has two objectives: minimising the TD error and penalising the OOD actions.  
1250 For the first objective, we adopt the conventional TD loss adapted to match the way we defined our  
1251 policy, which is represented by the equation

$$\ell_Q^{\text{TD}}(\psi_Q; s, a, r, s') = (Q_{\psi_Q}(s, a) - T(r, s'))^2,$$

1254 where  $s$ ,  $a$ ,  $r$ , and  $s'$  are the state, action, reward, and next state sampled from the dataset, respectively,  
1255 and the TD target  $T(r, s')$  is defined as

$$T(r, s') = r + \gamma \max_{\hat{\beta} \in \hat{\mathcal{V}}(s)} Q_{\psi_Q'} \left( s', \boldsymbol{\mu}_{\hat{\beta}}(s') + g_{\psi_\pi'}(s') \right).$$

1259  $\psi_Q'$  and  $\psi_\pi'$  in the preceding equation are the target critic network parameters and target actor net-  
1260 work parameters, respectively, which are updated by polyak averaging to gradually follow  $\psi_Q$  and  
1261  $\psi_\pi$ .

1262 The second objective of the critic learning step is to penalise the critic values of OOD actions to-  
1263 wards  $Q_{\hat{\mathcal{B}}}^{\text{LB}}(s, a)$ . In order to compute  $Q_{\hat{\mathcal{B}}}^{\text{LB}}$ , we need to compute  $Q_{\hat{\beta}}^{\text{LB}}$  for each  $\hat{\beta} \in \hat{\mathcal{B}}$ . However,  
1264 computing  $Q_{\hat{\beta}}^{\text{LB}}$  is not straightforward, due to the term  $\mathbb{E}_{a' \sim \hat{\beta}(s)} [\|a - a'\|]$  in (7). We discovered  
1266 that if we use a 1-norm and assume that  $\hat{\beta}$  has a diagonal covariance matrix, the expectation has the  
1267 following closed-form expression (Proposition 14):

$$\mathbb{E}_{a' \sim \hat{\beta}(s)} [\|a - a'\|_1] = \sum_{i=1}^{d_a} (y_i - \mu_i(s)) \operatorname{erf} \left( \frac{a_i - \mu_i(s)}{\sigma_i(s) \sqrt{2}} \right) + \sqrt{\frac{2}{\pi}} \sum_{i=1}^{d_a} \sigma_i \exp \left( -\frac{(a_i - \mu_i(s))^2}{2\sigma_i^2(s)} \right),$$

1271 where  $a = (a_1, a_2, \dots, a_{d_a})$  and  $\hat{\beta}(s)$  is a Gaussian distribution with a state-dependent mean vector  
1272  $\boldsymbol{\mu}(s) = (\mu_1(s), \mu_2(s), \dots, \mu_{d_a}(s))$  and a state-dependent covariance matrix whose main diagonal  
1273 is  $\boldsymbol{\sigma}(s) = (\sigma_1(s), \sigma_2(s), \dots, \sigma_{d_a}(s))$ . For  $r_{\min}$  and  $r_{\max}$ , following Mao et al. (2023), we estimate  
1275 them by the minimum and maximum rewards in all of the datasets of a given task, that is, for example  
1276  $r_{\min}$  and  $r_{\max}$  for a hopper-v2 dataset is computed by the minimum and maximum of the rewards  
1277 in hopper-expert-v2, hopper-medium-v2, and hopper-random-v2. To sum up, the loss function for  
1278 regularisation is

$$\ell_Q^{\text{reg}}(\psi_Q; \tilde{s}, \tilde{a}) = \mathbf{1}_{\tilde{a} \in \text{OOD}(\tilde{s})} (Q_{\psi_Q}(\tilde{s}, \tilde{a}) - Q_{\hat{\mathcal{B}}}^{\text{LB}}(\tilde{s}, \tilde{a}))^2,$$

1280 where  $\tilde{s}$  is a state sampled from the dataset and  $\tilde{a}$  is an action sampled from  $\pi_{\text{alg}}$ . The resulting total  
1281 loss can be written by the following equation

$$\ell_Q(\psi_Q) = \mathbb{E}_{(s, a, r, s') \sim \mathcal{D}} [\ell_Q^{\text{TD}}(\psi_Q; s, a, r, s')] + w_Q \mathbb{E}_{\tilde{s} \sim \mathcal{D}, \tilde{a} \sim \pi_{\text{alg}}(\cdot | \tilde{s})} [\ell_Q^{\text{reg}}(\psi_Q; \tilde{s}, \tilde{a})], \quad (13)$$

1285 where  $w_Q$  is a tunable hyperparameter. For  $\pi_{\text{alg}}$  we adopted

$$\pi_{\text{alg}}(a | s) = \frac{1}{2} \tilde{\beta}(a | s) + \frac{1}{2} \pi(a | s),$$

1288 where  $\pi$  is the current policy and  $\tilde{\beta}$  is defined as

$$\tilde{\beta}(\cdot | s) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\beta}^*}(s), \kappa^2 \boldsymbol{\Sigma}_{\hat{\beta}^*}(s)).$$

1290 Here,  $\boldsymbol{\mu}_{\hat{\beta}^*}(s)$  and  $\boldsymbol{\Sigma}_{\hat{\beta}^*}(s)$  are the mean vector and covariance matrix of the selected behaviour policy  
1292  $\hat{\beta}^*$  and  $\kappa$  is a tunable hyperparameter, which we set to 2 in most of our experiments. The first term  
1293 regularises the critic values over a broad range of actions to guide a randomly initialised network  
1294  $g_{\psi_\pi}$  towards producing near-zero values. The second term regularises the critic values in the vicinity  
1295 of the current policy allowing delicate control near the boundary of  $\text{OOD}(s)$ .

1296 C.4 IMPLEMENTATION DETAILS  
1297

1298 We standardised the observations following Fujimoto & Gu (2021) and normalised the rewards  
1299 so that the average empirical discounted return becomes  $\hat{V}$ , which we set to 100 for most of our  
1300 experiments. The algorithm was implemented upon the JAX (Bradbury et al., 2018) framework  
1301 using the Flax (Heek et al., 2024) library. The scikit-learn (Pedregosa et al., 2011) library was used  
1302 to compute ARIs and NMIs for Sections 5.3 and D.2.

1303

1304 D EXPERIMENT DETAILS  
13051306 D.1 MOTIVATION EXPERIMENT  
1307

1308 For the experiment in Section 3, we created a single-state infinite-horizon MDP with  $\gamma = 0.9$  and a  
1309 deterministic reward function

$$1310 R(\mathbf{a}) = \frac{1}{2} \exp(\|\mathbf{a} - \mathbf{a}^*\|_2^2).$$

1311 The optimal action is obviously  $\mathbf{a}^*$  and the optimal discounted return is  $1/(1 - \gamma) = 10$ . Since it  
1312 is a single-state MDP, we adopted a trainable six-dimensional vector instead of a policy network.  
1313 Similarly, the critic network  $Q_\psi: \mathcal{A} \rightarrow \mathbb{R}$  only takes the action as its input. The loss function  
1314 functions for the two algorithms are

$$1316 \ell_{\text{IS}}(\psi) = \mathbb{E}_{(r, \mathbf{a}) \sim \mathcal{D}} \left[ (Q_\psi(\mathbf{a}) - r - \gamma Q_{\bar{\psi}}(\mathbf{a}_\phi))^2 + \rho_\phi(\mathbf{a}) Q_\psi(\mathbf{a})^2 + \alpha \mathbb{E}_{\mathbf{a}' \sim \mathcal{N}(\mathbf{a}, \sigma^2 \mathbf{I}_6)} [Q_\psi(\mathbf{a}')^2] \right]$$

$$1318 \ell_{\text{HT}}(\psi) = \mathbb{E}_{(r, \mathbf{a}) \sim \mathcal{D}} \left[ (Q_\psi(\mathbf{a}) - r - \gamma Q_{\bar{\psi}}(\mathbf{a}_\phi))^2 + \alpha \mathbb{E}_{\mathbf{a}' \sim \pi_{\text{alg}}} [\mathbf{1}_{\mathbf{a}' \in \text{OOD}} Q_\psi(\mathbf{a}')^2] \right],$$

1320 where  $\mathbf{a}_\phi$  is the current policy,  $\bar{\psi}$  is the target critic parameter updated in an EMA fashion,  $\rho_\phi(\mathbf{a})$  is  
1321 the importance sampling ratio defined as

$$1322 \rho_\phi(\mathbf{a}) = \frac{\mu_\phi(\mathbf{a})}{\beta(\mathbf{a})} = \frac{1}{\sigma^6} \exp \left( \frac{1}{2} \|\mathbf{a}\|^2 - \frac{1}{2\sigma^2} \|\mathbf{a} - \mathbf{a}_\phi\|^2 \right),$$

1325  $\pi_{\text{alg}}$  is the sampling policy defined as

$$1327 \pi_{\text{alg}} = \frac{1}{2} \mathcal{N}(\mathbf{0}, \mathbf{I}_6) + \frac{1}{2} \mathcal{N}(\mathbf{a}_\phi, \sigma^2 \mathbf{I}_6),$$

1329 and

$$1330 \text{OOD} = \left\{ \mathbf{a} \in \mathbb{R}^6 : \|\mathbf{a}\|_2^2 \leq F_{\chi_6^2}^{-1}(0.5) \right\}.$$

1331 For the actor loss, we used

$$1332 \ell_\pi(\phi) = -Q_\psi(\mathbf{a}_\phi)$$

1334 in both cases.

1335 D.2 ADDITIONAL ANALYSIS ON THE TRAJECTORY CLUSTERING ALGORITHM  
1336

1337 Aside from the three random-medium-expert datasets mentioned in Section 5.3, we also created  
1338 custom D4RL datasets by concatenating medium and expert datasets of halfcheetah, hopper, and  
1339 walker2d tasks. The mean and standard deviation of ARIs and NMIs for each configuration over 5  
1340 different seeds are reported in Table 7. The configuration  $(\lambda_T, \lambda_R) = (1, 0)$  performs the best on  
1341 average even after we include the three medium-expert datasets.

1342 We also provide visualisations of our clustering results on different datasets of the D4RL benchmark  
1343 (Figures 4–9). Each row represents a different trajectory in the dataset.

1344

1345

1346

1347

1348

1349

1350  
 1351 Table 7: The impact of hyperparameters  $\lambda_T$  and  $\lambda_R$  on the performance of our trajectory clustering  
 1352 algorithm evaluated on custom D4RL datasets. The performance is measured in terms of adjusted  
 1353 rand index (ARI) and normalised mutual information score (NMI).  
 1354

	$\lambda_T = 1$		$\lambda_T = 0$	
	$\lambda_R = 1$	$\lambda_R = 0$	$\lambda_R = 1$	$\lambda_R = 0$
halfcheetah-medium-expert				
ARI	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00
NMI	1.00 $\pm$ 0.00	0.99 $\pm$ 0.01	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00
halfcheetah-random-medium-expert				
ARI	0.97 $\pm$ 0.04	0.99 $\pm$ 0.00	0.91 $\pm$ 0.19	0.91 $\pm$ 0.17
NMI	0.97 $\pm$ 0.03	0.97 $\pm$ 0.01	0.93 $\pm$ 0.12	0.92 $\pm$ 0.12
hopper-medium-expert				
ARI	0.80 $\pm$ 0.44	1.00 $\pm$ 0.00	0.99 $\pm$ 0.01	1.00 $\pm$ 0.00
NMI	0.80 $\pm$ 0.42	1.00 $\pm$ 0.00	0.97 $\pm$ 0.03	1.00 $\pm$ 0.00
hopper-random-medium-expert				
ARI	0.49 $\pm$ 0.29	0.98 $\pm$ 0.02	0.59 $\pm$ 0.33	0.97 $\pm$ 0.06
NMI	0.57 $\pm$ 0.26	0.97 $\pm$ 0.02	0.66 $\pm$ 0.29	0.98 $\pm$ 0.04
walker2d-medium-expert				
ARI	0.99 $\pm$ 0.01	1.00 $\pm$ 0.00	0.99 $\pm$ 0.02	1.00 $\pm$ 0.00
NMI	0.99 $\pm$ 0.02	1.00 $\pm$ 0.01	0.98 $\pm$ 0.04	1.00 $\pm$ 0.00
walker2d-random-medium-expert				
ARI	0.88 $\pm$ 0.16	0.98 $\pm$ 0.05	0.98 $\pm$ 0.03	1.00 $\pm$ 0.00
NMI	0.90 $\pm$ 0.11	0.98 $\pm$ 0.03	0.97 $\pm$ 0.04	1.00 $\pm$ 0.00
Average		ARI	0.86 $\pm$ 0.27	0.99 $\pm$ 0.02
		NMI	0.87 $\pm$ 0.24	0.98 $\pm$ 0.02
				0.98 $\pm$ 0.07
				0.98 $\pm$ 0.06

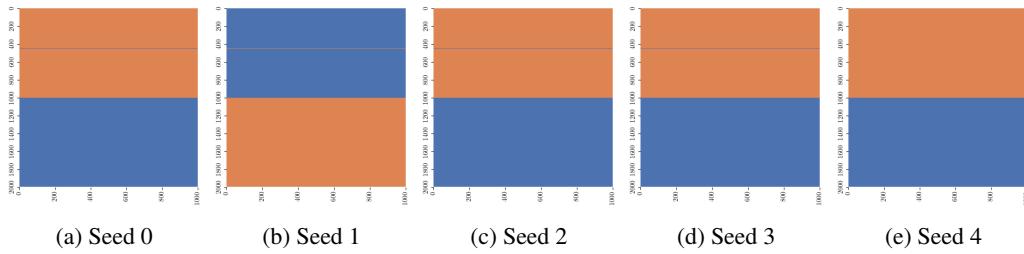


Figure 4: Clustering visualisations for halfcheetah-medium-expert-v2

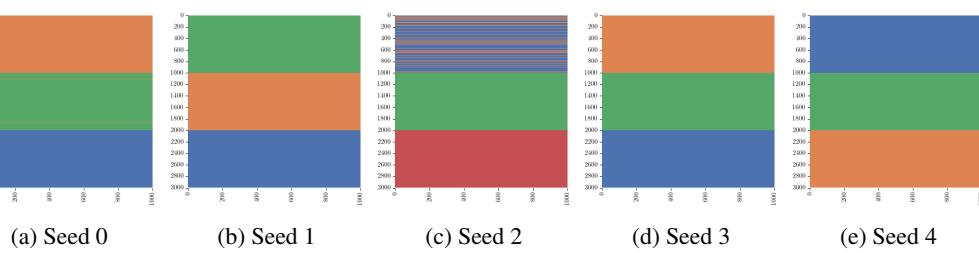


Figure 5: Clustering visualisations for halfcheetah-random-medium-expert-v2

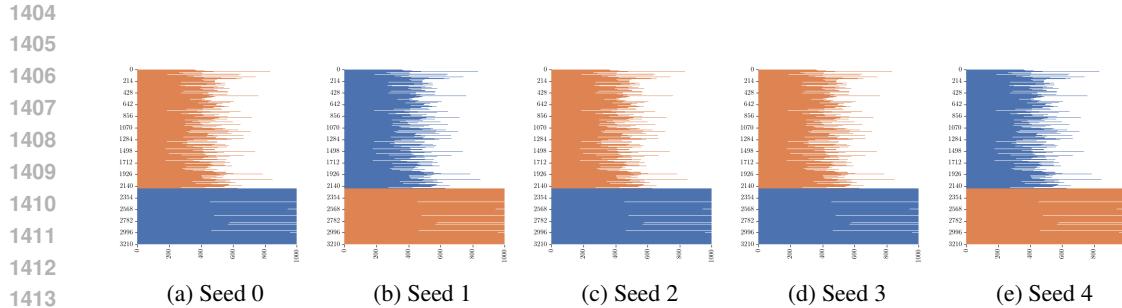


Figure 6: Clustering visualisations for hopper-medium-expert-v2

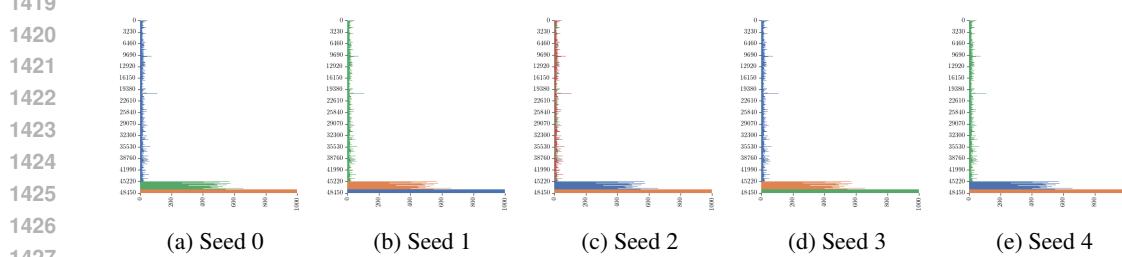


Figure 7: Clustering visualisations for hopper-random-medium-expert-v2

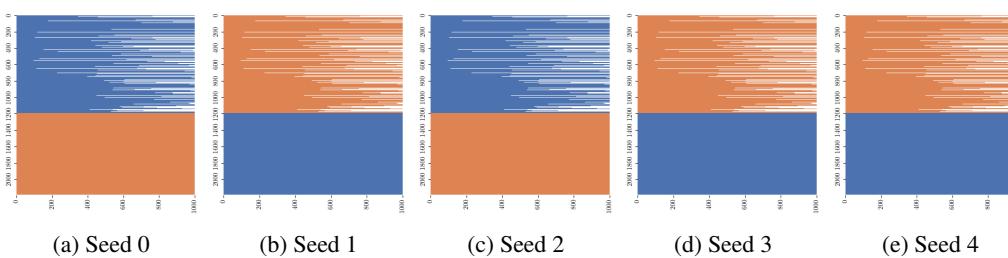


Figure 8: Clustering visualisations for walker2d-medium-expert-v2

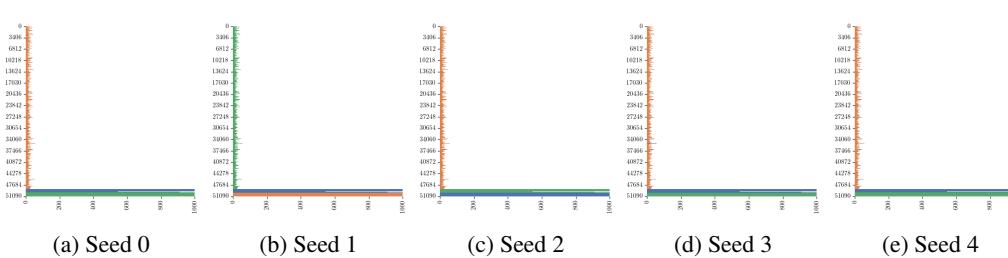
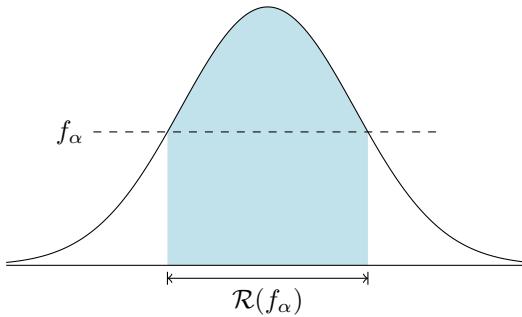
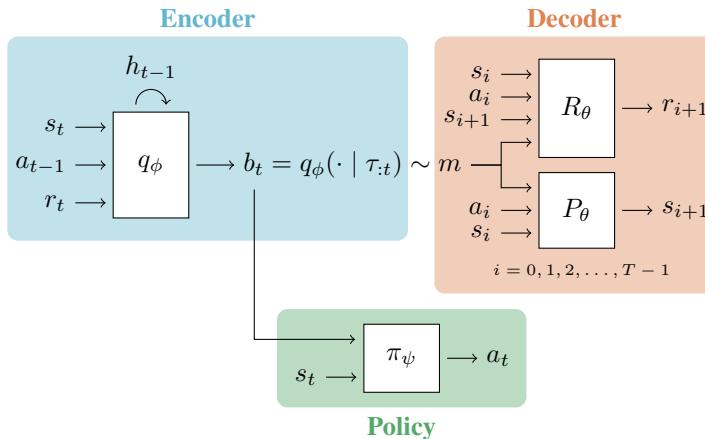


Figure 9: Clustering visualisations for walker2d-random-medium-expert-v2

1458 E ADDITIONAL FIGURES  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
14701471  
1472 Figure 10: A diagram showing the probability density function and the  $100(1 - \alpha)\%$  highest density  
1473 region of a normal distribution. The probability of the corresponding normal random variable to lie  
1474 inside  $\mathcal{R}(f_\alpha)$ , which corresponds to the area of the coloured region, is  $1 - \alpha$ .  
1475  
1476  
14771478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492 Figure 11: VariBAD architecture. This figure is a redrawn version of Figure 2 in Zintgraf et al.  
1493 (2021).  
1494  
1495  
1496  
1497

## F RELATED WORK

1498 **Value regularisation** Value regularisation aims to discourage the actor from choosing OOD actions by penalising their critic values. Conservative Q learning (CQL; Kumar et al. 2020) was one of the first works in this line of research, where they minimise the standard TD error together with the Q-values of OOD actions. Lyu et al. (2022) pointed out that the CQL excessively regularises the OOD Q-values to the extent that hampers the learning process. They suggested a milder regularisation term based on the critic values of ID actions. Supported value regularisation (SVR; Mao et al. 2023) proposed a penalisation scheme that maintains standard Bellman updates for ID actions while selectively penalising OOD actions' critic values. Most existing value regularisation algorithm, including the three works introduced in this section, sample the OOD actions from the current policy. However, as training progresses, the current policy will start to produce ID samples, so it is crucial to prevent unnecessary penalisation for those actions. CQL circumvents this issue through maximising the critic values for actions in the dataset. SVR does it by soft thresholding the regulariser based on the importance sampling ratio. In contrary, our method adopts a hard thresholding mechanism where ID actions are not penalised at all. This is possible due to our capability of explicitly identifying the OOD action set.

1512 **Heterogeneous datasets** There are multiple prior work concerned with offline RL datasets with  
 1513 heterogeneous behaviours. Wang et al. (2023) utilises a diffusion model (Sohl-Dickstein et al., 2015;  
 1514 Ho et al., 2020) to capture the multi-modality of the true behaviour policy. Li et al. (2023) trains  
 1515 a mixture of Gaussian policy on the dataset via likelihood maximisation and then obtains a closed-  
 1516 form estimate of the best possible action near the behaviour policy. These two works ignores the  
 1517 trajectory information and handles each transition individually. Mao et al. (2024) incorporates an  
 1518 expectation–maximisation algorithm to learn diverse policies from a given offline RL dataset. Wang  
 1519 et al. (2024) proposes a learning-based trajectory clustering algorithm that can also automatically  
 1520 determine the cluster size. Although these two works leverage the trajectory information, they obtain  
 1521 the trajectory representation by simply averaging the samples, causing a substantial loss of informa-  
 1522 tion. We incorporate a sequence modelling technique instead to learn an effective representation of  
 1523 each trajectory.

1524 **VQ-VAE** State-conditioned action quantisation (SAQ; Luo et al. 2023) is closely related to our  
 1525 work in the sense that they also leverage a VQ-VAE in the offline RL setting. However, their main  
 1526 focus is to discretise the actions because most of the challenges in offline RL originates from the  
 1527 ambiguity of continuous distributions. On the other hand, our algorithm uses VQ-VAE to cluster  
 1528 the trajectories and recover the behaviour policies. Also, SAQ discretises the actions individually,  
 1529 ignoring the trajectory information.

## 1531 G LIMITATIONS

1533 Our work is built upon the assumption that each trajectory in the dataset was sampled from a single  
 1534 behaviour policy. Although this assumption does not hold in general, as the behaviour policy may  
 1535 change mid-trajectory, the change is subtle enough for our algorithm to perform reasonably well.  
 1536 However, this might not be the case for real world scenarios. Future work could explore mechanisms  
 1537 to detect behaviour policy change and split the trajectory at those transition points.

1538  
 1539  
 1540  
 1541  
 1542  
 1543  
 1544  
 1545  
 1546  
 1547  
 1548  
 1549  
 1550  
 1551  
 1552  
 1553  
 1554  
 1555  
 1556  
 1557  
 1558  
 1559  
 1560  
 1561  
 1562  
 1563  
 1564  
 1565