




# Hard Negative Sample Mining for Contrastive Representation in Reinforcement Learning

Qihang Chen, Dayang Liang, and Yunlong Liu<sup>(✉)</sup> 

Department of Automation, Xiamen University, Xiamen, China  
ylliu@xmu.edu.cn

**Abstract.** In recent years, contrastive learning has become an important technology of self-supervised representation learning and achieved SOTA performances in many fields, which has also gained increasing attention in the reinforcement learning (RL) literature. For example, by simply regarding samples augmented from the same image as positive examples and those from different images as negative examples, instance contrastive learning combined with RL has achieved considerable improvements in terms of sample efficiency. However, in the contrastive learning-related RL literature, the source images used for contrastive learning are sampled in a completely random manner, and the feedback of downstream RL task is not considered, which may severely limit the sample efficiency of the RL agent and lead to sample bias. To leverage the reward feedback of RL and alleviate sample bias, by using gaussian random projection to compress high-dimensional image into a low-dimensional space and the Q value as a guidance for sampling the hard negative pairs, i.e. samples with similar representation but diverse semantics that can be used to learn a better contrastive representation, we propose a new negative sample method, namely Q value-based Hard Mining (QHM). We conduct experiments on the DeepMind Control Suite and show that compared to the random sample manner in vanilla instance-based contrastive method, our method can effectively utilize the reward feedback in RL and improve the performance of the agent in terms of both sample efficiency and final scores, on 5 of 7 tasks.

**Keywords:** Contrastive learning · Reinforcement learning · Random projection · Hard negative sample mining

## 1 Introduction

With the development of deep neural network and by combining its feature extraction power with the decision ability of reinforcement learning (RL), Deep RL (DRL) has been widely and successfully applied to dozens of tasks with high dimensional inputs [6]. However, how to endow an agent with the ability to quickly master the task with less interactions is still a challenge nowadays. While

model-based RL algorithms try to build and maintain an environment model which will help agent planning and making full use of precious interaction data, they usually suffer from enormous computations for planning and fragile model accuracy [7]. On the other side, model free RL algorithms directly learn from raw observation space, but plenty of training data is indispensable for good performance [8]. It's widely believed that a policy directly trained from the real state data will act better than those with raw and high dimensional inputs [9]. Hence a key to efficiency promotion for model free RL agents is to acquire good representations [1]. Lots of auxiliary tasks have been incorporated in traditional RL algorithms to accelerate better representation learning, such as auto-encoder [10], prediction [17, 18, 37], prototypes [11], goals [5] and so on. Recently, contrastive learning has made significant progress in natural language processing and computer vision areas [12, 32], which has also been incorporated into RL algorithms. Srinivas et al. proposed an instance contrastive method called Contrastive Unsupervised Representations for Reinforcement Learning (CURL), which is the first contrastive based algorithm that beats model-based methods like PlaNet [15] and Dreamer [16] on several tasks of DeepMind Control Suite (DMC) [9].

As a self-supervised learning method, contrastive learning tries to define and contrast semantically similar (positive) pairs and semantically dissimilar (negative) pairs in the embedding space [18]. The success of contrastive methods mainly depends on the design of correct positive and negative pairs [19]. In RL setting, naively we can regard the real states as the label of observations and let contrastive learning gather samples of the same state and pushes away those with different states. However, due to the limitation of perception, agents generally can not access to the real states. Under such circumstances, existing contrastive RL methods design positive or negative pairs in a random or unsupervised way, where the feedback of RL tasks is completely ignored and positive samples may sneak into negative samples. Such false-negative phenomenon is known as *sampling bias*. It may empirically induce to significant performance deterioration in some fields [20].

Moreover, a plenty of work in metric learning believe that *hard negative samples* dominate the quality and efficiency of the representation learning [22, 36], where *hard negative samples* are the true negative samples that mapped nearby the anchor sample in the embedding space [21]. To mine hard negative samples and improve the sample efficiency of RL agents, by observing that hard negative samples are samples that look similar but with different semantics, in this paper, we propose a new hard negative samples mining method, namely Q-value based Hard Mining (QHM).

In the detail, in order to seek for samples embedded similarly, gaussian random projection and KD-Tree are firstly applied in QHM for dimensionality reduction and search. Then to further filtering semantically different pairs among these similar samples, as real states are inaccessible, we can take the advantage of cumulative reward Q value as a guidance for mining since it is the most key feedback of any RL tasks. Consequently, with the assistance of a handy K-means cluster method, QHM approximately treats those similar observation-action pairs but with different Q value in recent trajectories as hard negative

samples pairs. Equipped with these unsupervised techniques, QHM is able to latch the RL task feedback and efficiently solve two key problems: (1) How to design task-relevant positive-negative pairs for contrastive representation learning in RL? (2) How to mine and exploit hard negative samples?

We conduct experiments on DMC and show that compared to contrastive learning with vanilla random sample method, our sample method combined with instance-based contrastive learning in RL can achieve better data efficiency and even better score performance on several tasks.

## 2 Related Work

### 2.1 Improving Sample Efficiency in RL

It is well known that learning policy directly from high dimensional data such as raw pixel images is inefficient [2]. Model-based RL agent builds an environment model and generates virtual rollouts to help better decisions, which is usually more data efficient than model-free agent. The related methods like SimPLe [2], PlaNet [15] and Dreamer [16] have successfully improve the data efficiency in Atari Games [13] and DMC [9], and even make a breakthrough on some challenging tasks such as MONTEZUMA’S REVENGE. For the model-free approaches, to improve performance and efficiency, the agent mainly focus on constructing and adopting various auxiliary tasks, such as predicting future [17, 18, 37], prototypes cluster [11], particle-based entropy maximization [4] or multi-goals [5].

In recent years, contrastive learning is also incorporated into RL as an auxiliary task. Typical works include instance contrastive learning based CURL [1] and CPC [18] that leverages prediction information. Subsequent works also tried to use contrastive learning to force agents to learn temporal features [23]. Although these approaches have achieved some successes in various domains, the pairs in these contrastive learning methods are sampled in a random or unsupervised manner, and the possible signals that may help representation learning are not considered. In the work of Guoqing et al. [25], a return-based contrastive representation for RL (RCRL) method is introduced, where observation-action samples with similar cumulative rewards are regarded as positive pairs and vice versa. While the cumulative reward is used for the sampling in both RCRL and our approach, QHM further considers about the *hard* property of negative samples and uses a more adaptive manner to partition the experience buffer, where the explicit model structure or learning objective does not need to be changed.

Besides using auxiliary tasks for learning a better representation, recent work such as RAD [26], DrQ [24], DrQ-v2 [27] also show that simple combination of image augmentation is conducive to the improvement of data efficiency.

### 2.2 Sample Strategy in Unsupervised Contrastive Learning

Contrastive learning encourages semantically similar pairs  $(x, x^+)$  to be close and semantically dissimilar pairs  $(x, x^-)$  to be more distant in embedding space

$f(*)$  [34]. Since the labels of data are unknown under unsupervised conditions, the main differences among these methods are their strategies of obtaining positive and negative pairs [20]. In the literature, strategies including random crop, jittering in images [26, 33] and random dropout in text missions [31] are commonly used to select positive samples, while less attention has been paid in the sampling of negative pairs and they are only simply sampled uniformly from the training data [19]. There exists two problems in randomly picking negative pairs. First, false negative samples will give rise to *sample bias* which is impossible to completely dismiss under unsupervised situations [20]. Second, we cannot ensure how informative the negative samples will be when they serve the downstream tasks [19]. The key to address the mentioned issues is hard negative mining, which in metric learning is well elucidated and proved to be most helpful for efficient representation learning [22]. But how to mine such hard negative samples for unsupervised contrastive learning? Based on Debaised Contrastive Loss (DCL) [20], Robinson et al. proposed to define the priority of a sample proportional to its similarity with the anchor to acquire hard samples, which has made a certain progress in images and sentences representations [19]. Wang et al. found that the choice of temperature  $\tau$  in contrastive loss controls the granularity of penalties on hard negative samples [35].

### 3 Background

#### 3.1 Instance Contrastive Learning in RL

In general, considering an embedding space  $f(*)$ , contrastive learning tries to gather the representations of positive pairs  $(x, x^+)$  but push away the representations of negative pairs  $(x, x^-)$ :

$$\mathbb{E}_{x, x^+, \{x_i^-\}_{i=1}^N} \left[ -\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + \sum_{i=1}^N e^{f(x)^T f(x^-)}} \right] \quad (1)$$

Given an anchor  $x$ , a corresponding positive sample  $x^+$  and  $N$  negative samples  $x^-$  will be used for contrast. For *Instance Discrimination* [14],  $x$  and  $x^+$  are different views generated from the same sample whilst  $x$  and  $x^-$  are from different samples. CURL is the first method that combines *Instance Discrimination* with RL where views of different images are accomplished by random crop. In detail, given a batch of randomly sampled  $K$  raw-pixel images, for each of them  $x_i (1 \leq i \leq K)$ , we have:

$$x_{i1} = \text{aug}(x_i) \quad x_{i2} = \text{aug}(x_i) \quad (2)$$

where  $\text{aug}(*)$  represents a fixed random method of data augmentation such as random crop. CURL simply takes samples  $(x_{i1}, x_{i2})$  as positive pairs and  $(x_{i*}, x_{j*}) (j \neq i)$  as negative pairs according to whether they are generated from the same image. All these generated  $2K$  samples will be used for the InfoNCE loss [18]:

$$\mathcal{L}_{CURL} = -\log \frac{e^{z_q^T W z_k^+}}{e^{z_q^T W z_k^+} + \sum_{i=1}^K e^{z_q^T W z_{k_i}^-}} \quad (3)$$

In Eq. (3),  $z_q$  are the encoded low-dimensional representations of cropped images  $x_{i1}$  through the query encoder  $f_{\theta_q}$  of the RL agent while  $z_k$  are from key encoder  $f_{\theta_k}$ . Query and key encoders share the same neural framework but have different parameters weights. Similar to Moco [12], CURL detaches the gradient of the key encoder  $f_{\theta_k}$  whose parameters  $\theta_k$  can be only updated by exponentially moving average (EMA) method as follow:

$$\theta_k = m\theta_k + (1 - m)\theta_q \quad (4)$$

where  $m \in [0, 1]$  is a factor of trading off and such an update method has been proved to be helpful in improving agent’s performance and avoiding model collapse [1, 12].

In DMC tasks, CURL takes a SAC agent as the base policy learner by using Eq. (3) to learn contrastive representations. Our work will build upon CURL and aim to improve the completely random sample method to acquire more task-relevant negative samples for the calculation of Eq. (3).

### 3.2 Gaussian Random Projection

Gaussian random projection is a simple and convenient projection method to reduce high-dimensional space to low dimension. It defines a mapping function  $\phi : x \rightarrow Px \in \mathbb{R}^F$  where  $x \in \mathbb{R}^D$  is the original data with dimension  $D$  and will be multiplied by a random initialized matrix  $P \in \mathbb{R}^{F \times D}$  to be transformed to a space of dimension  $F$ . Generally we have  $F \ll D$  and each element  $P_{ij}$  in  $P$  are sampled independently from a predefined gaussian distribution  $N(\mu, \sigma^2)$ . According to Johnson-Lindenstrauss lemma [28], for arbitrary  $x_i$  and  $x_j$ , there exists a  $\epsilon (0 \leq \epsilon \leq 1)$  and a map function  $\phi$  that satisfy:

$$(1 - \epsilon) \|x_i - x_j\| \leq \|\phi(x_i) - \phi(x_j)\| \leq (1 + \epsilon) \|x_i - x_j\| \quad (5)$$

As showed in Eq. (5), the distance relationship can be well preserved in the mapped low-dimensional space even with a random initialized matrix as long as  $D$  is sufficient [29]. In our method, in order to search for hard negative samples, a computation efficient search method is urgently-needed and as it is time consuming and computation expensive by directly searching in the raw pixel space, gaussian random projection are adopted for reducing the high-dimensional input to a low-dimensional space.

## 4 Q Value Based Hard Mining

In this section, we will introduce QHM, a contrastive learning sampling method based on cumulative rewards. Our intention is to improve the unreasonable random sampling method of contrastive learning in RL such as CURL and try to use

the reward feedback of a specific task to guide the sampling strategy of positive-negative pairs. So that the samples eventually used for contrastive training are semantically *mutually exclusive* in RL setting, which will also contribute to efficiency promotion. The most ideal result is that the samples divided into positive and negative pairs will not belong to the same real state.

#### 4.1 Construct Task-Relevant Positive and Negative Pairs in RL

In contrastive learning, given an arbitrary anchor sample  $x$ , a sample  $x^+$  is positive when it is semantically similar to anchor  $x$  and takes  $x^-$  as negative vice versa. In RL, considering an one-step observation  $o_t$  composed of successive images, through a given data augmentation method, accurate positive pairs can be guaranteed since we can simply generate two views of  $o_t$  and they do actually semantically matched. However, it is common to regard the augmentations of any two different observation  $o_{t1}$  and  $o_{t2}$  as negative pairs in existing contrastive method, where positive samples may be misdiagnosed as negative ones and that will inevitably lead to the *sample bias* problem and probably further, sample efficiency decline as mentioned before.

It is natural for an RL agent to distinguish different observations by their real states  $s$ , however, it is notoriously known that the real states is unavailable due to perceptual limitations in real world. When only high-dimensional observation  $o_t$  is available, we can turn to the most important feedback of RL tasks, i.e. Q value. Q value is the expected discount cumulative rewards after agent taking action  $a_t$  at observation  $o_t$ :  $Q(o_t, a_t) = \mathbb{E}(\sum_{\tau=t}^T \gamma^{\tau-t} r_\tau(o_\tau, a_\tau))$ , where  $\tau \in (0, 1]$  is the discount factor. Hence, to define pos-neg samples in RL, intuitively we have:

**Assumption 1.** *In RL, given a policy  $\pi_\psi : \mathbb{O} \rightarrow \mathbb{A}$  and **arbitrary** observation-action samples (augmented or not)  $(o_{t1}, \pi(a_{t1} | o_{t1}))$ ,  $(o_{t2}, \pi(a_{t2} | o_{t2}))$ , if they **share the same** Q value, we can approximately regard them as a positive pair.*

However, strict conditions are required for the establishment of this hypothesis including a perfect reward function of environment to disambiguate Q value. But on the contrary, we can define the negative samples:

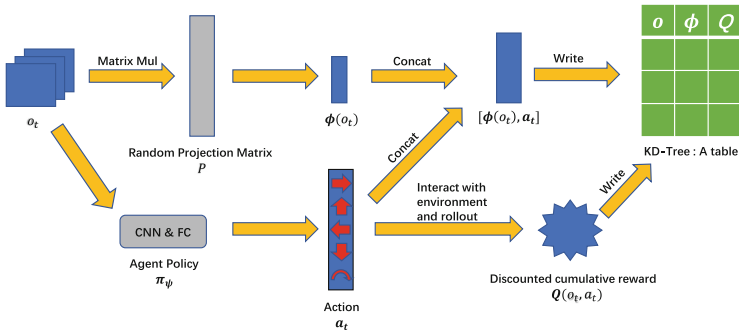
**Assumption 2.** *In RL, given a policy  $\pi_\psi : \mathbb{O} \rightarrow \mathbb{A}$  and **representations similar** observation-action samples (augmented or not)  $(o_{t1}, \pi(a_{t1} | o_{t1}))$ ,  $(o_{t2}, \pi(a_{t2} | o_{t2}))$ , if they have **quite different** Q values, we can approximately regard them as a negative pair.*

Please note that a sample mentioned above is composed of both observation and action. Such a definition for the negative pair is not perfect because of the uncertainty of the environment and the policy divergence, but it is still more reasonable than the random sampling method that is widely adopted for contrastive representation learning in RL. Since it is quite difficult to get the exact value of any  $Q(o_t, a_t)$ , in practice we simply use the cumulative rewards in historical trajectories to approximate Q.

### 4.2 Mine and Utilize Hard Negative Samples in RL

As mentioned, hard negative samples, i.e., the pairs with similar representation but different semantics are the key to efficient contrastive learning [21]. However, how to mine such samples from the data is still a challenging problem in the literature. In the following, how to mine and make use of hard negative samples in RL by using QHM will be introduced.

Given an anchor sample, it is infeasible to search the similar samples directly from the raw-pixel space due to heavy computational burden. We also should not search for the samples in agent’s encoder embedding space since frequent forward propagation in model may deteriorate the overall running time. In QHM, we just take the advantage of gaussian random projection to map raw-pixel images to a far-less dimensional space and subsequently a KD-Tree is utilized to execute  $k$ -nearest searching on the projection space. Specifically, KD-Tree is a table-like buffer which is independent of the agent’s replay buffer. Considering a gaussian projection function  $\phi(*)$ , QHM simply stores tuples  $\langle [\phi(o), a], o, Q(o, a) \rangle$  encountered in recent trajectories into the KD-Tree. Once the tree capacity hits the peak, samples visited most infrequently during training will be replaced. The specific process above is illustrated in Fig. 1.

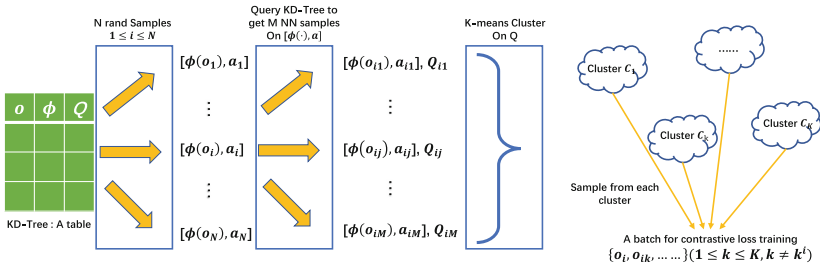


**Fig. 1.** KD tree storage process in QHM. For each observation  $o$ , QHM summarizes its cumulative reward  $Q$  in the trajectory after rollout and writes  $o$ ,  $z$  and  $Q$  into the KD tree where  $z$  is the concatenation of projection  $\phi(o)$  and corresponding action  $a$ .

In order to further screen hard negative samples, a simple  $K$ -means method is applied in QHM to cluster all these similar samples according to their  $Q$  value, and as all samples have been well scattered, QHM will eventually pick one sample at random from each cluster respectively. Then, all these left samples will share similar representations but with different  $Q$  values, which should be the hard negative samples we are seeking for. Please note that our QHM method mainly focus on the selection of the source images for negative samples generation. As for postive pairs, we adopt the same scheme as CURL, i.e. two views generating from a same image will be regard as positive to each other.

The implementation process is shown in Fig. 2: firstly QHM samples a batch of  $N$  samples  $x_i(1 \leq i \leq N) = [\phi(o_i), a_i, o_i, Q_i]$  at random from the KD-Tree. For each  $x_i$  of these  $N$  samples, QHM queries the KD-Tree for  $M$  nearest samples  $x_{ij}(1 \leq j \leq M)$  on  $\phi(*)$  to form a similar batch  $B_i^{in} = \{x_i, x_{i1}, x_{i2}, \dots, x_{iM}\}$  after absorbing the query sample  $x_i$ . Then K-Means cluster is applied based on their  $Q$  value to get  $K$  clusters  $C_k(1 \leq k \leq K)$ . Excluding the cluster  $C_{k^i}$  which the query sample  $x_i$  belongs to, we randomly pick one sample from each cluster to finally make up the hard negative batch  $B_i^{out} = \{x_i, x_{ik}, \dots\}(1 \leq k \leq K \ \& \ k \neq k^i)$  of  $x_i$ . Sequentially,  $N \times K$  samples will be acquired and cropped randomly to generate totally  $N \times K \times 2$  samples  $\{x_{ik1}, x_{ik2}, \dots\}(1 \leq i \leq N, 1 \leq k \leq K)$ , which will be used for contrastive loss Eq. (3) as following:

$$\mathcal{L}_{QHM-CURL} = \sum_{i=1}^N \mathcal{L}_{CURL}(x_{ik1}, x_{ik2}, \dots, x_{iK1}, x_{iK2}) \tag{6}$$



**Fig. 2.** The illustration of sample strategy in QHM. Firstly, several samples are sampled at random. For each of them, QHM queries the KD-Tree for  $M$  nearest samples to make up a state-similar batch including the query one. Then K-means cluster will conduct based on  $Q$  values to pick up the most divergent ones in each of these state-similar batches, which will finally form the hard negative samples batches for training.

## 5 Experiments

### 5.1 Environments

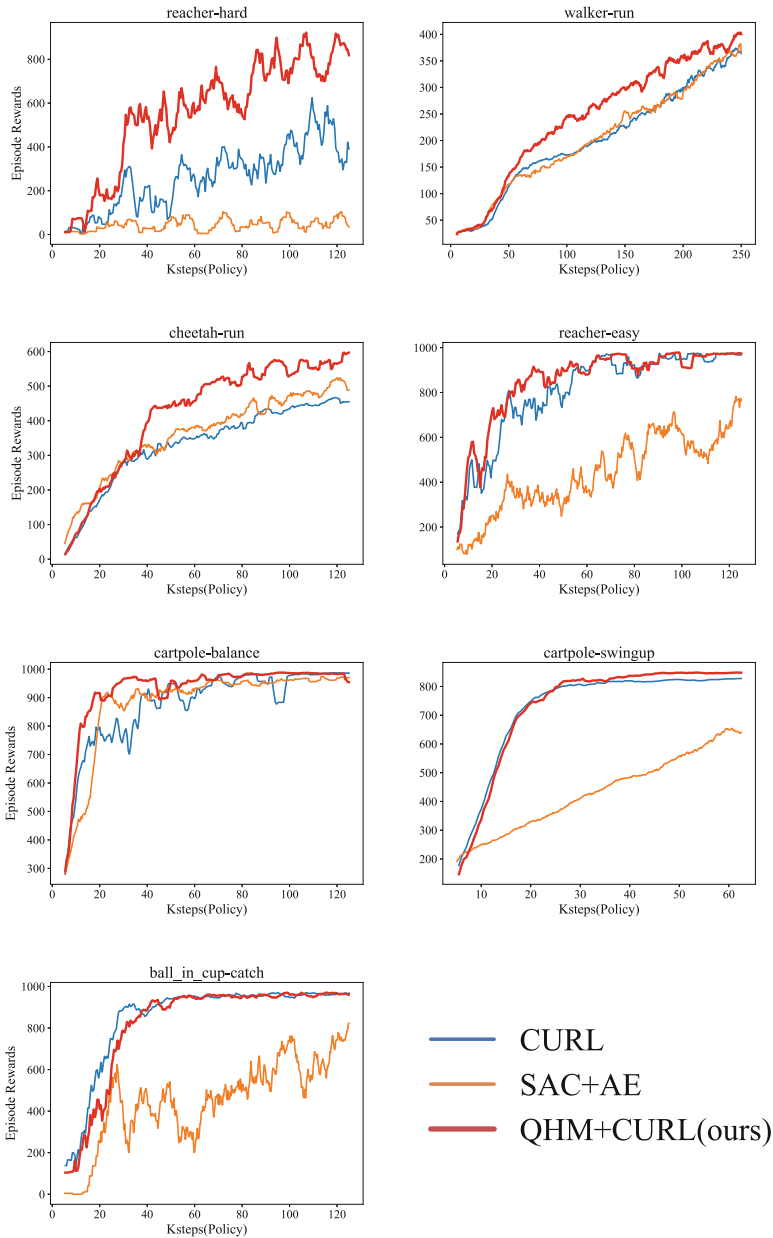
7 challenging tasks of DMC [9] are selected for evaluation. At every time step, the input of the agent is an 8-bits,  $100 \times 100$ , RGB image from the environment and 3 successive frames will be stacked as the observation to alleviate partially observable problems. And to accelerate training, the agent’s action repeat numbers is set as 8 for cartpole swingup, 2 for walker run and 4 for the rest respectively. Corresponding task policy step can be calculated by *total frames/action repeat*, which will be the abscissa of our experimental plot results.

### 5.2 Setting

QHM will be carried on CURL for experiments with the same neural network structure and hyperparameters. Specifically, encoders  $f_{\theta_q}$  and  $f_{\theta_k}$  composed of



successive convolution layers and full connection layers that are in charge of a mapping from raw-pixel images to embedding space with a dimensionality of 50. The capacity of replay buffer is  $100k$  and a batch of 512 samples will be randomly



**Fig. 3.** Evaluation scores results on 7 tasks of DeepMind Control Suite. QHM-CURL indicates CURL equipped with our method and vanilla CURL is our main competitor. SAC+AE is also included as a competitive method.

selected for SAC [3] updating. Adam [30] optimizer for update and  $84 \times 84$  random crop for image preprocessing. Specific QHM settings are shared across tasks: the dimensionality of the gaussian projection matrix  $P$  is  $[h, 9 \times 100 \times 100]$  where  $h$  is the projection dimension that we set 128 by default. Notes that the computational complexity of tree query is  $O(n^{(1-1/h)} + k)$ , where  $k$  is the population of total samples, it finally takes QHM almost twice as long as CURL to complete the training on the same device. Each element in projection matrix  $P$  is sampled independently from a predefined gaussian distribution  $N(0, (1/\sqrt{h})^2)$ . In KD-Tree, samples that are new-coming or frequently sampled for updating will be contained longer. In QHM sample strategy,  $N = 16, M = 31, K = 4$  are the numbers of query samples, M-nearest samples of query and clusters for K-Means, respectively. In practice we only take the last 16 samples of the  $M$  nearest of query for clustering in the next stage, which can effectively prevent identical samples. It’s also worth noticed that empty cluster may occur due to completely Q value duplicate. To get rid of biased negative samples, QHM will simply abandon this whole batch and the subsequent contrastive update.

### 5.3 Results

We conduct experiments on 7 different tasks of DMC and the supremum of tasks frames is limited to  $500k$  to evaluate the agent efficiency. The results are showed in Fig. 3. Every line is averaged over different random seeds and smoothed on abscissa interval. *QHM - CURL* represents the CURL [1] algorithm whose sampling strategy is replaced by our proposed QHM method. Hence, *CURL* is our main competitor which uses a completely random sample strategy. We also take another auxiliary task model-free method, *SAC+AE* [10], into account as a competitive baseline to further confirm the validity of our implementations. We implement CURL and SAC+AE from their official codes on *github* respectively.

As showed in Fig. 3, we can see that CURL combined with our sampling method QHM has superior sample efficiency and performance than vanilla CURL and SAC+AE on several tasks such as cheetah-run, walker-run and reacher-hard. All of these 3 tasks are defined in *medium* [27] subsets due to their greater action dimensions and  $500k$  environment steps are not yet sufficient for SAC-based agents to master them. In cartpole balance task, QHM-CURL acts more robust than baselines and has better convergence tendency. Most of the rest tasks are defined relatively *easy* [27]. Hence in these tasks, we can only see nuances among CURL and QHM-CURL.

Throughout all tasks results, we believe that our task-relevant hard negative mining strategy, QHM, can actually facilitate sample efficiency of the RL agent which may be suffering from the biased negative samples induced by a random sample strategy in contrastive-based reinforcement learning.

## 6 Conclusion

In this paper, we proposed QHM, a hard negative mining method dedicated to improving data-efficiency of RL agents. With the assistance of light components

such as KD-Tree, K-Means Cluster and Random Projection, when compared to vanilla instance-based contrastive sampling method, QHM can achieve further efficiency and even performance improvements on a certain number of tasks from DeepMind Control Suite. However, as in general we have no access to the real state of the environments, differentiating samples by their Q value stored would still be biased. There is a long way for us to acquire a near-real hard negative distribution and we leave this for future work. We believe that in the forthcoming future, better hard sampling strategies for contrastive learning in RL will be discovered and make significant contribution to representation learning.

**Acknowledgements.** This work was supported by the National Natural Science Foundation of China (No. 61772438 and No. 61375077). This work was also supported by the Innovation Strategy Research Program of Fujian Province, China (No. 2021R0012).

## References

1. Srinivas, A., Laskin, M., Abbeel, P.: CURL: contrastive unsupervised representations for reinforcement learning (2020)
2. Kaiser, L., et al.: Model-based reinforcement learning for atari. arXiv preprint [arXiv:1903.00374](https://arxiv.org/abs/1903.00374) (2019)
3. Haarnoja, T., et al.: Soft actor-critic algorithms and applications. arXiv preprint [arXiv:1812.05905](https://arxiv.org/abs/1812.05905) (2018)
4. Mutti, M., Pratissoli, L., Restelli, M.: A policy gradient method for task-agnostic exploration (2020)
5. Veeriah, V., Oh, J., Singh, S.: Many-goals reinforcement learning. arXiv preprint [arXiv:1806.09605](https://arxiv.org/abs/1806.09605) (2018)
6. Mnih, V., et al.: Playing atari with deep reinforcement learning. arXiv preprint [arXiv:1312.5602](https://arxiv.org/abs/1312.5602) (2013)
7. Talvitie, E.: Model regularization for stable sample rollouts. In: UAI (2014)
8. Racanière, S., et al.: Imagination-augmented agents for deep reinforcement learning. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (2017)
9. Tassa, Y., et al.: Deepmind control suite. arXiv preprint [arXiv:1801.00690](https://arxiv.org/abs/1801.00690) (2018)
10. Yarats, D., et al.: Improving sample efficiency in model-free reinforcement learning from images. arXiv preprint [arXiv:1910.01741](https://arxiv.org/abs/1910.01741) (2019)
11. Yarats, D., et al.: Reinforcement learning with prototypical representations. arXiv preprint [arXiv:2102.11271](https://arxiv.org/abs/2102.11271) (2021)
12. Chen, X., et al.: Improved baselines with momentum contrastive learning. arXiv preprint [arXiv:2003.04297](https://arxiv.org/abs/2003.04297) (2020)
13. Bellemare, M.G., et al.: The arcade learning environment: an evaluation platform for general agents. *J. Artif. Intell. Res.* **47**, 253–279 (2013)
14. Wu, Z., et al.: Unsupervised feature learning via non-parametric instance-level discrimination. arXiv preprint [arXiv:1805.01978](https://arxiv.org/abs/1805.01978) (2018)
15. Hafner, D., et al.: Learning latent dynamics for planning from pixels. In: International Conference on Machine Learning. PMLR (2019)
16. Hafner, D., et al.: Dream to control: learning behaviors by latent imagination. arXiv preprint [arXiv:1912.01603](https://arxiv.org/abs/1912.01603) (2019)

17. Lee, K.-H., et al.: Predictive information accelerates learning in RL. arXiv preprint [arXiv:2007.12401](https://arxiv.org/abs/2007.12401) (2020)
18. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint [arXiv:1807.03748](https://arxiv.org/abs/1807.03748) (2018)
19. Robinson, J., et al.: Contrastive learning with hard negative samples. arXiv preprint [arXiv:2010.04592](https://arxiv.org/abs/2010.04592) (2020)
20. Chuang, C.-Y., et al.: Debaised contrastive learning. arXiv preprint [arXiv:2007.00224](https://arxiv.org/abs/2007.00224) (2020)
21. Le-Khac, P.H., Healy, G., Smeaton, A.F.: Contrastive representation learning: a framework and review. *IEEE Access* **8**, 193907–193934(2020)
22. Suh, Y., et al.: Stochastic class-based hard example mining for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
23. Zhu, J., et al.: Masked contrastive representation learning for reinforcement learning. arXiv preprint [arXiv:2010.07470](https://arxiv.org/abs/2010.07470) (2020)
24. Kostrikov, I., Yarats, D., Fergus, R.: Image augmentation is all you need: regularizing deep reinforcement learning from pixels. arXiv preprint [arXiv:2004.13649](https://arxiv.org/abs/2004.13649) (2020)
25. Liu, G., et al.: Return-based contrastive representation learning for reinforcement learning. arXiv preprint [arXiv:2102.10960](https://arxiv.org/abs/2102.10960) (2021)
26. Laskin, M., et al.: Reinforcement learning with augmented data. arXiv preprint [arXiv:2004.14990](https://arxiv.org/abs/2004.14990) (2020)
27. Yarats, D., et al.: Mastering visual continuous control: improved data-augmented reinforcement learning. arXiv preprint [arXiv:2107.09645](https://arxiv.org/abs/2107.09645) (2021)
28. Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space 26. *Contemp. Math.* **26**, 28 (1984)
29. Blundell, C., et al.: Model-free episodic control. arXiv preprint [arXiv:1606.04460](https://arxiv.org/abs/1606.04460) (2016)
30. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
31. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. arXiv preprint [arXiv:2104.08821](https://arxiv.org/abs/2104.08821) (2021)
32. He, K., et al.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
33. Bachman, P., Devon Hjelm, R., Buchwalter, W.: Learning representations by maximizing mutual information across views. arXiv preprint [arXiv:1906.00910](https://arxiv.org/abs/1906.00910) (2019)
34. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: a new estimation principle for unnormalized statistical models. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. *JMLR Workshop and Conference Proceedings* (2010)
35. Wang, F., Liu, H.: Understanding the behaviour of contrastive loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
36. Wu, C.-Y., et al.: Sampling matters in deep embedding learning. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
37. Yan, W., et al.: Learning predictive representations for deformable objects using contrastive estimation. arXiv preprint [arXiv:2003.05436](https://arxiv.org/abs/2003.05436) (2020)