http://www.lamda.nju.edu.cn

ICLR 2022

Evolutionary Diversity Optimization with Clustering-based Selection for Reinforcement Learning

Yutong Wang, Ke Xue, Chao Qian

ICLR'22

LAMDA 11

Pre by Ke Xue

22/06/24

Introduction

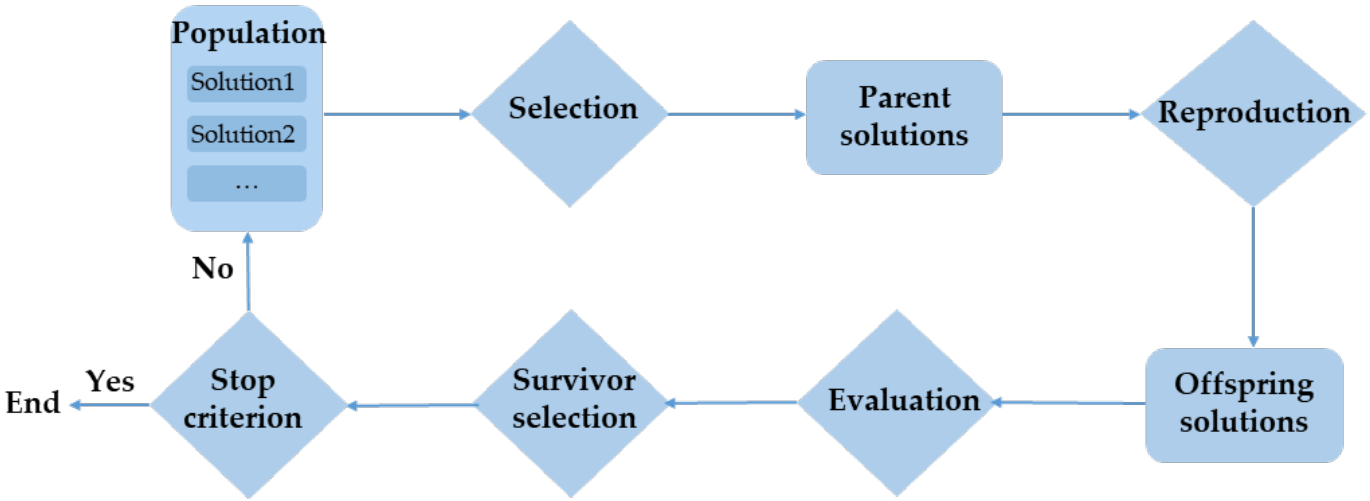
Reinforcement Learning (RL)

- Most RL methods aim to obtain a single optimal policy
- Some complex scenarios need **a set of diverse policies**
  - Exploration
  - Policy ensemble
  - Model/environment generation
  - Few-shot adaption

How to efficiently obtain a set of high-quality policies with diverse behaviors is a challenging problem in RL

Background

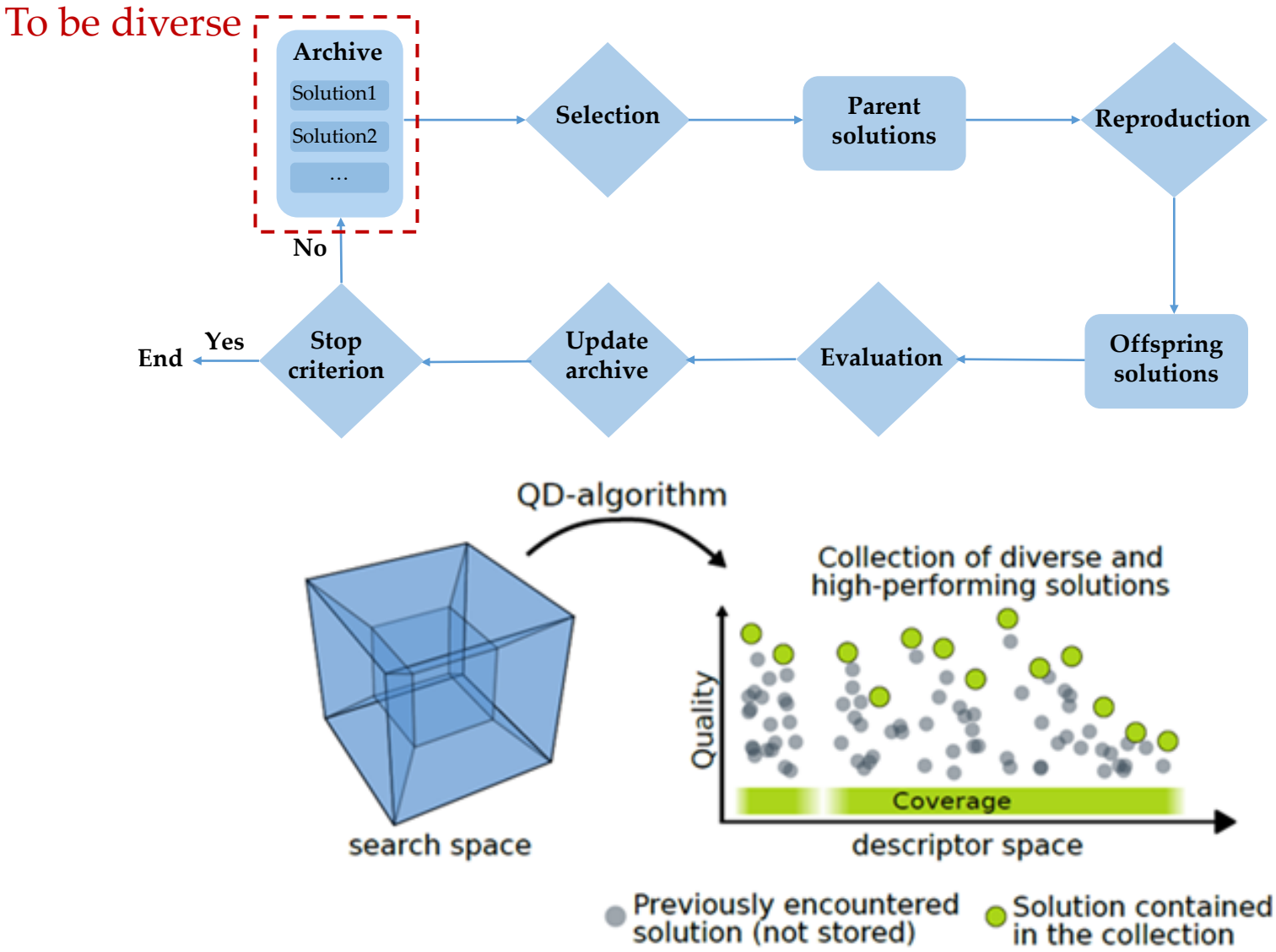
- Evolutionary Algorithm



- Key properties
  - Population-based search
  - Only evaluation is required
- Application: complex optimization problems
  - Black-box optimization problems
  - Multi-objective problems

Background

- Quality-diversity



Background - QD

Algorithm 2 QD-Optimization algorithm (I iterations)

$\mathcal{A} \leftarrow \emptyset$ ▷ *Creation of an empty container.*
for iter = 1 $\rightarrow$ I **do** ▷ *The main loop repeats during I iterations.*
  if iter == 1 **then** ▷ *Initialization.*
    $\mathcal{P}_{\text{parents}} \leftarrow$ random() ▷ *The first 2 batches of individuals are generated randomly.*
    $\mathcal{P}_{\text{offspring}} \leftarrow$ random()
  else ▷ *The next controllers are generated using the container and/or the previous batch.*
    $\mathcal{P}_{\text{parents}} \leftarrow \text{selection}(\mathcal{A}, \mathcal{P}_{\text{offspring}})$ Selection ▷ *Selection of a batch of individuals from the container and/or the previous batch.*
    $\mathcal{P}_{\text{offspring}} \leftarrow \text{variation}(\mathcal{P}_{\text{parents}})$ Reproduction ▷ *Creation of a randomly modified copy of $\mathcal{P}_{parents}$ (mutation and/or crossover).*
  for each $\theta \in \mathcal{P}_{\text{offspring}}$ **do**
    $\{f_\theta, \boldsymbol{b}_\theta\} \leftarrow f(\theta)$ ▷ *Evaluation of the individual and recording of its descriptor and performance.*
    if ADD_TO_CONTAINER(θ, $\mathcal{A}$) **then** ▷ *"ADD_TO_CONTAINER" returns true if the individual has been added to the container.*
      UPDATE_SCORES(parent(θ), Reward, $\mathcal{A}$) ▷ *The parent might get a reward.*
    else Update archive
      UPDATE_SCORES(parent(θ), -Penalty, $\mathcal{A}$) ▷ *Otherwise, it might get a penalty.*
  UPDATE_CONTAINER($\mathcal{A}$) ▷ *Update of the attributes of all the individuals in the container (e.g. novelty score).*
return $\mathcal{A}$

Background - QD

Algorithm 2 QD-Optimization algorithm (I iterations)

$\mathcal{A} \leftarrow \emptyset$ ▷ *Creation of an empty container.*
for iter = 1 $\rightarrow$ I **do** ▷ *The main loop repeats during I iterations.*
  if iter == 1 **then** ▷ *Initialization.*
    $\mathcal{P}_{\text{parents}} \leftarrow$ random() ▷ *The first 2 batches of individuals are generated randomly.*
    $\mathcal{P}_{\text{offspring}} \leftarrow$ random()
  else ▷ *The next controllers are generated using the container and/or the previous batch.*
    $\mathcal{P}_{\text{parents}} \leftarrow$ selection($\mathcal{A}$, $\mathcal{P}_{\text{offspring}}$) ▷ *Selection of a batch of individuals from the container and/or the previous batch.*
    $\mathcal{P}_{\text{offspring}} \leftarrow$ variation($\mathcal{P}_{\text{parents}}$) Reproduction ▷ *Creation of a randomly modified copy of $\mathcal{P}_{parents}$ (mutation and/or crossover).*
  for each $\theta \in \mathcal{P}_{\text{offspring}}$ **do**
    $\{f_\theta, \boldsymbol{b}_\theta\} \leftarrow f(\theta)$ ▷ *Evaluation of the individual and recording of its descriptor and performance.*
    if ADD_TO_CONTAINER(θ, $\mathcal{A}$) **then** ▷ *"ADD_TO_CONTAINER" returns true if the individual has been added to the container.*
      UPDATE_SCORES(parent(θ), Reward, $\mathcal{A}$) ▷ *The parent might get a reward.*
    else
      UPDATE_SCORES(parent(θ), -Penalty, $\mathcal{A}$) ▷ *Otherwise, it might get a penalty.*
  UPDATE_CONTAINER($\mathcal{A}$) ▷ *Update of the attributes of all the individuals in the container (e.g. novelty score).*
return $\mathcal{A}$

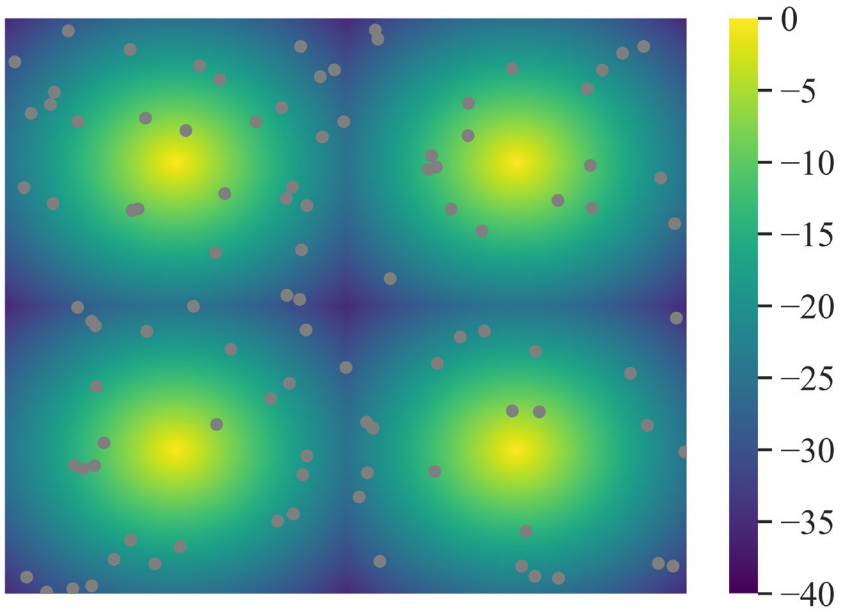
- Policy gradient assisted MAP-Elites (PGA-ME) [GECCO'21 Best paper]
  - Using PG to update the solutions
- Differentiable Quality-diversity (DQD) [NeurIPS'21 Oral, < 2.4%]
  - Using differentiable *behavior descriptor b*

Method - motivation

The **inefficient** selection

- Uniform selection
- Biased selection
- Pareto-based selection
- …

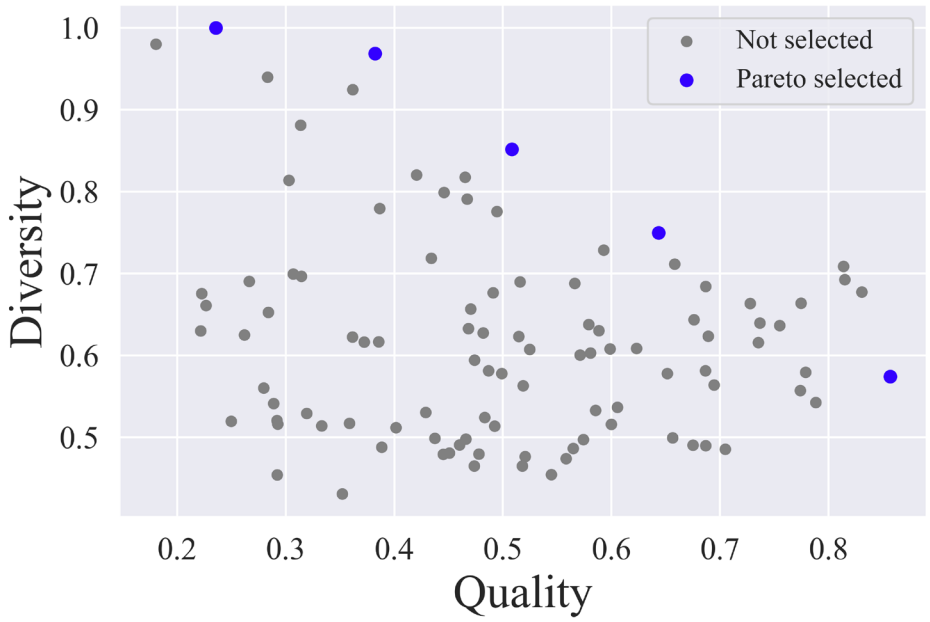
Method	Selection
Vanilla ES	The only parent solution
NSR-ES	Probabilistic selection
CVT-ES	Uniform selection
ME-ES	Biased selection
DvD-ES	All parent solutions
QD-RL	Pareto-based selection

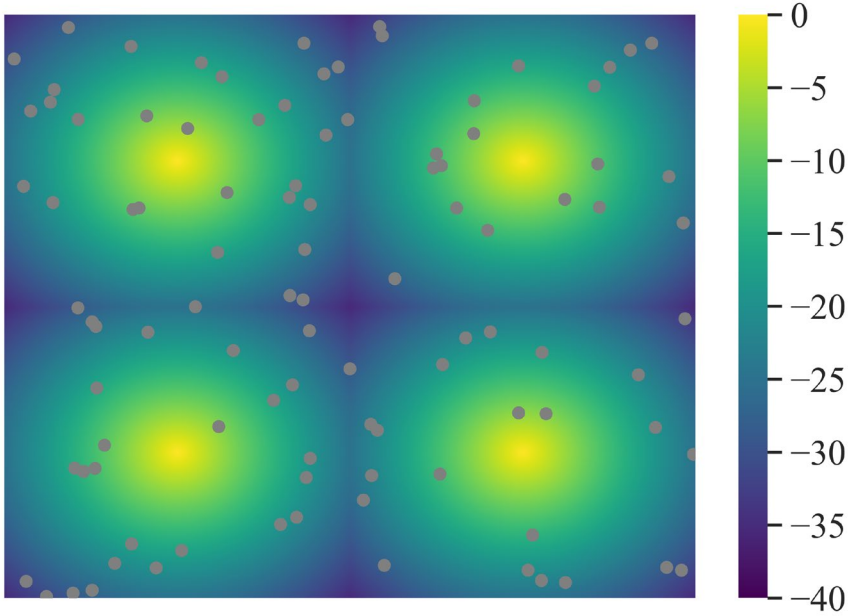


Method - motivation

The **inefficient** selection

- Uniform selection
- Biased selection
- Pareto-based selection

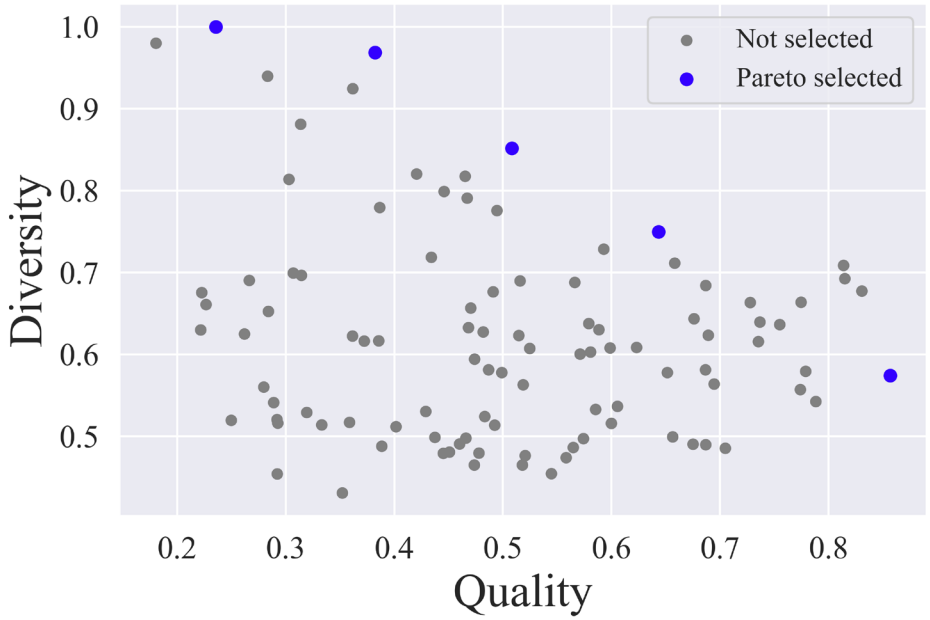


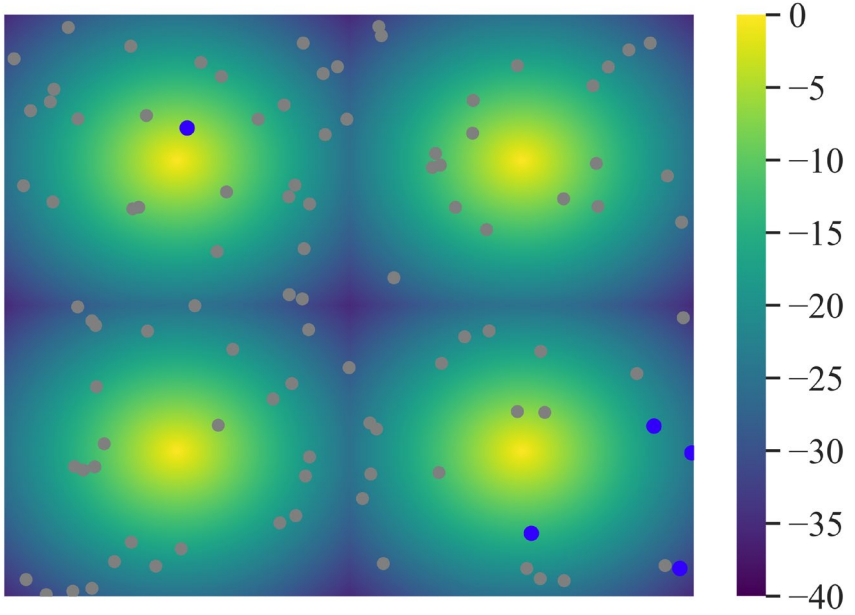


Method - motivation

The **inefficient** selection

- Uniform selection
- Biased selection
- Pareto-based selection





😄 😟

How to achieve efficient selection?

Method

Algorithm 1: EDO-CS

Input: number K of selected policies, number T of total iterations, number T' of updating iterations, behavior characterization $b(\pi_{\boldsymbol{\theta}})$, archive size l, archive A, bandit B

Output: archive A

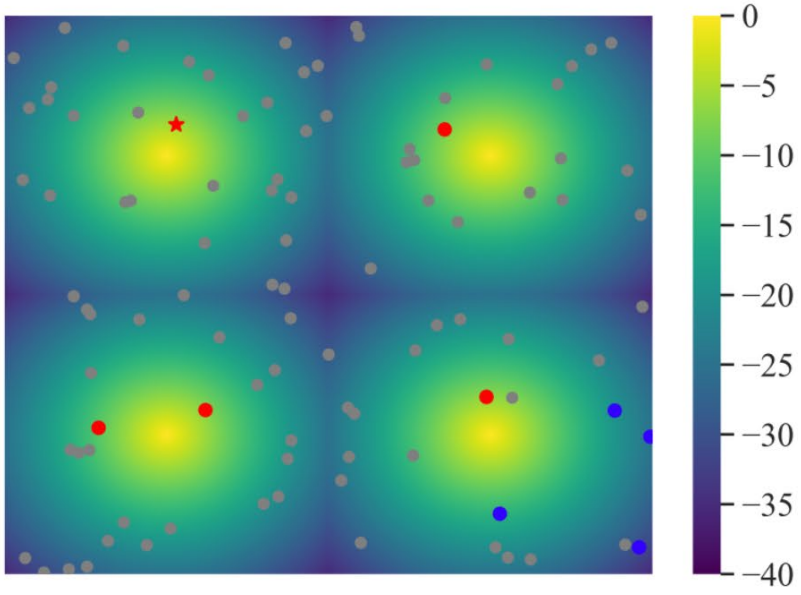
// Warm up

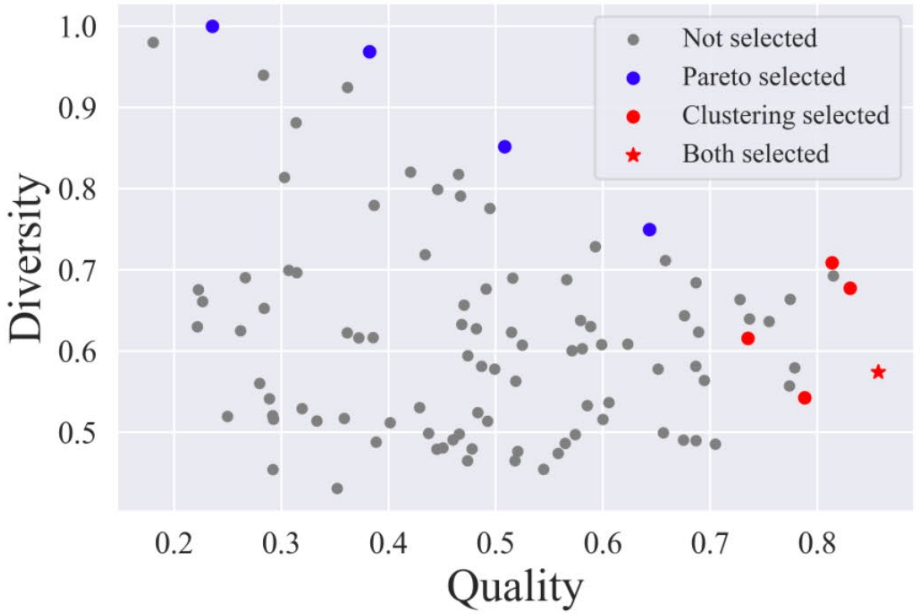
for $j = 1 : l$ **do**
  Randomly generate policy $\pi_{\boldsymbol{\theta}_j}$;
  Get cumulative rewards R and behavior $b(\pi_{\boldsymbol{\theta}_j})$ by evaluating the policy $\pi_{\boldsymbol{\theta}_j}$;
  Add $(\pi_{\boldsymbol{\theta}_j}, R, b(\pi_{\boldsymbol{\theta}_j}))$ into archive A
end
$t = 0$;
while $t < T$ **do**
  // Selection
  Use K-means to divide the policies in archive A into K clusters $\{C_k\}_{k=1}^{K}$;
  Select K policies $\{\pi_{\boldsymbol{\theta}_k}\}_{k=1}^{K}$, each one from a cluster;
  // Reproduction
  for $k = 1 : K$ **do**
    // Update in parallel
    Sample λ_k from the bandit B;
    for $i = 1 : T'$ **do**
      Set the objective function $J(\boldsymbol{\theta}_k) = (1 - \lambda_k)\mathbb{E}_{\tau \sim \pi_{\boldsymbol{\theta}_k}}[R(\tau)] + \lambda_k Div(\boldsymbol{\theta}_k)$;
      Use ES to update $\boldsymbol{\theta}_k$ as Eq. (6)
    end
    Get cumulative rewards R and behavior $b(\pi_{\boldsymbol{\theta}'_k})$ by evaluating the updated policy $\pi_{\boldsymbol{\theta}'_k}$
  end
  Update archive A and bandit B;
  $t = t + T'$
end

Clustering-based selection

$$Div(\boldsymbol{\theta}) = \frac{1}{k} \sum_{\pi'_{\boldsymbol{\theta}} \in A'_{\pi_{\boldsymbol{\theta}},k}} \|b(\pi_{\boldsymbol{\theta}}) - b(\pi'_{\boldsymbol{\theta}})\|_2.$$

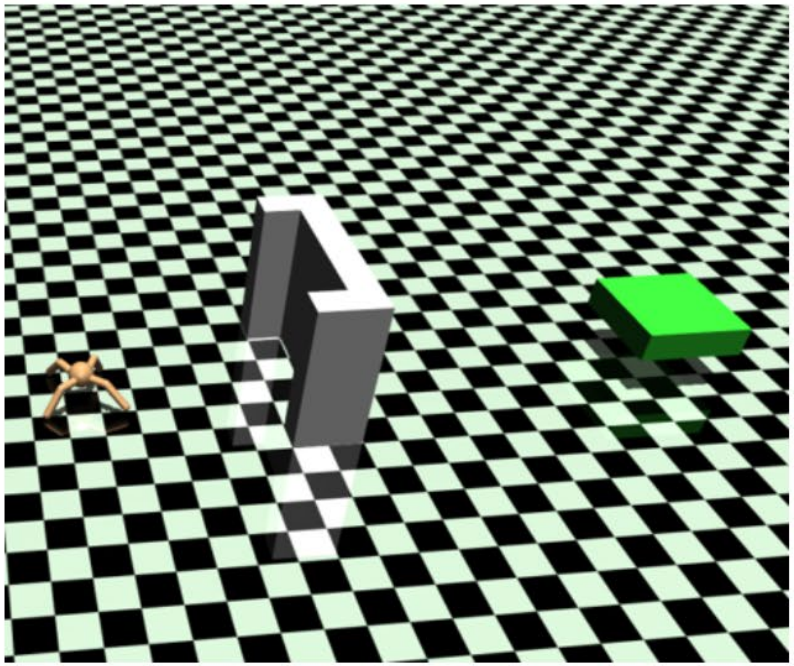
Method

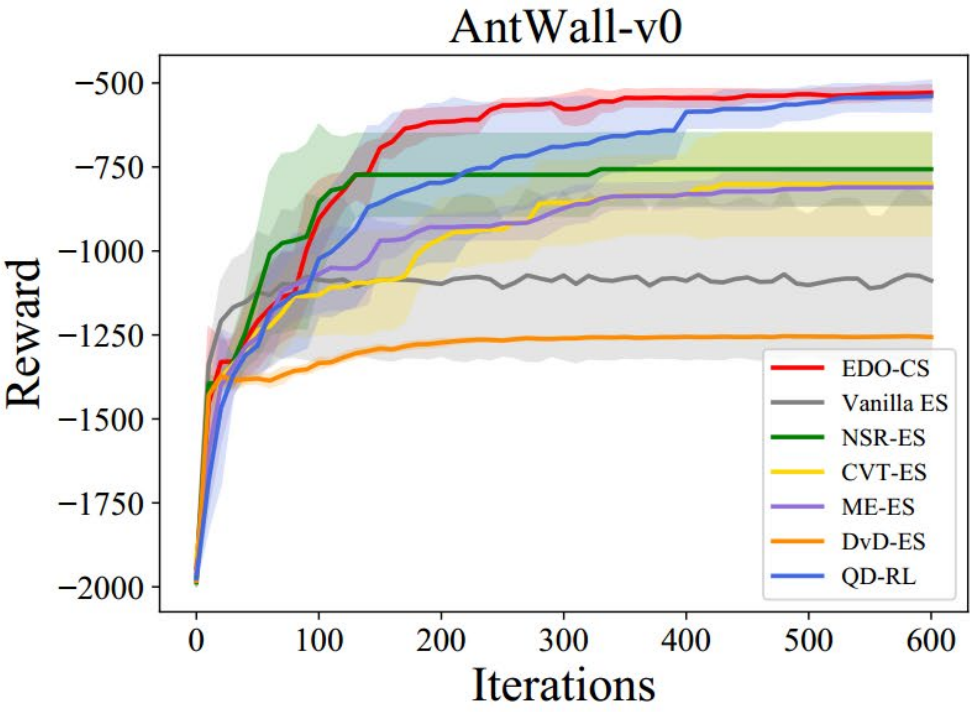


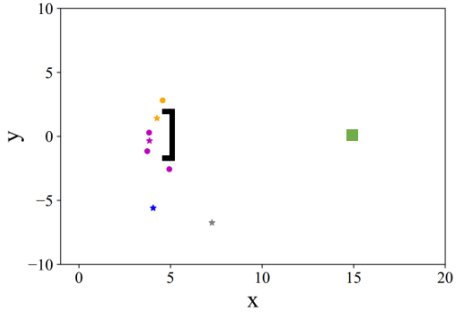


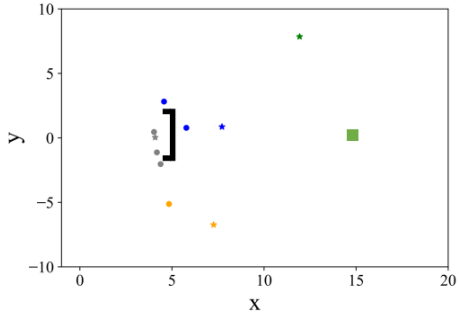
Method	Selection	Reproduction	EAs type	From archive
Vanilla ES	The only parent solution	Quality	$(1, 1)$	×
NSR-ES	Probabilistic selection	Quality and diversity	$(K, 1)$	×
CVT-ES	Uniform selection	Quality and diversity	$(K + K)$	✓
ME-ES	Biased selection	Quality or diversity	$(K + K)$	✓
DvD-ES	All parent solutions	Quality and diversity	(K, K)	×
QD-RL	Pareto-based selection	Quality or diversity	$(K + K)$	✓
EDO-CS	Clustering-based selection	Quality and diversity	$(K + K)$	✓

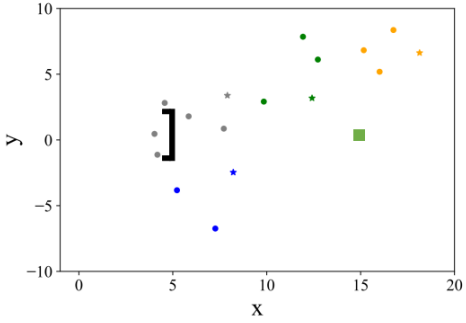
Experiment

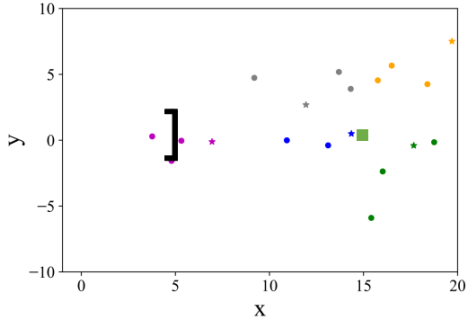



AntWall-v0
Reward
−500
−750
−1000
−1250
−1500
−1750
−2000
0
100
200
300
400
500
600
Iterations
EDO-CS
Vanilla ES
NSR-ES
CVT-ES
ME-ES
DvD-ES
QD-RL


y
x


y
x


y
x

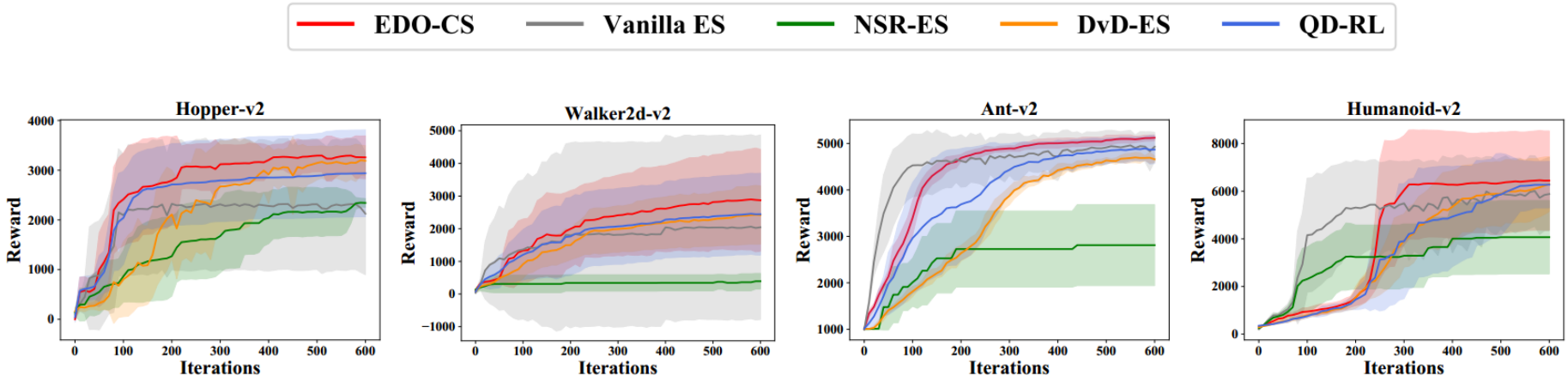

y
x

Experiment

- Multi-modal

Environment	EDO-CS	QD-RL	ME-ES	DvD-ES	CVT-ES	NSR-ES	Vanilla ES
HalfCheetahFwd	**4284**	2930	2700	-3419	3219	1346	-5543
HalfCheetahBwd	**6548**	6013	5953	6353	4672	5366	3911
AntFwd	**4617**	4291	4316	4507	3856	1737	1911
AntBwd	**4697**	4164	4123	3498	2958	3961	-851
Performance Ranking	**1**	3	3.5	3.75	4.75	5.25	6.75

- Single-modal



Experiment – ablation studies

- Adaptive λ

Setting of λ	*AntWall-v0*	*HalfCheetahFwd*	*HalfCheetahBwd*	*AntFwd*	*AntBwd*
Adaptive λ	**-529**	**4284**	**6548**	**4617**	**4697**
$\lambda = 0$	-650	3856	5931	4340	4426
$\lambda = 0.5$	-850	3877	6077	2800	3093

Experiment – ablation studies

- Adaptive λ
- Compared with direct optimization

$$f_{pair} = (1-\omega)\sum_{i=1}^{K}\mathbb{E}_{\tau\sim\pi_{\boldsymbol{\theta}_i}}\left[R(\tau)\right] + \omega\sum_{i=1}^{K}\sum_{j\neq i}\|b(\pi_{\boldsymbol{\theta}_i}) - b(\pi_{\boldsymbol{\theta}_j})\|_2$$

$$f_{det} = (1-\omega)\sum_{i=1}^{K}\mathbb{E}_{\tau\sim\pi_{\boldsymbol{\theta}_i}}\left[R(\tau)\right] + \omega\cdot det(\Pi)$$

Method	*AntWall-v0*	*HalfCheetahFwd*	*HalfCheetahBwd*	*AntFwd*	*AntBwd*
EDO-CS	**-529**	**4284**	**6548**	4617	**4697**
EDO-DOS$_{pair}$	-706	4188	5847	**5013**	4392
EDO-DOS$_{det}$	-536	-5591	6529	-530	2417

Method	*AntWall-v0*	*HalfCheetahFwdBwd*	*AntFwdBwd*
EDO-CS	**0.015**	**0.021**	**0.023**
EDO-DOS$_{pair}$	9.570	9.802	9.807
EDO-DOS$_{det}$	8.900	9.266	9.567

Experiment – ablation studies

- Adaptive λ
- Compared with direct optimization
- Clustering algorithm
- Number T' of updating iterations
- Population size K
- Archive size l

More results are shown in the Appendix

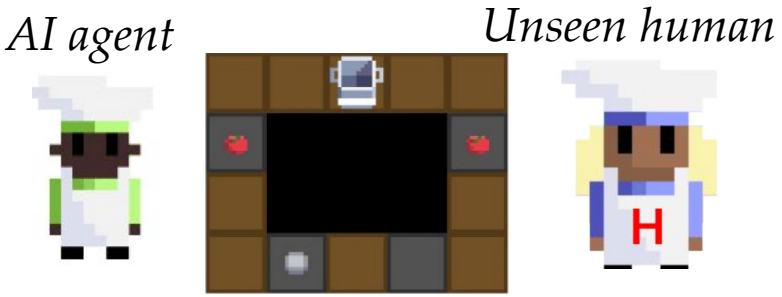
Discussion

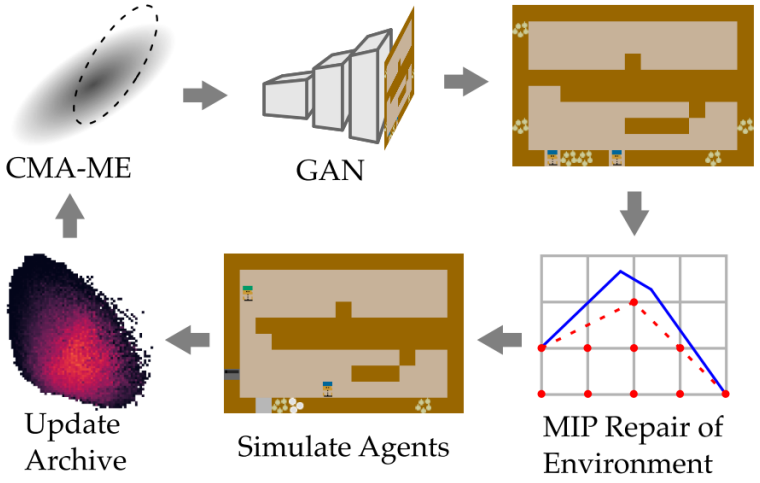
Conclusion

- QD is an interesting and attractive research area
- We proposed a *"Simple but efficient"* selection method for QD

Future work

- Application of QD
  - Human-AI coordination
  - Environment generation





Thanks you!

Our code can be found at:
www.lamda.nju.edu.cn/qianc/code-EDOCS.html

✉ xuek@lamda.nju.edu.cn