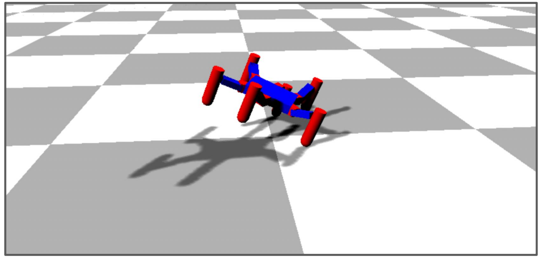
Individual Project Presentation

Tilman Hisarli

Imperial College London

Lifelong reinforcement learning in robotics for non-stationary environments

Lifelong Quality Diversity



Imperial College London

Problem formulation

- True robot autonomy requires ability to continuously adapt to changes in environment, but...

- ... such adaptation is difficult, as it should be:

⇨ **fast**

⇨ **possible in any environment**

  - No restrictive assumptions about future environments
  - May require continual learning of new skills

Objective

Build a **continuously** learning RL algorithm that can adapt to **any reasonable changes** in its environment **online**

⇒ **focus on Quality Diversity**

- Non convexity in robotics
- Multiple solutions useful

Imperial College London

Overview

1. Background literature
2. LLQD Method
3. Experiments & evaluation
4. Ethical considerations
5. Conclusion & future work

Imperial College London

Background I: Quality Diversity
Definition

Evolutionary algorithm that generates an archive of **diverse**, **high quality** solutions

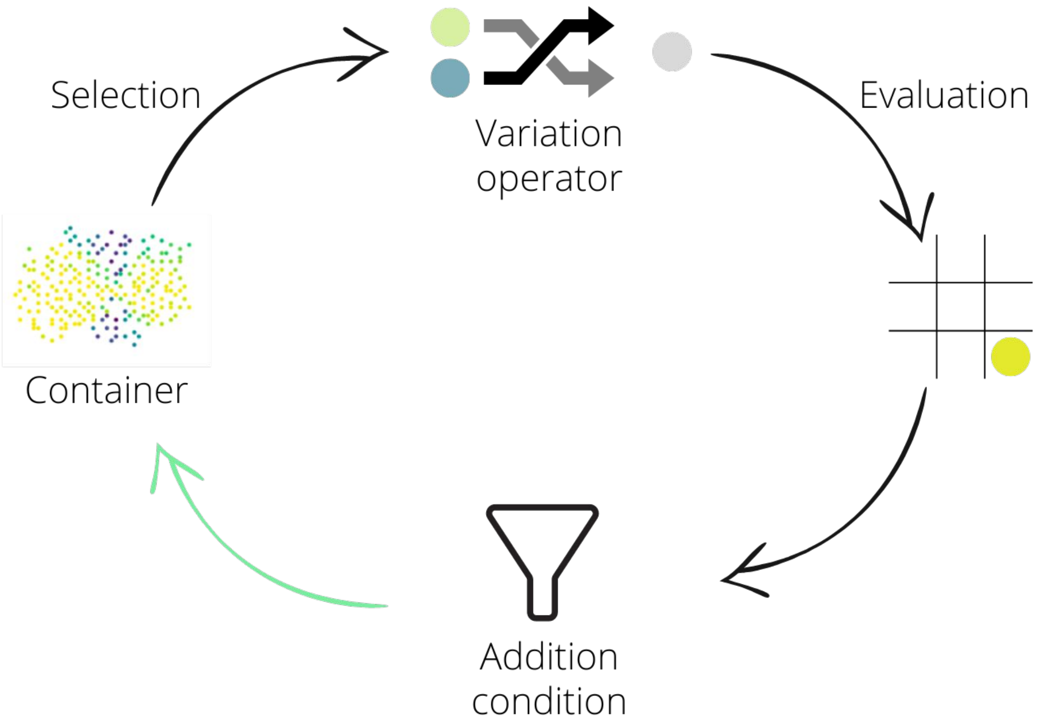
---

Quality – measured by some (task-specific) fitness function

Diversity – measured by low dimensional representation of solution's key features (Behavioural Descriptor)

J. K. Pugh, L. B. Soros, and K. O. Stanley, "Quality diversity: A new frontier for evolutionary computation," *Frontiers in Robotics and AI*, vol. 3, p. 40, 2016.

Background I: Quality Diversity
Generalized framework



Imperial College London

A. Cully and Y. Demiris, "Quality and diversity optimization: A unifying modular framework," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 245–259, 2018.

Background I: Quality Diversity
Generalized framework

Problem

- Low sample efficiency ⇒ requires tens of millions of evals
- Real evals are expensive / dangerous

Not suited for online learning

A. Cully and Y. Demiris, “Quality and diversity optimization: A unifying modular framework,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 245–259, 2018.

Background II: Archives as repertoires

1 Pre-compute archive in simulation

- No time constraints, parallelizable
- Non-dangerous

2 Adapt by intelligently choosing from archive at deployment

- Model difference btw sim and real performance

✓ **Adapt in minutes**

✗ **Effective adaptation not guaranteed**

Imperial College London

A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, pp. 503–507, 2015.
K. Chatzilygeroudis, V. Vassiliades, and J.-B. Mouret, "Reset-free trial-and- error learning for robot damage recovery," *Robotics and Autonomous Systems*, vol. 100, pp. 236–250, 2018.

Background II: Archives as repertoires

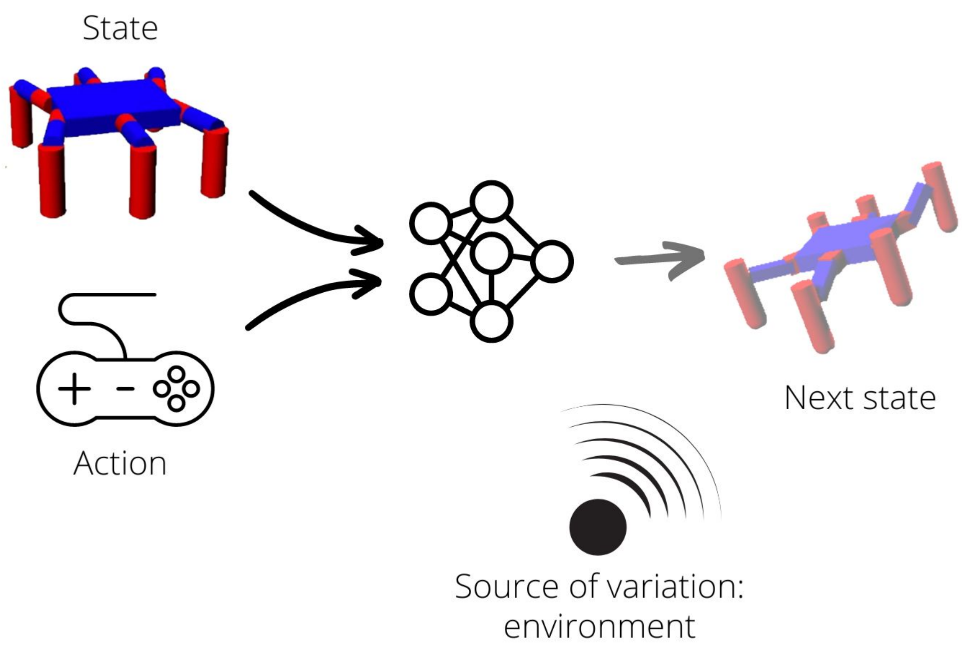
To guarantee effective adaptation, **continuously learn new skills**, rather than rely on old ones...

... but to ensure online viability, need to **solve issue of low sample efficiency** differently

⇨ Models

Imperial College London

Background III: Model-based QD
Learning dynamics



- Maps current state & action to next state
- Learns dynamics of interactions btw robot and environment
- Dynamics depend on environment

Imperial College London

Imperial College London

Background III: Model-based QD
DAQD

Dynamics-Aware Quality Diversity introduces key innovations ...

- Probabilistic dynamics model that **predicts the distribution of each dimension of the next state**
- Called iteratively to **evaluate in imagination** the robot trajectory ⇒ predicted BD and fitness of candidate solution
- Only candidate solutions added to **imaginary archive** (i.e. good model predicted performance) will be evaluated in real environment

⇒ 20x fewer real evals

✓ Online adaptation

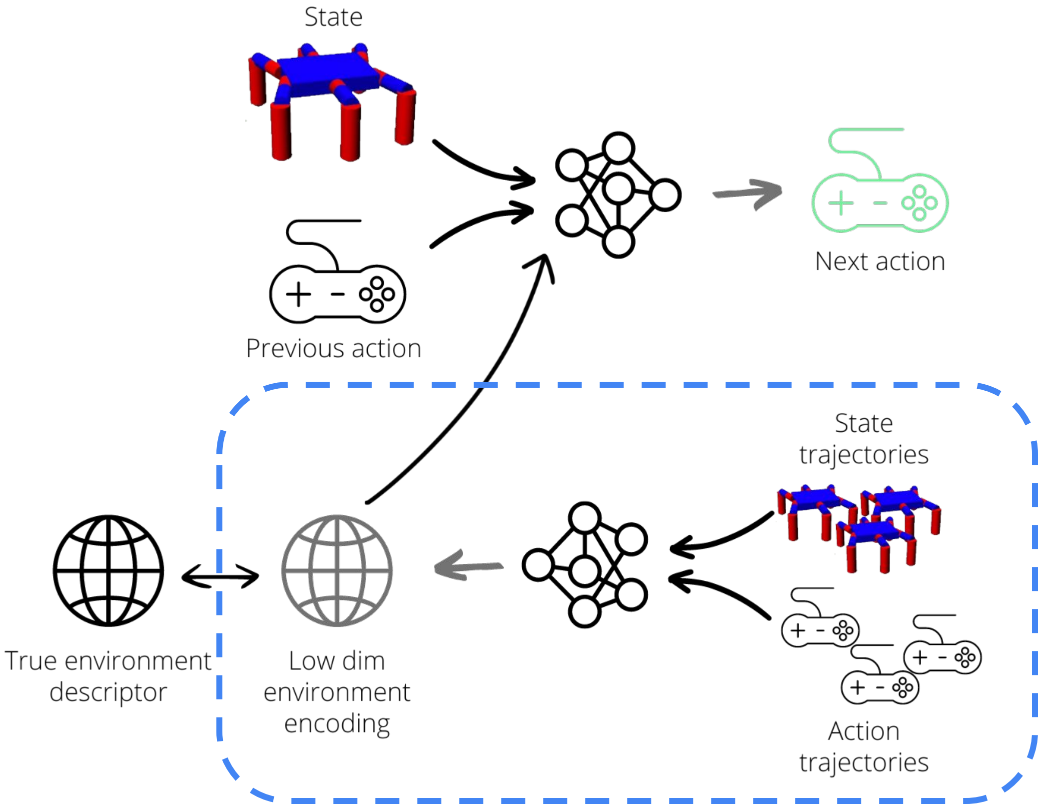
✓ Continual learning

✗ Mixes transitions

B. Lim, L. Grillotti, L. Bernasconi, and A. Cully, *Dynamics-aware quality-diversity for efficient learning of skill repertoires*, 2021. DOI: 10.48550/ARXIV.2109. 08522. [Online]. Available: https://arxiv.org/abs/2109.08522.

Imperial College London

Background IV: Adaptation
RMA



Adapts in seconds, but due to sim pre-training

Effectively uses state-action trajectories to infer environment

A. Kumar, Z. Fu, D. Pathak, and J. Malik, *Rma: Rapid motor adaptation for legged robots*, 2021. DOI: 10.48550/ARXIV.2107.04034. [Online]. Available: https://arxiv.org/abs/2107.04034.

LLQD Method
Inspiration

DAQD inspired

Sample efficient online skill learning

+

RMA inspired

Transition based environment information

- ✓ Learn continuously ...
- ✓ Online ...
- ✓ Specialised skills ...
- ✓ For any reasonable environment

Imperial College London

Imperial College London

LLQD Method
Key idea

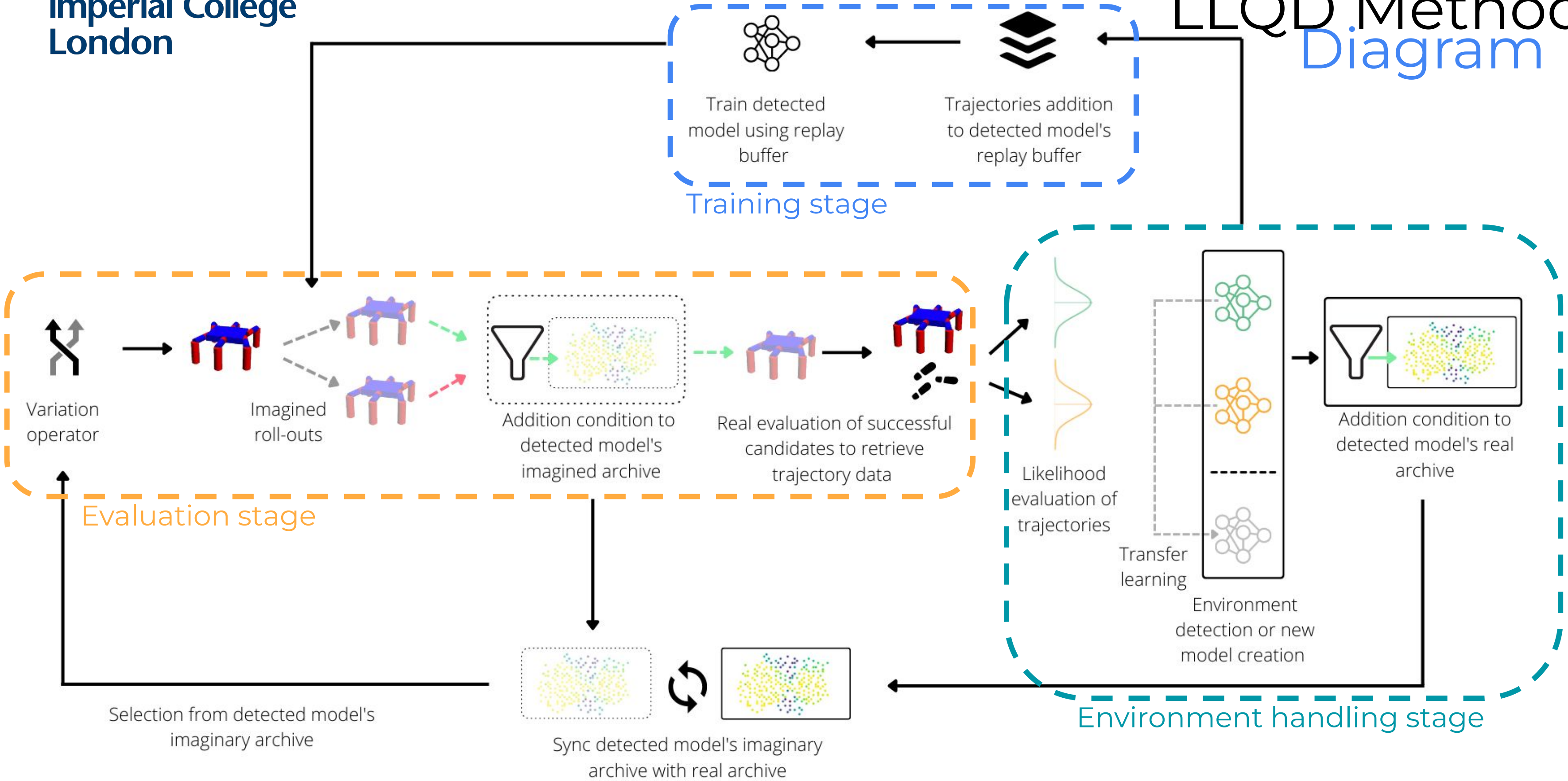
Generate **specialised archives** of behaviours online for each **distinct environment** the robot encounters, via …

1. **Detection** of previously encountered or new environments based on the robot’s recent **state action trajectories**

2. Sequential generation of **separate archives, replay buffers, and dynamics models** for each distinct environment as they are encountered (ideally one-to-one mapping of environment and model)

Imperial College London

LLQD Method Diagram



Some elements from B. Lim, L. Grillotti, L. Bernasconi, and A. Cully, *Dynamics-aware quality-diversity for efficient learning of skill repertoires*, 2021. DOI: 10.48550/ARXIV.2109. 08522. [Online]. Available: https://arxiv.org/abs/2109.08522.

Imperial College London

LLQD Method
Transition likelihoods

- Obtain state action trajectories from real evaluations ⇒ per eval, 299 transitions (state, action, next state)

- Use probabilistic dynamics model* …

$$\tilde{p}_\theta(s_{t+1} - s_t | s_t, a_t) = \mathcal{N}(\mu_\theta(s_t, a_t), \Sigma_\theta(s_t, a_t))$$

- … to obtain log likelihoods per transition (summed across state dims)

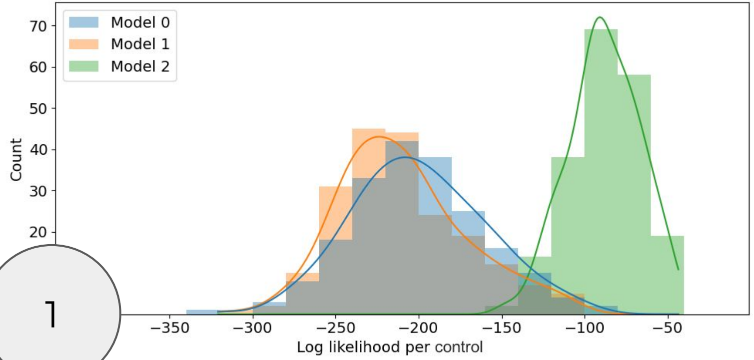
$$log\_likelihood = \sum_{i=1}^{48} \ln\left(\frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\left(-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right)}\right)$$

** Assuming independence across state dimensions*

Imperial College London

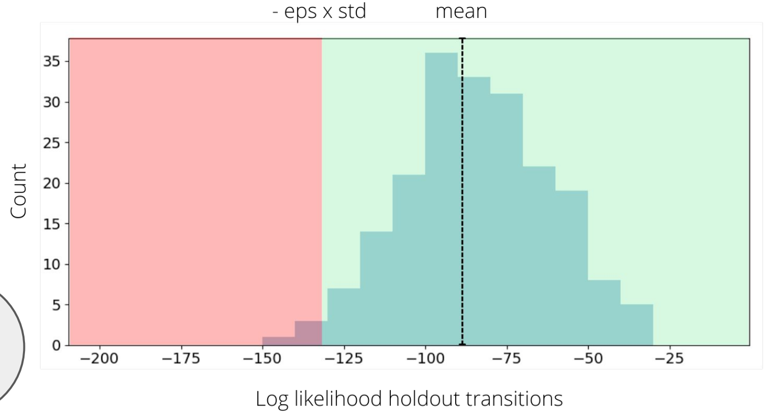
LLQD Method

Environment* detection



1

Obtain mean log likelihood of transitions for all existing models



2

In order of likeliest model m, assign if

$$\texttt{log\_likelihood\_mean}_m(\texttt{transitions}) > \texttt{log\_likelihood\_mean}_m(\texttt{holdout\_set}) - \texttt{epsilon} * \texttt{log\_likelihood\_std}_m(\texttt{holdout\_set})$$

3

If no model satisfies detection condition, sig. new env detected

Imperial College London

LLQD Method
Transfer learning

When a new environment is detected, instantiate new model with **weights and biases of next most likely model**

Why? Some **dynamics likely shared across environments** (e.g. effect of gravity on robot) especially when the environments are relatively similar

→ **Should improve sample efficiency**

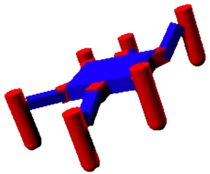
- Utilizes dynamics related information from previous samples
- Faster archive generation in terms of number of real evaluations

Imperial College London

Experiments & Evaluation I: Setup
Robot, environments & task

Robot

- 18 DoF hexapod
- 36 params in controller

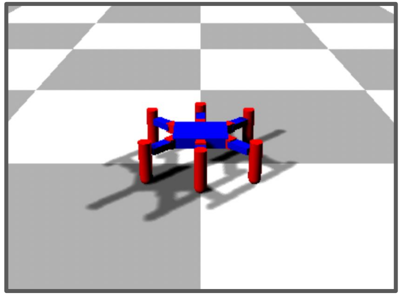


Task

- Learn omni directional repertoire
- Fitness: alignment of robot orientation with circular trajectory
- BD: pos in *xy*-space

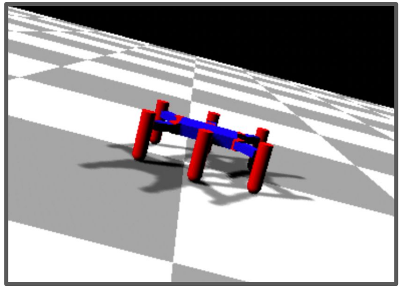
Environments

Environment 0



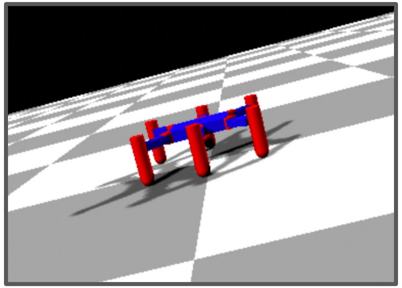
- Evenly balanced robot
- No payload
- Even floor

Environment 1



- Off-balance robot
- Heavy payload
- Slanted floor

Environment 2



- Evenly balanced robot
- No payload
- Slanted floor

Imperial College London

Experiments & Evaluation II: Main experiment

Benchmark & metrics

Benchmark

DAQD strong benchmark as it

- provides strong sample efficiency needed for **online** learning
- can acquire **new skills** continuously
- requires **no prior knowledge** of future environments

Goal

Show LLQD handles changing environments more effectively by assessing the final archives for a given number of real evaluations

Metrics

Coverage

- # solutions in archive
- indicates diversity

QD score

- sum of norm. fitness
- indicates quality

Reliability score

- % of reliable solutions wrt BD & fitness

Imperial College London

Experiments & Evaluation II: Main experiment

LLQD vs DAQD

Comparator baseline:

Continuously train DAQD on all environments encountered

	Outperformance LLQD vs DAQD		
	Coverage	**QD score**	**Reliability score**
Mean	13.8%	16.2%	38.4pp
P-value* **	0.002**	0.001**	0.000**
Median	7.2%	12.4%	47.4pp
25th percentile	1.1%	1.4%	20.0pp
75th percentile	22.4%	27.1%	64.5pp

Table 5.2: Outperformance of LLQD algorithm over DAQD on randomly generated sequence of environments over 20 runs. Minimum 200 simulations per environment, epsilon 0.5. ** indicates statistical significance at the 5% level. *** Wilcoxon signed rank test, one-sided, alternative hypothesis that distribution underlying the difference between LLQD and DAQD is "stochastically greater than a distribution symmetric about zero" [26]

	Absolute performance	
	LLQD	**DAQD**
Mean Coverage	179.7	160.8
Median Coverage	179.5	164.5
Mean QD score	110.5	96.5
Median QD score	111.0	95.5
Mean Reliability score	77.8%	39.4%
Median Reliability score	96.0%	32.8%

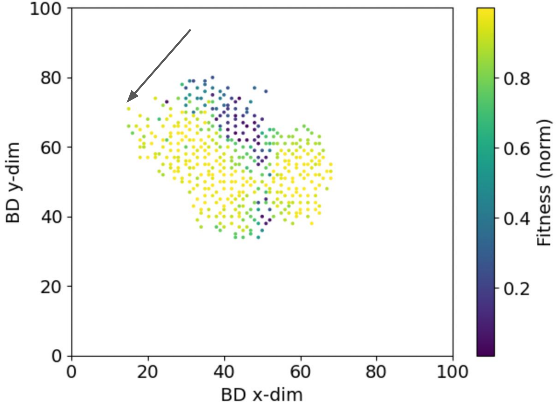
Table 5.1: LLQD and DAQD absolute total performance across environments averaged over 20 runs of randomly generated environment sequences. Per run, the same randomly generated sequence is used for both algorithms.

- Statistically significant outperformance of LLQD over DAQD **across all metrics**
- **Median outperformance** (@1.2k evals): +7% Cov., +12% QD score, +47pp Reliability

Why? LLQD's more accurate model predictions, DAQD's archive overwriting
⇒ outperformance increases at higher # real evals (+34% Cov., +33% QD @10k evals)

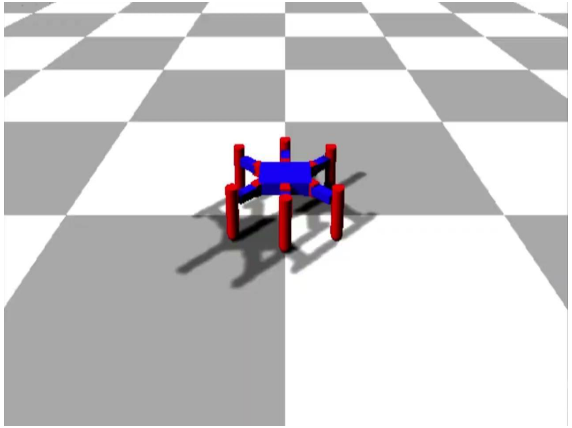
Imperial College London

Experiments & Evaluation II: Main experiment

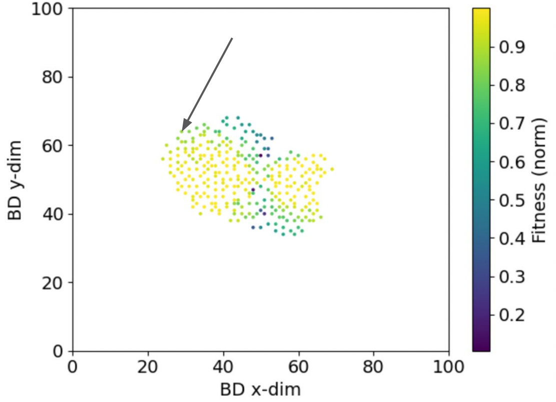


Repertoire for **DAQD** (all envs) after 10.8k real evaluations

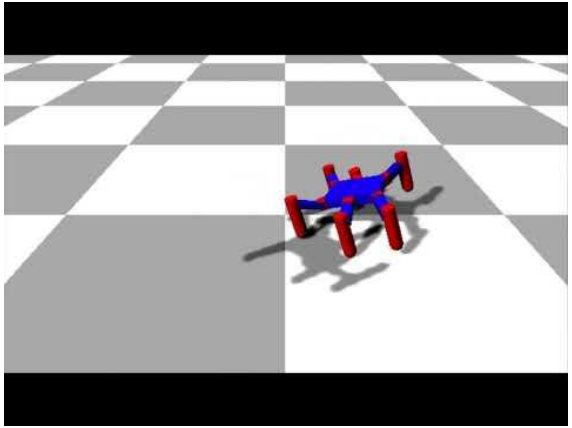
Env 0



Solution from DAQD archive not reliable in environment 0



Repertoire for **LLQD** env 0 after 5.1k real evaluations



Solution from env 0 specialised LLQD archive performs well as expected

Imperial College London

Experiments & Evaluation III: Ablation studies

Effect of epsilon

- Key parameter for **correctly detecting** environments; determines sensitivity with which existing environments are detected
- If **too high**, LLQD will map multiple models to the same env;
  if **too low**, LLQD will assign data from multiple envs to same model
- Detection results for 5 random environment switches over 20 runs:

% correct	Epsilon 0.3	Epsilon 0.5	Epsilon 0.7	Random
Mean	51.0%	77.0%	67.0%	31.4%
Median	40.0%	100.0%	80.0%	40.0%
25th percentile	35.0%	40.0%	40.0%	20.0%
75th percentile	65.0%	100.0%	100.0%	40.0%

Table 5.4: Percentage of correctly identified environments for various values of epsilon over 20 runs each, compared to a baseline of random classification. Per environment c. 100 simulations are performed, with c. 300 transitions of state, action, next state per simulation.

- epsilon of 0.5 performs best, with 77% mean and 100% median correct identification
- Strongly outperforms random baseline of 31% mean, 40% median

Imperial College London

Experiments & Evaluation III: Ablation studies

Effect of steps taken

- # steps taken in env prior to env classification affects performance wrt correctly detecting environments. This is because:
  - **Mean log likelihood of transitions becomes more robust** as number of transitions increases (some randomness in controls leads to outliers)
  - More transitions ⇒ **wider training distribution** per model, better generalization
- Detection results for 5 random environment switches over 20 runs:

% correct	Simulations per environment 50	100	200
Mean	63.0%	77.0%	81.0%
Median	60.0%	100.0%	90.0%
25th percentile	55.0%	40.0%	60.0%
75th percentile	80.0%	100.0%	100.0%

Table 5.5: Percentage of correctly identified environments for various numbers of simulations per visited environment. Per simulation c. 300 transitions of state, action, next state are recorded, used for environment detection, and ultimately added to the mapped model's replay buffer.

- Mean performance **rises to 81%**, as # real evals increases to 200, and becomes more **robust** as IQR tightens
- But note **trade-off** with online viability (!)

Experiments & Evaluation III: Ablation studies
Stretch performance

- Using correct hyperparameters, evaluate how good LLQD's environment detection capability is, in most **difficult stretch case of constantly randomly alternating environments**, as opposed to mere random environment sequencing
- Detection results for 5 random environment switches over 20 runs:

% correct	Base case	Stretch case
Mean	81.0%	77.0%
Median	90.0%	80.0%
25th percentile	60.0%	60.0%
75th percentile	100.0%	100.0%

Table 5.6: Percentage of correctly classified environments for base case of randomized environments and stretch case of constantly changing randomized environments. Performance over 20 runs. *Epsilon* set to 0.5, number of simulations per visited environment set to 200.

- **Slight drop** in mean performance by -4pp
- However, drop **not statistically significant**, with P value of 0.32 (ranksum test)

Imperial College London

Experiments & Evaluation III: Ablation studies
Transfer learning

- Examine whether LLQD's transfer learning allows it to **generate high quality archives with fewer real evaluations** than learning from scratch (as is the case under DAQD)

- Six permutations to consider
    - Init model for env 0 with params of model for env 1
    - Init model for env 0 with params of model for env 2
    - Init model for env 1 with params of model for env 0
    - Etc.

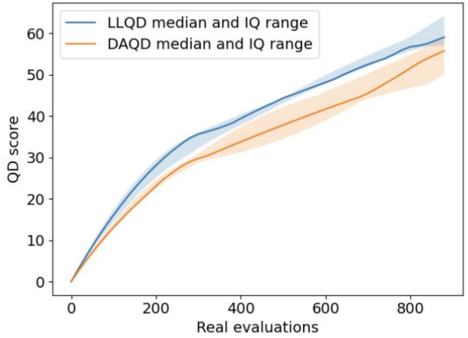


Figure 5.22: LLQD learning model for env 0 from env 2 vs DAQD baseline

- Plot median evolution of QD score and Coverage, vs DAQD baseline, over 10 runs (see Appendix for all plots)

Experiments & Evaluation III: Ablation studies

Transfer learning

P values for LLQD outperformance wrt Coverage

		P-value	
Learn env	**From env**	**@ 400 evals**	**@ 800 evals**
0	1	0.225	0.145
0	2	0.141	0.217
1	0	0.056*	0.163
1	2	0.003**	0.087*
2	0	0.008**	0.056*
2	1	0.353	0.500

Table 5.7: P-values for one-sided Wilcoxon rank-sum tests with alternative hypothesis that the distribution of archive Coverage underlying LLQD "is stochastically greater than the distribution underlying" [27] DAQD. Measured over 10 runs.

P values for LLQD outperformance wrt QD Score

		P-value	
Learn env	**From env**	**@ 400 evals**	**@ 800 evals**
0	1	0.248	0.056*
0	2	0.016**	0.010**
1	0	0.128	0.248
1	2	0.041**	0.113
2	0	0.014**	0.128
2	1	0.163	0.440

Table 5.8: P-values for one-sided Wilcoxon rank-sum tests with alternative hypothesis that the distribution of archive QD score underlying LLQD "is stochastically greater than the distribution underlying" [27] DAQD. Measured over 10 runs.

For **all except one** transfer scenario there is **at least some evidence (stat. significant at the 10% level) of outperformance** in relation to QD score, Coverage or both

Imperial College London

Ethical considerations

Dual use dilemma

- LLQD equips robot to continuously learn and adapt
- May be used to cause harm, e.g. in military use case
- Our research is fundamental, no steps taken twds military application

Environmental concerns

- Evolutionary algorithms require high number of computations
- May be energy / battery intensive
- Energy / battery may not come from renewable resources

Imperial College London

Conclusion

Original objective

"Build a **continuously** learning RL algorithm that can adapt to **any reasonable changes** in its environment **online**"

LLQD accomplishes all this by

- ✓ building specialised repertoires of skills
- ✓ online, with evidence of better sample efficiency than best practice
- ✓ for each distinct environment, through the use of an environment detection mechanism using only robot trajectory data
- ✓ that makes no restrictive assumptions about future environments

Future work

- Currently uses **resets** as environments have homogenous dynamics
  ⇒ make reset-free by merging with **RF-QD algorithm**

- Relax assumption of **sudden environment changes**

- Adjust selection rule to **prioritise less dense areas** of the BD space to generate solutions for more difficult to reach areas in the environment

- 3-dimensional, or **learnt behavioural descriptor**

Q&A

Imperial College London

Appendix: contributions to literature

Contributions to QD literature

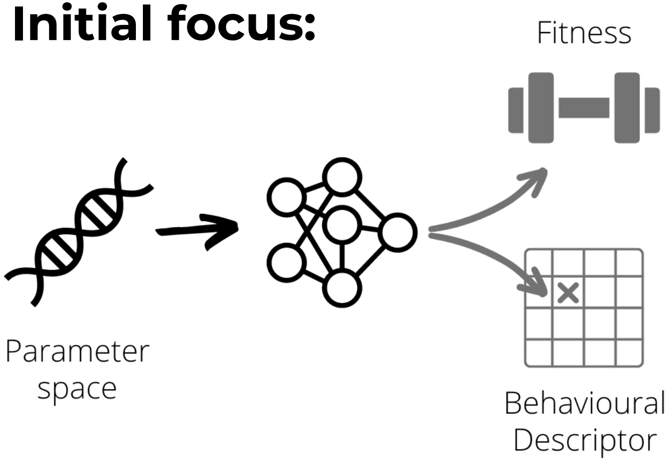
1. flexible, online continuous learning algorithm
2. transition likelihood estimation via probabilistic dynamics models
3. training-free environment detection mechanism
4. autonomous transfer learning mechanism for better sample efficiency

Appendix: early model based implementations

Learning BD or fitness

- Reduce cost of learning by **replacing expensive evaluation function** with its surrogate model
- Idea:
  - use model to **narrow down list of candidates** to those most likely to be successful
  - uses "stored" info from previous samples to **reduce number of real evaluations**

Initial focus:



✓ Fewer real evals needed

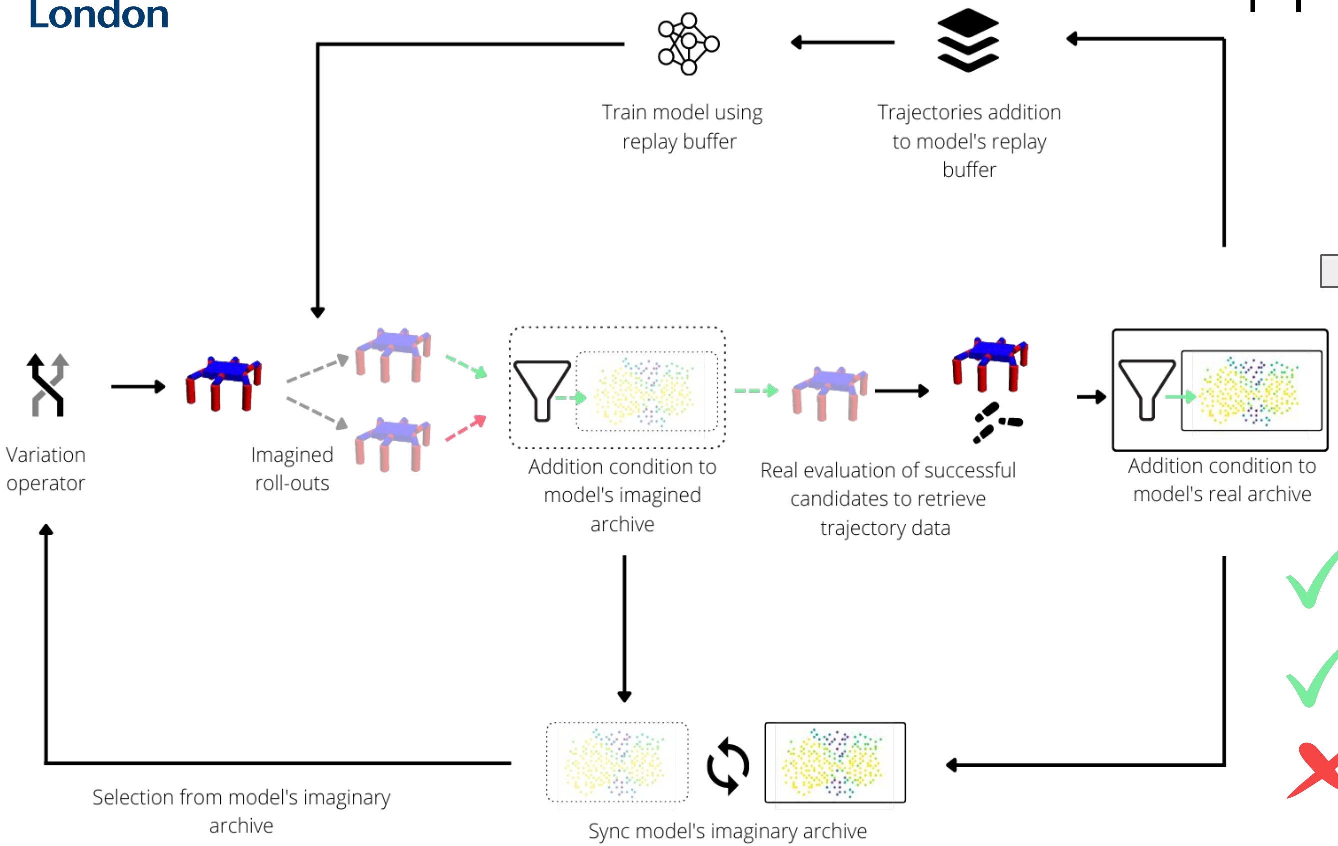
✗ Task-specific, error srcs

Imperial College London

L. Keller, D. Tanneberg, S. Stark, and J. Peters, "Model-based quality-diversity search for efficient robot learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
A. Gaier, A. Asteroth, and J.-B. Mouret, "Data-efficient design exploration through surrogate-assisted illumination," *Evolutionary computation*, pp. 1–30,

Imperial College London

Appendix: DAQD
Visual



20x fewer real evals

- ✓ Online adaptation
- ✓ Continual learning
- ✗ Mixes transitions

Some elements from B. Lim, L. Grillotti, L. Bernasconi, and A. Cully, *Dynamics-aware quality-diversity for efficient learning of skill repertoires*, 2021. DOI: 10.48550/ARXIV.2109. 08522. [Online]. Available: https://arxiv.org/abs/2109.08522.

Imperial College London

Appendix: LLQD Method
Stages

Evaluation stage

- Selection
- Mutation
- Model evaluation
- Real evaluation

Env handling stage

- Likelihood evaluation
- Detection of existing or new environment
- Model selection or new model creation

Training stage

- Add transitions to correct model buffer
- Train model
- Generate mean log likelihood and std of holdout set

Appendix: Main experiment naive benchmark I

LLQD vs DAQD

Naive comparator:

train DAQD on one environment; use resulting archive in all other envs

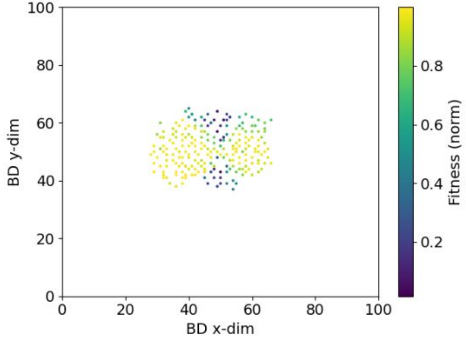
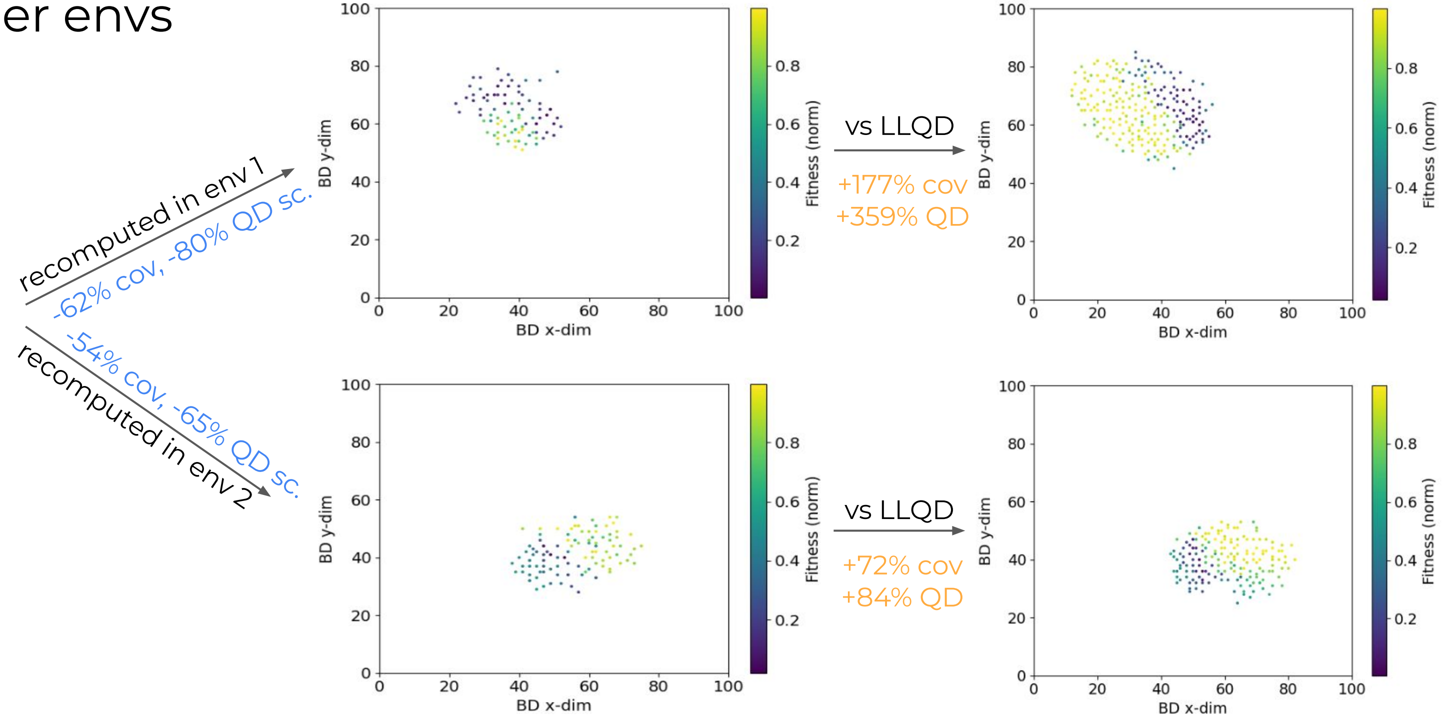


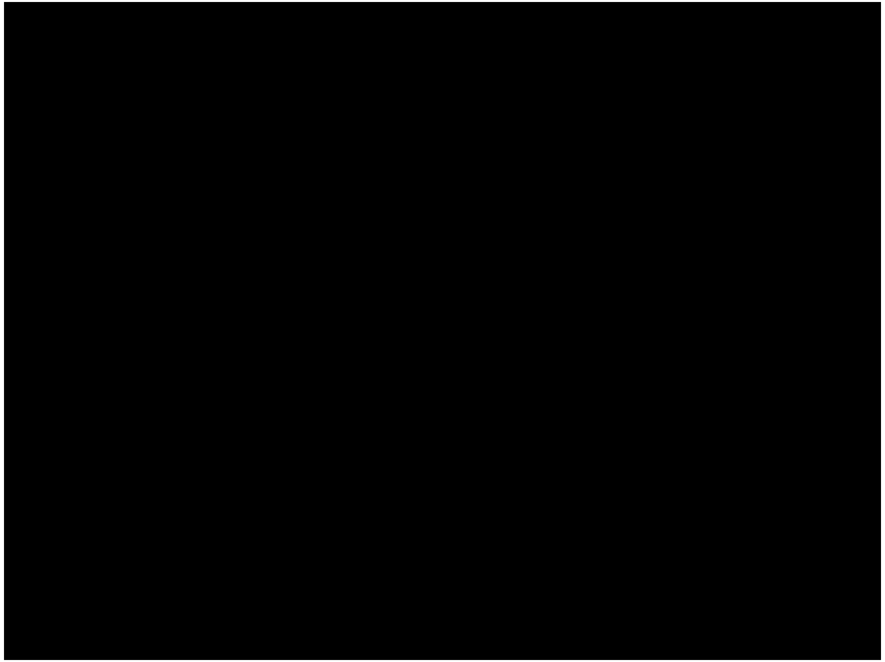
Figure 5.4: DAQD, Env 0, Evals 3k, Cov 216, QD Score 176.9

LLQD @ 3k evals



Imperial College London

Appendix main experiment: naive benchmark II
LLQD vs DAQD



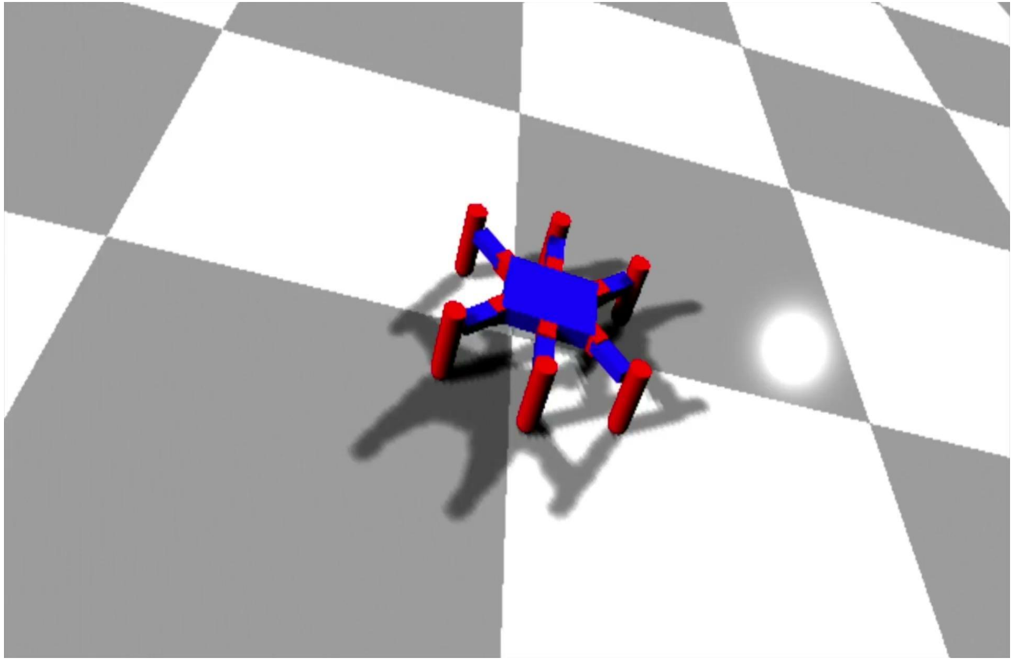
Visualisation of most north-western control in DAQD archive learnt in environment 0, applied in environment 1

Robot completely **wrongly aligned** after 3 second execution of control. Problematic for navigation tasks.

Imperial College London

Appendix: Main experiment naive benchmark III

LLQD vs DAQD



Visualisation of most north-western control in archive specialized on environment 1 (via LLQD)

Robot **“uses”** dynamics to jump. Final **alignment highly accurate**, thus useful for navigation tasks.

Appendix: LLQD pseudocode I

Algorithm 2 LLQD

Define LLQD class $\mathcal{L}$, which contains
  self.archive $\mathcal{A} = \emptyset$
  self.imagined_archive $\tilde{\mathcal{A}} = \emptyset$
  self.dynamics_model $\tilde{p}_\theta$
  self.replay buffer $\mathcal{B} = \emptyset$
  self.num_real_evals = 0

  self.compute_real_eval_ctrls():
    self.$\tilde{\mathcal{A}}$ = copy($\mathcal{A}$) ▷ sync imagined archive
    candidate_ctrls ← select_and_mutate($\mathcal{A}$) ▷ create candidates
    model_eval_ctrls ← model_eval(candidate_ctrls, $\tilde{p}_\theta$) ▷ model evaluate candidates
    succ_model_ctrls, $\tilde{\mathcal{A}}$ ← add_test(model_eval_ctrls, $\tilde{\mathcal{A}}$) ▷ try add cand. to $\tilde{\mathcal{A}}$, get succ. cand.
    self.real_eval_ctrls ← real_eval(succ_model_ctrls) ▷ evaluate cand. added to $\tilde{\mathcal{A}}$ in real env.
    return self.real_eval_ctrls ▷ return real eval. candidates

  self.update_buffer_and_train(real_eval_ctrls):
    succ_real_ctrls, $\mathcal{A}$ ← add_test(real_eval_ctrls, self.$\mathcal{A}$) ▷ try add cand. to $\mathcal{A}$, get succ. cand.
    self.$\mathcal{B}$ ← add_to_buffer(real_eval_ctrls.traj, self.$\mathcal{B}$) ▷ add traj. of real eval. cand. to $\mathcal{B}$
    self.$\tilde{p}_\theta$, ← train_from_buffer(self.$\tilde{p}_\theta$, self.$\mathcal{B}$) ▷ train model
    self.test_lhood_mean, self.test_lhood_std ← get_test_set_likelihood_stats(self.$\tilde{p}_\theta$, self.$\mathcal{B}$)
    self.num_real_evals += len(real_eval_ctrls) ▷ update number of real evaluations performed

env_list = get_rand_env_sequence() ▷ generate random sequence of environments to visit
llqd_list = [] ▷ list to store $\mathcal{L}$ objects

Appendix: LLQD pseudocode II

```
switch_env(env_list)            ▷ set real env to next env from random sequence; updates simulator
llqd = init_new_llqd_instance()                                    ▷ instantiate first ℒ object
real_eval_ctrls = llqd.compute_real_eval_ctrls()          ▷ real eval. selection of candidate ctrls
llqd.update_buffer_and_train(real_eval_ctrls)       ▷ first batch of ctrls always added to first ℬ
likeliest_llqd = llqd                   ▷ first ℒ object is always the likeliest object for first env
llqd_list.append(likeliest_llqd)                               ▷ add first ℒ object to storage list
while not reached end of env_list do
    switch_env(env_list)        ▷ set real env to next env from random sequence; updates simulator
    real_eval_ctrls = likeliest_llqd.compute_real_eval_ctrls() ▷ as bef.; next, get lhood of ctrls traj.
    ctrls_lhoods = [evaluate_likelihood(real_eval_ctrls.traj, llqd.p̃_θ) for llqd in llqd_list]
    test_lhood_means = [llqd.test_lhood_mean for llqd in llqd_list] ▷ get means of test set lhoods
    test_lhood_stds = [llqd.test_lhood_std for llqd in llqd_list]       ▷ get stds of test set lhoods
    num_llqds = len(llqd_list)                                  ▷ get number of existing ℒ objects
    transfer_candidate_idx = argmax(ctrls_lhoods)               ▷ get index of most likely ℒ object
    for i in range num_llqds do
        mx_id = argmax(ctrls_lhoods)                               ▷ index for most likely ℒ object
        if ctrls_lhoods[mx_id] > (test_lhood_means[mx_id] - EPS * test_lhood_stds[mx_id]): then
            likeliest_llqd = llqd_list[mx_id]
            break   ▷ if lhood of ctrls traj. close to mean lhood of ℒ's test set, ℒ ID-ed as relevant
        else           ▷ lhoods of ctrls traj. too low rel. to test set lhoods for likeliest ℒ, try next ℒ
            del ctrls_lhoods[mx_id]
            del test_lhood_means[mx_id]
            del test_lhood_stds[mx_id]
        end if
    end for
    if len(ctrls_lhoods) == 0 then      ▷ if no existing ℒ suitable, make new ℒ for observed traj.
        likeliest_llqd = init_new_llqd_instance()
        likeliest_llqd.p̃_θ ← copy_params(llqd_list[transfer_candidate_idx].p̃_θ)   ▷ transfer learning
```

Appendix: LLQD pseudocode III

```
        llqd_list.append(likeliest_llqd)                          ▷ add new ℒ to list of ℒ objects
    end if
    likeliest_llqd.update_buffer_and_train(real_eval_ctrls)  ▷ add ctrls traj. to likeliest ℒ's ℬ, train
end while
```

Imperial College London

Appendix: Transfer learning I

Coverage

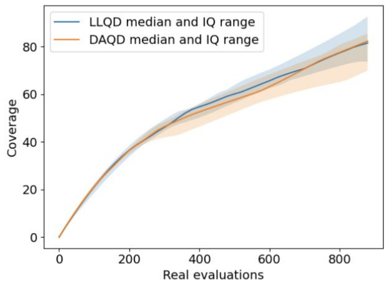


Figure 5.15: LLQD learning model for env 0 from env 1 vs DAQD baseline

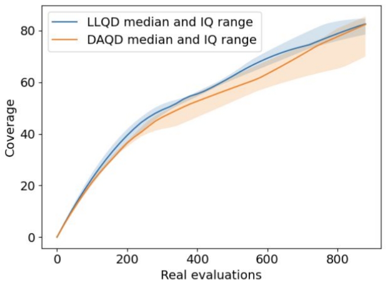


Figure 5.16: LLQD learning model for env 0 from env 2 vs DAQD baseline

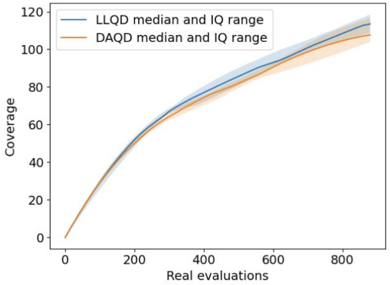


Figure 5.17: LLQD learning model for env 1 from env 0 vs DAQD baseline

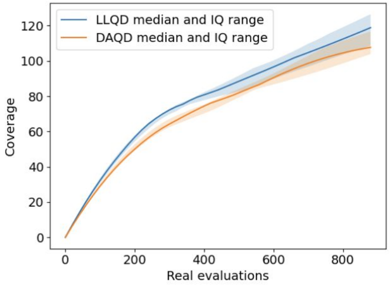


Figure 5.18: LLQD learning model for env 1 from env 2 vs DAQD baseline

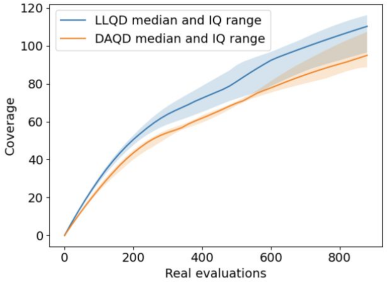


Figure 5.19: LLQD learning model for env 2 from env 0 vs DAQD baseline

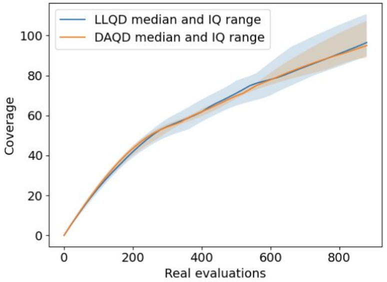


Figure 5.20: LLQD learning model for env 2 from env 1 vs DAQD baseline

Imperial College London

Appendix: Transfer learning I

QD Score

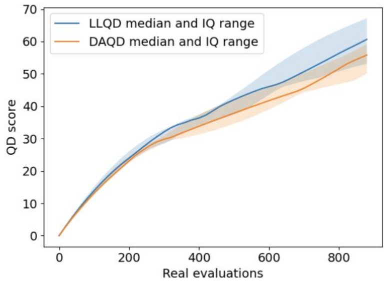


Figure 5.21: LLQD learning model for env 0 from env 1 vs DAQD baseline

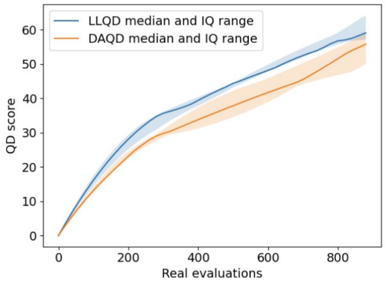


Figure 5.22: LLQD learning model for env 0 from env 2 vs DAQD baseline

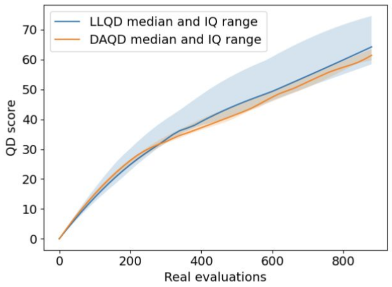


Figure 5.23: LLQD learning model for env 1 from env 0 vs DAQD baseline

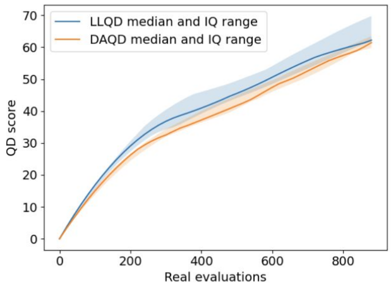


Figure 5.24: LLQD learning model for env 1 from env 2 vs DAQD baseline

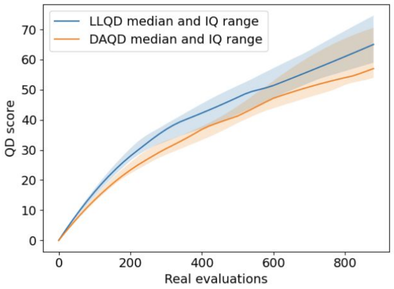


Figure 5.25: LLQD learning model for env 2 from env 0 vs DAQD baseline

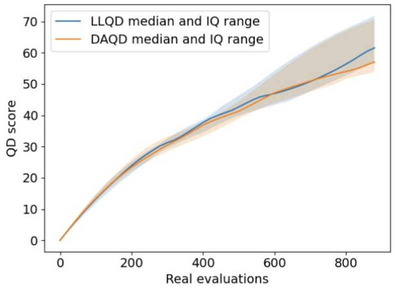


Figure 5.26: LLQD learning model for env 2 from env 1 vs DAQD baseline