# UCB-driven Utility Function Search for Multi-objective Reinforcement Learning Supplementary Material

May 28, 2025

# 1 Algorithms

**Algorithm 1** Fixed-MOPPO

**Input:** State $s_t$, weights $\mathbf{w}$
**Initialize:** $K$ weight-conditioned Actor-Critic Networks $\pi_k$ / $v^{\pi_k}$, scalarisation-vector subspaces $\mathbf{W}_k$ for $k = 1, \ldots, K$, and memory buffer $\mathcal{E}$ size of $D$.
**for** $k = 1$ **to** $K$ **do**
  **for** $t = 1$ **to** $D$ **do**
    $\mathbf{w}_{pivot} \leftarrow$ get pivot weight$(\mathbf{W}_k)$
    $a_t \leftarrow \pi_k(s_t, \mathbf{w}_{pivot})$
    $s_{t+1}, \mathbf{r_t} \leftarrow simulator(a_t)$
    $\mathcal{E} \leftarrow \mathcal{E} \cup \langle s_t, a_t, \mathbf{w}_{pivot}, \mathbf{r_t}, s_{t+1} \rangle$
    $s_t \leftarrow s_{t+1}$
  **end for**
  sample $\langle s_t, a_t, \mathbf{w}_{pivot}, \mathbf{r_t}, s_{t+1} \rangle \leftarrow \mathcal{E}$
  $\theta \leftarrow \theta + \eta \left(\nabla_\theta \log \pi_\theta(s_t, a; \mathbf{w}_{pivot})\right) \left(A^\pi(s_t, a_t; \mathbf{w}_{pivot})\right)$
  $\phi \leftarrow \phi + ||\boldsymbol{V}^{\pi_k}(s_t; \mathbf{w}_{\boldsymbol{pivot}}) - \boldsymbol{V}^{\pi_k}(s_{t+1}; \mathbf{w}_{\boldsymbol{pivot}})||^2$
  clear $\mathcal{E}$
**end for**

# 2 Benchmark Problems

This section provide detail objective return for each problems. Where $C$ in the following equations is live bonus.

## 2.1 Swimmer-v2

Observation and action space: $\mathcal{S} \in \mathbb{R}^8, \mathcal{A} \in \mathbb{R}^2$.
The first objective is forward speed in x axis:

$$R_1 = v_x \tag{1}$$

The second objective is energy efficiency:

$$R_2 = 0.3 - 0.15 \sum_i {a_i}^2, \quad a_i \in (-1, 1) \tag{2}$$

**Algorithm 2** Random-MOPPO

**Input:** State $s_t$, weights $\mathbf{w}$
**Initialize:** $K$ weight-conditioned Actor-Critic Networks $\pi_k$ / $v^{\pi_k}$, scalarisation-vector subspaces $\mathbf{W}_k$ for $k = 1, \ldots, K$, memory buffer $\mathcal{E}$ size of $D$, and scalarisation-vector re-sampling frequency $RF$.
**for** $k = 1$ **to** $K$ **do**
  **for** $t = 1$ **to** $D$ **do**
    **if** $t$ % $RF = 0$ **then**
      $\mathbf{w}_t \leftarrow$ uniform random sample($\mathbf{W}_k$)
    **end if**
    $a_t \leftarrow \pi_k(s_t, \mathbf{w_t})$
    $s_{t+1}, \mathbf{r_t} \leftarrow simulator(a_t)$
    $\mathcal{E} \leftarrow \mathcal{E} \cup \langle s_t, a_t, \mathbf{w}_t, \mathbf{r_t}, s_{t+1} \rangle$
    $s_t \leftarrow s_{t+1}$
  **end for**
  sample $\langle s_t, a_t, \mathbf{w}_t, \mathbf{r_t}, s_{t+1} \rangle \leftarrow \mathcal{E}$
  $\theta \leftarrow \theta + \eta \left(\nabla_\theta \log \pi_\theta(s, a; \mathbf{w}_t)\right) \left(A^\pi(s_t, a_t; \mathbf{w}_t)\right)$
  $\phi \leftarrow \phi + ||\boldsymbol{V}^{\pi_k}(s_t; \mathbf{w}_t) - \boldsymbol{V}^{\pi_k}(s_{t+1}; \mathbf{w}_t)||^2$
  clear $\mathcal{E}$
**end for**

## 2.2 HalfCheetah-v2

Observation and action space: $\mathcal{S} \in \mathbb{R}^{17}, \mathcal{A} \in \mathbb{R}^6$.
The first objective is forward speed in x axis:

$$R_1 = \min(v_x, 4) + C \tag{3}$$

The second objective is energy efficiency:

$$R_2 = 4 - \sum_i {a_i}^2 + C, \quad a_i \in (-1, 1) \tag{4}$$

$$C = 1 \tag{5}$$

## 2.3 Walker2d-v2

Observation and action space: $\mathcal{S} \in \mathbb{R}^{17}, \mathcal{A} \in \mathbb{R}^6$.
The first objective is forward speed in x axis:

$$R_1 = v_x + C \tag{6}$$

The second objective is energy efficiency:

$$R_2 = 4 - \sum_i {a_i}^2 + C, \quad a_i \in (-1, 1) \tag{7}$$

$$C = 1 \tag{8}$$

## 2.4 Ant-v2

Observation and action space: $\mathcal{S} \in \mathbb{R}^{27}, \mathcal{A} \in \mathbb{R}^8$.
The first objective is forward speed in x axis:

$$R_1 = v_x + C \tag{9}$$

The second objective is forward in y axis:

$$R_2 = v_y + C \tag{10}$$

$$C = 1 - 0.5 \sum_i a_i^2, \quad a_i \in (-1, 1) \tag{11}$$

**Algorithm 3** UCB-MOPPO

**Input:** Environment state $S_t$ and full weight set $\mathbf{W}$.
**Initialize:** $K$ weight-conditioned Actor-Critic networks $\{(\pi_k, v^{\pi_k})\}_{k=1}^{K}$; for each $k$, a predetermined subspace $\mathbf{W}_k \subset \mathbf{W}$ (of size $M$); working pool $\widetilde{\mathbf{W}}_k \leftarrow \mathbf{W}_k$; number of objectives $m$; warm-up iterations $Q$; evaluation interval $C$; for each $k$ and $j = 1, \ldots, m$, surrogate dataset $D^{k,j}_{\text{surrogate}} \leftarrow \emptyset$; surrogate models $\{f^{k,j}_{\text{bagging}}\}$; and dynamic weight pool $\mathcal{E}$ (of size $D$).
**for** $t = 0$ **to** $T$ **do** ▷ Warm-up Stage
  **for** $k = 1$ **to** $K$ **do**
    $\widetilde{\mathbf{W}}_k \leftarrow \text{getPivotWeights}(\mathbf{W}_k)$
    $\pi_k \leftarrow \text{FixWeightOptimization}(\pi_k, \widetilde{\mathbf{W}}_k)$
  **end for**
  ▷ Periodic Evaluation (every $C$ iterations)
  **if** $t \mod C = 0$ **then**
    **for** $k = 1$ **to** $K$ **do**
      **for all** $\mathbf{w} \in \widetilde{\mathbf{W}}_k$ **do**
        $V^{\pi_k}_{j,\mathbf{w}} \leftarrow \text{Simulate}(\pi_k, \mathbf{w})$ ▷ for each objective $j$
      **end for**
    **end for**
  **end if**
  **if** $t > Q$ **then** ▷ Construct Surrogate Training Data
    **for** $k = 1$ **to** $K$ **do**
      **for all** $\mathbf{w} \in \widetilde{\mathbf{W}}_k$ **do**
        **for** $z = 0$ **to** $\lfloor t/C \rfloor - 1$ **do**
          **for** $j = 1$ **to** $m$ **do**
            $\Delta V^{\pi_k,(z \to z+1)}_{j,\mathbf{w}} \leftarrow V^{\pi_k,z+1}_{j,\mathbf{w}} - V^{\pi_k,z}_{j,\mathbf{w}}$
            Append $(\mathbf{w}, \Delta V^{\pi_k,(z \to z+1)}_{j,\mathbf{w}})$ to $D^{k,j}_{\text{surrogate}}$
          **end for**
        **end for**
      **end for**
    **end for**
    ▷ Update Surrogate Models
    **for** $k = 1$ **to** $K$ **do**
      **for** $j = 1$ **to** $m$ **do**
        Update $f^{k,j}_{\text{bagging}}$ using $D^{k,j}_{\text{surrogate}}$
      **end for**
    **end for**
    ▷ Scalarisation-Vector Selection via UCB Acquisition
    **for** $k = 1$ **to** $K$ **do**
      $\mathcal{D}_k \leftarrow \emptyset$
      **for all** $\mathbf{w} \in \mathbf{W}_k$ **do** ▷ Candidates from the full subspace
        $V^{\pi_k}_{\mathbf{w}} \leftarrow \text{Simulate}(\pi_k, \mathbf{w})$
        **for** $j = 1$ **to** $m$ **do**
          $\hat{V}^{\pi_k}_{j,\mathbf{w}} \leftarrow V^{\pi_k}_{j,\mathbf{w}} + f^{k,j}_{\text{bagging}}(\mathbf{w})$
          $\widetilde{V}^{\pi_k}_{j,\mathbf{w}} \leftarrow \hat{V}^{\pi_k}_{j,\mathbf{w}} + \beta_{t'} \cdot \sigma_{k,j}(\mathbf{w})$
        **end for**
        Let $\mathbf{V}_{\mathbf{w}} \leftarrow \left(\widetilde{V}^{\pi_k}_{1,\mathbf{w}}, \ldots, \widetilde{V}^{\pi_k}_{m,\mathbf{w}}\right)$
        Compute $HV \leftarrow \texttt{HV}\Big(\texttt{Pareto}\big(\mathcal{L} \cup \{\mathbf{V}_{\mathbf{w}}\}\big)\Big)$, where $\mathcal{L}$ is the set of existing objective vectors.
        Add the pair $(\mathbf{w}, HV)$ to $\mathcal{D}_k$
      **end for**
      Sort $\mathcal{D}_k$ in descending order of $HV$
      Update working pool: $\widetilde{\mathbf{W}}_k \leftarrow$ top $N$ weights from $\mathcal{D}_k$
    **end for**
  **end if**
**end for**

### 2.5 Hopper-v2

Observation and action space: $\mathcal{S} \in \mathbb{R}^{11}, \mathcal{A} \in \mathbb{R}^{3}$.
The first objective is forward speed in x axis:

$$R_1 = 1.5v_x + C \tag{12}$$

The second objective is jumping height:

$$R_2 = 12(h - h_{init}) + C \tag{13}$$

$$C = 1 - 2e^{-4} \sum_i a_i^2, \quad a_i \in (-1, 1) \tag{14}$$

### 2.6 Hopper-v3

Observation and action space: $\mathcal{S} \in \mathbb{R}^{11}, \mathcal{A} \in \mathbb{R}^{3}$.
The first objective is forward speed in x axis:

$$R_1 = 1.5v_x + C \tag{15}$$

The second objective is jumping height:

$$R_2 = 12(h - h_{init}) + C \tag{16}$$

The third objective is energy efficiency:

$$R_3 = 4 - \sum_i a_i^2 + C \tag{17}$$

$$C = 1 \tag{18}$$

## 3 PPO Hyperparameters

Table 1: Hyper-parameter configuration of MOPPO algorithms.

| **Hyperparameters** | **Value** |
|---|---|
| Policy Number | 10 |
| Max Training Iterations | $2 \times 10^6$ |
| Number of Cells | 64 |
| Actor Learning Rate | $3 \times 10^{-4}$ |
| Critic Learning Rate | $3 \times 10^{-4}$ |
| Memory Size | 2500 |
| K Epochs | 10 |
| Gamma | 0.99 |
| Lambda | 0.95 |
| C1 Coefficient | 0.5 |
| C2 Coefficient | 0 |
| Epsilon Clip | 0.2 |
| Minibatch Size | 64 |

## 4 Convex Coverage Set Expansion

In this section, we illustrate the expansion of the Convex Coverage Set (CCS) throughout the training process using our proposed UCB-MOPPO algorithm. The graphs are arranged sequentially from left to right and top to bottom, showing the progressive evolution of the CCS. Each graph depicts 100 sub-space vectors representing a two-objective optimization problem. The detailed comparison of baselines is provided in Table 2, reporting the mean and standard deviation over three random seeds.

Table 2: Evaluation of HV and EU metrics for continuous MORL tasks over three independent runs. The best results are highlighted in **bold**.

| **Benchmark** | **Metric** | **UCB** | **Mean** | **Random** | **Fixed** | **PGMORL** | **PDMORL** | **CAPQL** | **GPI-LS** |
|---|---|---|---|---|---|---|---|---|---|
| Swimmer-V2 | HV ($10^4$) | **5.60** ± .18 | 4.72 ± .19 | 4.56 ± .23 | 4.45 ± .11 | 1.67 ± .09 | 1.77 ± .02 | 2.40 ± .03 | 4.75 ± .02 |
| | EU ($10^2$) | **2.18** ± .48 | 2.16 ± .25 | 2.08 ± .22 | 2.11 ± .12 | 1.23 ± .42 | 1.24 ± .49 | 1.79 ± .45 | 2.14 ± .41 |
| Halfcheetah-V2 | HV ($10^7$) | 1.78 ± .23 | 1.60 ± .07 | 1.21 ± .09 | 1.12 ± .05 | 0.58 ± .01 | 0.62 ± .02 | **2.20** ± .08 | 2.16 ± .02 |
| | EU ($10^3$) | 3.88 ± .22 | 3.58 ± .25 | 3.55 ± .30 | 3.44 ± .36 | 2.30 ± .44 | 2.42 ± .31 | **4.47** ± .03 | 4.30 ± .08 |
| Walker2d-V2 | HV ($10^7$) | **1.40** ± .03 | 1.29 ± .13 | 1.15 ± .06 | 1.13 ± .07 | 0.44 ± .02 | 0.56 ± .04 | 0.18 ± .05 | 1.22 ± .04 |
| | EU ($10^3$) | **3.54** ± .42 | 3.42 ± .29 | 3.34 ± .29 | 3.30 ± .33 | 1.98 ± .21 | 2.24 ± .34 | 1.50 ± .12 | 3.40 ± .30 |
| Ant-V2 | HV ($10^7$) | **1.07** ± .09 | 0.92 ± .04 | 0.65 ± .01 | 0.60 ± .01 | 0.61 ± .10 | 0.66 ± .02 | 0.45 ± .04 | 0.81 ± .01 |
| | EU ($10^3$) | **2.96** ± .43 | 2.61 ± .17 | 2.35 ± .24 | 2.21 ± .39 | 2.28 ± .35 | 2.41 ± .20 | 2.03 ± .42 | 2.84 ± .26 |
| Hopper-V2 | HV ($10^7$) | **0.84** ± .05 | 0.81 ± .04 | 0.79 ± .01 | 0.76 ± .02 | 0.23 ± .02 | 0.25 ± .02 | 0.25 ± .07 | 0.80 ± .02 |
| | EU ($10^3$) | 2.77 ± .28 | **2.84** ± .30 | 2.77 ± .18 | 2.71 ± .26 | 1.48 ± .47 | 1.51 ± .18 | 1.46 ± .11 | 2.75 ± .07 |
| Hopper-V3 | HV ($10^{10}$) | **3.62** ± .01 | 2.83 ± .22 | 2.90 ± .11 | 2.64 ± .18 | 0.63 ± .07 | 0.11 ± .02 | 0.31 ± .06 | 0.65 ± .03 |
| | EU ($10^3$) | 3.20 ± .02 | 3.19 ± .04 | **3.28** ± .20 | 2.90 ± .01 | 1.70 ± .04 | 1.74 ± .09 | 1.52 ± .29 | 2.14 ± .42 |

As training progresses, a clear trend of CCS growth emerges, characterized by an increasing spread and coverage of vectors across the objective space. Notably, the distribution of vectors provides valuable insights: regions where vectors are more widely scattered indicate areas with fewer vectors dominated by others, reflecting an expansion toward a more optimal and comprehensive CCS. This separation demonstrates the growing diversity and coverage of the vectors over time, effectively showcasing the CCS's ability to capture a wider range of trade-offs between objectives. Consequently, this illustrates the effectiveness of UCB-MOPPO in thoroughly exploring the objective space and improving the set of solutions throughout the training process.

## 4.1 Swimmer-V2

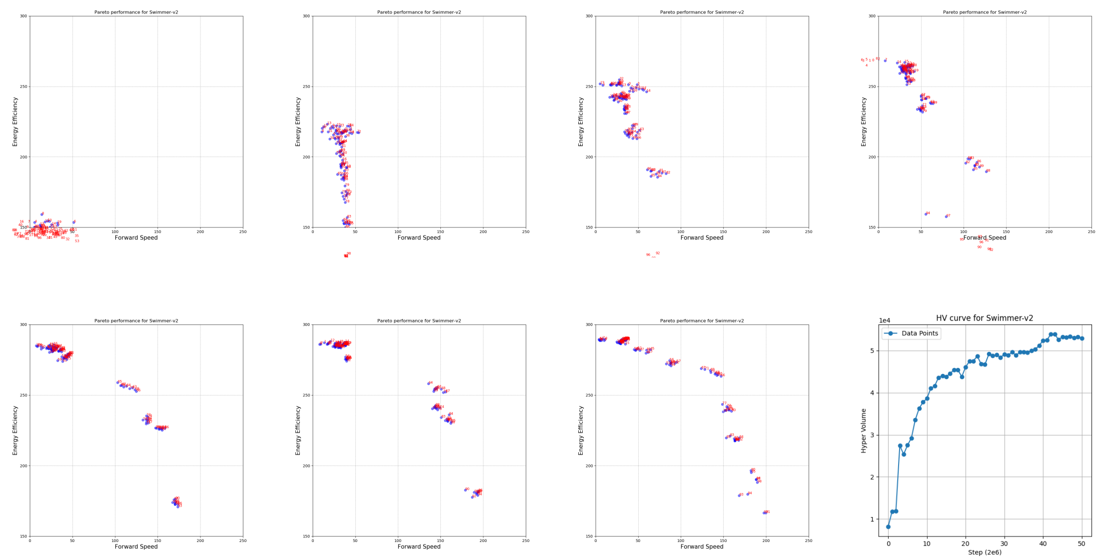


Figure 1: CCS expansion in Swimmer-V2 with one seed. The last graph shows the Hypervolume growth.

## 4.2 HalfCheetah-v2

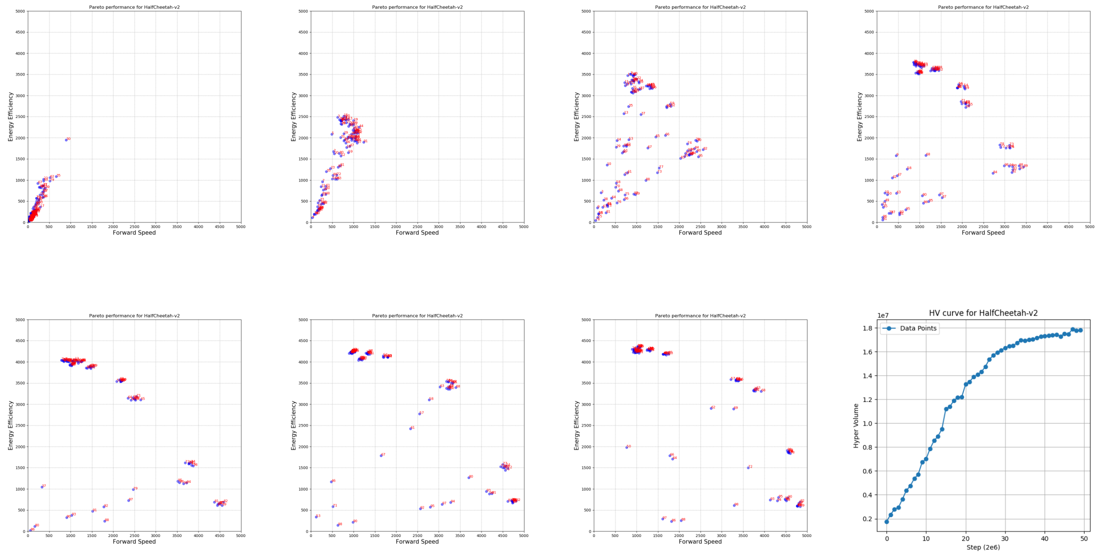


Figure 2: CCS expansion in HalfCheetah-V2 with one seed. The last graph shows the Hypervolume growth.

## 4.3 Walker2D-V2

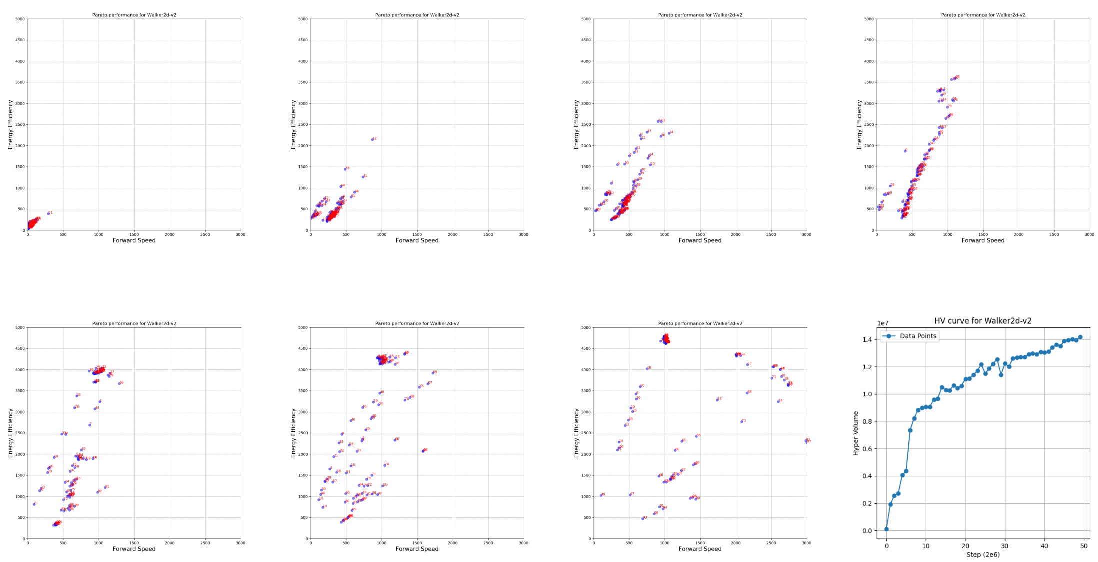


Figure 3: CCS expansion in Walker2D-V2 with one seed. The last graph shows the Hypervolume growth.

## 4.4 Ant-V2

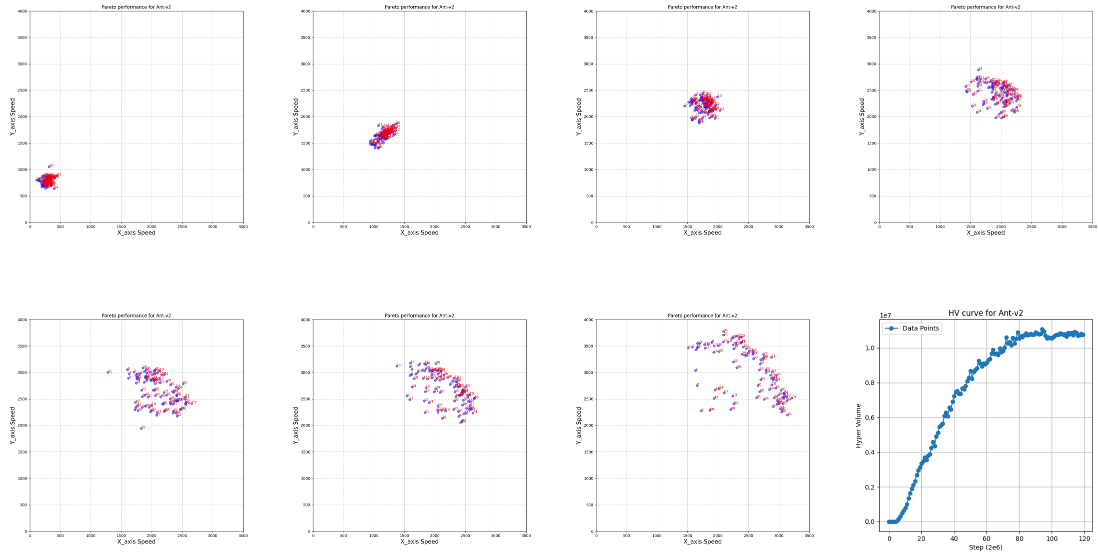


Figure 4: CCS expansion in Ant-V2 with one seed. The last graph shows the Hypervolume growth.

## 4.5 Hopper-V2

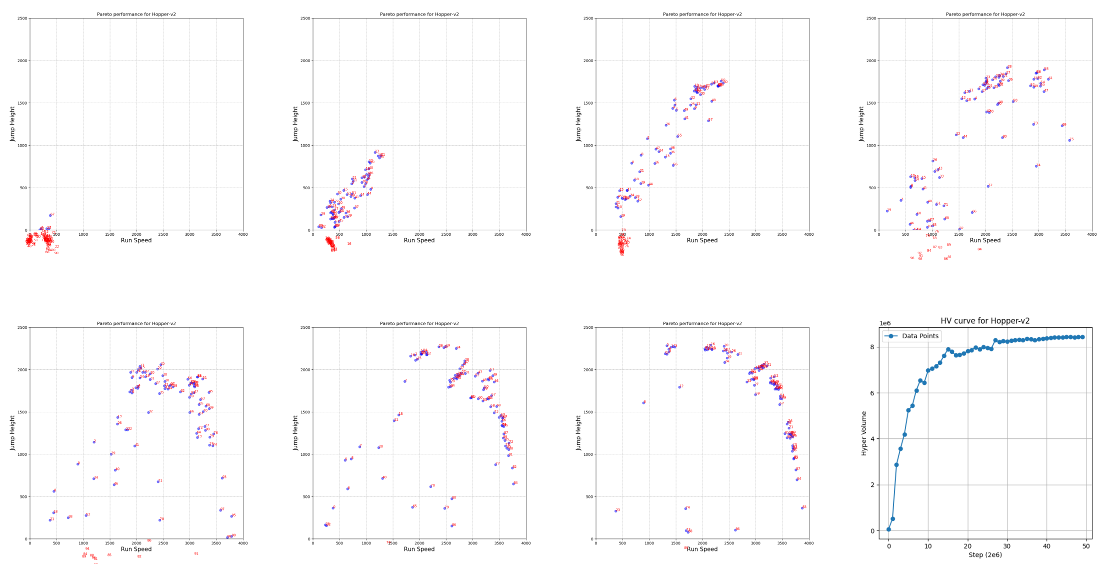


Figure 5: CCS expansion in Hopper-V2 with one seed. The last graph shows the Hypervolume growth.

## 4.6 Hopper-V3

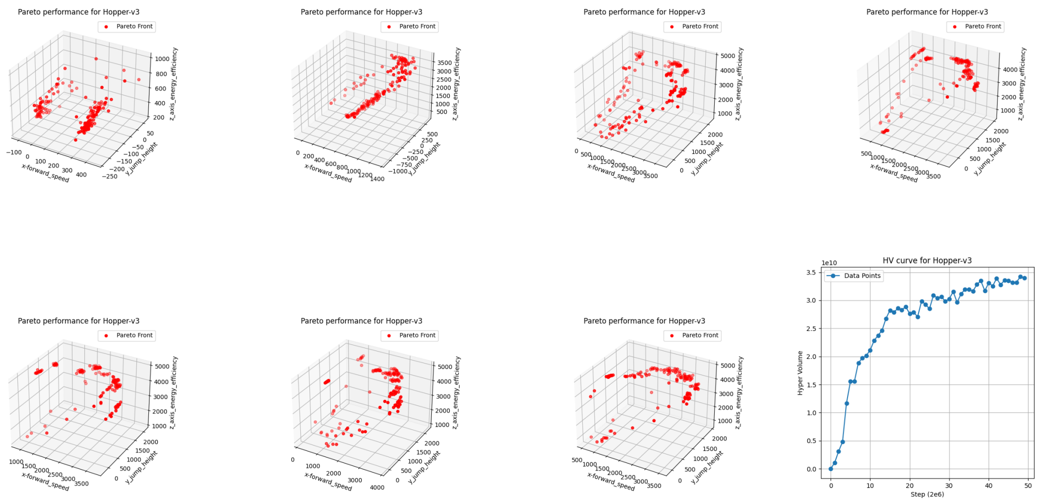


Figure 6: CCS expansion in Hopper-V3 with one seed. The last graph shows the Hypervolume growth.