

A TASK SPLITS

To evaluate the generalization capabilities of our approach, we curated distinct sets of training and testing scenarios for each environment. The specific scenarios comprising each train/test split are detailed in [Table 2](#).

Table 2: **Train/Test Task Splits for All Environments.** We list the specific scenarios used for training and out-of-distribution generalization testing.

Environment	Split	# of Tasks	Scenarios
LBF	Train	5	{8x8-2p-2f, 10x10-3p-3f, 15x15-3p-5f, 15x15-4p-5f, 16x16-5p-6f}
	Test	4	{12x12-4p-5f, 14x14-3p-3f, 17x17-6p-8f, 17x17-8p-10f}
RWARE	Train	15	{tiny-2ag, tiny-4ag, tiny-8ag, small-2ag, small-4ag, small-16ag, small-32ag, medium-8ag, medium-32ag, large-16ag, xlarge-8ag, xlarge-32ag, giant-32ag, colossal-8ag, titanic-16ag}
	Test	7	{tiny-16ag, medium-2ag, medium-16ag, xlarge-16ag, colossal-32ag, titanic-8ag, titanic-32ag}
Connector	Train	10	{12x12x4a, 15x15x3a, 18x18x4a, 21x21x5a, 24x24x6a, 27x27x7a, 30x30x10a, 33x33x11a, 36x36x12a, 39x39x13a}
	Test	11	{42x42x18a, 45x45x23a, 48x48x20a, 51x51x28a, 54x54x30a, 57x57x32a, 60x60x33a, 63x63x35a, 66x66x40a, 69x69x43a, 72x72x45a}

B DATASET QUALITY ABLATION

Do higher quality trajectories improve generalisation? As observed in [subsection 3.4](#), increasing dataset size does not lead to significant improvements in generalization to unseen tasks. A natural follow-up question is: *how does the quality of trajectories in the dataset affect training and test performance?* To investigate this, we conduct an experiment where training is performed with trajectories sampled from specific subsets of our dataset. Low-quality trajectories are those collected during the first two-thirds of the online training phase, while High-quality trajectories are those from the final third. Results on RWARE are shown in [Figure 7](#). For all algorithms, training performance improves with High-quality trajectories, though the gains on test tasks remain marginal. Across all three algorithms, training with Low-quality trajectories consistently yields the worst results on both training and test tasks. These results suggest that the most effective strategy is to prioritize High-quality trajectories while retaining a small fraction of Low-quality ones as negative examples.

C HYPERPARAMETERS

This section details the hyperparameters used for our experiments. [Table 3](#) lists the network architecture and environment-specific settings. The default training parameters are provided in [Table 4](#). Finally, algorithm-specific settings for MT-Oryx and MT-CQL-Sable are presented in [Table 5](#) and [Table 6](#), respectively.

Table 3: Default network settings for each environment.

Parameter	LBF	Connector	RWARE
Model embedding dimension	512	512	512
Number of transformer heads	4	4	4
Number of transformer blocks	1	1	1
HL-Gauss value support	$[-1, 1]$	$[-1, 1]$	$[-20, 20]$
HL-Gauss number of bins	51	51	51
Sable’s decay scaling factor	0.8	0.8	0.8

Table 4: Default training settings.

Hyperparameter	Value
Number of training updates	60 000
Number of evaluations	600
Number of evaluation episodes	32
Number of absolute evaluation episodes	320
Learning rate	1×10^{-3}
Discount (γ)	0.99
Polyak averaging coefficient (τ)	0.005
Maximum gradient norm	10
Sample sequence length	20
Sample batch size	480
Value temperature	1000
Policy temperature	0.1
Critic loss coefficient	1

Table 5: MT-Oryx specific settings.

Hyperparameter	Value
Value temperature	1000
Policy temperature	0.1
Critic loss coefficient	1
HL-Gauss smoothing ratio	0.75

Table 6: MT-CQL-Sable specific settings.

Hyperparameter	Value
CQL loss coefficient	10
HL-Gauss smoothing ratio	0.75

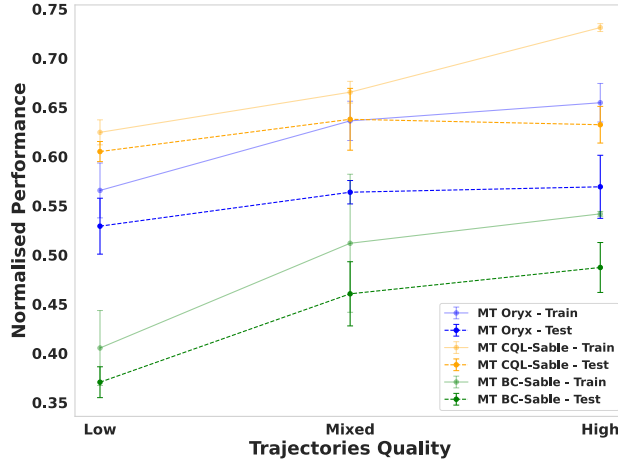


Figure 7: Performance of MT-Oryx, MT-CQL-Sable, and MT-BC-Sable on RWARE with different trajectory subsets. High-quality trajectories improve training performance, particularly for MT-CQL-Sable, but these gains do not transfer to the test tasks. Low-quality trajectories consistently yield the worst results.

D DATASETS

D.1 DATASET RELEASE PLAN

To guarantee the long-term reproducibility of this project, we will upload all of our datasets to a public HuggingFace repository². This will be done upon publication of this work.

D.2 DATASET STATISTICS

The following sections detail the statistics of the offline datasets for the RWARE, Connector, and LBF environments used in our experiments. Datasets were generated by recording rollouts from an online Sable (Mahjoub et al., 2025) agent at different intervals during its training. All data is collected from fixed intervals over training using an evaluation policy to vary the amount of data collected while maintaining a standard set of policies to sample from. For RWARE, we also create multiple datasets of different sizes by varying the number of evaluations sampled in order to perform our data-scaling experiments.

D.2.1 RWARE

For our data-scaling experiments in the RWARE environment, we generated three offline datasets of varying sizes. The datasets were constructed by collecting 122, 244, and 610 evaluation rollouts from a pre-trained online Sable agent (Mahjoub et al., 2025). Table 7 provides detailed statistics for each dataset size across all RWARE scenarios, illustrating how the number of episodes and transitions scales with the number of collected rollouts.

D.2.2 CONNECTOR

For the Connector environment, we generated 10 distinct offline datasets, one for each training scenario. Each dataset contains approximately 10 million transitions. The data collection process involved recording evaluation rollouts at 50 different checkpoints during the training of an online Sable agent. At each checkpoint, we generated 160 rollouts of 1280 timesteps each, resulting in a total of $50 \times 160 \times 1280 \approx 10.24$ million transitions per scenario. The ten scenarios used to create these datasets are listed in Table 9.

²<https://sites.google.com/view/multi-task-marl>

D.2.3 LBF

For LBF we collected all the the training data from an online Sable run for each LBF scenario.

Table 7: **RWARE dataset statistics across different data collection checkpoints.** We report the total number of episodes and timesteps (transitions) for each scenario, corresponding to datasets created from 122, 244, and 610 evaluation rollouts.

Scenario Name	122 Rollouts		244 Rollouts		610 Rollouts	
	Episodes	Timesteps	Episodes	Timesteps	Episodes	Timesteps
tiny-2ag	15,616	7,493,913	31,232	14,934,862	78,080	37,382,071
small-2ag	15,616	7,511,771	31,232	15,091,627	78,080	37,504,501
tiny-4ag	15,616	6,492,381	31,232	13,208,433	78,080	33,110,502
small-4ag	15,616	6,611,283	31,232	13,496,720	78,080	33,733,571
tiny-8ag	15,616	4,704,862	31,232	9,748,756	78,080	24,647,669
medium-8ag	15,616	2,502,476	31,232	5,148,947	78,080	12,747,091
xlarge-8ag	15,616	5,816,385	31,232	11,008,538	78,080	29,167,762
colossal-8ag	15,616	4,804,325	31,232	12,078,452	78,080	29,830,317
small-16ag	15,616	3,681,321	31,232	7,405,046	78,080	15,598,958
large-16ag	15,616	3,946,296	31,232	6,158,419	78,080	18,731,422
titanic-16ag	15,616	4,361,204	31,232	10,498,182	78,080	17,223,581
small-32ag	15,616	317,038	31,232	639,868	78,080	207,539
medium-32ag	15,616	4,147,400	31,232	8,386,685	78,080	20,855,336
xlarge-32ag	15,616	3,275,217	31,232	6,593,539	78,080	16,388,466
giant-32ag	15,616	3,682,013	31,232	6,513,235	78,080	12,706,872

Table 8: **Connector dataset statistics.** We generated a separate dataset of approximately 10.24 million transitions for each of the ten training scenarios.

Scenario Name	Total Timesteps
12x12x4a	$\approx 10.24 \times 10^6$
15x15x3a	$\approx 10.24 \times 10^6$
18x18x4a	$\approx 10.24 \times 10^6$
21x21x5a	$\approx 10.24 \times 10^6$
24x24x6a	$\approx 10.24 \times 10^6$
27x27x7a	$\approx 10.24 \times 10^6$
30x30x10a	$\approx 10.24 \times 10^6$
33x33x11a	$\approx 10.24 \times 10^6$
36x36x12a	$\approx 10.24 \times 10^6$
39x39x13a	$\approx 10.24 \times 10^6$

Table 9: **LBF dataset statistics.** We generated a separate dataset of approximately 4 million transitions for each of the 5 training scenarios.

Scenario Name	Total Timesteps
8x8-2p-2f	$\approx 3.99 \times 10^6$
10x10-3p-3f	$\approx 3.99 \times 10^6$
15x15-3p-3f	$\approx 3.99 \times 10^6$
15x15-4p-5f	$\approx 3.99 \times 10^6$
16x16-5p-6f	$\approx 3.99 \times 10^6$