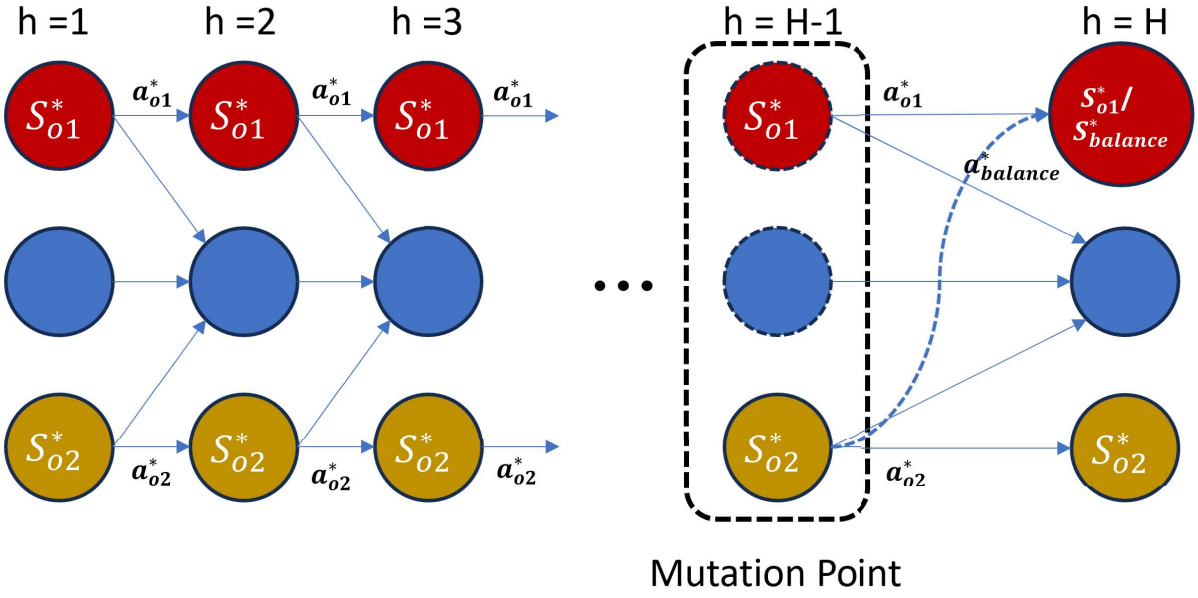

Figure 3: Lower bound example: MO combination lock

three preferences, it can achieve an expected utility of 0.3. There exist several (K) paths in this MOMDP instance delivering good rewards. To find all good policies, the agent needs at least to see $K \cdot 3^{H-1}$ samples. We formalize this in Theorem 2.

The scenario where preferences precisely match the three specific weights of $[1, 0]$, $[0.5, 0.5]$, and $[0, 1]$ is considerably rare, as preferences are typically drawn from an infinite weight simplex. In practical training situations, encountering a variety of preferences can lead to conflicting scenarios for the agent. This conflict arises because different preferences may necessitate different actions or strategies, potentially causing the agent's policy to experience "forgetfulness" or an inability to consistently apply learned behaviors across varying preferences. Such a situation complicates the training process and increases sample complexity.

Theorem 2. *(Koenig & Simmons, 1993; Uchendu et al., 2023) When using a 0-initialized exploration policy, there exists a MOMDP instance where the sample complexity is exponential to the horizon to see a partially optimal solution and a multiplicative exponential complexity to see an optimal solution.*

4.5.2 Upper bound of DG-MORL

We now give the upper bound of DG-MORL's regret. We start by making assumptions about the quality of the guide policy, i.e., it can cover some of the features that can be visited under the optimal policy, and the coverage rate is restricted by a concentrate ability coefficient. Then we give a Lemma to show that in a sequential decision making setting, the difference between the optimal policy and the guide policy is bounded and this difference will not increase under the self-evolution mechanism; but may decrease. Next, we provide several key definitions including Pareto suboptimality gap and sequence Pareto regret. Then we give a performance guarantee.

Assumption 1. *(Uchendu et al., 2023) Assume there exists a feature mapping function $\phi : \mathcal{S} \to \mathbb{R}^d$, that for any policy π, $Q^\pi(s, a)$ and $\pi(s)$ depends on s only through $\phi(s)$. The guide policy π_g[4] is assumed to cover the*

[4]Given that the policy set Π_g can be interpreted as a mixed policy tailored to specific scenarios, it retains its generality even when Π_g is considered as a singular policy.

states visited by the optimal policy corresponding to the preference weight vector w:

$$\sup_{s,h,w} \frac{d^{\pi^*}_{h,w}(\phi(s))}{d^{\pi_g}_{h,w}(\phi(s))} \leq C \tag{4}$$

$d^{\pi^*}_{h,w}(\phi(s))$ and $d^{\pi_g}_{h,w}(\phi(s))$ represents the probability distribution of feature $\phi(s)$ being visited at time step h under the preference weight w, while following the optimal policy π^* or guide policy π_g respectively. C is the concentratability coefficient (Rashidinejad et al., 2021; Guo et al., 2023). It quantifies the extent to which the guide policy π_g covers the optimal policy π^* in the feature space and restricts the probability of the optimal policy visiting a certain feature $\phi(s)$ to not exceed C times the probability of the guide policy visiting that feature.

Nonetheless, Assumption 1 is formulated within contextual bandit which does not sufficiently capture the complexities of sequential decision-making processes. We extend Assumption 1 to the context of sequential decision-making and incorporates a self-evolving mechanism, thereby establishing Lemma 3. As a result of the self-evolving mechanism, the guide policy π_g progressively enhances its performance over time.

Lemma 3. *For a MOMDP, the sequence concentration coefficient $C_{seq}(t)$ under preference weight w is defined as the maximum value of the concentration coefficient across all time steps h while with the self-evolving mechanism:*

$$\max_{h\in\{1,2,\ldots,H\}} \frac{d^{\pi^*}_{h,w}(\phi(s))}{d^{\pi_g}_{h,w}(\phi(s))} \leq C_{seq}(t,w) \leq C_{seq}(t_0,w) \tag{5}$$

Furthermore, we can give the Pareto concentration coefficient $C_{Pareto}(t) = \max_{w\in\mathcal{W}} C_{seq}(t,w) \leq C_{Pareto}(t_0)$
So we have:

$$\max_{h\in\{1,2,\ldots,H\}} \frac{d^{\pi^*}_{h}(\phi(s))}{d^{\pi_g}_{h}(\phi(s))} \leq C_{Pareto}(t) \leq C_{Pareto}(t_0) \tag{6}$$

We now give a few key definitions to calculate the upper bound of regret. One critical concept is the *Pareto suboptimality gap*. This term refers to the measure of distance between the current policy's performance on a specific arm of the multi-objective contextual bandit (MOCB) and the true Pareto optimal solution for that arm.

Definition 2. *(Pareto suboptimality gap (Lu et al., 2019)) Let x be an arm in $\mathcal{X}$, i.e. the arm set of the MOCB. The Pareto suboptimality gap Δx is defined as the minimal scalar $\sigma \geq 0$ so that x becomes Pareto optimal by adding σ to all entries of its expected reward.*

We now extend the Definition 2 to MOMDP to fit in sequence decision making process:

Definition 3. *(Sequence Pareto Regret) In a MOMDP, given a trajectory $\tau = (s_1, a_1, s_2, a_2, ..., s_H)$, the expectation of sequence Pareto regret Δ^{π}_{seq} for a finite horizon is:*

$$\Delta^{\pi}_{seq} = \Delta^{\pi}_{s_1} + \mathbb{E}[\Delta^{\pi}_{s_2} \mid s_1, a_1] + \mathbb{E}[\Delta^{\pi}_{s_3} \mid s_2, a_2] + \ldots + \mathbb{E}[\Delta^{\pi}_{s_H} \mid s_{H-1}, a_{H-1}] \tag{7}$$

where Δ_{s_h} is the Pareto suboptimiliy gap in state s_h, defined as follows:

$$\Delta^{\pi}_{s_h} = \min\left\{ \sigma \geq 0 \;\middle|\; \forall a \in A,\, \nexists a' \in A,\; \boldsymbol{Q}^*_{s_h,a} \geq \boldsymbol{Q}_{\pi}(s_h,a) + \sigma \cdot \mathbf{1} \;,\; \boldsymbol{Q}^*_{s_h,a} \neq \boldsymbol{Q}_{\pi}(s_h,a) + \sigma \cdot \mathbf{1}, \right\} \tag{8}$$

In Equation 7 , $\Delta^{\pi}_{\text{seq}}$ is the expected sequence Pareto regret of policy π. It quantifies the cumulative performance loss of policy π relative to the Pareto optimal policy throughout the decision-making process. $\Delta^{\pi}_{s_1}$ is the Pareto regret at the initial state s_1. $\mathbb{E}[\Delta^{\pi}_{s_h} \mid s_{h-1}, a_{h-1}]$ is the expected Pareto regret at state s_h which is conditioned on the previous state and action. By summing the initial Pareto regret and the conditional expected regrets at each subsequent state, we obtain the total expected sequence Pareto regret.

Equation 8 clarifies the conditions for Pareto dominance. $\boldsymbol{Q}^*_{s_h,a}$ represents the multi-objective Q-value of the optimal policy π^* for action a in state s_h. $\boldsymbol{Q}_{\pi}(s_h,a)$ is the multi-objective Q-value of policy π for action in

state s_h. σ indicates the minimum amount that needs to be added to all objectives for the current policy π to reach the level of the optimal policy across all actions while $\mathbf{1}$ is a full ones vector whose dimensionality is the same as the reward space. For each action $a \in A$ of strategy π, there does not exist an action $a' \in A$ such that $\boldsymbol{Q}^*_{s_h,a}$ is no worse than $\boldsymbol{Q}_\pi(s_h, a) + \sigma \cdot \mathbf{1}$ across all objectives, and strictly better on at least one objective. We accurately measure the Pareto suboptimality when following π relative to the optimal policy at state s_h.

Theorem 4. *The expectation of the sequence Pareto regret Δ^π_{seq} is the sum of the expectation of the Pareto regret for each step.*

$$\mathbb{E}[\Delta^\pi_{seq}] = \sum_{h=1}^{H} \mathbb{E}[\Delta^\pi_{s_h}] \tag{9}$$

Equation 9 indicates the additivity of regret, i.e. in MOMDP, the expected Pareto regret of a policy over the entire decision sequence can be decomposed into the sum of the expected Pareto regret at each time step.

We put the proof of Theorem 4 below.

Proof. Take expectations on both side:

$$\mathbb{E}[\Delta^\pi_{\text{seq}}] = \mathbb{E}[\Delta^\pi_{s_1}] + \mathbb{E}[\mathbb{E}[\Delta^\pi_{s_2} \mid s_1, a_1]] + \mathbb{E}[\mathbb{E}[\Delta^\pi_{s_3} \mid s_2, a_2]] + \ldots + \mathbb{E}[\mathbb{E}[\Delta^\pi_{s_H} \mid s_{H-1}, a_{H-1}]]$$

According to the Law of total expectation:

$$\mathbb{E}[\Delta^\pi_{\text{seq}}] = \mathbb{E}[\Delta^\pi_{s_1}] + \mathbb{E}[\Delta^\pi_{s_2}] + \mathbb{E}[\Delta^\pi_{s_3}] + \ldots + \mathbb{E}[\Delta^\pi_{s_H}] = \sum_{h=1}^{H} \mathbb{E}[\Delta_{s^\pi_h}]$$

□

We can now give the upper bound of the sum of the sequence Pareto regret during the training for T rounds.

Theorem 5. *(Performance Guarantee)*

Assuming that the environment satisfies the Markov property, and there exists a feature mapping function ϕ such that for any policy π both $\boldsymbol{Q}_\pi(s, a)$ and $\pi(s)$ depend only on $\phi(s)$. The DG-MORL algorithm guarantees that the guiding policy π_g progressively approaches the optimal policy π^* through self-evolution mechanisms. $f(t)$ is an abstract regret function, it depends on the specific algorithm we used to train π_e. During each training round $t \in [T]$, the algorithm executes policy π_t, and the sum of sequence Pareto regret $PR_{seq}(T)$ is bounded by:

$$PR_{seq}(T) = \sum_{t=1}^{T} \Delta^{\pi_t}_{seq} \leq THR_{\max}C \tag{10}$$

where $C = C_{Pareto}(t_0) + f(t_0)$, $C_{Pareto}(t_0)$ and $f(t_0)$ are the Pareto concentration coefficient and regret function at the first round of training. This is to get a conservative upper bound as $C_{Pareto}(t_0) \geq C_{Pareto}(t)$ because of the self-evolving mechanism and $f(t_0) \geq f(t)$ because of the policy improvement. $R_{\max} = \max_{s,a} \|r(s, a)\|_1$, we use this 1-norm to get a relative conservative and general upper bound[5]

Proof. of Theorem 5

Decomposing the regret into contributions from the guide policy π_g and the exploration policy π_e.

In the t-th round of training, the Pareto regret of the sequence of the mixed strategies π_t can be divided into two parts:

1. Regret from $\pi_{g,t}$

$$\Delta^{\pi_{g,t}}_{seq} = \sum_{h=1}^{h_g} \Delta^{\pi_{g,t}}_{s_h}$$

[5]While we acknowledge that using preference weight might add specificity, we prefer to use the 1-norm since it relaxes the upper bound. This approach ensures that our theorem remains valid even in the worst-case scenario.

where h_g is the last time step controlled by $\pi_{g,t}$

2.Regret from $\pi_{e,t}$

$$\Delta_{seq}^{\pi_{e,t}} = \sum_{h=h_g+1}^{H} \Delta_{s_h}^{\pi_{e,t}}$$

Thus, the total regret is :

$$\Delta_{seq}^{\pi_t} = \Delta_{seq}^{\pi_{g,t}} + \Delta_{seq}^{\pi_{e,t}}$$

With the help of the self-evolving mechanism, π_g can improve overtime. According to Lemma 3, at h timestep, the expected regret of $\pi_{g,t}$ is:

$$\mathbb{E}_{s_h}[\Delta_{s_h}^{\pi_{g,t}}] \leq R_{\max} \cdot C_{Pareto}(t)$$

In the worst case, the reward difference is $R_{\max}$, and the difference in state distribution is controlled by $C_{Pareto}(t)$. In the worst case, each time step can cause the maximized regret. The total expected regret for $\pi_{g,t}$ is:

$$\Delta_{h}^{\pi_{g,t}} \leq h \cdot R_{\max} \cdot C_{Pareto}(t)$$

Assuming we use ϵ-greedy strategy in the exploration policy π_e. For the exploration policy π_e, at each time step h, the expected regret of $\pi_{e,t}$ is:

$$\mathbb{E}_{s_h}[\Delta_{s_h}^{\pi_{e,t}}] \leq f(t) \cdot R_{\max}$$

This is because a random action is selected with probability ϵ, which may lead to a reward loss of up to $R_{\max}$ in the worst case. Therefore, the total expected regret for the exploration policy is:

$$\Delta_{H-h}^{\pi_{e,t}} \leq (H-h) f(t) \cdot R_{\max}$$

For T training rounds, the total regret is:

$$PR_{seq}(T) = \sum_{t=1}^{T} \Delta_{seq}^{\pi_t} \leq \sum_{t=1}^{T} (h_t \cdot R_{\max} \cdot C_{Pareto}(t) + (H-h_t) \cdot f(t) \cdot R_{\max})$$

where h_t is the number of time steps controlled by the guided policy during the t-th training round. During the training process, the number of time steps controlled by the guided policy decreases over time, i.e., h_t decreases with t. Since $C_{Pareto}(t)$ and $f(t)$ decreases with t, we approximate it by $C_{Pareto}(t_0)$ and $f(t_0)$, the value at the final round. We approximate that the steps controlled by π_g and π_e are all the maximized number of steps, i.e. H, the upper bound on the total regret is:

$$PR_{seq}(T) \leq THR_{\max}C$$

,where $C = C_{Pareto}(t_0) + f(t_0)$ □

5 Experiments

In this section, we introduce the baselines, benchmark environments and metrics. We then illustrate and discuss the results. Please note that, our DG-MORL does not require specific preferences to be covered by demonstrations. In MORL, as some preferences may share the same optimal policy, if the true PF is not known, it is impossible to cover all important preferences by demonstrations.