SWORD: AN EFFICIENT ADAPTATION OF MULTI-AGENT POLICY WITH SPARSE AND LOW RANK DECOMPOSITION FOR HETEROGENEOUS SATELLITE CLUSTER

Anonymous authors
Paper under double-blind review

ABSTRACT

This work investigates the adaptability of multi-agent policies to task changes in heterogeneous satellite clusters performing Earth observation missions. Since mission objectives and task requirements may vary across different mission scenarios, a policy trained for an initial task may experience a performance degradation when transferred to a new task, thereby requiring adaptation or fine-tuning. From a machine learning perspective, this problem can be formulated as a domain or task shift. Initially, the policy is trained using an on-policy multi-agent reinforcement learning (MARL) framework and subsequently transferred to a new task. To address the resulting performance degradation, we propose a policy adaptation approach based on Sparse and loW-Rank Decomposition (SWORD), which enables efficient fine-tuning of the previously learned policy by updating a reduced set of parameters. We first provide a theoretical analysis of the applicability of parameter-efficient fine-tuning within the MARL framework and subsequently validate the approach through experimental evaluation. The results demonstrate that SWORD's performance comparable to full-parameter fine-tuning while substantially reducing the number of trainable parameters. Furthermore, SWORD achieves improved performance compared with existing parameter-efficient fine-tuning baselines, demonstrating its potential for efficient policy adaptation in heterogeneous satellite cluster missions.

1 INTRODUCTION

The increasing deployment of satellite constellations reveals new opportunities for Earth Observation (EO) missions that require autonomous decision-making, dynamic resource management, and coordinated operation among multiple spacecraft (Herrmann & Schaub, 2023; Stephenson et al., 2024). In heterogeneous satellite clusters, satellites with different sensing and communication capabilities must coordinate the allocation of limited resources, including energy, storage capacity, sensing opportunities, and communication resources, to satisfy the mission-specific objectives. The complexity and dynamic nature of these environments motivates the application of multi-agent reinforcement learning (MARL), which enables satellite agents to learn coordination policies directly through interaction with the environment. MARL can provide a model-free decision-making framework that is potentially suitable for autonomous operation under uncertain and changing mission conditions (Hady et al., 2025).

However, satellite EO missions are not necessarily governed by a fixed set of objectives (Wang et al., 2026). In different missions the objective may prioritize different data collection objectives, or payload utilization criteria. Consequently, the reward functions and task objectives will also vary when the mission scenario changes. A trained policy for one mission may therefore no longer provides satisfactory performance when deployed to another mission with different objectives. As presented in Figure 1, transferring a trained policy to a new task with different mission objective decreases the average reward or policy's performance. This issue becomes a challenge in autonomous satellite systems application, where retraining a MARL policy from scratch for every new mission can be computationally expensive and may require extensive interaction with the environment. The ability

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

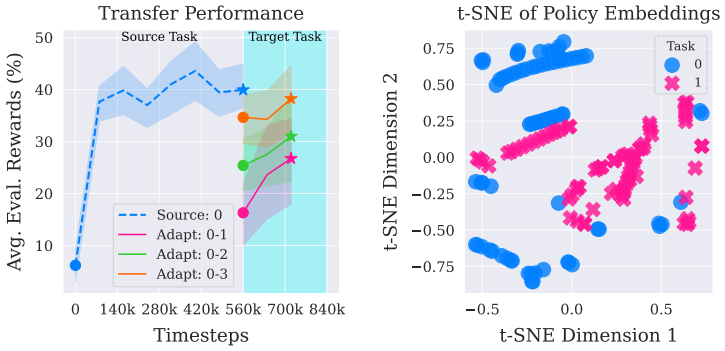

Figure 1: Policy Transfer Performance Degradation. Policy transfer to an unseen task with a different mission objective can cause performance degradation. Even after 160K timesteps of fine-tuning, the transferred policy remains below the target-task performance at 560K timesteps. The Task 0 $\rightarrow$ Task 1 transfer shows the largest degradation, while its t-SNE policy embeddings exhibit a distinct distribution from the source task. This observation motivates efficient policy adaptation for autonomous satellite clusters with changing mission objectives.

to reuse and efficiently adapt previously learned policies is therefore an important requirement for practical deployment of MARL in satellite systems.

Policy transfer and fine-tuning provide a potential solution to this problem by allowing knowledge acquired from a source task to be reused for a target task. Instead of learning a new policy from random initialization, the parameters of a previously trained policy can be adapted to the new task (Zhu et al., 2023; Hu et al., 2025). Task representation based policy transfer has been proposed (Qin et al., 2024). However, Parameter-efficient fine-tuning (PEFT) offers alternative by restricting adaptation to a small subset or low-dimensional representation of the original policy parameters (Han et al., 2024). In this work, we propose **Sparse and loW-Rank Decomposition (SWORD)** for efficient adaptation of previously learned MARL policies to new satellite mission objectives [1]. SWORD decomposes the policy update into sparse inspired from Zhang et al. (2026) and low-rank components with Singular Vector Decomposition (SVD) (Zhang et al., 2023), allowing the policy to adapt to the target task without updating the complete parameter space. This design aims to retain the transferable knowledge encoded in the source policy while providing sufficient flexibility to accommodate changes in the target mission and its reward structure. In particular, SWORD is designed for scenarios in which the reward function changes between the source and target tasks, representing different mission objectives in the satellite domain.

Moreover, this work investigates the theoretical basis of parameter-efficient policy adaptation in the MARL setting. We analyze the relationship between the restricted parameter update and the policy-gradient optimization process, providing a theoretical perspective on why sparse and low-rank updates can approximate the adaptation achieved by full-parameter fine-tuning. The theoretical analysis establishes the foundation for applying SWORD to MARL policy adaptation and motivates its use as an efficient alternative to conventional fine-tuning. Our contributions of this work are summarized as follows: **1) Different mission objective policy adaptation:** We formulate policy adaptation in heterogeneous satellite clusters as a transfer or domain adaptation problem between missions with different task objectives, enabling a previously trained MARL policy to be adapted to a new mission rather than retrained from scratch. **2) Efficient MARL policy adaptation:** We propose a parameter-efficient adaptation method that combines sparse and low-rank policy updates to reduce the number of trainable parameters while retaining the performance of the original policy. **3) Theoretical analysis:** We provide a theoretical analysis of parameter-efficient policy adaptation in the MARL framework and establish the relationship between restricted parameter updates and full-parameter policy optimization. **4) Experimental evaluation:** We evaluate SWORD on heterogeneous satellite cluster Earth observation missions with different reward formulations and compare

[1]Codes are made available at: `https://anonymous.4open.science/r/iclr27_pa-DF18/README.md`

its performance and adaptation efficiency with full-parameter fine-tuning and existing parameter-efficient adaptation methods.

The remainder of this paper is organized and separated into several sections. Section II reviews related work on MARL for satellite systems and parameter-efficient policy adaptation. Section III presents the satellite mission formulation and the MARL problem. Section IV introduces the proposed SWORD adaptation method and its theoretical formulation. Section V describes the experimental results and evaluations. Finally, Section VI concludes the paper and discusses potential directions for future research.

2 Related Works

MARL Policy Transfer and Adaptation. In recent years, transformer-based architectures have been increasingly explored for multi-agent reinforcement learning (MARL) due to their ability to decouple observation and action spaces, model dependencies and interactions among agents. Universal Policy Decoupling Transformer (UPDeT) was introduced for MARL in StarCraft Multi-Agent Challenge (SMAC) environments by representing agent observations using transformer-based architectures Hu et al. (2021b). Similarly, the Decision Transformer framework has been investigated for offline multi-agent reinforcement learning by learning policies from previously collected trajectories (Meng et al., 2023), while the Multi-Agent Transformer (MAT) formulates MARL as an autoregressive sequence modeling problem to capture interactions among agents (Wen et al., 2022). The transformer-based approaches have also been applied to spacecraft Earth observation missions, including an observation transformer encoder for strip-imaging tasks (Cheval & Schaub, 2026) and the Heterogeneous-Agent Differential Transformer (HADT) for heterogeneous satellite cluster coordination (Hady et al., 2026). Beyond transformer-based policy learning, policy reuse and adaptation have been investigated to reduce the cost of learning new tasks; for example, (Fu et al., 2026) proposed a policy reuse framework consisting of offline and online adaptation phases. However, these approaches primarily focus on transfer learning and reusing policies across tasks, while efficient adaptation of MARL policies to changes in mission objectives and reward functions remains less explored, particularly in heterogeneous satellite Earth observation scenarios.

Parameter Efficient Fine-tuning. Parameter-efficient fine-tuning (PEFT) has also recently been explored in MARL, including approaches that employ low-rank adaptation to enable agent-specific policy adaptation (Zhang et al., 2025). This methods primarily focus on adapting individual agents or policy components, while the adaptation of cooperative MARL policy to mission reward changes remains under explored. In addition, the Sparse fine-tuning methods update only a selected subset of model parameters, enabling efficient adaptation while preserving most of the pre-trained knowledge (Chen et al., 2021). Recently, sparse parameter updates have also been investigated, where they can improve the plasticity of pre-trained models when adapting to new tasks (Zhang et al., 2026). This approach uses Fisher information matrix to select the most relevant parameter to the current gradient updates. However, it focuses only during adaptation or fine-tuning phase. An alternative approach is low-rank adaptation, such as LoRA, which represents smaller set of parameter updates using low-rank matrices and thereby it can reduce the number of trainable parameters (Hu et al., 2021a). Building upon this principle, decomposed variants such as DoRA (Liu et al., 2024) and ADALORA (Zhang et al., 2023) further separate the magnitude and directional components of weight updates to improve adaptation performance. Despite these advances, existing decomposition-based methods typically determine the decomposition during initialization and subsequently fine-tune the selected ranks. This motivates the proposed **SWORD** framework, which combines the advantages of sparse and low-rank parameter updates through Singular Value Decomposition (SVD) for efficient policy adaptation. SWORD uses SVD to extract the low-rank structure of the parameters learned from the source task, while Fisher Information is used to identify and selectively retain the parameters that are most important for adaptation. This two components enables SWORD to preserve important knowledge from the previously learned source task while efficiently updating task-relevant parameters for the new target task.

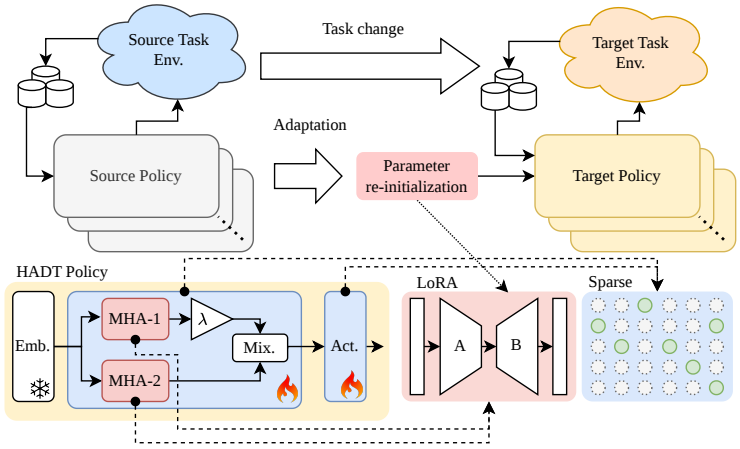

Figure 2: SWORD framework is constructed from low-rank adaptation with Singular Vector Decomposition (SVD) and sparse tuning. Some of pre-trained weights of HADT policy are keep in freeze, and we select some other weights to be adapted for a target task.

3 Policy Adaptation Under SWORD Framework

3.1 Transformer-based policy

We consider a general decentralized Partially Observable Markov Decision Process (Dec-POMDP), in which multiple heterogeneous satellite agents cooperatively perform an Earth observation mission under partial observability and limited onboard resources. In this setting, each satellite makes decisions based on its local observation, while the overall objective is to maximize the cooperative mission return. Let the heterogeneous satellite cluster be represented by a Dec-POMDP defined as: $\mathcal{M} = \langle \mathcal{N}, \mathcal{S}, \mathcal{O}_i, \mathcal{A}_i, P, R, \gamma \rangle$, where $\mathcal{N} = 1, \ldots, N$ denotes the set of heterogeneous satellite agents, $\mathcal{S}$ is the global state space, $\mathcal{O}_i$ and $\mathcal{A}_i$ denote the local observation and action spaces of satellite agent-i, respectively, P represents the state transition function, R is the cooperative reward function, and $\gamma \in [0, 1)$ is the discount factor. Due to the decentralized nature of satellite operation, agent-i does not directly observe the complete global state and instead receives a local observation $o_{i,t} \in \mathcal{O}_i$ at time step t.

The objective of MARL is to learn a joint policy $\boldsymbol{\pi} = \pi_{\theta_i}(a_i|o_i)$, parameterized by $\boldsymbol{\theta} = \theta_i$, that maximizes the expected discounted return:

$$J(\boldsymbol{\theta}; R) = \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}, P}\left[\sum_{t=0}^{T} \gamma^t R(s_t, \boldsymbol{a}_t)\right], \tag{1}$$

where $\boldsymbol{a}_t = (a_{1,t}, \ldots, a_{N,t})$ denotes the joint action of the satellite cluster.

In this work, we use HADT (Hady et al., 2026) as it is one of the transformer-based policy applied in the satellite cluster domain and achieve a good performance. It is optimised under the HAPPO (Zhong et al., 2024) update, ϕ is *not* advanced by a single joint gradient step. At each iteration a permutation $i_{1:n}$ is drawn and agent i_j maximises:

$$L_{i_j}^{\text{HAPPO}}(\phi^{i_j}) = \mathbf{E}_t\Big[\min\big(\varrho_{i_j,t}(\phi^{i_j})\hat{A}_t^{i_j},\ \text{clip}(\varrho_{i_j,t}(\phi^{i_j}), 1-\epsilon, 1+\epsilon)\hat{A}_t^{i_j}\big)\Big], \tag{2}$$

with $\varrho_{i_j,t} = \pi_{\phi^{i_j}}(a_t^{i_j} \mid o_t^{i_j})/\pi_{\phi^{i_j}}^{\text{old}}(a_t^{i_j} \mid o_t^{i_j})$ and $\hat{A}_t^{i_j}$ conditioned on the already-updated policies of $i_1, \ldots, i_{j-1}$ through the advantage decomposition $\hat{A}_{i_{1:m},t} = \sum_{j=1}^{m} \hat{A}_{i_j}(o_t, a_{i_{1:j-1},t}, a_{i_j})$.

In EO mission, the reward function is determined by the mission objectives and therefore may vary between missions denoted by $R^{(s)}$ and $R^{(t)}$ for the source and target tasks, respectively. Although the state space, observation space, action space, and transition dynamics may remain unchanged, the change in the optimization objective can result in different optimal policies: $\boldsymbol{\pi}^{*(s)} =$

$\arg\max_{\boldsymbol{\pi}} J(\boldsymbol{\pi}; R^{(s)})$ and $\boldsymbol{\pi}^{*(t)} = \arg\max_{\boldsymbol{\pi}} J(\boldsymbol{\pi}; R^{(t)})$. Therefore, a policy optimized for the source task is not necessarily optimal for the target task. Directly transferring $\boldsymbol{\pi}^{*(s)}$ to the target mission may lead to a performance degradation because the learned policy encodes the preferences induced by the source reward function. This motivates the policy adaptation problem considered in this work, where a pre-trained source policy is efficiently adapted to a target reward function without requiring complete retraining from random initialization.

3.2 SWORD Adaptation Algorithm

SWORD methods restricting the set of parameters (or regularization) that can be updated during adaptation as shown in Figure 2. SWORD consists of two major components: A Singular Vector Decomposition LoRA inspired by (Zhang et al., 2023) and Sparse tuning based on Fisher information importance (Zhang et al., 2026) detailed as follow:

1) Fisher-guided sparse tuning identifies the parameters that are most important for the source policy and selectively updates only a small subset of them during fine-tuning. Let $\boldsymbol{\theta} \in \mathbb{R}^d$ denotes the pre-trained policy parameters and $\mathcal{D}$ denote the trajectory data collected from the target task. The importance of parameter θ_j is estimated using the Fisher Information Matrix (FIM), given by:

$$\mathcal{I}_j(\boldsymbol{\theta}) = \mathbb{E} * \tau \sim \mathcal{D}\left[\left(\frac{\partial \log \pi * \boldsymbol{\theta}(a_t|o_t)}{\partial \theta_j}\right)^2\right], \tag{3}$$

where $\pi_{\boldsymbol{\theta}}$ denotes the policy and τ represents a sampled trajectory. A binary importance mask $\mathbf{m} \in 0, 1^d$ is then constructed by selecting the parameters with the largest Fisher information values:

$$m_j = \begin{cases} 1, & \mathcal{I}_j \in \mathrm{topK}(\mathcal{I}, K), \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where K denotes the number of parameters selected for adaptation. The sparse-tuned parameters are consequently updated according to: $\boldsymbol{\theta}' = \boldsymbol{\theta} + \mathbf{m} \odot \Delta\boldsymbol{\theta}$, where $\odot$ denotes element-wise multiplication. Thus, parameters with high estimated importance are prioritized for adaptation, while the remaining parameters are preserved from the pre-trained policy.

2) LoRA with Singular Value Decomposition (LoRA-SVD) to represent policy adaptation through a low-rank parameter update. For a pre-trained weight matrix $\mathbf{W} * 0 \in \mathbb{R}^{d*\mathrm{out} \times d_{\mathrm{in}}}$, LoRA keeps $\mathbf{W}_0$ frozen and represents the task-specific update as:

$$\mathbf{W}' = \mathbf{W}_0 + \Delta\mathbf{W}, \qquad \Delta\mathbf{W} = \alpha\mathbf{B}\mathbf{A}, \tag{5}$$

where $\mathbf{A} \in \mathbb{R}^{r \times d_{\mathrm{in}}}$ and $\mathbf{B} \in \mathbb{R}^{d_{\mathrm{out}} \times r}$ are trainable low-rank matrices, $r \ll \min(d_{\mathrm{in}}, d_{\mathrm{out}})$ is the adaptation rank, and α is a scaling factor. In SWORD, Singular Value Decomposition is used to obtain a compact representation of the adaptation directions. Given a weight or update matrix $\mathbf{M}$, its decomposition is $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ contain the left and right singular vectors and $\boldsymbol{\Sigma}$ contains the corresponding singular values. The rank-r approximation is obtained by retaining the dominant singular components: $\mathbf{M}_r = \mathbf{U}_r\boldsymbol{\Sigma}_r\mathbf{V} * r^\top$, thereby preserving the principal adaptation directions while reducing the number of trainable parameters.

Additionally, a complete algorithm flow is provided in a Pseudocode form (see Appendix A.9.) for more detail of our proposed method.

3.3 SWORD Theoretical Analysis

SWORD consists of two principle components which have different structural constraints. However, both can be formulated as restricted policy-gradient optimization derived from Fu et al. (2023). Under the SWORD framework, the source policy parameters $\boldsymbol{\theta}^{(s)}$ are used as the initialization for the target task, and the adaptation is performed through a restricted parameter update.

Let $J(\theta)$ be a differentiable RL objective and let $g(\theta) = \nabla_\theta J(\theta)$ denote the policy gradient. Let $\mathcal{U} \subseteq \mathbb{R}^d$ denote a feasible parameter-update space induced by a parameter-efficient tuning method, and let $P_{\mathcal{U}}$ be the orthogonal projection onto $\mathcal{U}$. For a sufficiently small learning rate $\alpha > 0$, the locally optimal update within $\mathcal{U}$ is: $\Delta\theta_{\mathcal{U}} = \alpha P_{\mathcal{U}} g$.

Proposition 3.1 (Restricted Policy-Gradient Update)**.** The corresponding first-order improvement of the objective function is:

$$J(\theta + \Delta\theta_{\mathcal{U}}) - J(\theta) = \alpha \left\| P_{\mathcal{U}} g \right\|_2^2 + O(\alpha^2). \tag{6}$$

Proof of Proposition 3.1 can be found in Appendix A.5.

Consider a parameter or weight matrix $W \in \mathbb{R}^{m \times n}$. For sparse tuning, let $\mathcal{S}$ denote the set of trainable weight indices. The corresponding update space is: $\mathcal{U}_S = \{\Delta W : \Delta W_{ij} = 0, \quad (i,j) \notin \mathcal{S}\}$ for LoRA with rank r, the adapted weight as in Equation 5.

Proposition 3.2 (LoRA and Sparse Tuning as Restricted Updates)**.** The corresponding local LoRA update space is the tangent space:

$$\mathcal{U}_L = \left\{ \Delta B A + B \Delta A : \Delta B \in \mathbb{R}^{m \times r},\ \Delta A \in \mathbb{R}^{r \times n} \right\}. \tag{7}$$

Consequently, both sparse tuning and LoRA can be formulated as restricted policy-gradient updates:

$$\Delta W_S = \alpha P_S \left(\nabla_W J \right), \qquad \Delta W_L = \alpha P_L \left(\nabla_W J \right), \tag{8}$$

where P_S and P_L denote the corresponding projection operators.

Proof of Proposition 3.2 can be found in Appendix A.6.

Let $J(\theta)$ be continuously differentiable in a neighbourhood of θ, with policy gradient $g = \nabla_\theta J(\theta)$ and Hessian $H = \nabla_\theta^2 J(\theta)$. Let P_L and P_S denote the effective local projection operators associated with LoRA and sparse tuning, respectively. Define the corresponding restricted policy-gradient updates as: $\Delta\theta_L = \alpha P_L g$, and $\Delta\theta_S = \alpha P_S g$.

Theorem 1 (Local Performance Equivalence of LoRA and Sparse Tuning)**.** *Suppose that* $\left| \|P_L g\|_2^2 - \|P_S g\|_2^2 \right| \le \epsilon_g$ *and* $\left| g^\top P_L H P_L g - g^\top P_S H P_S g \right| \le \epsilon_H$*. Then, for sufficiently small* α*, yields boundary range:*

$$|\Delta J_L - \Delta J_S| \le \alpha \epsilon_g + \frac{\alpha^2}{2} \epsilon_H + O(\alpha^3), \tag{9}$$

In Theorem 1, $\Delta J_L = J(\theta + \Delta\theta_L) - J(\theta)$ and $\Delta J_S = J(\theta + \Delta\theta_S) - J(\theta)$ are defined. Consequently, if $\epsilon_g \to 0, \epsilon_H \to 0, \alpha \to 0$, then $\Delta J_L - \Delta J_S \to 0$. Thus, LoRA and sparse tuning achieve asymptotically equivalent local improvements of the RL objective when their projected policy-gradient magnitudes and projected curvature terms are equivalent. Proof of Theorem 1 is provided in the Appendix A.7. This theorem provides a foundation for SWORD (Sparse and loW-rank Decomposition) by showing that LoRA and sparse tuning can achieve comparable local objective improvements when their projected policy-gradient and curvature terms are sufficiently similar. This is important because SWORD combines low-rank adaptation with sparse parameter selection rather than relying exclusively on either strategy. The theorem therefore supports the underlying assumption that restricting adaptation to a carefully selected low-dimensional parameter space does not necessarily sacrifice RL performance. In particular, it motivates SWORD to identify parameters that preserve important policy-gradient information while representing the adaptation using a low-rank structure, enabling parameter-efficient task adaptation with limited performance degradation.

4 Experimental setup

Our experiments performed using Intel(R) Core(TM) i9-14900K CPU, 32 GB RAM, and 24 GB GPU NVIDIA GeForce RTX 4090 machine. To evaluate the adaptability of SWORD to changes in mission objectives, four task configurations are defined by varying the relative importance of target priority and cloud coverage in the reward function. The simulation is build using BSK-RL (Stephenson & Schaub, 2024) and Basilisk simulator (Kenneally et al., 2020) (detailed explanation provided in the Appendix A.1 and A.2. The mission specific payload reward at time t is formulated as:

$$R_t = \begin{cases} q_i - \eta_t + c_t, & \text{if captured} \\ -\eta_t + \delta_t, & \text{if downlinked} \\ -100, & \text{if failure} \\ -\eta_t, & \text{if power is used} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

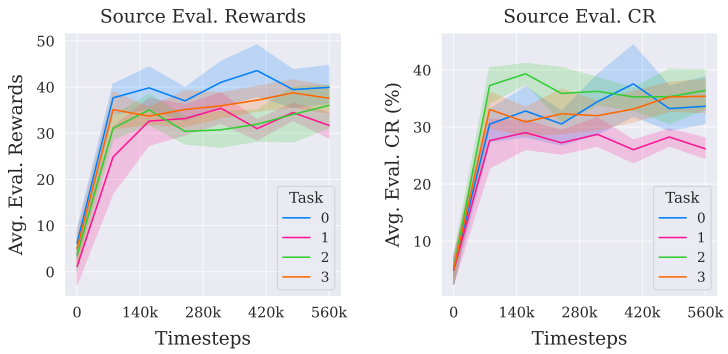

Figure 3: Source tasks evaluation performance using HADT policy.

where, q_i is capturing reward, η_t is power usage reward, δ_t is downlinked data reward (more information is provided in Appendix A.3. We specifically design the capturing reward q_i for a different types of tasks that consists of several components: $q_{i,t} = \sum_{i=1}^{N} \left(\omega_1 \rho_{i,t} + \omega_2 \sigma_{i,t}\right) / \tau_t$, where N denotes the number of satellites, $\rho_{i,t}$ represents the priority of the target captured by satellite i at time t, $\sigma_{i,t}$ denotes the cloud-coverage-related reward or cost associated with the captured target, and τ_t represents the total number of captured targets at time t. The coefficients ω_1 and ω_2 determine the relative contributions of target priority and cloud coverage, respectively as summarized in Table 1. Based on these coefficients, four tasks are constructed to represent different mission preferences.

Table 1: Definition of the four task configurations based on the relative weighting of target priority and cloud coverage.

Task Id.	**Mission Objective**	$\boldsymbol{\omega_1}$	$\boldsymbol{\omega_2}$
Task 0	Target priority only	1.0	0.0
Task 1	Target priority $>$ Cloud coverage	0.8	0.2
Task 2	Target priority $=$ Cloud coverage	0.5	0.5
Task 3	Target priority $<$ Cloud coverage	0.2	0.8

5 Experimental Results

5.1 Source-Task Policy

We train HADT policy under four different tasks as in Figures 3 in terms of completion rate (CR) and Rewards. The evaluation results demonstrate that the pre-trained policy can achieve meaningful performance across different task configurations, although the performance varies according to the underlying mission objective. This variation is expected because each task is associated with a different reward formulation, resulting in different optimization objectives for the learned policy. Overall, these results establish the capability of HADT in the source tasks to be used for different tasks policy adaptation experiments.

5.2 Different-Task Policy Adaptation

The results in Table 2 demonstrate the effectiveness of the proposed SWORD strategy for adapting the pre-trained HADT policy across different task changes. Overall, SWORD achieves an average completion rate very close to Full Fine-Tuning, with only a 0.54 percentage-point difference. This result is particularly notable considering that SWORD updates only 16.3% of the total model parameters, whereas Full Fine-Tuning updates all 44,226 parameters (Hyper-parameters are provided in the Appendix A.2).

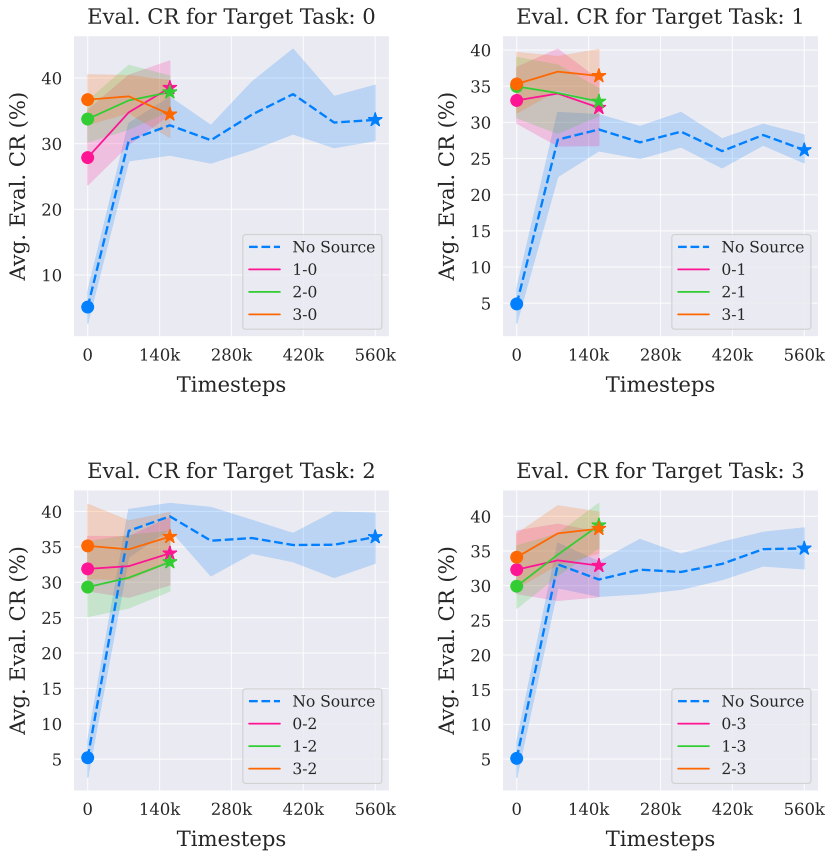

Figure 4: SWORD adaptation completion rate in different source and target tasks.

The comparison with the baselines highlights the benefit of our proposed adaptation approach. LoRA-FFT (Gao et al., 2024) and LoRA-SVD (Zhang et al., 2023) achieve average completion rates of 32.90% and 32.35%, respectively, while Sparse Fine-Tuning (Zhang et al., 2026) achieves 33.45%. In contrast, SWORD increases the average completion rate to 35.45%, representing improvements of 2.55, 3.10, and 2.00 percentage points over LoRA-FFT, LoRA-SVD, and Sparse Fine-Tuning, respectively. Interestingly, this performance is also comparable to the strategy that freezes the embedding parameters, which achieves 35.56% while requiring substantially more trainable parameters (89.4%). These results suggest that the combination of low-rank updates and sparse parameter selection can exploit complementary adaptation mechanisms, allowing the policy to modify task-relevant knowledge without requiring extensive updates to the entire network.

The consistency of SWORD can also be observed across different target tasks. For Target 1 and Target 2, SWORD achieves average completion rates of 33.78% and 34.46%, respectively, slightly exceeding Full Fine-Tuning at 33.51% and 34.29%. For Target 0 and Target 3, its performance remains close to Full Fine-Tuning, with differences of 1.36 and 1.23 percentage points, respectively. This indicates that the effectiveness of SWORD is not limited to a particular transfer direction and that the proposed adaptation mechanism can maintain a relatively stable performance under different reward-function changes. his provides empirical support for the proposed approach as a practical solution for efficient task adaptation in heterogeneous satellite MARL.

5.3 The Impact of SWORD

To investigate the contribution of the individual components of SWORD, an ablation study is conducted by comparing the complete SWORD method with two variants: **SWORD w/o SVD**, which

Table 2: Completion Rate (CR) percentage after 160K training steps and parameter efficiency across different adaptation task pairs (Source – Target).

Task Pair	Full FT	w/o Emb. FT	LoRA FFT	LoRA SVD	Sparse FT	SWORD (ours)
1–0	40.00 ± 6.90	36.58 ± 3.63	26.70 ± 2.87	26.67 ± 3.24	36.21 ± 6.17	38.52 ± 4.82
2–0	38.56 ± 3.18	37.94 ± 2.50	34.73 ± 4.17	32.31 ± 2.26	38.37 ± 2.32	37.84 ± 3.15
3–0	36.34 ± 4.10	35.98 ± 2.93	37.09 ± 1.93	36.17 ± 6.71	30.00 ± 4.71	34.47 ± 4.95
Target 0 Avg.	**38.30**	36.83	32.84	31.72	34.86	**36.94**
0–1	30.98 ± 8.38	33.07 ± 7.36	33.41 ± 5.99	34.05 ± 5.23	30.25 ± 6.23	32.03 ± 5.25
2–1	32.86 ± 6.49	31.43 ± 3.86	33.30 ± 3.60	31.55 ± 2.20	31.67 ± 2.86	32.88 ± 2.20
3–1	36.70 ± 2.02	36.48 ± 5.57	36.63 ± 2.68	34.41 ± 6.39	31.67 ± 4.02	36.42 ± 4.00
Target 1 Avg.	**33.51**	33.66	34.45	33.34	31.20	**33.78**
0–2	33.64 ± 5.91	35.87 ± 8.23	32.05 ± 3.25	34.70 ± 6.03	30.72 ± 2.78	34.09 ± 5.26
1–2	31.63 ± 3.42	31.47 ± 3.16	27.65 ± 1.95	27.83 ± 2.23	33.51 ± 5.47	32.86 ± 5.03
3–2	37.61 ± 5.73	35.49 ± 4.49	36.78 ± 2.13	35.72 ± 6.32	32.92 ± 5.04	36.44 ± 4.29
Target 2 Avg.	**34.29**	34.28	32.16	32.75	32.38	**34.46**
0–3	36.36 ± 5.65	36.94 ± 6.99	32.12 ± 3.33	34.81 ± 4.99	29.47 ± 2.56	32.89 ± 4.97
1–3	38.79 ± 6.18	36.29 ± 3.24	27.59 ± 3.33	27.08 ± 1.13	37.12 ± 2.48	38.71 ± 3.89
2–3	38.37 ± 2.69	39.18 ± 4.08	36.74 ± 4.87	32.91 ± 1.90	39.51 ± 5.13	38.22 ± 3.42
Target 3 Avg.	**37.84**	**37.47**	32.15	31.60	35.37	**36.61**
% Param.	100.0%	89.4%	81.3%	81.3%	20.0%	16.3%

Table 3: Ablation study of SWORD and its individual components.

Method	Avg. CR (%)
SWORD (LoRA-Sparse)	35.45
SWORD w/o SVD	35.11
SWORD w/o Fisher	32.93

removes the SVD-based low-rank decomposition and become standard LoRA, and **SWORD w/o Fisher**, which replaces Fisher-information-guided parameter selection with random parameter selection. The comparison is designed to examine whether the performance of SWORD results from the joint use of low-rank adaptation and informed sparse parameter selection rather than parameter sparsity alone. In particular, the SWORD w/o SVD variant evaluates the contribution of the low-rank representation, while the SWORD w/o Fisher variant evaluates whether selecting parameters according to their estimated importance provides an advantage over randomly selecting the same number of parameters. The completion rate (CR) after fine-tuning drops 0.34% without SVD and 2.52% without Fisher information. This results provide evidence regarding the individual and complementary contributions of SWORD main components.

6 Conclusion

Our results demonstrate that the proposed SWORD algorithm provides an effective and parameter-efficient solution for adapting pre-trained MARL policies to different satellite mission objectives. SWORD achieves an average completion rate close to Full Fine-Tuning, while updating only 16.3% of the total model parameters. This indicates that the combination of low-rank adaptation and sparse parameter updates can substantially reduce the fine-tuning cost while preserving adaptation performance across different source–target task pairs. These findings further support the potential of parameter-efficient fine-tuning for heterogeneous satellite cluster missions, where frequent task changes may require rapid policy adaptation. Future work will investigate the scalability of SWORD to larger and more heterogeneous satellite clusters, more diverse mission objectives, and continuous or dynamically changing reward functions. In addition, future research will explore adaptive mechanisms for automatically identifying the most task-relevant parameters and low-rank components during fine-tuning, as well as extending the framework to online and continual multi-task adaptation under realistic orbital uncertainties and communication constraints.

AI USE STATEMENT

In this work, we used generative AI tools for proof reading, shaping language and writing style. We have not used generative AI tools for generating ideas in this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide comprehensive materials to facilitate the reproduction of the proposed SWORD framework and experimental results. The main paper describes the problem formulation, task definitions, SWORD architecture, and training and evaluation methodology, while the appendix provides the complete mathematical formulation, assumptions, and proofs underlying the theoretical analysis of sparse and low-rank adaptation. The implementation details, including the model architecture, fine-tuning configurations, hyper-parameters, and experimental settings, are provided in the supplementary material. We further provide the source code for SWORD and the baseline fine-tuning methods, together with the scripts required for training and evaluation, as anonymous supplementary materials `https://anonymous.4open.science/r/iclr27_pa-DF18`. The supplementary material also documents the satellite-cluster environment, task and reward definitions, data generation and preprocessing procedures, and evaluation protocols used to obtain the reported results. All experimental comparisons are conducted using the same environment enabling the reported results to be reproduced.

REFERENCES

Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in neural information processing systems*, 34:19974–19988, 2021.

Anais Cheval and Hanspeter Schaub. Reinforcement learning with hybrid action representation for autonomous strip imaging task scheduling in super-agile satellites. In *AIAA SCITECH 2026 Forum*, pp. 0132, 2026.

Ke Fu, Hao Chen, Yu Wang, Quan Liu, and Jian Huang. Efficient multi-agent policy adaptation with bayesian policy reuse and view-invariant contrastive awareness. *Neural Networks*, pp. 108755, 2026.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 12799–12807, 2023.

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. *arXiv preprint arXiv:2405.03003*, 2024.

Mohamad A Hady, Siyi Hu, Mahardhika Pratama, Zehong Cao, and Ryszard Kowalczyk. Multi-agent reinforcement learning for resources allocation optimization: a survey. *Artificial Intelligence Review*, 58(11):354, 2025.

Mohamad A Hady, Muhammad Anwar Masum, Siyi Hu, Mahardhika Pratama, Jimmy Cao, and Ryszard Kowalczyk. Hadt: A heterogeneous multi-agent differential transformer for autonomous earth observation satellite cluster. *arXiv preprint arXiv:2605.31023*, 2026.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Adam Herrmann and Hanspeter Schaub. Reinforcement learning for the agile earth-observing satellite scheduling problem. *IEEE Transactions on Aerospace and Electronic Systems*, 59(5):5235–5247, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021a.

Siyi Hu, Fengda Zhu, Xiaojun Chang, and Xiaodan Liang. Updet: Universal multi-agent reinforcement learning via policy decoupling with transformers. *arXiv preprint arXiv:2101.08001*, 2021b.

Siyi Hu, Mohamad A Hady, Jianglin Qiao, Jimmy Cao, Mahardhika Pratama, and Ryszard Kowalczyk. Adaptability in multi-agent reinforcement learning: A framework and unified review. *arXiv preprint arXiv:2507.10142*, 2025.

Patrick W Kenneally, Scott Piggott, and Hanspeter Schaub. Basilisk: A flexible, scalable and modular astrodynamics simulation framework. *Journal of aerospace information systems*, 17(9):496–507, 2020.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024.

Linghui Meng, Muning Wen, Chenyang Le, Xiyun Li, Dengpeng Xing, Weinan Zhang, Ying Wen, Haifeng Zhang, Jun Wang, Yaodong Yang, et al. Offline pre-trained multi-agent decision transformer. *Machine Intelligence Research*, 20(2):233–248, 2023.

Rongjun Qin, Feng Chen, Tonghan Wang, Lei Yuan, Xiaoran Wu, Yipeng Kang, Zongzhang Zhang, Chongjie Zhang, and Yang Yu. Multi-agent policy transfer via task relationship modeling. *Science China Information Sciences*, 67(8):182101, 2024.

Mark Stephenson, Lorenzzo Mantovani, and Hanspeter Schaub. Intent sharing for emergent collaboration in autonomous earth observing constellations. In *AAS Guidance and Control Conference, Breckenridge, CO*, pp. 24–192, 2024.

Mark A Stephenson and Hanspeter Schaub. Bsk-rl: Modular, high-fidelity reinforcement learning environments for spacecraft tasking. In *75th International Astronautical Congress, Milan, Italy, IAF*, pp. 1186–1197, 2024.

Zhao Wang, Zheng Zhang, Xiaojun Shan, Hong-an Wei, and Ping Tang. Generative models for sar–optical image translation: A systematic review. *International Journal of Applied Earth Observation and Geoinformation*, 146:105009, 2026.

Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35:16509–16521, 2022.

Beining Zhang, Aditya Kapoor, and Mingfei Sun. Low-rank agent-specific adaptation (lorasa) for multi-agent policy learning. *arXiv preprint arXiv:2502.05573*, 2025.

Huan Zhang, Shenghua Fan, Shuyu Dong, Yujin Zheng, Dingwen Wang, and Fan Lyu. Sparse tuning enhances plasticity in ptm-based continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28230–28238, 2026.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *JMLR*, 25:1–67, 2024.

Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11): 13344–13362, 2023.

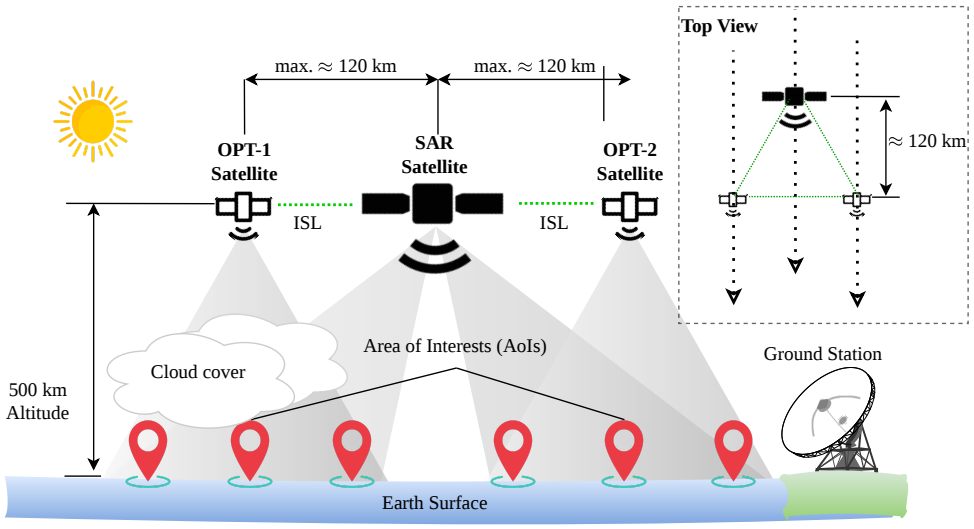

Figure 5: Heterogeneous Satellite Cluster Configuration.

A Appendix

A.1 Heterogeneous Satellite Cluster Using Basilisk Simulator Environment

The multi-agent reinforcement learning framework for heterogeneous satellite tasking is built on **Basilisk** (Kenneally et al., 2020), an open-source, modular astrodynamics simulation framework developed by the Autonomous Vehicle Systems (AVS) Laboratory at the University of Colorado Boulder. Basilisk provides high-fidelity, physics-based modeling of spacecraft dynamics, including orbital propagation, attitude determination and control, power generation and consumption, on-board data storage, thermal effects, and sensor/actuator behavior. Its modular C/C++ architecture, exposed through a Python interface, allows individual spacecraft subsystems—reaction wheels, batteries, data buffers, communication payloads, and imaging or synthetic aperture radar (SAR) sensors—to be composed independently, which makes it well suited for representing a *heterogeneous* satellite cluster in which agents differ in sensing modality, power budget, and data-handling capacity.

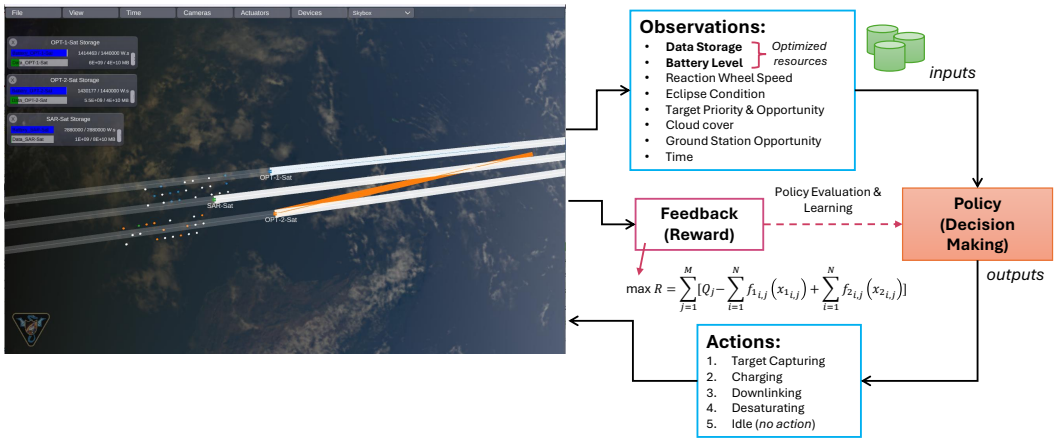

Figure 6: BSK-RL as MARL enviroment.

BSK-RL (Basilisk Reinforcement Learning) (Stephenson & Schaub, 2024) extends this simulator into a Gymnasium-compatible environment specifically designed for satellite tasking problems. It wraps the underlying Basilisk dynamics in a Markov Decision Process (or, in the multi-agent case, a Decentralized Partially Observable Markov Decision Process, Dec-POMDP) interface, exposing per-satellite observations, discrete or continuous action spaces, and configurable reward functions while Basilisk continues to propagate the true physical state in the background (see Figure 6. This separation allows the RL policy to reason over abstracted, mission-relevant quantities—data storage level, battery state of charge, reaction wheel speed, eclipse condition, target priority, cloud cover, and ground station visibility—without needing to directly solve the underlying orbital and attitude dynamics at each decision step.

In the heterogeneous cluster setting as illustrated in Figure 5 the OPT-1-Sat, OPT-2-Sat, and SAR-Sat configuration, each agent i maintains its own local observation vector o_i drawn from this shared set of state features, reflecting differences in sensor type (optical imaging versus SAR) and on-board resource constraints. At each timestep, BSK-RL evaluates a discrete action set common to all agents—*target capturing*, *charging*, *downlinking*, *desaturating* (reaction wheel momentum dumping), and *idle*—and forwards the selected action to the corresponding Basilisk subsystem models, which update the true spacecraft state accordingly.

A.2 Parameters for Simulation Environment

Several satellite parameters to define the satellite specifications are adjustable from the BSK-RL's environment definition as shown in Table 4.

Table 4: Satellite Default Parameters

Parameter Name	Optical	SAR	Unit
Data Storage and Transmission:			
Data Storage Capacity	5	10	GB
Initial Data Storage Level	0	0	%
Instrument Baud Rate	500	1000	Mbps
Transmitter Baud Rate	100	100	Mbps
Power System:			
Battery Capacity	400	800	W.h
Initial Battery Level	100	100	%
Solar Panel Area	1	2	m^2
Solar Panel Efficiency	20	20	%
Base Power Draw	-10	-10	W
Instrument Power Draw	-30	-120	W
Transmitter Power Draw	-25	-25	W
Thruster Power Draw	-80	-100	W
Satellite Attitude:			
Image Attitude Error	0.1	0.1	Degree
Image Rate Error	0.1	0.1	Degree
Attitude Control Input Max.	0.4	0.4	-
Attitude Rate Control Input Max.	0.1	0.1	-
Servo P constant	30	30	-
Servo Ki constant	5	5	-
Disturbance Vector	[0, 0, 0]	[0, 0, 0]	-
Max. Reaction Wheel speed	6000	6000	RPM
Initial Reaction Wheel speed	[0, 0, 0]	[0, 0, 0]	RPM

The orbital parameters in BSK-RL used for our experiments are shown in Table 5.

A.3 Reward Function

The reward function R_t as defined in Equation 10 governing agent behaviour at each timestep t is defined piecewise, reflecting the distinct outcomes an agent can encounter when tasking an Area of Interest (AoI):

Table 5: Orbital Parameters for 1 Cluster

Parameter	Values	Unit
Num. of Satellites	3	satellite
Inclination	[39, 41, 40]	degree
Offsets	[-74, -74, -75]	degree
Num. of Planes	3	plane
Altitude	500	km

Table 6: Hyperparameter settings used for SWORD training.

Parameter	Value
Training Settings:	
Number of rollout threads	20
Total environment steps	560,000
Episode length	400
Log interval	5
Evaluation interval	10
Data chunk length	10
Optimizer Settings:	
Actor learning rate	5×10^{-4}
Critic learning rate	5×10^{-4}
Adam ϵ	1×10^{-5}
Weight decay	0
std_x coefficient	1
std_y coefficient	0.5
PPO Settings:	
Actor epochs	5
Critic epochs	5
Clipping parameter	0.2
Actor mini-batches	1
Critic mini-batches	1
Entropy coefficient	0.01
Value loss coefficient	1
Maximum gradient norm	10.0
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
Huber loss δ	10.0
LoRA and Sparse Settings:	
LoRA rank	4
Sparsity	0.8

Data Acquisition. The dominant term when an AoI is successfully captured is $q_i \in (0, 1)$, the priority weight assigned to that AoI. This directly encodes the objective of maximising the capture of high-priority targets: an agent that repeatedly captures low-value AoIs accrues less reward than one that prioritises scarce, high-value targets, even at equal capture counts.

Resource Utilisation. Three complementary terms shape how efficiently the heterogeneous satellite agent uses its onboard resources:

i) Power usage is penalised through

$$\eta_t = \gamma \, \Delta Q_t \, (1 - Q_t), \qquad \Delta Q_t = Q_{t-1} - Q_t, \tag{11}$$

where Q_t denotes the remaining battery/power state and ΔQ_t the power consumed between consecutive steps. The $(1 - Q_t)$ factor scales the penalty upward as the battery depletes, so identical

power draws are penalised more heavily when reserves are already low, discouraging the agent from exhausting its power margin.

ii) Data downlinking is rewarded via:

$$\delta_t = \beta\,\Delta D_t, \qquad \Delta D_t = D_t - D_{t-1}, \tag{12}$$

where D_t is cumulative data collected, so ΔD_t measures the incremental data downlinked at step t. This term appreciates the agent to offload collected data promptly rather than accumulate it without transmission, which may cause out of memory condition.

iii) Payload selection is rewarded according to whether the chosen sensing modality matches cloud coverage $\sigma \in (0, 1)$ over the AoI:

$$c_t = \begin{cases} -1+\sigma, & \sigma < 50\%,\ \text{payload} = \text{SAR} \\ \sigma, & \sigma > 50\%,\ \text{payload} = \text{SAR} \\ 1-\sigma, & \sigma < 50\%,\ \text{payload} = \text{OPT} \\ -\sigma, & \sigma > 50\%,\ \text{payload} = \text{OPT} \end{cases} \tag{13}$$

Since SAR can penetrate cloud cover while optical (OPT) sensing cannot, this structure rewards choosing SAR under heavy cloud ($\sigma > 50\%$) and OPT under clear skies ($\sigma < 50\%$), and penalises the mismatched choice in each regime — encouraging the agent to condition payload selection on environmental state rather than using a fixed sensor.

Failure Avoidance. A fixed penalty of -100 is applied whenever a resource constraint is violated (e.g. reaction wheel saturation or battery falling below the 80% safety threshold). This term dominates the reward landscape by design, ensuring the policy strongly avoids unsafe operating states regardless of the acquisition or efficiency gains otherwise on offer.

Together, these terms balance four competing objectives that prioritise target capture, power efficiency, timely data downlink, and payload usage while enforcing hard safety constraints through the failure penalty.

A.4 Task Difference

The t-SNE visualization of the policy embeddings provides a qualitative representation of the relationships among policies trained under the four different reward configurations (see Figure 7. Since the underlying satellite environment and decision-making problem remain unchanged, the primary source of variation between the policies is the relative weighting of target priority and cloud coverage in the reward function. As defined in Table 1, Task 0 emphasizes target priority exclusively (100:0), Task 1 assigns greater importance to priority while incorporating cloud coverage (80:20), Task 2 balances the two objectives equally (50:50), and Task 3 places greater emphasis on cloud coverage (20:80). Therefore, the position and distribution of each task's policy embeddings can be interpreted as an indication of how changes in the reward objective influence the learned policy representation.

The t-SNE plot shows that the four task policies occupy distinguishable regions of the embedding space, indicating that the learned representations capture task-dependent characteristics. In particular, the gradual change in reward weighting from Task 0 toward Task 3 represents a continuous shift in the mission objective, from strongly prioritizing high-priority targets to increasingly favouring targets with lower cloud coverage. Consequently, policies associated with adjacent task definitions are expected to exhibit greater similarity than policies trained under more substantially different reward configurations. This relationship is important for the proposed task-adaptation setting because it suggests that the policy representation contains information related to the underlying reward objective rather than solely representing the common satellite environment.

The embedding visualization also provides motivation for parameter-efficient fine-tuning. Although the four tasks share the same environment dynamics, observation space, and action space, their policies are differentiated by the mission-specific reward preference. Thus, adapting a source policy to a target task does not necessarily require relearning the complete policy representation. Instead, the adaptation can be viewed as modifying a relatively small portion of the learned parameters

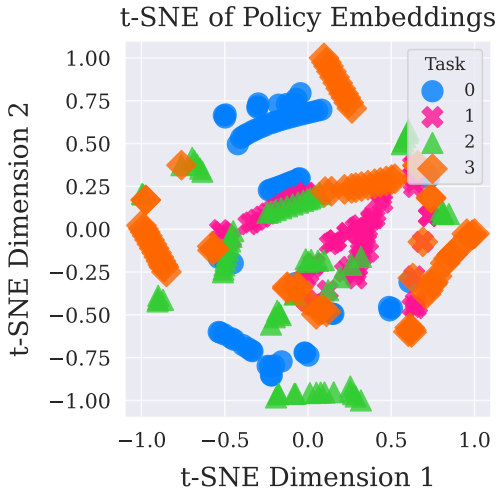

Figure 7: Embedding space of the trained Policy in Four Different Tasks.

to shift the policy toward the target task's objective. In this context, the task-dependent structure observed in the t-SNE embedding supports the use of SWORD, which combines sparse parameter selection with low-rank adaptation to efficiently modify an existing policy while preserving the task-general knowledge learned from the source task.

To sum up, the t-SNE analysis suggests that changing the reward weighting produces measurable changes in the learned policy representation. The four tasks therefore provide a controlled experimental setting for studying policy adaptability, where task differences arise from a systematic modification of the mission objective rather than from changes to the environment or action space. This makes the task set suitable for evaluating whether SWORD can efficiently move a source policy toward the representation required by a target reward configuration while avoiding the computational cost of full policy fine-tuning.

A.5 Proof of Proposition 3.1

Proof. Using the first-order Taylor expansion of $J(\theta)$ around θ, we have

$$J(\theta + \Delta\theta) = J(\theta) + g^\top \Delta\theta + O(\|\Delta\theta\|_2^2). \tag{14}$$

Since the update is restricted to $\mathcal{U}$, the first-order objective improvement is maximized by an update in the direction of the projected gradient. Thus,

$$\Delta\theta_{\mathcal{U}} = \alpha P_{\mathcal{U}} g. \tag{15}$$

Substituting this update into the Taylor expansion gives

$$J(\theta + \Delta\theta_{\mathcal{U}}) - J(\theta) = \alpha g^\top P_{\mathcal{U}} g + O(\alpha^2). \tag{16}$$

Because $P_{\mathcal{U}}$ is an orthogonal projection,

$$P_{\mathcal{U}}^\top = P_{\mathcal{U}}, \qquad P_{\mathcal{U}}^2 = P_{\mathcal{U}}, \tag{17}$$

and therefore

$$g^\top P_{\mathcal{U}} g = \|P_{\mathcal{U}} g\|_2^2 . \tag{18}$$

Consequently,

$$J(\theta + \Delta\theta_{\mathcal{U}}) - J(\theta) = \alpha \, \|P_{\mathcal{U}} g\|_2^2 + O(\alpha^2). \tag{19}$$

□

A.6 Proof of Proposition 3.2

Proof. For sparse tuning, only the parameters indexed by $\mathcal{S}$ are allowed to change. Hence, every feasible update belongs to $\mathcal{U}_S$. Applying Proposition 3.1 gives

$$\Delta W_S = \alpha P_S \left(\nabla_W J\right). \tag{20}$$

For LoRA, the adapted weight matrix is

$$W' = W + BA. \tag{21}$$

Consider a small perturbation of the LoRA parameters:

$$B \to B + \Delta B, \qquad A \to A + \Delta A. \tag{22}$$

The resulting change in the weight matrix is

$$\Delta W = (B + \Delta B)(A + \Delta A) - BA \tag{23}$$

$$= \Delta BA + B\Delta A + \Delta B\Delta A. \tag{24}$$

For sufficiently small perturbations, the second-order term $\Delta B\Delta A$ can be neglected. Therefore, the first-order LoRA update belongs to the tangent space

$$\mathcal{U}_L = \{\Delta BA + B\Delta A\}. \tag{25}$$

Applying Proposition 3.1 to this local update space gives

$$\Delta W_L = \alpha P_L \left(\nabla_W J\right). \tag{26}$$

Thus, although LoRA and sparse tuning impose different structural constraints, both can be interpreted as restricted policy-gradient optimization. □

A.7 Proof of Theorem 1

Proof. Using the second-order Taylor expansion of J around θ,

$$J(\theta + \Delta\theta) = J(\theta) + g^\top \Delta\theta + \frac{1}{2}\Delta\theta^\top H \Delta\theta + O(\|\Delta\theta\|_2^3). \tag{27}$$

For LoRA, substituting

$$\Delta\theta_L = \alpha P_L g \tag{28}$$

gives

$$\Delta J_L = \alpha g^\top P_L g + \frac{\alpha^2}{2} g^\top P_L^\top H P_L g + O(\alpha^3). \tag{29}$$

Since P_L is an orthogonal projection, $P_L^\top = P_L$, and therefore

$$g^\top P_L g = \|P_L g\|_2^2. \tag{30}$$

Hence,

$$\Delta J_L = \alpha \|P_L g\|_2^2 + \frac{\alpha^2}{2} g^\top P_L H P_L g + O(\alpha^3). \tag{31}$$

Similarly, for sparse tuning,

$$\Delta\theta_S = \alpha P_S g, \tag{32}$$

which gives

$$\Delta J_S = \alpha \|P_S g\|_2^2 + \frac{\alpha^2}{2} g^\top P_S H P_S g + O(\alpha^3). \tag{33}$$

Subtracting the two expressions gives

$$\Delta J_L - \Delta J_S = \alpha \left(\|P_L g\|_2^2 - \|P_S g\|_2^2 \right) + \frac{\alpha^2}{2} \left(g^\top P_L H P_L g - g^\top P_S H P_S g \right) + O(\alpha^3). \tag{34}$$

Taking the absolute value and applying the assumptions,

$$|\Delta J_L - \Delta J_S| \leq \alpha \epsilon_g + \frac{\alpha^2}{2} \epsilon_H + O(\alpha^3). \tag{35}$$

As

$$\epsilon_g, \epsilon_H, \alpha \to 0, \tag{36}$$

the right-hand side converges to zero. Therefore,

$$\Delta J_L - \Delta J_S \to 0. \tag{37}$$

This proves the local performance equivalence of LoRA and sparse tuning under the stated conditions. □

A.8 Training time comparison

The Table 7 shows that the differences in the number of trainable parameters are relatively modest among all methods, while Sparse Fine-Tuning and SWORD reduce the number of trainable parameters. Importantly, however, all methods operate with a relatively small parameter budget, with even Full Fine-Tuning updating only 44,226 parameters. Thus, the parameter-count difference alone does not indicate a major computational distinction in this relatively small HADT model. Notably, SWORD updates only 7,194 parameters, corresponding to approximately 16.3% of the parameters updated by Full Fine-Tuning and about 18.1% of those updated by Sparse Fine-Tuning. Despite this reduction, SWORD achieves an average training time of 0.47 ± 0.05 hours, which is comparable to Full Fine-Tuning (0.47 ± 0.06 hours) and slightly lower than Sparse Fine-Tuning (0.49 ± 0.08 hours). The LoRA variants update 35,970 parameters and exhibit somewhat higher training times (0.61 hours), suggesting that the number of trainable parameters does not directly translate into proportional training-time savings in this relatively small network. Overall, the results indicate that the primary advantage of SWORD is not a dramatic reduction in absolute training time, but rather achieving comparable adaptation with substantially fewer trainable parameters, which becomes more meaningful as the underlying policy network and deployment scenario scale.

Table 7: Number of trainable parameters for each fine-tuning method.

Algorithm	**Trainable Parameters**	**Avg. Training Time (hours)**
Full Fine-Tuning	44,226	0.47 ± 0.06
w/o Embedding Fine-Tuning	39,554	0.59 ± 0.08
LoRA-FFT	35,970	0.61 ± 0.12
LoRA-SVD	35,970	0.61 ± 0.1
Sparse Fine-Tuning	8,845	0.49 ± 0.08
SWORD	**7,194**	$\mathbf{0.47 \pm 0.05}$

Algorithm 1 SVD-Initialized LoRA with Fisher-Guided Sparse Update

Require: Pretrained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, rank r, sparsity budget s (fraction of parameters), learning rate η, Fisher estimation batch $\mathcal{B}_F$, training data $\mathcal{D}$
Ensure: Fine-tuned weight $W = W_0 + \Delta W$

// Step 1: SVD-based LoRA Initialization
Compute $W_0 = U\Sigma V^\top$ ▷ Singular Vector Decomposition (SVD)
$U_r \leftarrow U[:, :r]$, $\Sigma_r \leftarrow \Sigma[:r, :r]$, $V_r \leftarrow V[:, :r]$
Initialize $A \leftarrow \Sigma_r^{1/2} V_r^\top \in \mathbb{R}^{r\times k}$
Initialize $B \leftarrow U_r \Sigma_r^{1/2} \in \mathbb{R}^{d\times r}$ ▷ $BA \approx$ top-r component of W_0, so $\Delta W = BA$ starts aligned with W_0's dominant subspace
for training step $t = 1, 2, \ldots, T$ **do**
  Sample batch $b_t \sim \mathcal{D}$
  Compute HAPPO loss $\mathcal{L}(b_t; W_0 + \Delta W_t)$ (as in Equation 2.)
  Compute $g^{(b)} \leftarrow \nabla_{\Delta W}\mathcal{L}(b; W_0 + BA)$
  // Step 2: Fisher Information Estimation
  $F \leftarrow \|\mathbf{g^{(b)}}\|_2$ for all W
  // Step 3: Determine Sparse Update Mask
  Flatten F and select the top-$s \cdot (d \cdot k)$ entries by magnitude
  Construct binary mask $M \in \{0,1\}^{d\times k}$: $M_{ij} = 1$ if F_{ij} among top-s entries, else 0 ▷ mask updates on parameters most sensitive to the loss
  $\Delta W_t \leftarrow BA + M \odot S$
  $g_A \leftarrow \nabla_A \mathcal{L}, \quad g_B \leftarrow \nabla_B \mathcal{L}, \quad g_S \leftarrow \nabla_S \mathcal{L}$
  $A \leftarrow A - \eta\, g_A$
  $B \leftarrow B - \eta\, g_B$
  $S \leftarrow S - \eta\,(M \odot g_S)$ ▷ gradient masked so only selected entries update
end for

A.9 Pseudocode of SWORD

Pseudocode is provided as Algorithm 1.