Appendices

Concurrent Constrained Optimization of Unknown Rewards for Heterogeneous Task Allocation

APPENDIX A
EXPERIMENTAL EVALUATION DATA

1. *Numerical Simulations*

We simulated a task-allocation problem with:

- M = 3 (three tasks)
- S = 4 (four species)
- U = 3 (three traits)

In order to simulate the ground truth GP reward functions, $f_m \sim \mathcal{N}(\mu_m, \sigma_m)$ we used the following hand-crafted mean (μ) and standard deviation(σ) matrices:

$$\mu_{(M\ X\ U)} = \begin{bmatrix} 158.675 & 1816.802 & 168.409 \\ 177.846 & 2221.625 & 212.906 \\ 183.533 & 2092.250 & 193.702 \end{bmatrix}$$

$$\sigma_{(M\ X\ U)} = \begin{bmatrix} 2.491e+03 & 2.576e+05 & 2.275e+03 \\ 2.440e+03 & 2.331e+05 & 1.989e+03 \\ 2.559e+03 & 2.649e+05 & 2.366e+03 \end{bmatrix}$$

rewards were then computed as:

$$r_m = (f_m * 2 * 10^9) + \epsilon_m \ \forall m \in \{1, 2, \cdots, M\} \quad (1)$$

where the scaling factor of $2 * 10^9$ is used merely to bring the reward values into an easily-interpretable scale, and ϵ captures the effects of inherent stochasticity in multi-robot tasks in the form of noise. Equation (1) is used only to evaluate a given assignment post-hoc. We otherwise assume no access to these ground truth reward functions.

We generated different teams by uniformly randomly sampling the number of agents of each species, N_s, between 1 and 10. The species-trait matrix, Q, for these teams is uniformly randomly sampled by using a mean and standard deviation for each trait. We use [28.50, 384.04, 36.15] for the mean and [16.454, 43.149, 7.198] for the standard deviation. For the experiments, we chose six teams from this set of randomly generated teams so that our data has a good mixture of the maximum sum of rewards these teams can yield.

For the teams used to present and analyze results in the paper, the team compositions and species-trait matrices are mentioned below. The teams are arranged in increasing order of their competence. Their competence is measured as the maximum ground truth reward that an allocation X from the given team can obtain,i.e.

$$r_{max} = \max_X \sum_{m=1}^{M} r_m \quad (2)$$

where r_m is computed as defined in Equation(1) of the paper.

1) **Team 1**

Team composition				
Species	A	B	C	D
No. of robots	7	3	5	4

Species-Trait matrix			
Species	Trait 1	Trait 2	Trait 3
A	28.0826	363.1913	51.1745
B	11.2092	434.2397	32.4508
C	16.6207	378.5524	31.4128
D	15.7114	437.0867	31.5072

2) **Team 2**

Team composition				
Species	A	B	C	D
No. of robots	7	5	5	5

Species-Trait matrix			
Species	Trait 1	Trait 2	Trait 3
A	21.91620	334.3167	26.3267
B	0.3228	420.5752	28.6841
C	19.7711	411.8564	33.54407
D	29.6150	318.1833	38.4799

3) **Team 3**

Team composition				
Species	A	B	C	D
No. of robots	8	5	2	8

Species-Trait matrix			
Species	Trait 1	Trait 2	Trait 3
A	35.6443	360.2783	40.6767
B	42.3553	379.7987	44.6118
C	13.4724	385.0184	35.1182
D	26.7633	341.3894	40.0623

4) **Team 4**

Team composition				
Species	A	B	C	D
No. of robots	5	5	6	8

Species-Trait matrix			
Species	Trait 1	Trait 2	Trait 3
A	25.3149	419.1320	37.2515
B	20.3573	452.9782	30.6338
C	14.7630	388.0228	37.0236
D	71.1834	340.6735	45.7495

5) **Team 5**

Team composition				
Species	A	B	C	D
No. of robots	10	3	3	5

Species-Trait matrix			
Species	Trait 1	Trait 2	Trait 3
A	28.0132	401.1998	30.4497
B	32.2405	398.9287	34.2543
C	8.9981	276.0692	44.8429
D	36.6740	404.8579	30.7181

6) **Team 6**

Team composition				
Species	A	B	C	D
No. of robots	4	5	5	5

Species-Trait matrix			
Species	Trait 1	Trait 2	Trait 3
A	10.3987	449.1855	42.6299
B	57.0252	362.2864	31.9146
C	44.9961	427.1479	44.5305
D	38.4274	502.6917	40.5852

We ran each baseline for 400 iterations for the above six teams.The results of these simulations have been presented in the paper in detail.

2. *Robotarium simulations for Emergency Response Tasks*
We developed an emergency response scenario in the Robotarium simulator with:

- M = 3 (three tasks)
  1) Fire fighting : A coalition of robots carries water from one location to douse fire at another location.
  2) Debris-removal : A coalition of robots carries pieces of debris to a drop-off location.
  3) Coverage-control : A coalition of robots covers an environment by sensing the area around comprising individual robots.
- S = 4 (four species) : [Species A, Species B, Species C, Species D]
- U = 4 (four traits)
  1) Speed
  2) Water-carrying capacity
  3) Payload capacity
  4) Sensing radius

We simulated the tasks with varied requirements in terms of:

- Units of fire to be doused
- Units of debris to be cleared
- Area to be covered
- Location of fire and debris

For the task set-up referred to in the paper, we had the following requirements:

1) Task 1: Douse 45 units of fire
2) Task 2: Clear 45 units of debris
3) Task 3: Cover a rectangular area of length=3.2 units and width=2 units

The rewards were calculated as a fraction of the task requirement that was fulfilled. For example, if an allocation of robots was able to douse 20 out of 45 units of fire, the reward was 20/45 = 0.444 for the fire-fighting task.

We generated multiple teams where each species had between 3-6 robots with non-varying traits between them. The team that is discussed in the paper had the following composition and species-trait matrix:

Team composition				
Species	A	B	C	D
No. of robots	4	4	4	3

Species-Trait matrix				
Species	Speed	Water-carrying capacity	Payload Capacity	Sensing radius
A	0.1	4	1	0.18
B	0.1	1	5	0.12
C	0.1	1	2	0.54
D	0.2	2	2	0.24

APPENDIX B
BOOTSTRAPPING WITH OFFLINE DATA

In the addition to the experiments presented in Sections VI A and VI B of the paper, we conducted experiments with historical data (i.e. *demonstrations*) to bootstrap the learning process in CMTAB as well as the other baselines. In this appendix, we present the results and evaluation of these experiments.

1. *Numerical Simulations*
We consider the same experimental design described in Section VI A of the paper and in Section A of Appendix I. For each of the six teams, we generate a set of $N_D = 70$ passive demonstrations by uniformly randomly allocations and collecting the corresponding rewards. We then used the N_D demonstrations to compute a prior on the GPs before using each method for online optimization. We repeat the experiment for five rounds for each of the six teams over $N = 400$ iterations.

As can be seen from Fig. 1, CMTAB consistently and efficiently learns to allocate agents better than the best-performing

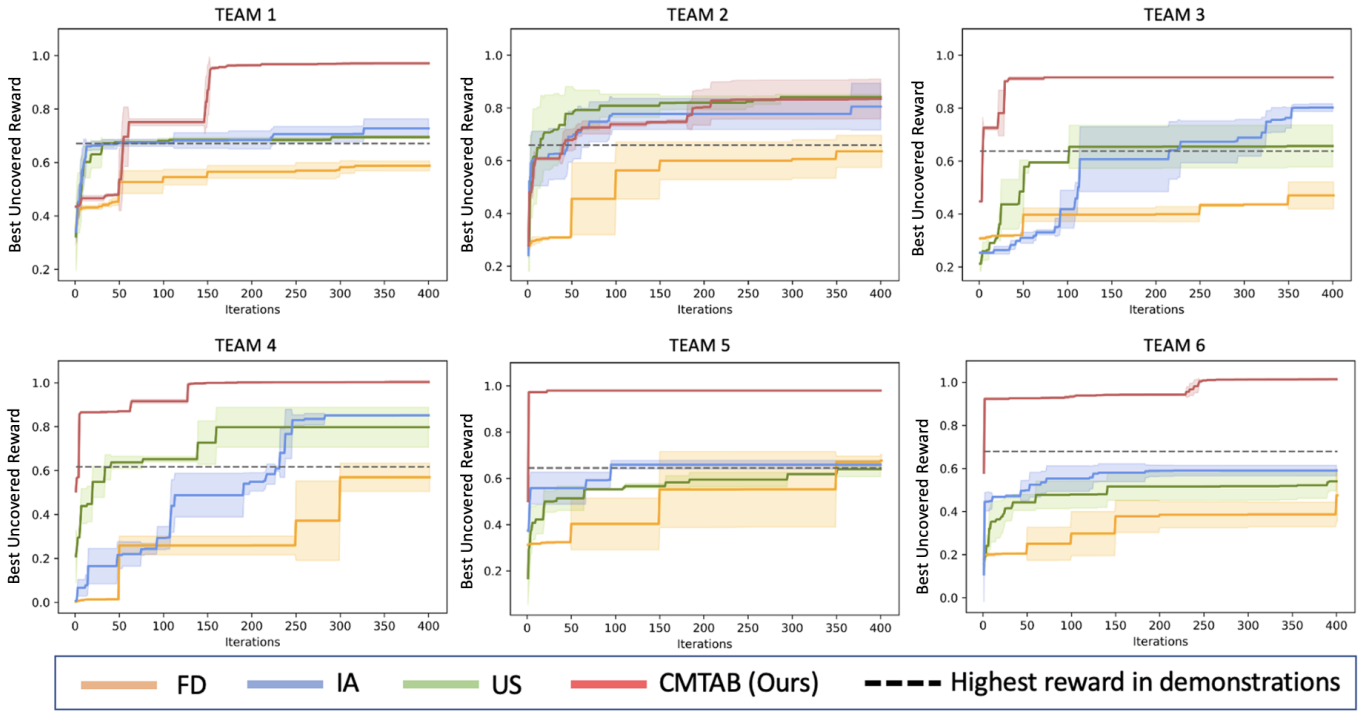

Fig. 1. Best uncovered reward as a function of interactions with the environment when the GPs are initialized using offline demonstrations.

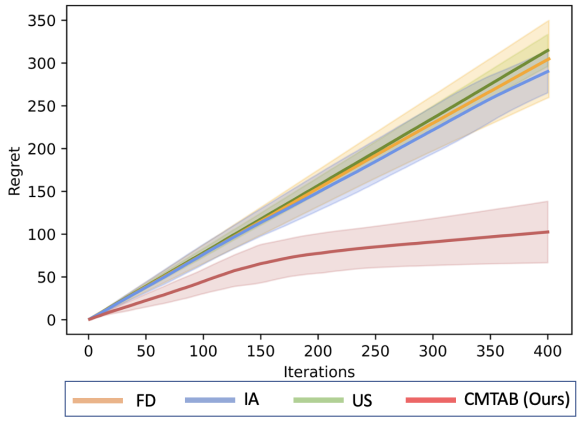

Fig. 2. Multi-task regret averaged over all teams for each baseline when the reward functions are initialized using initial demonstrations

demonstrations, irrespective of the underlying team. Comparing the team-wise BURs in Fig. 2 in the paper and Fig. 1 in this appendix, we see that FD benefits from the offline demonstrations across all teams. The US baseline is unable to leverage offline demonstrations since it engages in pure exploration (i.e., no exploitation) which can result in high rewards only by getting "lucky" when sampling allocations. Such a baseline would evidently not benefit from offline demonstrations. CMTAB and IA exhibit similar trend for Teams 4, 5, and 6 (an average increase of 5.67%). However, this trend does not hold for Teams 1, 2 and 3. This is likely due to the fact that Teams 1, 2, and 3 are of lower competencies and thus were more likely to produce low-scoring demonstrations which might have caused CMTAB and IA to be stuck exploiting a local minimum. From Fig. (2), we once again find that CMTAB comfortably outperformed the baselines in terms of CMR, suggesting that CMTAB accumulates the least regret even when initialized with passive demonstrations.

2. *Robotarium simulations for Emergency Response Tasks*
We setup the emergency response task scenario as described in Section VI B of the paper and in Section B of Appendix I. For the set of experiments presented in this section, we first generated $N_D = 70$ random demonstrations in the simulator to use these as an input for the baselines to learn a prior of the reward functions. We then ran the experiments for five rounds and 400 iterations for every baseline.

In Figs. (3) and (4), we plot the BUR and CMR for each method when the GPs were initialized with a prior learned from demonstrations. Echoing the results from numerical simulations, we see that all methods except the US baseline have been able to leverage the prior from historical data. In stark contrast, the FD baseline results in a 872.5% improvement in its highest BUR compared to when there is no historical data. However, it reaches a suboptimal steady state (normalized BUR = 0.778) in less than 100 iterations and continues to exploit it instead of exploring to improve the rewards. When there is no historical data, CMTAB exhibits increased *exploration* in this initial iterations as evidenced by the increasing BUR. In contrast, such an exploration phase is

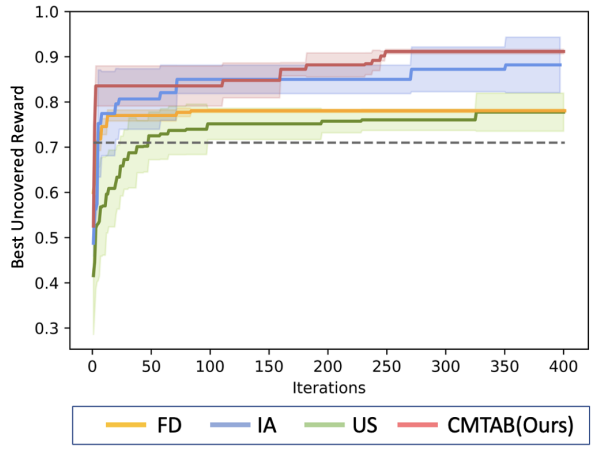

Fig. 3. Best uncovered reward over iterations in the simulated emergency response environment

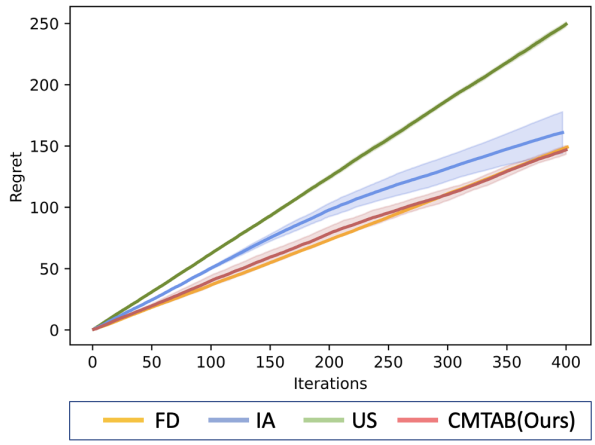

Fig. 4. Cumulative multi-task regret over iterations in the simulated emergency response environment

not seen in Fig. (3) when CMTAB has been able to leverage the demonstrations to identify regions of high-reward. In the further iterations, CMTAB continues to balance exploration and exploitation to achieve rewards higher than those seen in the demonstrations. Since IA selects selects the allocations for each task individually, it is not always able to identify high-reward region. As such, IA considerably more iterations to catch up with the CMTAB.

An interesting observation from the CMR plots in Fig. (4), is the similarity between the regret accumulated by FD and CMTAB. In fact, FD has a slightly lower regret than CMTAB up to 300 iterations This improved performance is likely due to the fact that FD is able to exploit the high-reward region exposed by the demonstrations. Further, CMTAB's selection of under-explored regions in the initial iterations can yield a lower reward, thus adding to the regret. As iterations progress however, CMTAB's tendency to select high-reward allocations increases, damping the accumulation of regret.

APPENDIX C
ROBOTARIUM SIMULATIONS FOR ADDITIONAL TEAMS

In this appendix, we present the additional experiments we conducted in the simulated emergency response scenario for two teams of robots (besides the one mentioned in the paper). The task setup ($M = 3$), species ($S = 4$) and species-trait matrices of the teams were the same as mentioned in section 2 of Appendix A. The composition of the teams were as follows:

1) **Team 1**

Team composition				
Species	A	B	C	D
No. of robots	3	3	3	3

2) **Team 2**

Team composition				
Species	A	B	C	D
No. of robots	5	4	6	3

In Figure (5), we plot the metrics BUR (Best Uncovered Reward) and CMR (Cumulative Multi-task Regret) for both the teams when the reward function GPs were initialised with zero mean. CMTAB continues to be the best-performing algorithm for these teams as well. From the BUR plot for Team 2, we observe that IA performs marginally similar to CMTAB highlighting the advantage of adaptive discretization. However, the accumulated cost of suboptimal coalitions continues to gradually increase for IA unlike CMTAB at the end of 400 iterations.

APPENDIX D
GENERALIZING TRAIT-REWARD MAPS TO NEW TEAM

We use the traits of robots to model the requirements of tasks, enabling the approach to be generalizable to new teams of robots. We conducted a set of experiments using numerical simulations to corroborate this design choice. In this appendix, we present the associated results.

We used the numerically simulated task-allocation problem, team compositions and species-trait matrices as mentioned in Appendix A. In the first instance of the experiment, we started with Team 2 and changed the team to a more competent Team 6 at the 250^{th} iteration. In the second case, we started with Team 6 and changed it to Team 2 after 250 iterations. Figure (6) shows the BUR and CMR plotted for these experiments. As we can see from the plots, the new teams of robots in both cases were able to form high-rewarding coalitions in lesser than 100 iterations without accumulating additional regret in the process (CMR for CMTAB continues to be the least). The other baselines show similar trends as they model trait-reward maps too, however they take more iterations to catch up to CMTAB due to the lack of one or more of its properties.

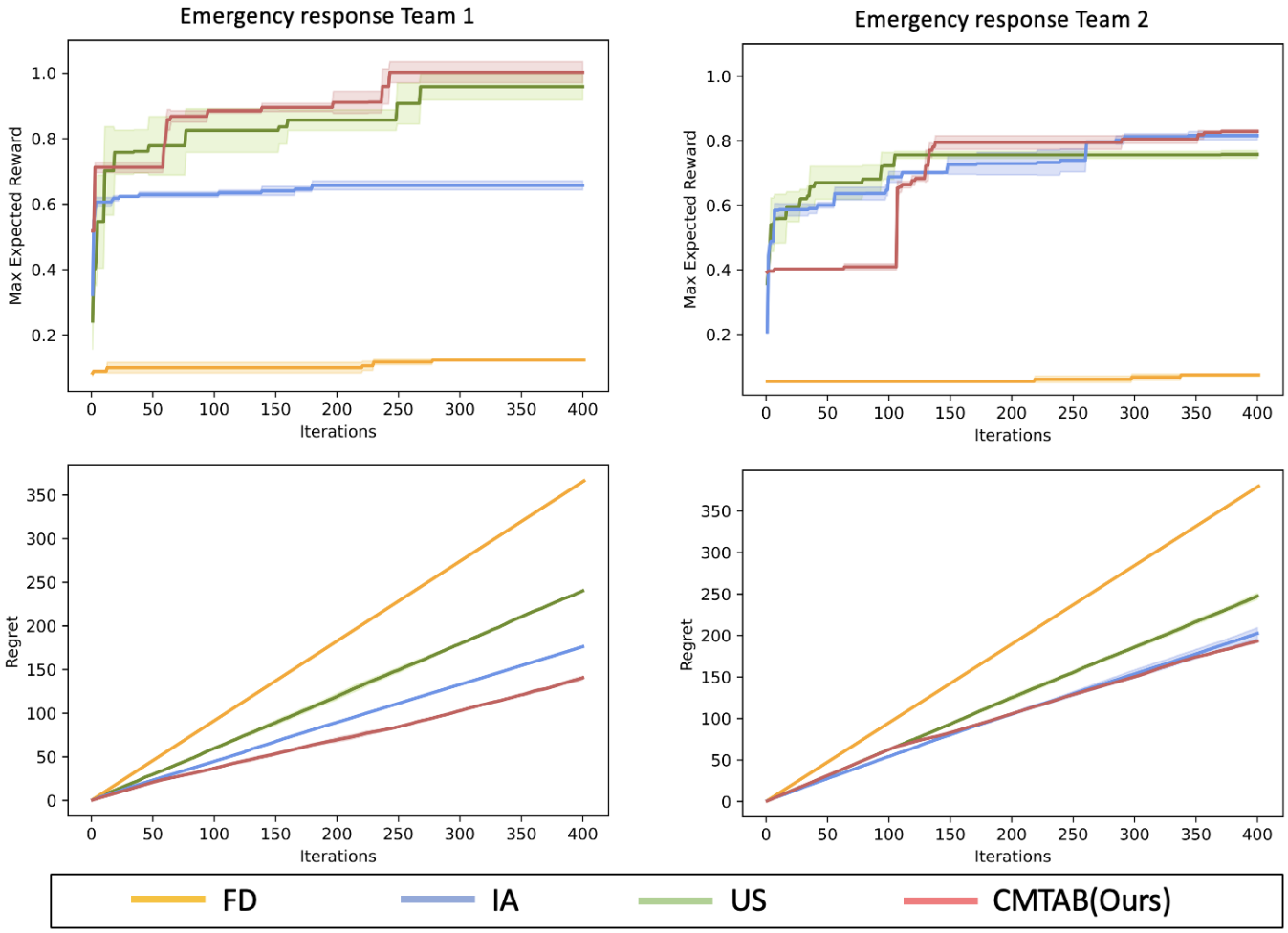

Fig. 5. Best uncovered reward (top) and cumulative multi-task regret (bottom) as a function of iterations in the simulated emergency response environment, when the GPs are initialized with zero mean..

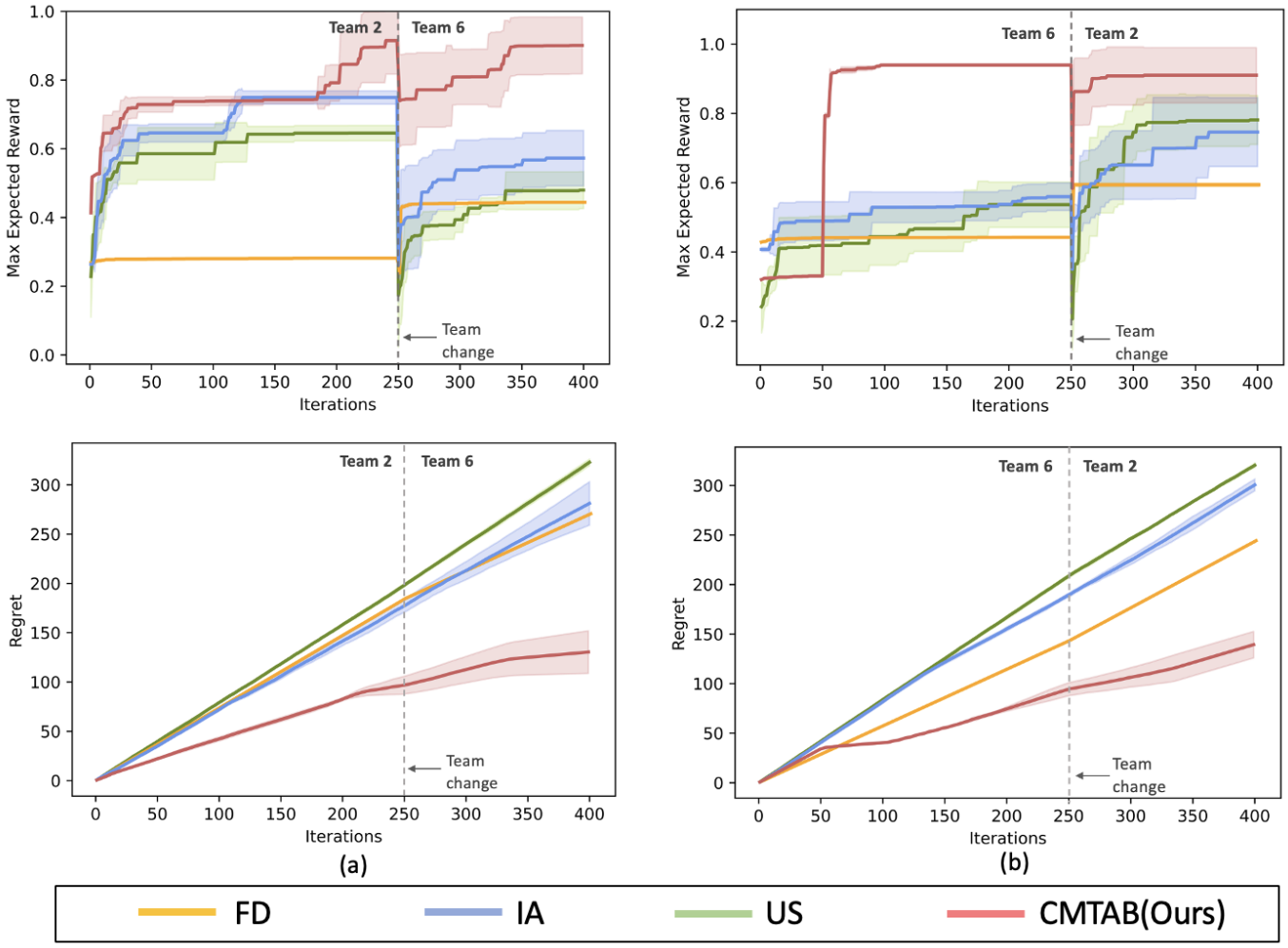

Fig. 6. Best uncovered reward (top) and cumulative multi-task regret (bottom) as a function of iterations, when the team was changed at the 250^{th} iteration.