Appendix: Coordinated Multi-Agent Exploration using Shared Goals

In the following we provide implementation details (Sec. A) and additional training curves (Sec. C).

A Implementation Details

***Pass*, *Secret-room*, and *Push-box* environments:** We combine CMAE with plain tabular Q-learning (Sutton & Barto, 2018). Note, we don't apply any existing tabular exploration techniques, such as optimistic initialization. The Q-table is initialized to zero. The update step size for exploration policies and target policies are 0.1 and 0.05 respectively. The replay buffer size is 50,000. The bonus b for reaching a goal is 1, and the discount factor γ is 0.95.

***Island* environment:** We combine CMAE with DQN (Mnih et al., 2013; 2015). The Q-function is parameterized by a three-layer perceptron (MLP) with 64 hidden units per layer and ReLU activation function. We use 32 parallel environments to collect data. We implement CMAE in Pytorch (Paszke et al., 2017). The optimizer is Adam (Kingma & Ba, 2015), and the learning rate for both exploration policies and target policies are 10^{-4}. The size of the replay memory is $50,000$, the batch size is 1024, the bonus for reaching goals is 10, and the discount factor γ is 0.95. The target network is four updates behind the latest network.

SMAC environment: We combine CMAE with the official code for QMIX (Rashid et al., 2018). Following their default setting, for both exploration and target policies, the agent is a two-layer MLP with 64 hidden units per layer and ReLU non-linearity. The mix network is eight units. The discounter factor γ is 0.99. The replay memory stores the latest 500 episodes, and the batch size is 32. RMSProp is used with a learning rate of 5×10^{-4}. The target network is updated every 100 episodes.

B Comparison to QMIX with RND

Following the reviewers' suggestion, we run QMIX Rashid et al. (2018) with RND Burda et al. (2019) on the SMAC 8m-sparse task. Both our approach and QMIX + RND are trained for 2M environment steps. Our approach achieves $80.1\% \pm 1.3\%$ win rate while QMIX + RND achieves only $1.5\% \pm 0.4\%$ win rate. The training curves are shown in Fig. 7.

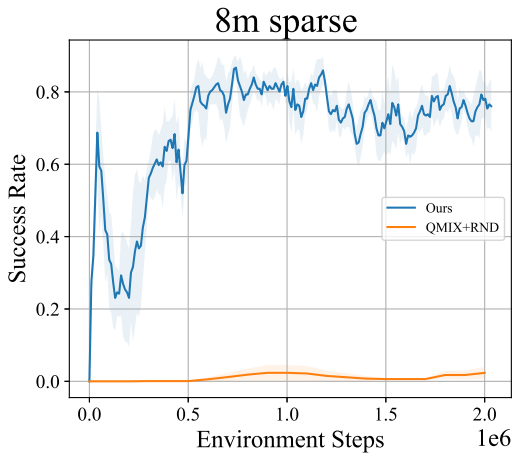

Figure 7: Results of our approach and QMIX+RND on *8m-sparse* task.

C TRAINING CURVES

We provide training curves that summarize test episode return, number of eliminated enemies, and number of dead allies for our CMAE, and baselines QMIX, VDN, MAVEN, QTRAN on *3m-sparse* (Fig. 8), *8m-sparse* (Fig. 9), *3m-dense* (Fig. 10), and *8m-dense* (Fig. 11).

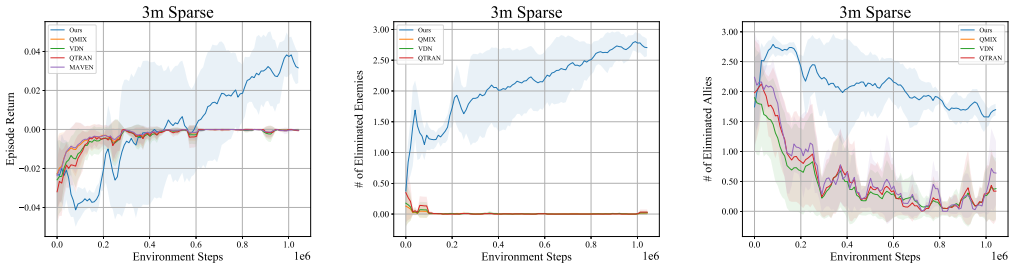

Figure 8: Episode return, number of eliminated enemies, and number of eliminated allies of CMAE and baselines on *3m-sparse*.

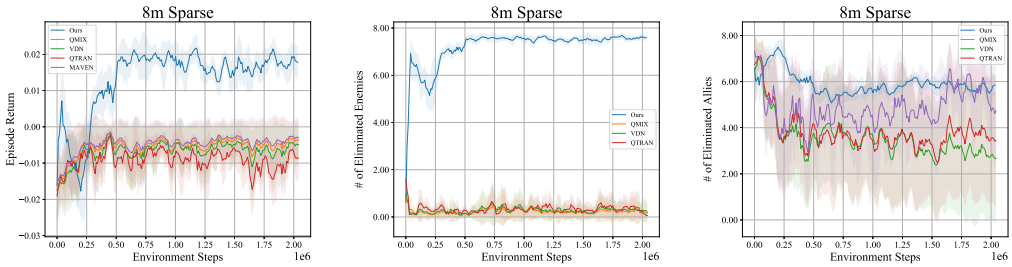

Figure 9: Episode return, number of eliminated enemies, and number of eliminated allies of CMAE and baselines on *8m-sparse*.

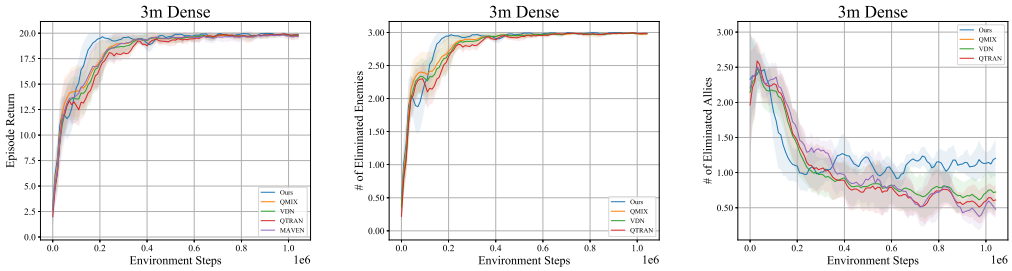

Figure 10: Episode return, number of eliminated enemies, and number of eliminated allies of CMAE and baselines on *3m-dense*.

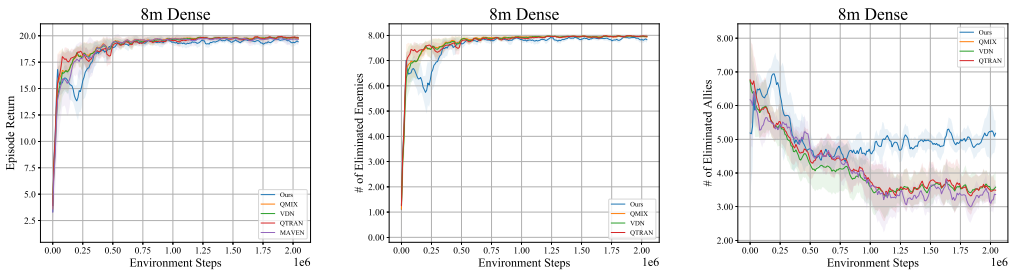

Figure 11: Episode return, number of eliminated enemies, and number of eliminated allies of CMAE and baselines on *8m-dense*.