

1 We’ve made some changes and submit an updated version here.

2 A PSEUDOCODE of our method

3 The pseudocode of our method is detailed in 1, and our code is provided in the supplemental material.

Algorithm 1 BiKT for Multi-Task MARL

```

1: Inputs:
2: The offline dataset  $\mathcal{D}_{\text{src}} = \{\mathcal{D}_{\text{src}}^n\}_{n=1}^{N_{\text{src}}}$ , Individual Skill Encoder  $p_{\text{skill}}$ , Action Decoder  $q_{\text{act}}$ , cooperative tactic
   encoder  $p_{\text{tac}}$ , skill decoder  $q_{\text{skill}}$ , the tactic codebook embeddings  $\mathcal{C} = \{c_k\}_{k=1}^K$ , the skill-based decision
   transformer  $\{\pi_i\}_{i=1}^N$ , the number of source tasks  $N_{\text{src}}$ , the number of agents  $N$ , the tactic codebook number
    $K$ , the content length of bi-level decision transformer  $L$ , learning rates  $l_1, l_2, l_3$ 
3: Training:
4: for each timestep  $t$  in  $1..H$  do
5:   Sample  $\delta_t^i = (s_t, \tau_t^i, \mathbf{a}_t) \sim \mathcal{D}_{\text{src}}$  # Individual Skill Learning
6:   Use  $\delta_t^i$  to compute  $\mathcal{L}_{\text{skill}}(\phi_1)$  in Eq. 3.
7:   Calculate gradients to update  $p_{\text{skill}}$  and  $q_{\text{act}}$ , with learning rate  $l_1$ 
8: end for
9: for each timestep  $t$  in  $1..H$  do
10:  Sample  $\zeta_t^i = (s_t, \tau_t^i, z_t^1, \dots, z_t^N) \sim \mathcal{D}_{\text{src}}$  # Cooperative Tactic Codebook Learning
11:  Use  $\zeta_t^i$  to compute  $\mathcal{L}_{\text{tactic}}(\phi_2)$  in Eq. 4.
12:  Calculate gradients to update  $p_{\text{tac}}$  and  $q_{\text{skill}}$ , with learning rate  $l_2$ 
13: end for
14: for each timestep  $t$  in  $1..H$  do
15:  Sample  $\Omega_t^i = (\tau_{\leq t}^i, c_{\leq t}^i, z_{\leq t}^i, \hat{R}_t^i) \sim \mathcal{D}_{\text{src}}$  # Bi-level Decision Transformer Learning
16:  Use  $\Omega_t^i$  to compute  $\mathcal{L}_{\text{policy}}(\theta)$  in Eq. 5.
17:  Calculate gradients to update  $p_{\text{skill}}$  and  $q_{\text{act}}$ , with learning rate  $l_3$ 
18: end for
19: Execution:
20: for each timestep  $t$  in source task  $\mathcal{T}_{\text{src}}^n$  do
21:   Given return-to-go  $\{\hat{R}_t^i\}_{i=1}^N$ 
22:    $\{c_t^i\}_{i=1}^N \leftarrow \pi_{\theta}(c_t^i | \hat{R}_{\leq t}^i, o_{\leq t}^i, c_{\leq t}^i, z_{\leq t}^i)$  # Select team tactic
23:    $\{z_t^i\}_{i=1}^N \leftarrow \pi_{\theta}(z_t^i | \hat{R}_{\leq t}^i, o_{\leq t}^i, c_{\leq t}^i, z_{\leq t}^i)$  # Take Individual skills
24:    $\{a_t^i\}_{i=1}^N \leftarrow q_{\text{act}}(\cdot | \tau_t^i, \dots, \tau_t^N, z_t^1, \dots, z_t^N)$  # Take Individual actions
25: end for

```

4

5 B Experiment Setting Details

6 B.1 SMAC

7 **Environment Overview** SMAC is derived from the real-time strategy game StarCraft II, focusing
8 on micromanagement. Unlike typical StarCraft II games that involve both macromanagement
9 (strategic planning) and micromanagement (fine control of units), SMAC is structured to emphasize
10 decentralized control by requiring each unit to be managed by an independent agent based solely on
11 local, limited observations. This setup necessitates multi-agents learning sophisticated cooperative
12 behaviors under the challenge of partial observability. SMAC consists of diverse micro scenarios
13 designed to assess how well agents coordinate to solve complex tasks. Each scenario involves two
14 opposing armies with variations in initial positioning, unit types, and terrain features.

15 **Observations, Actions and Team Goal.** At each timestep, agents gain local observations within
16 their field of view, providing information such as distance, health, shields, and unit type of visible units,
17 as well as terrain features. During centralized training, the global state includes comprehensive data
18 on all units, including energy levels and attack cooldowns. Agents have a discrete action set including
19 movement, attacks, healing by Medivacs with certain constraints ensuring decentralization. The
20 shooting range of units is limited compared to their sight range, necessitating strategic maneuvering.

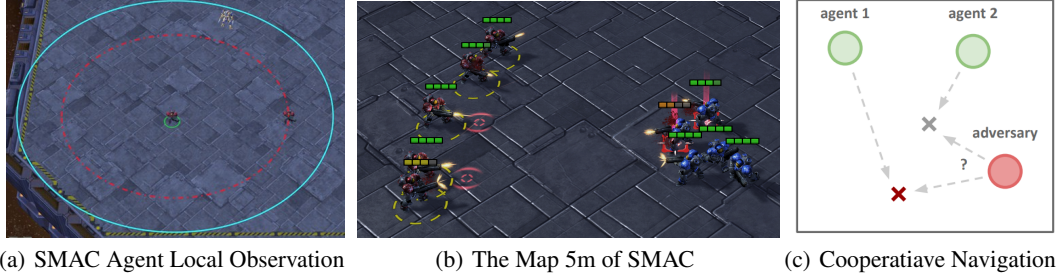


Table 1: Descriptions of tasks in the Marine-Hard task set.

| Task type | Task | Ally units | Enemy units | Properties |
|--------------|------------|------------|-------------|--------------------------|
| Source tasks | 3m | 3 Marines | 3 Marines | homogeneous & symmetric |
| | 5m_vs_6m | 5 Marines | 6 Marines | homogeneous & asymmetric |
| | 9m_vs_10m | 9 Marines | 10 Marines | homogeneous & asymmetric |
| Unseen tasks | 4m | 4 Marines | 4 Marines | homogeneous & symmetric |
| | 5m | 5 Marines | 5 Marines | homogeneous & symmetric |
| | 10m | 10 Marines | 10 Marines | homogeneous & symmetric |
| | 12m | 12 Marines | 12 Marines | homogeneous & symmetric |
| | 7m_vs_8m | 7 Marines | 8 Marines | homogeneous & asymmetric |
| | 8m_vs_9m | 8 Marines | 9 Marines | homogeneous & asymmetric |
| | 10m_vs_11m | 10 Marines | 11 Marines | homogeneous & asymmetric |
| | 10m_vs_12m | 10 Marines | 12 Marines | homogeneous & asymmetric |
| | 13m_vs_15m | 13 Marines | 15 Marines | homogeneous & asymmetric |

21 The allied units are controlled by agents trained to maximize the win rate against enemy units
 22 governed by the game’s AI using scripted strategies.

23 **Multi-Task Settings.** To assess multi-task generalization, we follow the ODIS setting with three
 24 task sets: *Marine-Hard*, *Marine-Easy*, and *Stalker-Hard*. Each task set consists of distinct training
 25 source tasks and testing scenarios. Specific details for each task set are presented in the Tables 1, 2
 26 and 3. The *Marine-Hard* and *Marine-Easy* sets comprise different marine battle scenarios where
 27 the learned multi-agent strategy must guide allied marines against enemy marines controlled by the
 28 game’s AI, matching or exceeding in number. The *Stalker-Zealot* set consists of several challenges
 29 involving equal numbers of stalkers and zealots on either side.

30 **Offline Dataset.** In our experiments, we use the same dataset collected by ODIS for the fair
 31 comparison. Within each task set, the offline data is collected by pre-trained QMIX policies with
 32 different levels of performance: *Expert*, *Medium*, *Medium-Expert* and *Medium-Replay*. The *Expert*
 33 policy is trained with 2,000,000 environment steps. The *Medium* policy is trained until it achieves
 34 approximately a 50% win rate. The *Medium-Expert* dataset is a mixture of trajectories from both the *Expert* and
 35 *Medium* policies. The *Medium-Replay* dataset is obtained from the replay buffer of the *Medium*
 36 policy, which contains a larger proportion of lower-quality trajectories. Table 4 summarizes the full
 37 settings of the offline datasets.

38 B.2 Cooperative Navigation

39 **Environment Overview** To further evaluate our method, we consider a task set based on the
 40 Cooperative Navigation (CN) scenario, a representative cooperative task from the Multi-Agent
 41 Particle Environment (MPE). The environment consists of N agents and L landmarks situated in a
 42 two-dimensional continuous space with discrete time steps. Agents must coordinate their physical
 43 actions to reach the L landmarks. Each agent observes the relative positions of other agents and
 44 landmarks, and the team receives a shared reward based on the proximity of any agent to each
 45 landmark—that is, the goal is for all landmarks to be ‘covered’ by the team. Agents occupy physical

Table 2: Descriptions of tasks in the Marine-Easy task set.

| Task type | Task | Ally units | Enemy units | Properties |
|--------------|------|------------|-------------|-------------------------|
| Source tasks | 3m | 3 Marines | 3 Marines | homogeneous & symmetric |
| | 5m | 5 Marines | 5 Marines | homogeneous & symmetric |
| | 10m | 10 Marines | 10 Marines | homogeneous & symmetric |
| Unseen tasks | 4m | 4 Marines | 4 Marines | homogeneous & symmetric |
| | 6m | 6 Marines | 6 Marines | homogeneous & symmetric |
| | 7m | 7 Marines | 7 Marines | homogeneous & symmetric |
| | 8m | 8 Marines | 8 Marines | homogeneous & symmetric |
| | 9m | 9 Marines | 9 Marines | homogeneous & symmetric |
| | 11m | 11 Marines | 11 Marines | homogeneous & symmetric |
| | 12m | 12 Marines | 12 Marines | homogeneous & symmetric |

Table 3: Descriptions of tasks in the Stalker-Zealot task set.

| Task type | Task | Ally units | Enemy units | Properties |
|--------------|------|--------------------------|--------------------------|---------------------------|
| Source tasks | 2s3z | 2 Stalkers, 3 Zealots | 2 Stalkers, 3 Zealots | heterogeneous & symmetric |
| | 2s4z | 2 Stalkers, 4 Zealots | 2 Stalkers, 4 Zealots | heterogeneous & symmetric |
| | 3s5z | 3 Stalkers, 5 Zealots | 3 Stalkers, 5 Zealots | heterogeneous & symmetric |
| Unseen tasks | 1s3z | 1 Stalkers, 3 Zealots | 1 Stalkers, 3 Zealots | heterogeneous & symmetric |
| | 1s4z | 1 Stalkers, 4 Zealots | 1 Stalkers, 4 Zealots | heterogeneous & symmetric |
| | 1s5z | 1 Stalkers, 5 Zealots | 1 Stalkers, 5 Zealots | heterogeneous & symmetric |
| | 2s5z | 2 Stalkers, 5 Zealots | 2 Stalkers, 5 Zealots | heterogeneous & symmetric |
| | 3s3z | 3 Stalkers, 3 Zealots | 3 Stalkers, 3 Zealots | heterogeneous & symmetric |
| | 3s4z | 3 Stalkers, 4 Zealots | 3 Stalkers, 4 Zealots | heterogeneous & symmetric |
| | 4s3z | 4 Stalkers, 3 Zealots | 4 Stalkers, 3 Zealots | heterogeneous & symmetric |
| | 4s4z | 4 Stalkers, 4 Zealots | 4 Stalkers, 4 Zealots | heterogeneous & symmetric |
| | 4s5z | 4 Stalkers, 5 Zealots | 4 Stalkers, 5 Zealots | heterogeneous & symmetric |
| | | | | |

space and are penalized for collisions with one another, encouraging coordinated but non-overlapping behaviors. In this setting, agents must infer which landmark to cover and navigate there while avoiding others. The agents can execute discrete actions of moving towards four directions and a “none” operation.

Multi-Task settings. The task set of CN consists of different numbers of agents. Specifically, CN- n denotes a CN map containing n agents. Offline datasets are collected using the QMIX algorithm. Detailed dataset settings are summarized in Table 5.

B.3 Computing Resources

For computing resources, we utilize the *Intel(R) Xeon(R) Gold 5220* CPU and *NVIDIA TITAN RTX* GPU in the experiments. Each experiment in per task set lasts on average for 8 hours.

Table 4: Properties of offline datasets in SMAC with different qualities.

| Tasks | Quality | Trajectories | Average Return | Average Win Rate |
|-----------|---------------|--------------|----------------|------------------|
| 3m | expert | 2000 | 19.8929 | 0.9910 |
| | medium | 2000 | 13.9869 | 0.5402 |
| | medium-expert | 4000 | 16.9399 | 0.7656 |
| | medium-replay | 3603 | N/A | N/A |
| 5m | expert | 2000 | 19.9380 | 0.9937 |
| | medium | 2000 | 17.3288 | 0.7411 |
| | medium-expert | 4000 | 18.6334 | 0.8674 |
| | medium-replay | 711 | N/A | N/A |
| 10m | expert | 2000 | 19.9438 | 0.9922 |
| | medium | 2000 | 16.6297 | 0.5413 |
| | medium-expert | 4000 | 18.2595 | 0.7626 |
| | medium-replay | 571 | N/A | N/A |
| 5m_vs_6m | expert | 2000 | 17.3424 | 0.7185 |
| | medium | 2000 | 12.6408 | 0.2751 |
| | medium-expert | 4000 | 14.9916 | 0.4968 |
| | medium-replay | 32607 | N/A | N/A |
| 9m_vs_10m | expert | 2000 | 19.6140 | 0.9431 |
| | medium | 2000 | 15.5049 | 0.4146 |
| | medium-expert | 4000 | 17.5594 | 0.6789 |
| | medium-replay | 13731 | N/A | N/A |
| 2s3z | expert | 2000 | 19.7655 | 0.9602 |
| | medium | 2000 | 16.6279 | 0.4465 |
| | medium-expert | 4000 | 18.1967 | 0.7034 |
| | medium-replay | 4505 | N/A | N/A |
| 2s4z | expert | 2000 | 19.7402 | 0.9509 |
| | medium | 2000 | 16.8735 | 0.4965 |
| | medium-expert | 4000 | 18.3069 | 0.7237 |
| | medium-replay | 6172 | N/A | N/A |
| 3s5z | expert | 2000 | 19.7850 | 0.9518 |
| | medium | 2000 | 16.3126 | 0.3114 |
| | medium-expert | 4000 | 18.0488 | 0.6316 |
| | medium-replay | 11528 | N/A | N/A |

Table 5: Properties of offline datasets in Cooperative Navigation with different qualities.

| Tasks | Quality | Trajectories | Average Return | Average Win Rate |
|-------|---------|--------------|----------------|------------------|
| CN-2 | expert | 2000 | 1.0000 | 1.0000 |
| | medium | 2000 | 0.6152 | 0.6152 |
| CN-4 | expert | 2000 | 0.7173 | 0.7173 |
| | medium | 2000 | 0.4273 | 0.4273 |

56 C Implementation Details

57 In this section, we will provide the model structure, the hyperparameters, and other training details of
58 ODIS. We present each part of BiKT in the following sections.

59 C.1 Multi-Head Attention

60 We utilize Multi-Head Attention (MHA) to represent the embeddings of skills and tactics. This
61 mechanism enables the model to jointly attend to different representation subspaces, making it

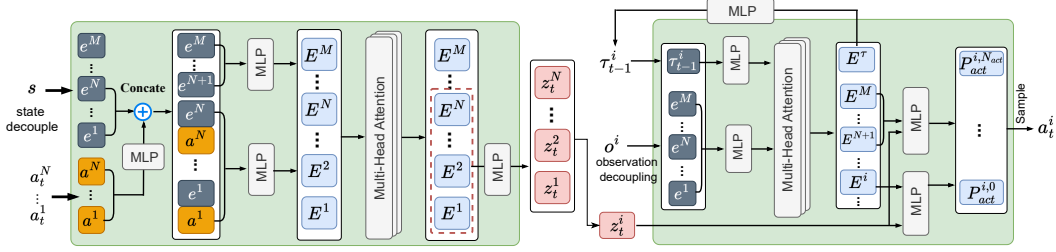


Figure 1: The detailed model structure of our individual skill learning.

effective for modeling contextual dependencies. Given query, key, and value matrices $Q \in \mathbb{R}^{T \times d}$, $K \in \mathbb{R}^{S \times d}$, and $V \in \mathbb{R}^{S \times d}$, the scaled dot-product attention is computed as in Equation 1. Multi-head attention applies this operation across h heads, where each head has its own projection matrices W_i^Q , W_i^K , and W_i^V . The result of each head is shown in Equation 2, and the final output is formed by concatenating all heads and applying a linear projection, as shown in Equation 3.

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right) V \quad (1)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (3)$$

C.2 Details of Individual Skill Learning

The detailed architecture for individual skill learning is illustrated in Figure 1. During the skill encoding phase, each agent employs the shared encoder $p_{\text{skill}}(\cdot \mid s, \mathbf{a}, i)$ to compute its latent skill embedding $z^i \in \mathbb{R}^{N_s}$. Specifically, the encoder takes as input the global state s , the joint action \mathbf{a} , and the agent index i .

To construct the encoder input, we first concatenate the actions and entity features of the N allied agents. These N concatenated representations, along with the remaining $M - N$ entity features (e.g., enemies or neutral units), are mapped into a unified set of M embeddings. These embeddings are then passed through the Multi-Head Attention module of the Transformer. Finally, we take the N attention outputs corresponding to the allied agents and feed them into an MLP to generate the individual skill embeddings.

To reconstruct actions, each agent is expected to infer its action based on its trajectory history and individual skill embedding. To achieve this, we first extract entity features from the observation o^i , and process the history τ_t^i and entity features through separate multilayer perceptrons (MLPs). The resulting representations are then fed into a MHA module to capture relevant contextual dependencies. The MHA output is subsequently concatenated with the individual skill embedding z_t^i and passed through another set of MLPs to produce the action logits. These logits define the action distribution P_{act} over N_{act} discrete action dimensions. Finally, the action is sampled from this distribution for execution.

C.3 Details of Cooperative Tactic Learning

The detailed architecture for the cooperative tactic codebook is illustrated in Figure 2. We construct the *Cooperative Tactic Encoder* $p_{\text{tac}}(\cdot \mid s, \{z^i\}_{i=1}^N) \rightarrow \hat{c}_t \in \mathbb{R}^{N_s}$, the *Skill Decoder* $q_{\text{skill}}(\cdot \mid \tau^i, c_k) \rightarrow \hat{z}_t^i$, and the tactic codebook $\mathcal{C} = \{c_k\}_{k=1}^K$. The tactic embeddings in codebook \mathcal{C} is randomly initialized.

The Tactic Encoder maps the global state s_t and the individual skill embeddings $[z_t^1, \dots, z_t^N]$ into a continuous tactic representation \hat{c}_t . Specifically, each individual skill z_t^i is concatenated with the corresponding ally's entity features, forming N enriched ally representations. These, along with the remaining entity features, are separately embedded into vectors and then fed into the MHA module. The outputs are averaged to produce the final tactic embedding. This tactic embedding is then discretized by searching the nearest neighbor in the tactic codebook \mathcal{C} , resulting in the selected tactic $c_k \in \mathcal{C}$.

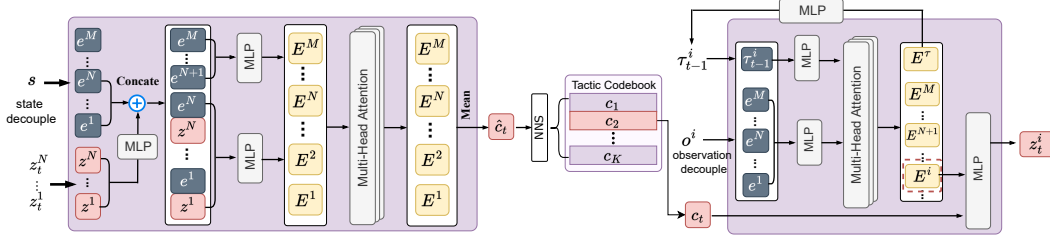


Figure 2: The detailed model structure of our cooperative tactic codebook.

To decode the individual skills, the entity features extracted from observations and agent i 's trajectory history τ^i are fed into the skill decoder q_{skill} . These inputs are first embedded separately using MLPs and then passed through the MHA module. The output embedding corresponding to agent i and the individual skill representation z_t are passed through an MLP to generate the final decoded skill.

Subsequently, the Skill Decoder reconstructs each agent's latent skill \hat{z}_t^i from its local trajectory history τ^i and the selected tactic c_k . This process enables the integration of global cooperative strategies with decentralized agent behavior during execution.

During the tactic encoding phase, each agent employs the encoder $p_{\text{tac}}(\cdot \mid s, \mathbf{a}, i)$ to compute its latent skill embedding $z^i \in \mathbb{R}^{N_s}$. Specifically, the encoder takes as input the global state s , the joint action \mathbf{a} , and the agent index i .

Stop-Gradient Operator. The stop-gradient operator, denoted as $\text{sg}[\cdot]$, is used to block gradients during backpropagation while preserving values during the forward pass. Formally, for a variable x , the stop-gradient operation behaves as:

$$\text{Forward: } \text{sg}[x] = x, \quad \text{Backward: } \frac{\partial \text{sg}[x]}{\partial x} = 0. \quad (4)$$

That is, in the forward pass, $\text{sg}[x]$ evaluates to the same value as x , but during the backward pass, no gradients are propagated through $\text{sg}[x]$.

C.4 Hyper-parameters of BiKT

The hyperparameters of our method are detailed in Table 6

Table 6: Hyperparameters of our method.

| Hyperparameter | Value |
|--|--------|
| Individual skill dimension N_s | 4 |
| Tactic embedding N_c | 64 |
| The number of tactics in \mathcal{C} : K | 16 |
| Hidden layer dimension of BDT | 64 |
| The multi-head number of BDT | 2 |
| The content length of BDT | 10 |
| Optimizer | Adam |
| Training steps for L_{skill} | 15000 |
| Training steps for L_{tactic} | 8000 |
| Training steps for L_{policy} | 30000 |
| Batch size | 32 |
| learning rate l_1 | 0.0004 |
| learning rate l_2 | 0.0001 |
| learning rate l_3 | 0.0002 |
| β_1 | 1 |
| β_2 | 0.01 |
| α | 0.05 |

Table 7: The performance of different methods in Task set *Stalker Zealot*.

| Tasks | Expert | | | | Medium | | | |
|-------|---------------|-----------|-----------|-----------|---------------|-----------|-----------|-----------|
| | UPDeT-m | ODIS | Hi-SSD | BiKT | UPDeT-m | ODIS | Hi-SSD | BiKT |
| | Source Tasks | | | | | | | |
| 2s3z | 50.0±33.4 | 97.7±2.6 | 95.2±1.0 | 97.9±2.3 | 35.0±23.0 | 49.2±8.4 | 32.3±11.7 | 51.6±3.3 |
| 2s4z | 23.4±26.6 | 60.9±6.8 | 79.8±6.0 | 93.2±5.1 | 18.8±10.3 | 32.8±12.2 | 17.0±2.2 | 25.0±7.4 |
| 3s5z | 17.2±19.8 | 87.5±9.6 | 92.8±5.0 | 93.0±4.5 | 25.6±24.2 | 28.9±6.8 | 24.4±7.9 | 29.8±2.3 |
| | Unseen Tasks | | | | | | | |
| 1s3z | 1.6±1.6 | 76.6±3.5 | 81.6±15.2 | 77.0±4.2 | 3.8±5.0 | 41.4±18.8 | 44.2±9.9 | 32.4±0.4 |
| 1s4z | 26.6±19.3 | 17.2±10.5 | 42.0±26.1 | 52.6±15.1 | 2.5±3.6 | 50.7±7.5 | 18.1±11.0 | 22.6±0.5 |
| 1s5z | 29.7±26.4 | 2.5±2.3 | 16.7±12.3 | 8.7±4.3 | 5.0±4.2 | 14.1±8.4 | 2.5±2.2 | 18.8±3.9 |
| 2s5z | 23.4±22.2 | 27.3±6.0 | 79.7±2.2 | 75.3±3.2 | 16.9±14.1 | 32.0±4.6 | 11.3±3.7 | 24.2±6.5 |
| 3s3z | 20.3±10.9 | 89.1±5.2 | 88.0±4.5 | 98.4±1.6 | 24.4±28.6 | 23.4±9.2 | 21.9±10.7 | 34.4±2.2 |
| 3s4z | 12.5±19.9 | 96.9±2.2 | 88.1±9.0 | 97.7±2.3 | 28.8±31.6 | 50.8±15.5 | 17.2±4.5 | 54.8±0.5 |
| 4s3z | 6.2±4.4 | 64.1±13.0 | 88.6±4.1 | 96.9±2.3 | 11.2±18.0 | 13.3±7.5 | 31.9±23.2 | 18.7±0.1 |
| 4s4z | 7.8±13.5 | 79.7±10.9 | 73.4±5.2 | 75.5±5.3 | 1.2±1.5 | 12.5±7.0 | 13.2±6.5 | 16.7±1.4 |
| 4s5z | 5.5±7.8 | 86.7±12.6 | 65.6±3.7 | 44.5±6.8 | 5.6±8.5 | 7.0±4.1 | 4.5±1.3 | 8.3±1.4 |
| 4s6z | 4.7±6.4 | 88.3±8.4 | 68.4±4.9 | 68.2±6.9 | 1.9±2.5 | 1.6±1.6 | 0.9±0.9 | 2.5±2.5 |
| | Medium-Expert | | | | Medium-Replay | | | |
| | Source Tasks | | | | | | | |
| 2s3z | 57.5±27.1 | 58.6±15.5 | 68.1±8.1 | 81.3±7.4 | 14.4±13.2 | 15.6±18.2 | 9.0±1.5 | 30.2±8.4 |
| 2s4z | 53.1±24.6 | 41.4±7.8 | 41.9±10.2 | 73.8±7.8 | 12.5±9.7 | 7.8±5.2 | 6.0±1.2 | 30.8±7.1 |
| 3s5z | 35.0±23.5 | 41.4±18.5 | 57.8±10.7 | 59.1±8.7 | 20.0±16.6 | 18.8±3.1 | 17.5±2.0 | 19.3±7.1 |
| | Unseen Tasks | | | | | | | |
| 1s3z | 4.4±8.8 | 72.7±12.2 | 73.0±10.2 | 75.9±9.1 | 0.0±0.0 | 21.1±20.4 | 36.3±7.1 | 30.9±9.1 |
| 1s4z | 11.9±9.8 | 44.5±20.3 | 32.3±30.5 | 37.9±5.9 | 7.5±10.0 | 6.2±7.7 | 24.8±9.1 | 26.3±7.2 |
| 1s5z | 3.8±4.6 | 42.2±31.4 | 9.4±9.5 | 14.4±19.4 | 11.9±9.6 | 7.8±6.4 | 4.4±2.2 | 12.5±4.7 |
| 2s5z | 37.5±22.5 | 43.0±10.7 | 25.6±7.8 | 19.0±5.2 | 20.0±16.8 | 14.1±8.1 | 16.5±2.8 | 17.2±8.4 |
| 3s3z | 33.8±15.0 | 50.0±13.3 | 56.6±25.6 | 57.9±8.2 | 17.5±12.3 | 25.0±20.1 | 9.6±3.3 | 27.6±4.5 |
| 3s4z | 43.1±20.7 | 52.3±9.5 | 71.7±9.7 | 75.6±13.3 | 15.6±11.2 | 19.5±16.6 | 22.5±10.6 | 19.4±11.1 |
| 4s3z | 23.8±21.0 | 17.2±7.2 | 60.5±15.1 | 28.8±9.4 | 11.2±15.0 | 8.6±14.9 | 11.0±10.4 | 10.4±5.1 |
| 4s4z | 10.6±13.8 | 20.3±6.8 | 37.3±9.4 | 39.9±4.9 | 5.6±9.8 | 4.7±8.1 | 9.4±1.8 | 8.3±2.9 |
| 4s5z | 11.9±16.1 | 21.9±2.2 | 17.0±4.1 | 24.3±5.3 | 10.6±19.7 | 0.8±1.4 | 0.8±0.8 | 4.4±3.5 |
| 4s6z | 5.0±8.5 | 18.0±5.1 | 19.7±5.9 | 14.8±3.2 | 6.9±13.8 | 2.3±4.1 | 2.3±4.1 | 3.5±2.9 |

Table 8: The performance of different methods in Task set *Marine Easy*

| Tasks | Expert | | | | Medium | | | |
|-------|---------------|-----------|-----------|-----------|---------------|-----------|-----------|-----------|
| | UPDeT-m | ODIS | Hi-SSD | BiKT | UPDeT-m | ODIS | Hi-SSD | BiKT |
| | Source Tasks | | | | | | | |
| 3m | 83.6±12.6 | 97.7±2.6 | 99.5±8.1 | 99.4±1.3 | 60.2±29.9 | 57.8±9.2 | 74.7±14.6 | 87.2±4.7 |
| 5m | 74.8±22.9 | 95.3±5.2 | 99.9±0.0 | 99.9±0.0 | 67.8±5.9 | 82.8±5.2 | 81.6±10.8 | 74.2±5.9 |
| 10m | 83.6±19.2 | 88.3±20.3 | 95.2±8.4 | 99.9±0.0 | 48.8±7.9 | 71.9±6.6 | 84.8±8.6 | 82.1±7.2 |
| | Unseen Tasks | | | | | | | |
| 4m | 53.0±32.3 | 90.6±7.0 | 94.4±2.9 | 96.9±1.5 | 41.7±17.4 | 63.3±16.1 | 74.5±15.5 | 75.5±8.6 |
| 6m | 37.9±8.6 | 79.7±17.5 | 99.7±0.3 | 99.7±0.1 | 75.8±22.7 | 89.8±17.6 | 88.0±10.0 | 83.0±4.7 |
| 7m | 44.2±13.2 | 72.7±16.9 | 99.1±0.7 | 99.7±0.1 | 65.2±25.2 | 96.1±1.4 | 97.3±2.3 | 89.9±0.0 |
| 8m | 51.7±26.2 | 80.9±14.4 | 99.8±0.3 | 99.1±0.1 | 88.4±13.7 | 97.7±2.6 | 93.8±5.2 | 98.9±1.2 |
| 9m | 76.3±13.4 | 99.2±1.4 | 99.9±0.0 | 99.9±0.0 | 64.8±35.6 | 87.5±2.2 | 75.2±15.5 | 88.9±11.8 |
| 11m | 53.6±22.4 | 83.6±12.4 | 99.2±0.8 | 99.3±1.0 | 23.4±11.8 | 64.7±3.1 | 62.0±21.8 | 68.2±4.7 |
| 12m | 44.3±22.8 | 70.3±30.2 | 99.7±1.1 | 99.6±1.0 | 13.5±11.7 | 41.4±6.0 | 55.5±25.7 | 49.7±13.4 |
| | Medium-Expert | | | | Medium-Replay | | | |
| | Source Tasks | | | | | | | |
| 3m | 48.4±36.8 | 89.8±9.7 | 90.9±5.9 | 91.3±4.8 | 29.7±10.0 | 79.7±4.7 | 87.7±2.9 | 78.8±3.2 |
| 5m | 64.1±17.9 | 83.7±16.0 | 79.4±6.9 | 85.3±5.9 | 6.2±10.8 | 3.1±5.4 | 87.5±2.9 | 88.5±1.6 |
| 10m | 68.8±23.8 | 93.8±4.4 | 60.2±21.1 | 83.6±3.1 | 0.0±0.0 | 0.0±0.0 | 84.2±4.9 | 85.2±2.5 |
| | Unseen Tasks | | | | | | | |
| 4m | 43.7±25.0 | 57.8±18.8 | 70.9±9.1 | 72.2±8.3 | 25.0±22.6 | 25.0±5.4 | 71.6±4.1 | 77.2±4.7 |
| 6m | 47.7±30.0 | 76.0±6.0 | 70.6±6.1 | 78.2±1.6 | 0.0±0.0 | 3.1±5.4 | 99.8±0.3 | 86.8±3.2 |
| 7m | 57.8±32.9 | 66.4±14.6 | 85.0±11.7 | 85.6±16.4 | 0.0±0.0 | 0.0±0.0 | 99.8±0.3 | 84.2±1.4 |
| 8m | 40.6±19.3 | 43.8±11.5 | 72.8±9.5 | 68.3±4.6 | 0.0±0.0 | 1.6±1.6 | 96.7±0.3 | 87.6±1.7 |
| 9m | 47.7±24.8 | 73.4±16.2 | 80.0±14.6 | 70.8±6.6 | 0.0±0.0 | 0.0±0.0 | 88.8±1.3 | 86.6±2.4 |
| 11m | 85.9±14.2 | 68.8±20.3 | 70.9±5.9 | 75.5±12.4 | 0.0±0.0 | 0.0±0.0 | 45.6±4.5 | 52.3±3.6 |
| 12m | 46.1±15.5 | 62.5±8.0 | 62.7±7.8 | 59.7±9.8 | 0.0±0.0 | 0.0±0.0 | 38.0±3.7 | 41.5±4.3 |

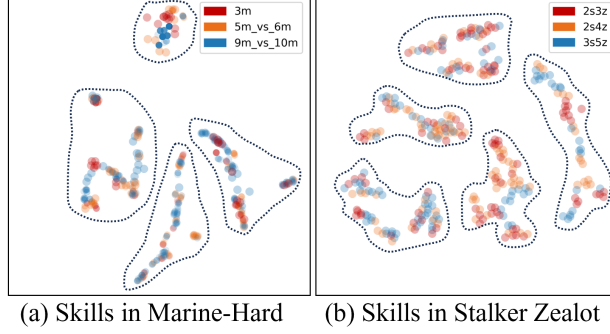


Figure 3: The individual skill embeddings.

D Additional Results

D.1 The performance comparison of other task sets.

The results for task set *Stalker Zealot* and *Marine Easy* are shown in Table 7 8. The *Stalker Zealot* requires different tactics in different tasks, which brings big challenge in policy transfer. The results show that our BiKT overall outperforms other baselines in both task sets. However, in *Marine Easy*, the tactics required for each task are similar, resulting that all methods can achieve high performance. It makes that Hi-SSD and BiKT can both obtain high performance in Expert setting.

D.2 Visualization of individual skills from source tasks

We additionally show the individual skill embeddings from the task labels, in Figure 3.

D.3 Ablation study

We conduct ablation experiments on task set *Marine Hard* to evaluate different variants of our method, and the results are shown in Table 9.

- **MADT_w_OD**: To evaluate the impact of tactic and skill learning, we remove them and let the decision transformer learns to take action directly, which naturally degrades into the MADT method with observation decoupling, denoted by MADT_w_OD. For fairness, we utilize the same hyperparameter in MADT_w_OD. For convenience we also provide the results of MADT.
- **$L = 5$** : We set the context length of skill-based Decision Transformer π_i as 5.
- **$C_{K=32}$** : During the team tactic learning process, we set the tactic number of codebook K as 32.
- **w/o C** : We overpass the learning process of team tactic and directly let the skill-based decision transformer to learn the individual skills. It is achieved by removing the c_t^i embedding token in Figure ?? . At this time, the SDT policy learns to directly output the individual skills and then takes its action.
- **Con-Tac**: Compared with continuous individual skill embeddings, we utilize a fixed number of tactics. For that we employ a VAE to learn team-level tactics and use *Continuous Tactic* (Con-Tac) embeddings instead of a discrete tactic set.

Ablation study: The action based policy struggles to generalize to different tasks. The results of MADT_w_OD in Table 9 indicate that our proposed tactic and skill learning components play a crucial role in the overall performance. Using an action-based policy introduces the challenge that the agent must take different actions under similar observations across diverse tasks, which cannot be addressed effectively without additional guiding information. As a result, MADT performs well only in tasks with similar numbers of agents and comparable problem settings, such as *3m*, *4m*, and *5m*. Although the observation decoupling in MADT_w_OD leads to performance improvements, it is not the primary contributing factor to generalization.

Ablation Study: Individual Skills Alone Cannot Transfer Diverse Team-Level Knowledge The results of *w/o C* in Table 9 show that without the guidance of team tactics, our method’s performance drops. This is because the policy must execute different skills without any external team-level information. In this setting, agents tend to learn a fixed combination of skills that only adapts well to a limited set of tasks, such as *3m*, *4m*, and *5m*. However, this approach fails to capture diverse team tactics required for more complex tasks like *7m_vs_8m* and *8m_vs_9m*, leading to a noticeable decline in policy transfer performance.

Ablation Study: Discrete Tactic Codebook Outperforms Continuous Tactic Embeddings The results for Con-Tac in Table 9 indicate that using continuous tactic embeddings can improve performance on some tasks. However, it still falls short of the results achieved with a discrete tactic codebook (BiKT). We argue that each tactic should correspond to a clear, meaningful, and reusable coordination pattern. Different tasks often share common tactics, which provide stable team-level guidance and assist agents in selecting appropriate individual skills across varied scenarios. Continuous tactic embeddings tend to blur this clarity, thereby weakening the effectiveness of team strategies.

Ablation Study: Impact of Content Length and Tactic Codebook Size The results with a content length $L = 5$ for the BDT model demonstrate that our method’s success does not heavily depend on a complex Transformer architecture. Instead, the key factor is the way we incorporate bi-level knowledge transfer in multi-task MARL, which proves to be highly effective. Regarding the tactic codebook size, the results with $C = 32$ show that our tactic learning process converges to meaningful and useful tactic embeddings. This indicates that a moderately sized codebook is sufficient to achieve both efficiency and performance.

Table 9: The results of ablation study in task set *Marine Hard*

| | Source Tasks | | | Unseen Tasks | | |
|------------|------------------|-----------------|-----------------|-----------------|------------------|-----------------|
| | 3m | 5m_vs_6m | 9m_vs_10m | 4m | 5m | 10m |
| MADT | 88.5±3.9 | 3.1±0.0 | 1.0±1.5 | 83.3±5.3 | 75.0±6.8 | 1.0±1.5 |
| MADT_w_OD | 90.2±2.8 | 10.2±3.5 | 16.2±6.8 | 88.4±2.3 | 83.2±2.7 | 12.7±4.2 |
| $L = 5$ | 100.0±0.0 | 78.9±3.5 | 98.4±1.5 | 99.3±0.9 | 99.9±0.8 | 97.2±1.5 |
| $C_{K=32}$ | 100.0±0.0 | 80.9±3.2 | 99.2±0.3 | 99.2±0.1 | 99.3±0.2 | 99.3±0.2 |
| w/o C | 98.2±0.2 | 68.2±5.3 | 83.2±2.5 | 90.2±1.9 | 88.6±2.7 | 86.2±3.2 |
| Con-Tac | 99.2±0.4 | 64.3±6.7 | 80.3±4.4 | 92.2±2.4 | 90.6±4.6 | 88.3±4.2 |
| BiKT | 100.0±0.0 | 81.3±4.5 | 99.4±0.4 | 99.3±0.1 | 100.0±0.0 | 99.4±0.1 |

| Unseen Tasks | | | | | | |
|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|
| | 12m | 7m_vs_8m | 8m_vs_9m | 10m_vs_11m | 10m_vs_12m | 13m_vs_15m |
| MADT | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 |
| MADT_w_OD | 8.2±3.2 | 8.4±3.9 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 |
| $L = 5$ | 98.7±0.9 | 64.8±8.2 | 48.2±8.3 | 90.2±2.3 | 12.0±1.8 | 3.2±1.5 |
| $C_{K=32}$ | 99.0±0.2 | 70.2±8.0 | 46.2±7.4 | 90.2±2.1 | 12.3±2.2 | 3.3±1.6 |
| w/o C | 57.2±4.2 | 23.3±5.5 | 20.3±9.2 | 40.7±9.2 | 1.6±1.6 | 1.6±1.3 |
| Con-Tac | 60.5±3.8 | 18.2±7.4 | 17.8±5.2 | 38.7±8.3 | 0.9±0.5 | 0.5±0.4 |
| BiKT | 99.0±0.2 | 68.0±9.9 | 50.0±6.2 | 90.6±1.1 | 14.6±1.5 | 4.2±2.1 |

D.4 Semantic of Individual skills and tactics

We provide more examples for our learned skills and tactics, as shown in Figure 4, 5 and 7.

E Limitations

While our proposed method demonstrates strong generalization across the evaluated tasks, it remains an open question whether it can consistently maintain performance when scaled to highly diverse and large-scale task distributions. Exploring more expressive or adaptive embedding mechanisms could be a promising direction for future work.



Figure 4: The semantics of individual skills in *Marine-Hard*

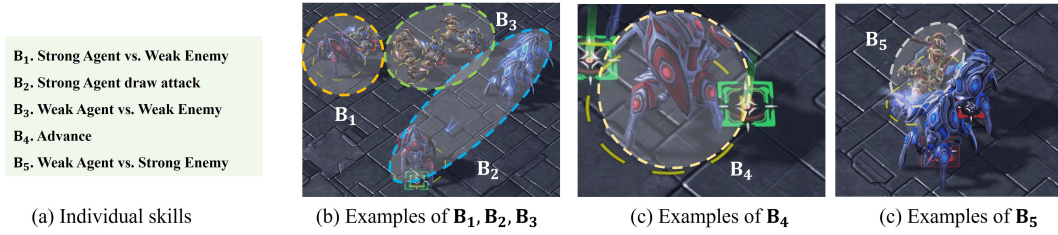


Figure 5: The semantics of individual skills in *Stalker-Zealot*.



Figure 6: The semantics of some tactics. The tactic ids correspond to Figure 4, which are learned from the offline trajectories. In (a), agents are learned to take all fire to a single enemy target, which can quickly eliminate an enemy and make up for the disadvantage in agent numbers. This tactic is more aggressive, and the win rate is not stable. In (b), the agents are learned to attack their enemy targets locally, without considering the disadvantage in agent numbers. It provides a more stable win rate, but it falls in 5m_vs_6m.

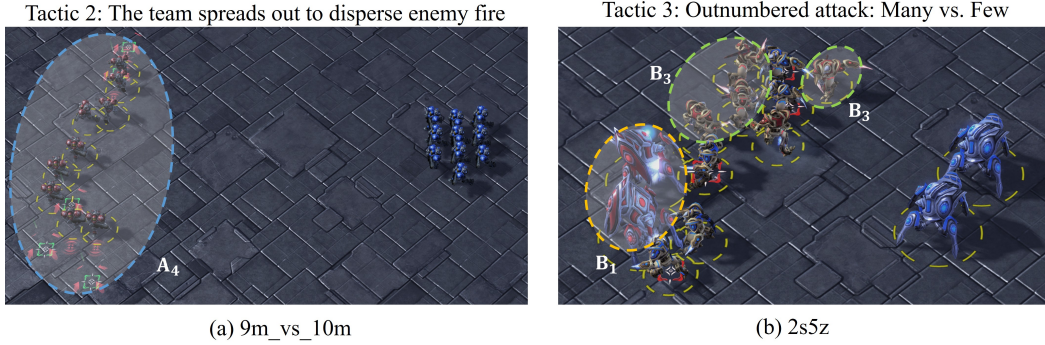


Figure 7: The semantics of some tactics. The tactic ids correspond to Figure 4. In (a), agents are learned to take skill A_4 to keep away from neighbors, which forms the team tactic that the team spreads out to disperse enemy fire. In (b), the agents learn to take skills B_1 and B_3 , allowing the stronger agent to attack a weaker enemy, while the weaker agent also targets a weak enemy. This tactic leverages numerical superiority to quickly eliminate weaker opponents.