DISCOVERING SEMANTICALLY GROUNDED AND VARIABLE-DURATION MULTI-AGENT COOPERATION SKILLS FROM OFFLINE MULTI-TASK DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline multi-task MARL learns cooperative policies from static source-task datasets for zero-shot deployment to unseen tasks. Recent skill-based methods improve cross-task generalization by discovering reusable latent coordination skills. However, existing methods learn distinguishable latent skill representations without semantic guidance; this leaves the skill-discovery space overly broad, and the resulting distinctions may not correspond to semantically meaningful coordination behaviors. Moreover, step-wise or fixed-interval skill inference prevents the skill durations from adapting to changes in the team's high-level coordination intention. Accordingly, we propose **S**emantically **G**rounded and **V**ariable-**D**uration skill learning (SGVD). We uses an LLM-assisted pipeline to iteratively refine Label Functions (LFs), which provide weak supervision through semantic labels and transition boundaries. Semantic labels guide latent skill representations to capture distinct and recognizable coordination behaviors, while transition boundaries supervise an observation-conditioned predictor that detects changes in the team's high-level coordination intention during execution and triggers coordination-skill transitions accordingly. Furthermore, we introduce segment-terminal prediction and dynamics distillation as two auxiliary tasks to enhance the ability of coordination-skill representations to capture segment-level transition dynamics. Experiments on offline multi-task SMAC task sets under four data qualities show that SGVD improves zero-shot performance while discovering semantically grounded, variable-duration coordination skills.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has achieved remarkable progress in complex decision-making tasks, such as multi-robot controlGu et al. (2023); Orr & Dutta (2023), autonomous drivingZhou et al. (2021); Palanisamy (2020), and real-time strategy games Samvelyan et al. (2019); Vinyals et al. (2019). However, real-world multi-agent systems are often open and dynamic, where the number of agents, opponents, targets, and environmental configurations may vary substantially across deployment scenarios. Therefore, learning generalizable MARL policies that can effectively transfer across diverse tasks is crucial for deploying MARL in real-world multi-agent systems.

A promising way to address this problem is to learn reusable coordination skills from offline trajectories across multiple source tasks and reuse them for decision-making in unseen tasks. Rather than pursuing generalization directly at the primitive-action level, this paradigm transfers reusable patterns of high-level coordination across tasks Zhang et al. (2023). Concretely, existing studies have extended this paradigm along several perspectives. HiSSD jointly learns common skills that incorporate cooperative temporal knowledge and task-specific skills that guide fine-grained action execution Liu et al. (2025), while VO-MASD captures subgroup- and temporal-level abstractions to extract subgroup coordination patterns as multi-agent skills Chen et al. (2025). HyGen integrates offline data and online interactions to refine general skills and learn high-level skill-selection policies Zhang et al. (2024), whereas BiKTZhang et al. (2025) transfers reusable knowledge at both the individual and team levels by extracting individual skill embeddings from offline trajectories and modeling their coordinated combinations as team-level tactics with a bi-level decision transformer.

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.

Despite these advances, existing skill-learning paradigms still face two limitations. First, they typically discover distinguishable latent skills through representation-learning objectives, while overlooking explicit semantic supervision. This leaves the skill-discovery space overly broad, such that differences among latent skills may capture incidental behavioral variations rather than semantically meaningful coordination patterns. Second, skills are typically inferred at every timestep or from fixed-length trajectory segments, imposing a predefined temporal constraint. Consequently, skill transitions may be misaligned with changes in the team's high-level coordination intention, fragmenting coherent behaviors or merging distinct coordination phases.

To this end, we propose **S**emantically **G**rounded and **V**ariable-**D**uration skill learning (SGVD), which learns coordination skills that are explicitly grounded in recognizable cooperative behaviors and persist for variable durations. To realize these properties, we leverage the reasoning capabilities of LLMs to identify semantically coherent coordination phases in offline trajectories. Rather than directly using LLM-generated annotations for policy learning, we use an LLM to assist the construction and iterative refinement of Label Functions (LFs) Yu & Bach (2023); Guan et al. (2025), which map behavioral signals extracted from trajectory segments to semantic phase labels and transition boundaries. The semantic labels provide supervision that grounds the learned skill representations in recognizable coordination behaviors, while the transition boundaries delineate changes in the team's high-level coordination intention. To enable variable-duration coordination skills, we introduce a boundary-guided switching module that retains the active skill between transitions and updates it only when a boundary is detected. During training, LF-derived transition boundaries determine the skill-switching points and supervise an observation-conditioned boundary predictor. During execution, this predictor detects changes in high-level coordination intention from agents' observations and triggers coordination-skill transitions accordingly. Furthermore, we introduce segment-terminal prediction and dynamics distillation as two auxiliary tasks to enhance the ability of coordination-skill representations to capture segment-level transition dynamics.

Our main **contributions** are fourfold: **(1)** We propose SGVD, which uses an LLM-assisted iterative LF pipeline to annotate semantic coordination phases and their transition boundaries, explicitly grounding team-level skills in recognizable cooperative behaviors. **(2)** We develop a boundary-guided switching mechanism that supervises an observation-conditioned boundary predictor with LF-derived boundaries, enabling variable-duration skill execution. **(3)** We introduce segment-terminal prediction and dynamics distillation to encode segment-level transition dynamics into coordination-skill representations. **(4)** Experiments on three offline multi-task SMAC task sets across four data qualities demonstrate SGVD's zero-shot generalization and its ability to discover semantically grounded, variable-duration coordination skills.

2 Related Work

2.1 Offline MARL and Multi-Task Generalization

Offline reinforcement learning (RL) learns policies from static datasets without further environment interaction. Existing offline RL and MARL methods address distribution shift through sequence modeling, conservative value estimation, implicit constraints, or behavior regularization Meng et al. (2023); Kumar et al. (2020); Yang et al. (2021); Kostrikov et al. (2022); Fujimoto & Gu (2021). Multi-task MARL accommodates structural variations across tasks through entity-based architectures or training curricula Hu et al. (2021); Iqbal et al. (2021), but generally transfers knowledge at the primitive-policy level and relies on online training or target-task adaptation. In contrast, our work learns reusable team-level coordination skills solely from offline source-task data for zero-shot generalization.

2.2 Multi-Agent Skill Discovery and Temporal Abstraction

Multi-agent skill discovery abstracts primitive-action sequences into reusable behaviors. Early methods, including HSD, MASD, and HMASD, learn latent skills online to facilitate exploration and coordination Yang et al. (2020); He et al. (2020); Yang et al. (2023). More recently, ODIS extracts task-invariant skills from offline multi-task data, HiSSD separates common and task-specific skills, and VO-MASD models subgroup coordination patterns. HyGen further refines general skills through online interactions, while BiKT transfers individual skills and team-level tactics using a

bi-level decision transformer Zhang et al. (2023); Liu et al. (2025); Chen et al. (2025); Zhang et al. (2024; 2025). However, these coordination skills remain predominantly latent, with their behavioral meanings typically interpreted post hoc. Language-guided methods such as LGSD provide semantic supervision but focus on single-agent skill discovery Rho et al. (2025). SGVD instead explicitly grounds team-level skills in recognizable coordination behaviors.

2.3 LLM-Assisted Programmatic Weak Supervision

Programmatic weak supervision encodes domain heuristics as Label Functions (LFs) that produce noisy labels for downstream learning Ratner et al. (2016; 2017). Recent methods such as ALFRED and DataSculpt employ LLM to facilitate LF construction Yu & Bach (2023); Guan et al. (2025), but mainly target static classification. SGVD instead uses an LLM to iteratively refine trajectory-aware LFs over multi-agent behavioral statistics, producing both semantic phase labels and transition boundaries. The resulting annotations supervise skill learning only during training, without requiring the LLM or LFs during execution.

3 Preliminary

3.1 Offline Multi-task Cooperative MARL

We consider a family of cooperative multi-agent tasks divided into a set of source tasks $\mathcal{M}_{\mathrm{src}} = \{\mathcal{M}_k\}_{k=1}^{K}$ and a set of unseen target tasks $\mathcal{M}_{\mathrm{tgt}}$. Each task $\mathcal{M}_k$ is modeled as a decentralized partially observable Markov decision process (Dec-POMDP) Oliehoek & Amato (2016), defined by $\mathcal{M}_k = \langle \mathcal{I}_k, \mathcal{S}_k, \{\mathcal{O}_k^i\}_{i=1}^{n_k}, \{\mathcal{A}_k^i\}_{i=1}^{n_k}, P_k, r_k, \gamma \rangle$, where $\mathcal{I}_k = \{1, \ldots, n_k\}$ denotes the set of agents, and the number of agents n_k may vary across tasks.

At timestep t, the environment has a global state $s_t \in \mathcal{S}_k$, and each agent i receives a local observation $o_t^i \in \mathcal{O}_k^i$. The local information available to agent i can be represented by its action-observation history $\tau_t^i = (o_1^i, a_1^i, \ldots, o_{t-1}^i, a_{t-1}^i, o_t^i)$. The joint action $\mathbf{a}_t = (a_t^1, \ldots, a_t^{n_k})$ induces the environment transition $s_{t+1} \sim P_k(\cdot \mid s_t, \mathbf{a}_t)$ and produces a shared reward $r_t = r_k(s_t, \mathbf{a}_t)$. The objective is to learn a joint policy $\boldsymbol{\pi}_k$ that maximizes the expected discounted return: $\boldsymbol{\pi}_k^* = \arg\max_{\boldsymbol{\pi}_k} \mathbb{E}_{\boldsymbol{\pi}_k, P_k}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$.

For each source task, the learner is given a static dataset $\mathcal{D}_k$ containing L_k trajectories. Each trajectory is represented as $\xi_m^k = \{(s_t, \mathbf{o}_t, \mathbf{a}_t, r_t)\}_{t=1}^{T_m}$, where $\mathbf{o}_t = (o_t^1, \ldots, o_t^{n_k})$ denotes the collection of agents' local observations, $m \in \{1, \ldots, L_k\}$ indexes the trajectories, T_m is the length of ξ_m^k. The task-specific and complete offline datasets are respectively defined as $\mathcal{D}_k = \{\xi_m^k\}_{m=1}^{L_k}, \mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$. No additional environment interaction is allowed during training.

The objective is to learn a cooperative policy from $\mathcal{D}$ that can be deployed directly to tasks in $\mathcal{M}_{\mathrm{tgt}}$ without target-task data, online interaction, or task-specific fine-tuning. Source and target tasks may differ in the numbers of agents, opponents, targets, and other entities.

3.2 Coordination Skills with Variable Duration

We use a team-level coordination skill c_t to represent the shared coordination intent of all agents at timestep t. Conditioned on this skill, agent i selects its primitive action according to $a_t^i \sim \pi_{\theta_l}\left(\cdot \mid o_t^i, c_t\right)$. Although c_t is shared by the team, agents may exhibit different primitive behaviors due to their distinct local observations.

Since a coordination intent may persist across multiple primitive timesteps, we partition a trajectory of T primitive transitions into J consecutive variable-duration coordination segments. Let t_j denote the starting timestep of the j-th segment and d_j its duration, measured by the number of primitive transitions. The j-th segment covers the decision timesteps $\sigma_j = \{t_j, \ldots, t_j + d_j - 1\}$ and reaches its terminal state at timestep $t_j + d_j$. Consecutive segments therefore satisfy $t_{j+1} = t_j + d_j$, with $\sum_{j=1}^{J} d_j = T$. Each segment σ_j is assigned a coordination skill $\mathbf{c}^{(j)}$, which remains fixed throughout the segment: $\mathbf{c}_t = \mathbf{c}^{(j)}$ for all $t \in \sigma_j$. To mark skill transitions, we define a boundary indicator b_t, where $b_t = 1$ if $t = t_j$ for some $j \in \{1, \ldots, J\}$, and $b_t = 0$ otherwise. Thus, each

coordination skill persists for d_j primitive transitions and is updated at the beginning of the next segment.

4 METHOD

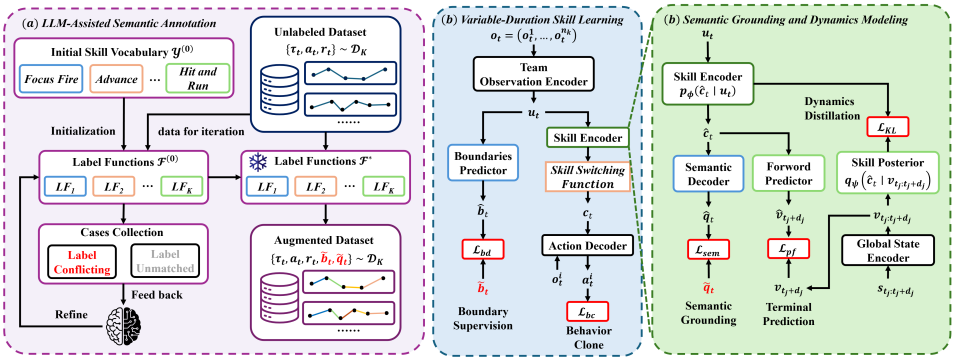

Figure 1: Overview of the SGVD framework. (a) An LLM iteratively refines LFs on a small sample trajectories, which are then used to annotate the full offline dataset with skill semantic labels and transition boundaries. (b) The framework infers coordination team skills from joint observations and switches them only at detected boundaries, enabling variable-duration decentralized execution. (c) We design three auxiliary tasks—semantic grounding, terminal prediction, and dynamics distillation—to enhance the learned skills' understanding of coordination semantics and awareness of environmental dynamics.

4.1 LLM-ASSISTED SEMANTIC ANNOTATION

In offline multi-task MARL, semantically meaningful coordination patterns are difficult to identify directly from original trajectories which consisting of global states, local observations, joint actions, and rewards. We therefore employ an LLM to assist in constructing deterministic label functions (LFs), which map interpretable trajectory statistics to weak semantic skill annotations.

For each task family, we first specify an initial semantic skill vocabulary $\mathcal{Y}^{(0)}$, containing representative coordination behaviors such as FOCUS FIRE, ADVANCE, and HIT AND RUN. Given the definition of each skill and the semantics of the available trajectory statistics, the LLM generates an executable label function LF_y for each skill $y \in \mathcal{Y}^{(0)}$. The resulting initial LF set is defined as $\mathcal{F}^{(0)} = \{\mathrm{LF}_y \mid y \in \mathcal{Y}^{(0)}\}$.

Each trajectory is partitioned into fixed, non-overlapping windows $\{\mathcal{W}_m\}_{m=1}^{M}$, from which a deterministic extractor Φ derives interpretable statistics $\mathbf{x}_m = \Phi(\mathcal{W}_m)$ on team movement, attacks, target selection, health changes, and spatial relations. The current LF set labels unique matches, while conflicting and unmatched windows guide the LLM to refine overlapping LFs or introduce recurring, behaviorally distinct skills. Each revision is validated on both failure cases and previously stable annotations, yielding $\mathcal{F}^{(0)} \to \mathcal{F}^{(1)} \to \cdots \to \mathcal{F}^\star$. Further details on the statistics, prompts, and refinement criteria are provided in Appendix A.

We then freeze $\mathcal{F}^\star$ and apply it to the complete offline dataset. Each window $\mathcal{W}_m$ is assigned a skill label $\tilde{y}_m$, which is propagated to its timesteps as $\tilde{y}_t = \tilde{y}_m$ for all $t \in \mathcal{W}_m$. Consecutive windows with the same label are merged into a single coordination segment, thereby allowing segment durations to vary with the underlying behavior. For each timestep, the boundary annotation is defined as $\tilde{b}_1 = 1$ and $\tilde{b}_t = \mathbb{I}[\tilde{y}_t \neq \tilde{y}_{t-1}]$ for $t > 1$, while the semantic target is defined as $\tilde{\mathbf{q}}_t = E_{\text{text}}(\tilde{y}_t)$. Here, E_{text} is a frozen pretrained Sentence-Transformer that maps each textual skill label to its semantic embedding. Accordingly, the augmented dataset is written as $\widetilde{\mathcal{D}} = \{(\tau_t, \tilde{b}_t, \tilde{\mathbf{q}}_t)\}_{t=1}^{T}$, where $\tilde{b}_t$ marks the beginning of a new coordination segment and $\tilde{\mathbf{q}}_t$ provides semantic supervision for the learned team-level skill.

4.2 VARIABLE-DURATION COORDINATION SKILL LEARNING

In this section, we use the LF-derived boundary annotations in $\widetilde{\mathcal{D}}$ to learn variable-duration coordination skills, whose durations are determined by coordination boundaries rather than a predefined skill horizon.

Given the collection of agents' local observations $\mathbf{o}_t = (o_t^1, \ldots, o_t^{n_k})$, we first encode each local observation and aggregate the resulting representations into a team-level representation: $\mathbf{e}_t^i = f_{\text{obs}}(o_t^i)$ and $\mathbf{u}_t = f_{\text{team}}(\{\mathbf{e}_t^i\}_{i=1}^{n_k})$. Following ODIS Zhang et al. (2023), f_{obs} employs a Transformer-based observation encoder to accommodate the variable numbers of agents and entities across tasks.

Based on $\mathbf{u}_t$, the boundary predictor estimates the probability that a new coordination segment begins: $\hat{b}_t = f_{\text{bp}}(\mathbf{u}_t) \in [0, 1]$. The boundary predictor is supervised by the LF-derived annotation $\tilde{b}_t$ through

$$\mathcal{L}_{\text{bnd}} = -\frac{1}{T-1} \sum_{t=2}^{T} \left[\tilde{b}_t \log \hat{b}_t + (1 - \tilde{b}_t) \log(1 - \hat{b}_t) \right]. \tag{1}$$

In parallel, the skill encoder generates a candidate coordination skill $\hat{\mathbf{c}}_t = f_{\text{skill}}(\mathbf{u}_t)$. We refer to $\hat{\mathbf{c}}_t$ as a candidate because it replaces the currently active coordination skill only when a transition boundary is identified. Specifically, the coordination skill is updated through a switching function:

$$\mathbf{c}_t = \begin{cases} \tilde{b}_t \hat{\mathbf{c}}_t + (1 - \tilde{b}_t)\mathbf{c}_{t-1}, & \text{during training,} \\ I_t \hat{\mathbf{c}}_t + (1 - I_t)\mathbf{c}_{t-1}, & \text{during execution,} \end{cases} \tag{2}$$

where $I_t = \mathbb{I}[\hat{b}_t > \delta]$ and δ is the boundary-detection threshold. Because skill switching occurs only at coordination boundaries, $\mathbf{c}_t$ remains unchanged within each segment, and its duration is determined by the interval between successive boundaries rather than by a predefined skill horizon.

Finally, an action decoder f_{act} parameterizes each agent's policy $\pi_{\text{act}}(\cdot \mid o_t^i, \mathbf{c}_t)$. The policy is trained on the offline trajectories using the behavior cloning objective:

$$\mathcal{L}_{\text{BC}} = -\frac{1}{T n_k} \sum_{t=1}^{T} \sum_{i=1}^{n_k} \log \pi_{\text{act}} \left(a_t^i \mid o_t^i, \mathbf{c}_t \right). \tag{3}$$

During execution, the learned policy selects each agent's action conditioned on its local observation and the active coordination skill.

4.3 SEMANTIC GROUNDING AND DYNAMICS MODELING

The preceding section enables variable-duration coordination skills. In this section, we use the LF-derived semantic annotations to ground these skills in recognizable team-level coordination behaviors, and we introduce dynamics modeling to enhance the coordination skill representations.

Semantic Grounding

We model a semantic decoder f_{sem} to projects the candidate coordination skill $\hat{\mathbf{c}}_t$ into the same embedding space as the semantic target $\tilde{\mathbf{q}}_t$, yielding $\hat{\mathbf{q}}_t = f_{\text{sem}}(\hat{\mathbf{c}}_t)$. We then align the decoded representation with its semantic target using the cosine distance:

$$\mathcal{L}_{\text{sem}} = \frac{1}{|\mathcal{V}_{\text{sem}}|} \sum_{t \in \mathcal{V}_{\text{sem}}} \left[1 - \cos\left(\hat{\mathbf{q}}_t, \tilde{\mathbf{q}}_t \right) \right], \tag{4}$$

where $\mathcal{V}_{\text{sem}}$ denotes the set of timesteps with valid semantic annotations. Candidate skills assigned the same LF skill label share the same semantic target. Therefore, minimizing $\mathcal{L}_{\text{sem}}$ grounds semantically equivalent coordination behaviors from different tasks in a common semantic space, establishing an explicit correspondence between the learned coordination skills and their behavioral meanings.

Dynamics Modeling

We combine segment-conditioned posterior alignment with terminal prediction. An entity-wise, permutation-invariant global-state encoder Zhang et al. (2023) produces $\mathbf{v}_{t_j:t_j+d_j} =$

$f_{\text{state}}(\mathbf{s}_{t_j:t_j+d_j})$. The training-only posterior $q_\psi(\widehat{\mathbf{c}}_{t_j} \mid \mathbf{v}_{t_j:t_j+d_j})$ conditions on the complete segment, whereas the prior $p_\phi(\widehat{\mathbf{c}}_{t_j} \mid \mathbf{u}_{t_j})$ uses the current observation representation. Both are diagonal Gaussians, aligned through:

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}\left[q_\psi\left(\hat{\mathbf{c}}_t \mid \mathbf{v}_{t_j:t_j+d_j}\right) \middle\| p_\phi\left(\hat{\mathbf{c}}_t \mid \mathbf{u}_{t_j}\right)\right]. \tag{5}$$

Since the posterior observes the complete state transition whereas the skill encoder uses only the current observation representation, this objective distills segment-level dynamics into the candidate skill available during execution.

We further use the terminal global-state representation as a direct prediction target for the candidate coordination skill. Specifically, a forward predictor f_{pf} estimates $\hat{\mathbf{v}}_{t_j+d_j} = f_{\text{pf}}(\hat{\mathbf{c}}_t)$, and is trained through

$$\mathcal{L}_{\text{pf}} = \left\|\hat{\mathbf{v}}_{t_j+d_j} - \text{sg}\left(\mathbf{v}_{t_j+d_j}\right)\right\|_2^2, \tag{6}$$

Here, sg stops gradients through the target. The global-state encoder, posterior, and forward predictor are used only during training.

4.4 Overall Training Objective

Given the LF-augmented offline dataset, we jointly optimize all trainable components of our framework. Specifically, $\mathcal{L}_{\text{bnd}}$ supervises the boundary predictor; $\mathcal{L}_{\text{sem}}$ grounds the candidate coordination skill in the predefined skill semantics; $\mathcal{L}_{\text{dyn}}$ incorporates segment-level dynamics through posterior–prior alignment and terminal-state prediction; and $\mathcal{L}_{\text{BC}}$ trains the skill-conditioned low-level policy. We define the dynamics modeling loss as $\mathcal{L}_{\text{dyn}} = \mathcal{L}_{\text{KL}} + \beta_{\text{pf}}\mathcal{L}_{\text{pf}}$. The overall objective is

$$\mathcal{L}_{\text{total}} = \lambda_{\text{bnd}}\mathcal{L}_{\text{bnd}} + \lambda_{\text{sem}}\mathcal{L}_{\text{sem}} + \lambda_{\text{dyn}}\mathcal{L}_{\text{dyn}} + \lambda_{\text{BC}}\mathcal{L}_{\text{BC}}, \tag{7}$$

where β_{pf} balances the two dynamics objectives, while the λ coefficients balance the main training objectives. The complete training procedure is provided in Appendix C.

5 Experiment

In this section, we conduct experiments to answer three questions: **Q1:** How does our method compare with existing offline multi-task MARL methods in policy performance and zero-shot generalization to unseen tasks? **Q2:** How do the key components contribute to its overall performance? **Q3:** Do the learned coordination skills exhibit recognizable behavioral semantics and variable-duration temporal structures across tasks?

5.1 Baselines

We compare against representative offline multi-task MARL baselines. **ODIS** Zhang et al. (2023) discovers task-general coordination skills from multi-source offline data for decentralized action generation. **HiSSD** Liu et al. (2025) hierarchically decomposes cooperative behaviors into task-shared common skills and task-specific individual skills. **BiKT** Zhang et al. (2025) transfers reusable knowledge through individual skill embeddings and team-level tactics inferred by a bi-level decision transformer.

5.2 Benchmark

We evaluate all methods on the StarCraft Multi-Agent Challenge (SMAC) benchmark Samvelyan et al. (2019), following the offline multi-task evaluation protocol used in ODIS Zhang et al. (2023), HiSSD Liu et al. (2025), and BiKT Zhang et al. (2025). We consider three task sets: *marine-easy*, *marine-hard*, and *stalker-zealot*, covering homogeneous and heterogeneous battles with varying team sizes and unit compositions. For each task set, agents are trained on offline trajectories from three source tasks and evaluated on both source and unseen tasks without further interaction, adaptation, or fine-tuning. Detailed task configurations and evaluation splits are provided in Appendix D.

Table 1: Average test win rates of the best policies over five random seeds in the task set *Marine-Hard* with different data qualities.

Tasks	Expert				Medium			
	ODIS	HiSSD	BiKT	SGVD	ODIS	HiSSD	BiKT	SGVD
Source Tasks								
3m	98.4±2.7	99.5±0.3	**99.8±0.2**	99.6±0.4	85.9±10.5	62.7±5.7	**86.1±9.6**	66.4±6.2
5m6m	53.9±5.1	66.1±7.0	**80.3±4.5**	68.2±5.3	22.7±7.1	26.4±3.8	36.9±4.5	**37.2±3.4**
9m10m	80.4±8.7	95.5±2.7	99.4±0.4	**96.9±2.2**	**78.1±3.8**	73.9±2.3	72.5±5.5	74.7±2.6
Unseen Tasks								
4m	95.3±3.5	99.2±1.2	99.3±0.1	**99.4±1.4**	61.7±17.7	77.3±10.2	**91.3±6.2**	80.1±6.5
5m	89.1±10.0	99.2±1.2	**99.9±0.1**	**99.9±0.1**	85.9±11.8	88.4±8.4	94.4±5.4	**95.8±4.2**
10m	93.8±2.2	98.4±0.8	99.4±0.1	**99.6±0.2**	61.3±11.3	**98.0±0.3**	96.3±2.5	92.4±6.3
12m	58.6±11.8	75.5±19.7	**99.0±0.2**	89.5±10.2	35.9±8.1	86.4±6.0	**92.5±4.0**	88.0±4.1
7m8m	25.0±15.1	35.3±9.8	**68.0±9.9**	42.4±8.5	**28.1±22.0**	14.2±10.1	4.3±2.5	15.9±8.0
8m9m	19.6±6.0	47.0±6.2	50.0±6.2	**57.1±4.3**	4.7±2.7	15.3±2.8	17.0±1.9	**17.6±3.7**
10m11m	42.4±7.2	86.3±14.6	90.6±1.1	**91.1±4.2**	29.7±15.4	43.6±4.6	25.6±4.6	**45.5±5.4**
10m12m	1.6±1.6	14.5±9.1	14.6±3.5	**15.4±4.3**	1.6±1.6	0.6±0.5	2.2±1.3	**2.4±1.8**
13m15m	2.3±2.6	1.3±2.5	4.2±2.1	**5.2±3.8**	1.6±1.6	1.4±2.4	2.5±1.3	**2.8±1.6**
Tasks	Medium-Expert				Medium-Replay			
	ODIS	HiSSD	BiKT	SGVD	ODIS	HiSSD	BiKT	SGVD
Source Tasks								
3m	73.6±22.0	86.6±3.7	**99.4±1.3**	92.2±5.7	83.6±14.0	78.8±5.3	78.7±6.3	**84.5±6.2**
5m6m	9.4±2.2	41.9±9.7	49.4±6.0	**50.2±8.4**	16.6±4.7	25.3±10.3	28.1±3.1	**28.6±4.0**
9m10m	31.3±14.5	83.6±6.9	58.3±2.4	**85.2±7.2**	34.4±8.0	45.8±3.9	47.6±7.0	**48.6±4.5**
Unseen Tasks								
4m	82.8±13.5	91.1±6.1	**98.1±0.2**	96.4±4.2	55.6±14.5	77.3±1.9	94.4±2.8	**95.6±3.4**
5m	82.8±17.7	98.3±1.8	98.8±0.1	**98.9±1.2**	96.1±4.1	88.1±13.4	97.9±0.2	**98.1±5.4**
10m	82.8±16.8	96.4±2.1	96.5±1.2	**97.7±3.4**	84.4±15.1	94.7±2.6	96.7±1.5	**97.6±4.1**
12m	81.3±20.6	88.4±11.8	95.6±0.3	**96.5±3.7**	84.4±6.6	90.3±3.6	92.7±2.6	**93.6±2.4**
7m8m	15.6±4.4	30.5±10.4	40.6±4.2	**42.6±6.2**	9.4±2.2	21.7±4.7	19.3±6.4	**22.6±4.3**
8m9m	10.9±4.7	35.2±18.3	22.4±3.6	**40.7±12.4**	11.7±8.7	14.5±4.0	10.5±4.0	**15.6±4.3**
10m11m	33.6±8.9	54.7±6.8	42.3±3.2	**62.4±9.3**	35.9±5.2	42.5±4.4	43.8±6.6	**47.2±5.3**
10m12m	1.6±1.6	2.5±1.0	2.3±1.3	**5.7±2.5**	2.3±1.4	0.5±0.3	0.6±0.2	**3.2±2.6**
13m15m	2.3±2.6	5.2±3.7	2.8±1.4	**5.5±2.4**	2.4±1.4	3.6±2.1	2.4±1.2	**4.0±2.8**

5.3 Offline Datasets

We use the offline datasets constructed by ODIS Zhang et al. (2023), which were collected from the source tasks using QMIX Rashid et al. (2018). Following D4RL Fu et al. (2020), the datasets contain four quality levels: *expert*, *medium*, *medium-expert*, and *medium-replay*, representing different levels of policy quality and behavioral diversity. All methods are trained using the same offline trajectories without additional environment interaction. Further details on data collection, composition, and statistics are reported in Appendix D.

5.4 Performance(Q1)

Table 1 presents the average test win rates on *Marine-Hard* under different offline data qualities. Results on *Marine-Easy* and *Stalker-Zealot* are provided in Appendix B. Overall, SGVD method achieves better performance across most of data qualities on both source and unseen tasks. The advantage is particularly pronounced on Medium-Expert and Medium-Replay datasets, whose trajectories contain mixed-quality and heterogeneous coordination behaviors. We attribute this robustness to semantic grounding and variable-duration modeling, which jointly improve the semantic consistency and temporal coherence of the learned coordination skills, yielding higher-quality and

more transferable skills that support stronger zero-shot generalization to unseen tasks. Moreover, the improved performance on unseen tasks with different team sizes and asymmetric configurations indicates that the learned skills encode transferable coordination knowledge rather than task-specific behavior patterns.

Table 2: Ablation studies on *Marine-Hard* under different offline data qualities. Each entry reports the mean and standard deviation across the corresponding source or unseen tasks, computed from task-level win rates averaged over five random seeds. BC-best is included as a skill-free reference baseline. We present only these aggregated results in this table; complete abaltion results for all ablated variants are provided in Appendix E.

Data Quality	BC-best	w/o Semantic Grounding	Fixed Duration	w/o Terminal Prediction	w/o Dynamics Distillation	SGVD
			Source Tasks			
Expert	81.1±21.8	87.3±15.5	86.6±15.7	87.4±14.7	87.2±15.2	**88.2±14.2**
Medium	50.4±20.1	56.8±15.8	53.4±15.6	57.6±15.1	58.1±15.5	**59.4±16.1**
Medium-Expert	41.7±18.5	73.2±17.9	71.8±19.4	73.3±17.8	73.9±17.5	**75.9±18.4**
Medium-Replay	46.5±24.7	48.8±24.7	44.7±25.9	51.5±23.1	50.5±23.6	**53.9±23.1**
			Unseen Tasks			
Expert	46.8±36.8	63.9±34.9	61.6±35.6	64.3±35.2	64.7±35.0	**66.6±35.7**
Medium	25.6±26.8	45.9±37.8	42.6±37.6	46.2±37.8	46.4±37.7	**48.9±37.9**
Medium-Expert	38.8±33.1	57.3±36.5	52.4±37.7	57.9±36.6	57.7±37.0	**60.7±36.8**
Medium-Replay	41.2±33.7	49.9±40.3	47.6±41.3	51.4±40.6	51.0±40.5	**53.1±40.4**

5.5 Ablation(Q2)

To evaluate the contribution of each component, we compare the full model with four ablated variants and include BC-best as a reference baseline. **w/o Semantic Grounding** removes the semantic decoder f_{sem} and $\mathcal{L}_{\text{sem}}$, such that the learned coordination skills are no longer explicitly aligned with the predefined skill semantics. **Fixed Duration** removes the boundary predictor and $\mathcal{L}_{\text{bnd}}$, and instead resamples the team skill every $H = 3$ timesteps during both training and execution. **w/o Terminal Prediction** removes the forward predictor f_{pf} and $\mathcal{L}_{\text{pf}}$, eliminating the explicit supervision that encourages coordination skills to capture segment outcomes. **w/o Dynamics Distillation** removes the training-only segment posterior q_ψ and $\mathcal{L}_{\text{KL}}$, thereby preventing the execution-time skill encoder from receiving segment-transition guidance. **BC-best** serves as a skill-free imitation baseline, allowing us to examine whether the proposed skill-learning framework provides benefits beyond direct behavior cloning. **SGVD** retains all the proposed components.

As shown in Table 2, the full model consistently outperforms all ablated variants on both source and unseen tasks. Fixed Duration causes the most pronounced degradation because resampling team skills every three timesteps imposes artificial boundaries that may interrupt an ongoing coordination behavior or delay adaptation when the team intention changes. This temporal mismatch is particularly harmful for the heterogeneous behaviors in *medium-replay* data, where inaccurate skill abstraction can even make the hierarchical policy less effective than skill-free BC. Semantic grounding improves cross-task transfer by aligning behaviorally similar trajectory segments with shared and task-invariant skill semantics. Terminal prediction and dynamics distillation further enhance the learned skill representations. These results confirm the complementary contribution of all proposed components.

6 Visualization(Q3)

We examine the semantic structure of the learned skills across tasks and their temporal evolution during execution through two complementary visualizations.

6.1 Skill Representation

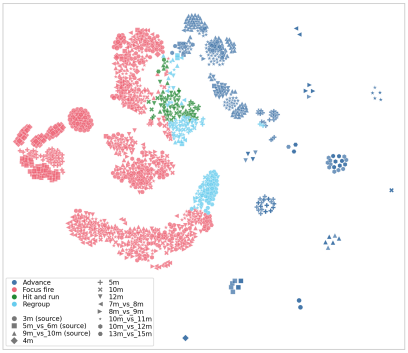

Figure 2: Skill representation visualization

As show in Figure 2. To analyze the semantic structure and cross-task transferability of the learned skills, we visualize the semantic representations produced by the model trained on the *medium-replay* dataset at 35k training steps. We evaluate the trained policy for 50 episodes on each source and unseen task without parameter updates. During evaluation, skill boundaries are determined by predicted boundary probabilities rather than ground-truth annotations. At every predicted skill boundary, we record the active skill embedding and its predicted semantic category. We jointly visualize the skill representations from all source and unseen tasks using t-SNE. Colors denote predicted semantic skill categories, while marker shapes distinguish different tasks. Representations associated with the same semantic skill tend to form consistent clusters across tasks, suggesting that the learned skill space captures task-invariant semantic structures and generalizes to unseen tasks.

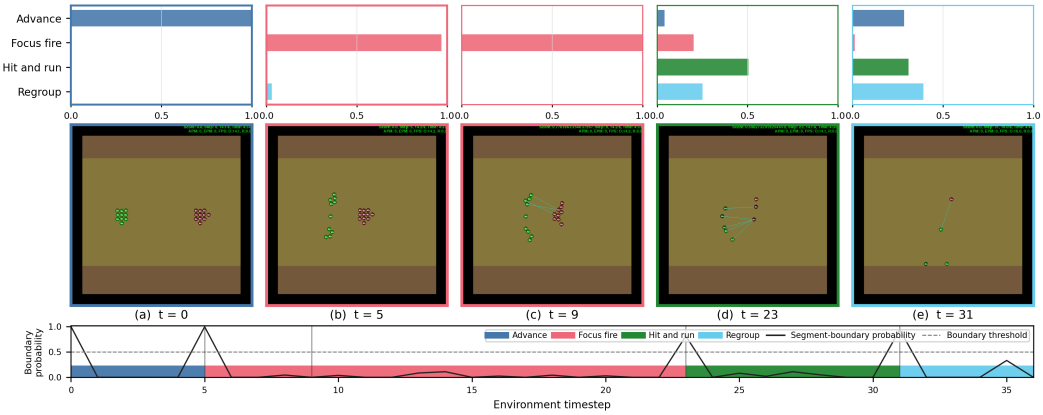

Figure 3: A practical demonstration of the transition of coordination intention

6.2 Transition of Coordination Intention

As show in Figure 3 we visualize coordination-skill evolution through predicted skill distributions, corresponding SMAC battle states, and temporal boundary probabilities. The active skill is maintained below the 0.5 boundary threshold and switched when the threshold is exceeded. This figure demonstrates that the learned coordination skills are temporally persistent and adaptively switched according to changes in the battle state, providing an interpretable link between boundary prediction, skill selection, and coordinated team behavior.

7 Conclusion

We present SGVD for learning semantically grounded and variable-duration coordination skills from offline multi-task data. An LLM-assisted LF pipeline provides semantic labels and transition boundaries for skill grounding and boundary-guided switching, while dynamics modeling captures segment-level transitions. During execution, SGVD requires neither the annotation pipeline nor training-only global information. Experiments across SMAC task sets and four data qualities demonstrate competitive zero-shot performance, while ablations validate the effectiveness of its key components.while visualizations reveal the representation and coordination intention transition of skill.

AI USE STATEMENT

We used ChatGPT to assist with language editing, refinement of methodological and experimental descriptions, LaTeX formatting, organization and summarization of experimental results, review of tool-use trajectories, and implementation of repetitive coding tasks. All AI-assisted content was carefully reviewed and revised by the authors. We verified organized results against the original experimental outputs, checked trajectory analyses against the source traces, and validated all assisted code before use. The use of LLMs as evaluated systems or automated judges is separately described in the methodology and experimental setup. The authors take full responsibility for the accuracy, originality, and integrity of the final manuscript.

REFERENCES

Jiayu Chen, Bhargav Ganguly, Tian Lan, and Vaneet Aggarwal. Variational offline multi-agent skill discovery. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2025.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

Shangding Gu, Jakub Grudzien Kuba, Yuanpei Chen, Yali Du, Long Yang, Alois Knoll, and Yaodong Yang. Safe multi-agent reinforcement learning for multi-robot control. *Artificial Intelligence*, 319:103905, 2023. doi: 10.1016/j.artint.2023.103905.

Naiqing Guan, Kaiwen Chen, and Nick Koudas. DataSculpt: Cost-efficient label function design via prompting large language models. In *Proceedings of the 28th International Conference on Extending Database Technology*, pp. 226–232, 2025. doi: 10.48786/edbt.2025.18.

Shuncheng He, Jianzhun Shao, and Xiangyang Ji. Skill discovery of coordination in multi-agent reinforcement learning. *arXiv preprint arXiv:2006.04021*, 2020.

Siyi Hu, Fengda Zhu, Xiaojun Chang, and Xiaodan Liang. UPDeT: Universal multi-agent reinforcement learning via policy decoupling with transformers. In *International Conference on Learning Representations*, 2021.

Shariq Iqbal, Christian A. Schroeder de Witt, Bei Peng, Wendelin Böhmer, Shimon Whiteson, and Fei Sha. Randomized entity-wise factorization for multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2021.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Sicong Liu, Yang Shu, Chenjuan Guo, and Bin Yang. Learning generalizable skills from offline multi-task data for multi-agent cooperation. In *International Conference on Learning Representations*, 2025.

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

Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Springer, 2016. doi: 10.1007/978-3-319-28929-8.

James Orr and Ayan Dutta. Multi-agent deep reinforcement learning for multi-robot applications: A survey. *Sensors*, 23(7):3625, 2023. doi: 10.3390/s23073625.

Praveen Palanisamy. Multi-agent connected autonomous driving using deep reinforcement learning. In *Proceedings of the International Joint Conference on Neural Networks*, 2020. doi: 10.1109/IJCNN48605.2020.9207663.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4295–4304. PMLR, 2018.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282, 2017.

Alexander J. Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, 2016.

Seungeun Rho, Laura Smith, Tianyu Li, Sergey Levine, Xue Bin Peng, and Sehoon Ha. Language guided skill discovery. In *The Thirteenth International Conference on Learning Representations*, 2025.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 2186–2188, 2019.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354, 2019. doi: 10.1038/s41586-019-1724-z.

Jiachen Yang, Igor Borovikov, and Hongyuan Zha. Hierarchical cooperative multi-agent reinforcement learning with skill discovery. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 1566–1574, 2020.

Mingyu Yang, Yaodong Yang, Zhenbo Lu, Wengang Zhou, and Houqiang Li. Hierarchical multi-agent skill discovery. In *Advances in Neural Information Processing Systems*, 2023.

Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

Peilin Yu and Stephen H. Bach. Alfred: A system for prompted weak supervision. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 479–488. Association for Computational Linguistics, 2023.

Fuxiang Zhang, Chengxing Jia, Yi-Chen Li, Lei Yuan, Yang Yu, and Zongzhang Zhang. Discovering generalizable multi-agent coordination skills from multi-task offline data. In *International Conference on Learning Representations*, 2023.

Junkai Zhang, Jinmin He, Yifan Zhang, Yifan Zang, Ning Xu, and Jian Cheng. Bi-level knowledge transfer for multi-task multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Mingliang Zhang, Sichang Su, Chengyang He, and Guillaume Sartoretti. Hybrid training for enhanced multi-task generalization in multi-agent reinforcement learning. *arXiv preprint arXiv:2408.13567*, 2024.

Ming Zhou, Jun Luo, Julian Villella, Yaodong Yang, David Rusu, Jiayu Miao, Weinan Zhang, Montgomery Alban, Iman Fadakar, Zheng Chen, Chongxi Huang, Ying Wen, Kimia Hassanzadeh, Daniel Graves, Zhengbang Zhu, Yihan Ni, Nhat Nguyen, Mohamed Elsayed, Haitham Ammar, Alexander Cowen-Rivers, Sanjeevan Ahilan, Zheng Tian, Daniel Palenicek, Kasra

Rezaee, Peyman Yadmellat, Kun Shao, Dong Chen, Baokuan Zhang, Hongbo Zhang, Jianye Hao, Wulong Liu, and Jun Wang. SMARTS: An open-source scalable multi-agent RL training school for autonomous driving. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 264–285. PMLR, 2021. URL https://proceedings.mlr.press/v155/zhou21a.html.

A LLM-ASSISTED SEMANTIC ANNOTATION PROCESS

This appendix describes the annotation pipeline used to produce the weak semantic and temporal supervision introduced in the main text. The LLM is used only during label-function (LF) development: it reads skill definitions, statistic semantics, and a small set of diagnostic windows, and returns executable deterministic rules. Once the rules have been finalized, both the LLM and the refinement set are removed from the learning pipeline. The frozen LFs are then applied offline to the complete source-task dataset. Thus, the LLM neither labels the full dataset directly nor participates in policy training or execution.

A.1 SEMANTIC SKILL VOCABULARIES AND BEHAVIORAL STATISTICS

Initial skill vocabularies. We define a compact initial vocabulary separately for the two unit families because the coordination patterns that can be expressed by homogeneous ranged units differ from those of a heterogeneous melee–ranged team. Table 3 lists the initial labels and their operational meanings. These labels are intended to describe team-level coordination intentions rather than the primitive action of any single agent. The vocabulary is not required to be exhaustive: as described below, a new label can be introduced when unmatched windows repeatedly exhibit a coherent behavior that is not captured by the current vocabulary.

Task family	Semantic skill	Behavioral interpretation
Marine	ADVANCE	The team forms up and approaches the enemy before sustained combat; movement is prevalent and little enemy damage is inflicted.
	FOCUS FIRE	Multiple attacking agents consistently select the same target, typically causing a marked health reduction or a kill.
	HIT AND RUN	Attacks are interleaved with movement while the team maintains or increases its distance from the enemy.
	GROUPING	Agents reduce or maintain a small team spread to restore a coordinated formation, usually with less sustained fire than in FOCUS FIRE.
	DISPERSE LINEUP	The team increases its spatial spread to form a wider firing line or reduce local crowding.
Stalker–Zealot	ADVANCE	The heterogeneous team approaches the enemy while preserving a coordinated formation.
	FOCUS FIRE	Concurrent attackers concentrate their fire on a common enemy target.
	HIT AND RUN	Ranged pressure is interleaved with repositioning so that Stalkers maintain or increase their distance from the enemy.
	GROUPING	Stalkers and Zealots remain mutually coordinated or regroup, reflected by stable/decreasing spread and high ally visibility.
	LAYERED ENGAGEMENT	Zealots occupy the close-range frontline while Stalkers contribute ranged attacks from behind.

Table 3: Initial semantic skill vocabularies. The names shown here are converted to natural-language phrases before being embedded for semantic supervision.

Window construction. For LF development and full-dataset annotation, each valid episode is divided into consecutive, non-overlapping windows of $H_{\text{ann}} = 3$ primitive transitions. The last window is retained whenever it contains a valid transition. Fixed windows provide a uniform input unit for computing robust short-horizon behavioral evidence; they do not impose a fixed execution duration. After labeling, adjacent windows with the same semantic label are merged, so a resulting coordination segment may span any positive multiple of H_{ann} (or include a shorter terminal window).We further examine the effect of the annotation-window length H_{ann} through the ablation study reported in Table 14.

Behavioral statistic extractor. For a window W_m, the deterministic extractor Φ maps the stored global states, local observations, joint actions, action availability, and validity mask to a statistic vector $\mathbf{x}_m = \Phi(W_m)$. Table 4 summarizes the statistics exposed to the LLM. Ratios are normalized by the number of valid decisions of living agents, health losses are normalized by the corresponding effective health at the beginning of the window, and geometric quantities are computed only over living units. These normalizations make the rules less sensitive to the number of agents and allow the same LF structure to be used across maps within a task family.

Statistic category	Representative statistics	Behavioral evidence
Action patterns	Attack and movement ratios; target consistency	Whether the team attacks, repositions, or concentrates fire on a common target.
Combat outcomes	Normalized health variation; unit deaths	Whether the team inflicts or receives meaningful damage within the window.
Spatial relations	Enemy-distance change; team-spread change	Whether the team approaches, retreats, disperses, or regroups.
Unit-type relations	Type-specific attack and distance statistics	Role-dependent coordination between Stalkers and Zealots in heterogeneous tasks.

Table 4: Representative behavioral statistics supplied to the label functions. The predefined extractor computes all statistics deterministically from the original trajectory windows.

For example, the LF for FOCUS FIRE jointly considers the attack ratio, same-target ratio, and enemy health loss, whereas the LF for HIT AND RUN considers attack and movement ratios together with the change in enemy distance. For heterogeneous teams, the LF for LAYERED ENGAGEMENT additionally uses the frontline gap and type-specific attack ratios. Each LF combines multiple statistics to distinguish behaviors with superficially similar action frequencies.

A.2 LLM-ASSISTED LF CONSTRUCTION AND ITERATIVE REFINEMENT

Executable LF contract. At refinement round r, let $\mathcal{Y}^{(r)}$ denote the current semantic skill vocabulary. Consistent with the LF sequence defined in the main text, the corresponding LF set is

$$\mathcal{F}^{(r)} = \left\{ \mathrm{LF}_y^{(r)} \mid y \in \mathcal{Y}^{(r)} \right\}, \tag{8}$$

where $\mathcal{Y}^{(0)}$ and $\mathcal{F}^{(0)}$ are the initial vocabulary and LF set, respectively. For every current label $y \in \mathcal{Y}^{(r)}$, $\mathrm{LF}_y^{(r)}$ is a deterministic Python function

$$\mathrm{LF}_y^{(r)}(\mathbf{x}_m) \to (z_{m,y}, s_{m,y}, e_{m,y}), \tag{9}$$

where $z_{m,y} \in \{0,1\}$ is a match decision, $s_{m,y} \in [0,1]$ is a diagnostic confidence score, and $e_{m,y}$ is a textual explanation of the matched evidence. Only $z_{m,y}$ is used to determine the weak label. Each generated program is self-contained, deterministic, and restricted to the statistics and window metadata described above; it cannot call the LLM, access the environment, or use policy returns. Thresholds are represented by named constants or explicit predicates so that revisions remain inspectable.

Given the matched-label subset

$$\mathcal{Y}_{m,+}^{(r)} = \{y \in \mathcal{Y}^{(r)} : z_{m,y} = 1\}, \tag{10}$$

a window is *uniquely labeled* when $|\mathcal{Y}_{m,+}^{(r)}| = 1$, an *abstention* when $|\mathcal{Y}_{m,+}^{(r)}| = 0$, and a *conflict* when $|\mathcal{Y}_{m,+}^{(r)}| > 1$. Both failure types are retained during development: conflicts expose overlapping behavioral definitions, while abstentions reveal missing coverage or thresholds that are too restrictive.

Prompt contents. After the semantic descriptions have been generated, the LF-construction prompt contains (i) a short description of the SMAC environment and task family, (ii) the current skill names and their LLM-generated natural-language behavioral interpretations, (iii) the complete

predefined statistic schema and its semantics, (iv) the executable LF interface, and (v) a seed implementation illustrating the required code organization. The LLM is instructed to use interpretable conjunctions of positive and negative evidence and to balance coverage with exclusivity. It must return a complete executable file containing one LF per skill, a dispatcher, human-readable evidence descriptions, and a change log.

At refinement round r, the prompt additionally includes the current LF source, aggregate diagnostic rates, and compact records of selected failure windows. Each record contains its map, temporal indices, statistic vector, and current matched labels. For a conflict, the LLM is asked to make the smallest targeted change that leaves one semantically appropriate match. For an abstention, it is asked to cover a recurring pattern without broadening an existing rule so far that it creates new conflicts. Revisions therefore modify explicit predicates or thresholds rather than assigning labels to individual examples.

Iterative refinement and validation. Let $N^{(r)}_{\text{single}}$, $N^{(r)}_{\text{conflict}}$, and $N^{(r)}_{\text{abstain}}$ denote the three outcome counts on the LF-development sample. We monitor

$$R^{(r)}_{\text{single}} = N^{(r)}_{\text{single}}/N, \tag{11}$$

$$R^{(r)}_{\text{conflict}} = N^{(r)}_{\text{conflict}}/N, \tag{12}$$

$$R^{(r)}_{\text{abstain}} = N^{(r)}_{\text{abstain}}/N. \tag{13}$$

Failure cases are grouped by their conflicting label set or by recurring regions of the statistic space. A representative group is supplied to the LLM, and the proposed program is first checked for syntax and interface validity. It is then re-evaluated on the entire development sample, not only on the prompted cases. We accept a revision only when it resolves the targeted pattern without materially degrading previously stable unique matches. This replay check prevents local edits from silently shifting unrelated skill definitions. The process continues until the remaining cases are sparse or behaviorally ambiguous, or until the preset refinement budget is reached.

The vocabulary may be expanded during this process. Specifically, we consider a new skill only when a sizeable group of abstained windows recurs across trajectories, is internally consistent in its behavioral statistics, and cannot be expressed as a minor variant of an existing label. The LLM must provide a name, an operational definition, and an LF based on observable evidence. This conservative criterion avoids creating a separate semantic category for isolated outliers. All accepted versions are checkpointed, making the final LF set $\mathcal{F}^\star$ and its revision history reproducible.

A.3 Full-Dataset Annotation and Supervision Generation

After refinement, we freeze the LF set and apply it once to every valid window in the complete offline source-task dataset. Let $\tilde{y}_m$ be the unique LF label of window W_m. Windows for which the final LFs abstain or conflict receive dedicated diagnostic identifiers and are masked out of semantic supervision rather than being resolved by an additional LLM query. Consequently, annotation cost is independent of the number of full-dataset windows, and uncertain cases do not introduce forced semantic targets.

For each valid window, its label is propagated to the primitive timesteps that it covers. Consecutive valid windows with identical labels are merged into one coordination segment. Writing $m(t)$ for the window containing timestep t, the timestep-level label and boundary are

$$\tilde{y}_t = \tilde{y}_{m(t)}, \qquad \tilde{b}_t = \begin{cases} 1, & t = 1, \\ \mathbb{I}[\tilde{y}_t \neq \tilde{y}_{t-1}], & t > 1. \end{cases} \tag{14}$$

For abstention and conflict identifiers, a validity mask excludes the corresponding timestep from $\mathcal{L}_{\text{sem}}$ and prevents either identifier from being learned as a semantic skill category. The same validity information is used when constructing the effective boundary-training mask, so diagnostic identifiers are not themselves treated as coordination skills.

Finally, each valid label is converted from its identifier (e.g., `HIT_AND_RUN`) to a natural-language phrase and encoded by the frozen text encoder E_{text}:

$$\tilde{\mathbf{q}}_t = E_{\text{text}}(\text{text}(\tilde{y}_t)). \tag{15}$$

The augmented trajectory therefore stores the semantic target $\tilde{\mathbf{q}}_t$, the boundary target $\tilde{b}_t$, and their validity masks alongside the original state, observation, action, reward, and availability fields. The semantic target supervises the decoder in Eq. (4), while the boundary target supervises Eq. (1) and controls the training-time switch in Eq. (2). The symbolic skill identifier is retained only for diagnostics and visualization. Neither it nor the text encoder, LFs, global state, or LLM is required during decentralized execution.

B ADDITIONAL RESULTS

The main text reports the complete comparison on MARINE-HARD. To further evaluate the generality of the learned coordination skills, Tables 5 and 6 present the task-wise results on MARINE-EASY and STALKER-ZEALOT, respectively, under all four offline data qualities.

Table 5: Average test win rates of the best policies over five random seeds in the MARINE-EASY task set under different offline data qualities.

Tasks	Expert				Medium			
	ODIS	HiSSD	BiKT	SGVD	ODIS	HiSSD	BiKT	SGVD
Source Tasks								
3m	97.7±2.6	99.5±8.1	99.4±1.3	**99.6±1.2**	57.8±9.2	74.7±14.6	87.2±4.7	**79.6±8.7**
5m	95.3±5.2	**99.9±0.0**	**99.9±0.0**	98.9±1.2	82.8±5.2	81.6±10.8	74.2±5.9	**84.5±9.1**
10m	88.3±20.3	95.2±8.4	**99.9±0.0**	96.2±7.4	71.9±6.6	84.8±8.6	82.1±7.2	**86.7±7.4**
Unseen Tasks								
4m	90.6±7.0	94.4±2.9	**96.9±1.5**	94.5±3.3	63.3±16.1	74.5±15.5	75.5±8.6	**76.8±1.5**
6m	79.7±17.5	99.7±0.3	**99.7±0.1**	99.6±0.4	89.8±17.6	88.0±10.0	83.0±4.7	**89.4±9.8**
7m	72.7±16.9	99.1±0.7	**99.7±0.1**	98.8±1.2	96.1±1.4	97.3±2.3	89.9±0.0	98.4±1.2
8m	80.9±14.4	99.8±0.3	99.1±0.1	**99.8±0.1**	97.7±2.6	93.8±5.2	**98.9±1.2**	94.8±4.3
9m	99.2±1.4	**99.9±0.0**	**99.9±0.0**	99.5±0.4	87.5±2.2	75.2±15.5	88.9±11.8	**89.9±7.8**
11m	83.6±12.4	99.2±0.8	99.3±1.0	**99.4±0.6**	64.7±3.1	62.0±21.8	68.2±4.7	**74.4±11.4**
12m	70.3±30.2	99.7±1.1	99.6±1.0	**99.8±0.4**	41.4±6.0	55.5±25.7	49.7±13.4	**57.8±17.9**
Tasks	Medium-Expert				Medium-Replay			
	ODIS	HiSSD	BiKT	SGVD	ODIS	HiSSD	BiKT	SGVD
Source Tasks								
3m	89.8±9.7	90.9±5.9	91.3±4.8	**92.3±6.7**	79.7±4.7	87.7±2.9	78.8±3.2	**89.2±3.1**
5m	83.7±16.0	79.4±6.9	**85.3±5.9**	79.6±7.8	3.1±5.4	87.5±2.9	88.5±1.6	**89.4±3.2**
10m	**93.8±4.4**	60.2±21.1	83.6±3.1	86.7±4.2	0.0±0.0	84.2±4.9	85.2±2.5	86.6±3.7
Unseen Tasks								
4m	57.8±18.8	70.9±9.1	72.2±8.3	**73.4±6.3**	25.0±5.4	71.6±4.1	**77.2±4.7**	74.3±5.1
6m	76.0±6.0	70.6±6.1	78.2±1.6	**81.4±4.3**	3.1±5.4	**99.8±0.3**	86.8±3.2	94.3±5.3
7m	66.4±14.6	85.0±11.7	85.6±16.4	**87.5±9.3**	0.0±0.0	**99.8±0.3**	84.2±1.4	93.8±2.1
8m	43.8±11.5	72.8±9.5	68.3±4.6	**77.6±8.2**	1.6±1.6	96.7±0.3	87.6±1.7	**97.4±1.4**
9m	73.4±16.2	80.0±14.6	70.8±6.6	**81.3±9.8**	0.0±0.0	88.8±1.3	86.6±2.4	**89.3±3.2**
11m	68.8±20.3	70.9±5.9	75.5±12.4	**79.8±6.7**	0.0±0.0	45.6±4.5	**52.3±3.6**	46.2±4.9
12m	62.5±8.0	**62.7±7.8**	59.7±9.8	61.3±5.7	0.0±0.0	38.0±3.7	41.5±4.3	**48.7±6.8**

Table 6: Average test win rates of the best policies over five random seeds in the STALKER-ZEALOT task set under different offline data qualities.

Tasks	Expert				Medium			
	ODIS	HiSSD	BiKT	SGVD	ODIS	HiSSD	BiKT	SGVD
Source Tasks								
2s3z	97.7±2.6	95.2±1.0	97.9±2.3	**98.1±2.2**	49.2±8.4	32.3±11.7	**51.6±3.3**	49.3±5.4
2s4z	60.9±6.8	79.8±6.0	**93.2±5.1**	84.5±4.4	32.8±12.2	17.0±2.2	**25.0±7.4**	19.0±3.2
3s5z	87.5±9.6	92.8±5.0	93.0±4.5	**94.1±3.1**	28.9±6.8	24.4±7.9	29.8±2.3	**34.5±2.2**
Unseen Tasks								
1s3z	76.6±3.5	81.6±15.2	77.0±4.2	**85.1±8.3**	41.4±18.8	**44.2±9.9**	32.4±0.4	39.4±12.4
1s4z	17.2±10.5	42.0±26.1	52.6±15.1	**66.3±11.7**	**50.7±7.5**	18.1±11.0	22.6±0.5	19.2±7.6
1s5z	2.5±2.3	16.7±12.3	8.7±4.3	33.4±17.3	14.1±8.4	2.5±2.2	18.8±3.9	**19.3±6.3**
2s5z	27.3±6.0	**79.7±2.2**	75.3±3.2	71.1±2.4	**32.0±4.6**	11.3±3.7	24.2±6.5	15.3±4.8
3s3z	89.1±5.2	88.0±4.5	**98.4±1.6**	89.2±3.5	23.4±9.2	21.9±10.7	34.4±2.2	27.4±5.1
3s4z	96.9±2.2	88.1±9.0	**97.7±2.3**	94.3±3.9	50.8±15.5	17.2±4.5	**54.8±0.5**	45.8±3.5
4s3z	64.1±13.0	88.6±4.1	**96.9±2.3**	89.4±4.5	13.3±7.5	31.9±23.2	18.7±0.1	**34.7±7.5**
4s4z	79.7±10.9	73.4±5.2	75.5±5.3	**80.6±7.3**	12.5±7.0	13.2±6.5	16.7±1.4	**23.6±4.3**
4s5z	**86.7±12.6**	65.6±3.7	44.5±6.8	67.2±7.4	7.0±4.1	4.5±1.3	8.3±1.4	**14.7±9.2**
4s6z	**88.3±8.4**	68.4±4.9	68.2±6.9	69.3±6.8	1.6±1.6	0.9±0.9	2.5±2.5	**6.6±3.4**

Tasks	Medium-Expert				Medium-Replay			
	ODIS	HiSSD	BiKT	SGVD	ODIS	HiSSD	BiKT	SGVD
Source Tasks								
2s3z	58.6±15.5	68.1±8.1	81.3±7.4	**83.1±4.2**	15.6±18.2	9.0±1.5	**30.2±8.4**	17.1±7.6
2s4z	41.4±7.8	41.9±10.2	**73.8±7.8**	47.1±8.3	7.8±5.2	6.0±1.2	30.8±7.1	19.1±9.5
3s5z	41.4±18.5	57.8±10.7	59.1±8.7	**65.9±7.7**	18.8±3.1	17.5±2.0	19.3±7.1	**32.9±8.5**
Unseen Tasks								
1s3z	72.7±12.2	73.0±10.2	75.9±9.1	**76.8±6.3**	21.1±20.4	36.3±7.1	30.9±9.1	**39.7±8.8**
1s4z	**44.5±20.3**	32.3±30.5	37.9±5.9	27.0±12.7	6.2±7.7	24.8±9.1	26.3±7.2	**31.5±11.3**
1s5z	**42.2±31.4**	9.4±9.5	14.4±19.4	19.4±11.4	7.8±6.4	4.4±2.2	**12.5±4.7**	9.3±5.7
2s5z	**43.0±10.7**	25.6±7.8	19.0±5.2	38.4±8.9	14.1±8.1	16.5±2.8	17.2±8.4	**18.4±7.4**
3s3z	50.0±13.3	56.6±25.6	57.9±8.2	**83.7±7.9**	25.0±20.1	9.6±3.3	27.6±4.5	29.4±7.6
3s4z	52.3±9.5	71.7±9.7	75.6±13.3	**89.3±2.4**	19.5±16.6	22.5±10.6	19.4±11.1	**29.4±6.8**
4s3z	17.2±7.2	60.5±15.1	28.8±9.4	**84.6±4.8**	8.6±14.9	11.0±10.4	10.4±5.1	12.7±3.1
4s4z	20.3±6.8	37.3±9.4	39.9±4.9	**42.6±7.6**	4.7±8.1	9.4±1.8	8.3±2.9	8.7±1.6
4s5z	21.9±2.2	17.0±4.1	24.3±5.3	**27.6±8.1**	0.8±1.4	0.8±0.8	4.4±3.5	**5.6±4.9**
4s6z	18.0±5.1	**19.7±5.9**	14.8±3.2	16.7±4.3	2.3±4.1	0.4±0.3	3.5±2.9	**4.7±3.8**

C Additional Method Details

C.1 Complete Training Procedure

Algorithm 1 Training SGVD on LF-Augmented Offline Data

Require: Offline dataset $\mathcal{D}$; frozen LFs $\mathcal{F}^\star$; frozen text encoder E_{text}
Require: Annotation window length H_{ann}; loss weights in Eq. equation 7
Ensure: Trained model parameters Θ

Stage 1: Offline annotation

$\widetilde{\mathcal{D}} \leftarrow \emptyset$
for each trajectory $\xi \in \mathcal{D}$ **do**
  Partition ξ into windows of length H_{ann}
  Apply $\mathcal{F}^\star$ to window statistics $\Phi(W_m)$ to obtain labels and validity masks
  Propagate labels and construct boundaries, semantic targets, and valid segments
  Add ξ, its terminal state, annotations, and masks to $\widetilde{\mathcal{D}}$
end for
Fix $\widetilde{\mathcal{D}}$ and initialize Θ

Stage 2: Model optimization

while the stopping criterion is not met **do**
  Sample a masked minibatch, including terminal states, from $\widetilde{\mathcal{D}}$
  for $t = 1, \dots, T$ **do**
    $\mathbf{e}_t^i \leftarrow f_{\text{obs}}(o_t^i)$; $\mathbf{u}_t \leftarrow f_{\text{team}}(\{\mathbf{e}_t^i\}_{i=1}^n)$
    $\widehat{b}_t \leftarrow f_{\text{bp}}(\mathbf{u}_t)$
    $(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t) \leftarrow f_{\text{skill}}(\mathbf{u}_t)$
    $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\widehat{\mathbf{c}}_t \leftarrow \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t$
    if $t = 1$ or $\widetilde{b}_t = 1$ **then**
      $\mathbf{c}_t \leftarrow \widehat{\mathbf{c}}_t$
    else
      $\mathbf{c}_t \leftarrow \mathbf{c}_{t-1}$
    end if
    Decode $\pi_{\text{act}}^i(\cdot \mid o_t^i, \mathbf{c}_t)$ and $\widehat{\mathbf{q}}_t \leftarrow f_{\text{sem}}(\widehat{\mathbf{c}}_t)$
  end for
  Compute masked $\mathcal{L}_{\text{bnd}}$, $\mathcal{L}_{\text{BC}}$, and $\mathcal{L}_{\text{sem}}$; exclude $t = 1$ from boundary loss
  for each valid annotated segment σ_j **do**
    $\mathbf{v}_{t_j:t_j+d_j} \leftarrow f_{\text{state}}(\mathbf{s}_{t_j:t_j+d_j})$
    Infer $q_\psi(\widehat{\mathbf{c}}_{t_j} \mid \mathbf{v}_{t_j:t_j+d_j})$
    Compute the segment KL loss using Eq. equation 5
    Compute terminal prediction loss using Eq. equation 6
  end for
  Average segment losses over valid segments; use zero if none are available
  Form $\mathcal{L}_{\text{total}}$ using Eq. equation 7
  Update Θ along the gradient paths in Table 7
end while

C.2 Optimization and Gradient Flow

Fixed offline supervision. Algorithm 1 separates offline annotation from model optimization. The finalized LFs and pretrained text encoder remain frozen, and their outputs provide fixed semantic and boundary targets. The trainable neural modules are optimized using the weighted objective in Eq. equation 7.

Loss-to-module mapping. Let θ_{obs} and θ_{team} parameterize the observation and team encoders. The boundary predictor, action decoder, semantic decoder, and forward predictor have parameters θ_{bp}, θ_{act}, θ_{sem}, and θ_{pf}, respectively. The skill prior, segment posterior, and global-state encoder have parameters ϕ, ψ, and θ_{state}. Table 7 summarizes the gradient paths, including upstream en-

coders. Candidate skills are sampled through a differentiable reparameterization, allowing downstream objectives to update the skill prior.

Table 7: Gradient paths for individual training objectives. The LFs and text encoder remain frozen for all objectives.

Objective	Parameters receiving gradients
$\mathcal{L}_{\mathrm{bnd}}$	$\theta_{\mathrm{obs}}, \theta_{\mathrm{team}}, \theta_{\mathrm{bp}}$
$\mathcal{L}_{\mathrm{BC}}$	$\theta_{\mathrm{obs}}, \theta_{\mathrm{team}}, \phi, \theta_{\mathrm{act}}$
$\mathcal{L}_{\mathrm{sem}}$	$\theta_{\mathrm{obs}}, \theta_{\mathrm{team}}, \phi, \theta_{\mathrm{sem}}$
$\mathcal{L}_{\mathrm{KL}}$	$\theta_{\mathrm{obs}}, \theta_{\mathrm{team}}, \phi, \theta_{\mathrm{state}}, \psi$
$\mathcal{L}_{\mathrm{pf}}$	$\theta_{\mathrm{obs}}, \theta_{\mathrm{team}}, \phi, \theta_{\mathrm{pf}}$

Boundary supervision and skill switching. During training, the fixed annotations $\widetilde{b}_t$ control skill switching. Thus, the switching decision provides no gradient path from $\mathcal{L}_{\mathrm{BC}}$ to the boundary predictor. The predictor's parameters θ_{bp} receive gradients only from $\mathcal{L}_{\mathrm{bnd}}$, whereas its upstream encoders also receive gradients from the other objectives. The active skill $\mathbf{c}_t$ conditions action prediction, while semantic supervision is applied to the candidate skill $\widehat{\mathbf{c}}_t$. During execution, predicted boundaries replace the annotations, with the active skill initialized at the first timestep.

Posterior alignment and terminal prediction. The KL objective updates both the observation-conditioned prior and the segment-conditioned posterior, together with their input encoders. Neither distribution is detached; the posterior is learned rather than treated as a fixed teacher. For terminal prediction, the target $\mathrm{sg}(\mathbf{v}_{t_j+d_j})$ blocks gradients to the global-state encoder. Consequently, $\mathcal{L}_{\mathrm{pf}}$ updates the forward predictor, skill prior, and observation and team encoders, while the global-state encoder and posterior are updated through $\mathcal{L}_{\mathrm{KL}}$. The global-state encoder, posterior, and forward predictor are used only during training.

D EXPERIMENTAL SETTINGS

D.1 LLM AND COMPUTATIONAL RESOURCES

We use DeepSeek-V3 for the offline construction and iterative refinement of label functions. The finalized LFs are applied to the offline dataset to generate semantic labels and transition boundaries. DeepSeek-V3 is not used during policy training or execution. SGVD training experiments are conducted using four NVIDIA GeForce RTX 4090 GPUs.

D.2 BENCHMARKS AND TASK SPLITS

We evaluate SGVD on the StarCraft Multi-Agent Challenge (SMAC), a cooperative micromanagement benchmark with decentralized partial observations and a shared team reward. Following the offline multi-task protocol introduced by ODIS and subsequently adopted by HiSSD and BiKT Zhang et al. (2023); Liu et al. (2025); Zhang et al. (2025), we construct three task sets: MARINE-EASY, MARINE-HARD, and STALKER-ZEALOT. Each task set contains three source tasks used for offline training and a collection of held-out tasks used only for zero-shot evaluation. Source and unseen tasks differ in team size, enemy-team size, or unit composition. No trajectory from an unseen task is used for representation learning, policy learning, model selection, or LF construction.

Tables 8–10 give the complete task configurations. In MARINE-EASY, both teams contain the same number of homogeneous ranged units, and generalization mainly requires adapting to previously unseen team sizes. MARINE-HARD additionally contains asymmetric battles in which the allied team is outnumbered. STALKER-ZEALOT contains heterogeneous teams composed of ranged Stalkers and melee Zealots and therefore requires the policy to transfer role-dependent coordination patterns.

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 8: 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_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

Table 9: Descriptions of tasks in the MARINE-HARD 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 Stalker, 3 Zealots	1 Stalker, 3 Zealots	heterogeneous & symmetric
	1s4z	1 Stalker, 4 Zealots	1 Stalker, 4 Zealots	heterogeneous & symmetric
	1s5z	1 Stalker, 5 Zealots	1 Stalker, 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
	4s6z	4 Stalkers, 6 Zealots	4 Stalkers, 6 Zealots	heterogeneous & symmetric

Table 10: Descriptions of tasks in the STALKER-ZEALOT task set.

D.3 OFFLINE DATASETS

We use the public offline datasets collected by ODIS Zhang et al. (2023), which are also used by HiSSD Liu et al. (2025). All trajectories were generated on the source tasks by QMIX policies and no additional environment interaction is used for training. Following the D4RL-style protocol, we evaluate four data qualities:

- **Expert**: trajectories collected by a QMIX policy trained for two million environment steps;
- **Medium**: trajectories collected from a checkpoint whose test win rate is approximately half that of the expert policy;
- **Medium-Expert**: the union of the Medium and Expert datasets, combining two policy-quality levels;
- **Medium-Replay**: transitions retained in the replay buffer of the Medium policy, which contain a broader mixture of early and intermediate behaviors.

Table 11 reports the number of trajectories, average return, and average win rate of each source-task dataset. The Medium-Expert dataset contains the 2,000 Medium and 2,000 Expert trajectories. Medium-Replay sizes vary because they correspond to the available replay buffers rather than a fixed trajectory count; following ODIS and HiSSD, their aggregate return and win rate are not reported. We train a separate model for each task set and data quality; within a run, the model is trained jointly on the three corresponding source-task datasets.

Before policy training, the frozen task-set-specific LFs are applied to these source trajectories as described in Appendix A. This preprocessing only appends semantic targets, boundary targets, and diagnostic label identifiers; it does not alter the original states, observations, actions, rewards, or action-availability masks. Unseen-task trajectories are never annotated because they are not available during training.

D.4 IMPLEMENTATION DETAILS

We build SGVD on the entity-oriented offline multi-task architecture used by ODIS and HiSSD. To accommodate different observation sizes across tasks, agent i's local observation is decomposed into its own features o_i^{own} and a variable-size collection of ally and enemy entity features $\{o_{i,j}^{\text{entity}}\}$. Each component is projected to a 64-dimensional embedding and processed by a single-layer Transformer. Let

$$X_i = \left[e_i^{\text{own}}, e_{i,1}^{\text{entity}}, \ldots, e_{i,M_i}^{\text{entity}}\right] \tag{16}$$

denote the resulting entity-token sequence. The observation encoder applies scaled dot-product attention:

$$Q_i = X_i W^Q, \qquad K_i = X_i W^K, \qquad V_i = X_i W^V, \tag{17}$$

Task	Quality	Trajectories	Average Return	Average Win Rate
3m	Expert	2,000	19.8929	0.9910
	Medium	2,000	13.9869	0.5402
	Medium-Expert	4,000	16.9399	0.7656
	Medium-Replay	3,603	N/A	N/A
5m	Expert	2,000	19.9380	0.9937
	Medium	2,000	17.3288	0.7411
	Medium-Expert	4,000	18.6334	0.8674
	Medium-Replay	711	N/A	N/A
10m	Expert	2,000	19.9438	0.9922
	Medium	2,000	16.6297	0.5413
	Medium-Expert	4,000	18.2595	0.7626
	Medium-Replay	571	N/A	N/A
5m_vs_6m	Expert	2,000	17.3424	0.7185
	Medium	2,000	12.6408	0.2751
	Medium-Expert	4,000	14.9916	0.4968
	Medium-Replay	32,607	N/A	N/A
9m_vs_10m	Expert	2,000	19.6140	0.9431
	Medium	2,000	15.5049	0.4146
	Medium-Expert	4,000	17.5594	0.6789
	Medium-Replay	13,731	N/A	N/A
2s3z	Expert	2,000	19.7655	0.9602
	Medium	2,000	16.6279	0.4465
	Medium-Expert	4,000	18.1967	0.7034
	Medium-Replay	4,505	N/A	N/A
2s4z	Expert	2,000	19.7402	0.9509
	Medium	2,000	16.8735	0.4965
	Medium-Expert	4,000	18.3069	0.7237
	Medium-Replay	6,172	N/A	N/A
3s5z	Expert	2,000	19.7850	0.9518
	Medium	2,000	16.3126	0.3114
	Medium-Expert	4,000	18.0488	0.6316
	Medium-Replay	11,528	N/A	N/A

Table 11: Properties of the offline SMAC datasets under different data qualities.

$$\overline{X}_i = \mathrm{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d_k}}\right) V_i. \tag{18}$$

This entity-wise computation shares parameters across entity slots and therefore permits the same encoder to process varying numbers of agents and enemies. The team representation is obtained by mean aggregation over the encoded agent representations. Both the candidate coordination skill and active coordination skill have dimension 64. The action decoder is shared across tasks and produces an action distribution for each agent from its local observation and the active team skill.

The boundary predictor is a binary classifier over the team representation. During training, LF-derived boundaries control skill switching through teacher forcing; during evaluation, the predicted boundary probability is thresholded at $\delta = 0.5$. The semantic decoder maps the candidate skill to the text-target space. The training-only segment posterior is parameterized as a diagonal Gaussian, with its log variance clipped to $[-5, 2]$. Unless otherwise stated, all models are optimized for 35,000 gradient steps with AdamW. Table 12 summarizes the principal hyperparameters.

For MARINE-EASY and the Expert, Medium, and Medium-Expert variants of MARINE-HARD, we use a learning rate of 3×10^{-4} and a minibatch of 32 trajectories. For Medium-Replay on MARINE-HARD and all STALKER-ZEALOT settings, whose source trajectories are more heterogeneous, we use a learning rate of 10^{-4} and a minibatch of 16 trajectories. Model checkpoints are saved every 5,000 training steps. All ablated variants retain the same architecture and optimization settings as the full model except for the component explicitly removed.

Hyperparameter	Setting
Entity embedding dimension	64
Team-skill dimension	64
Transformer layers	1
Attention heads	1
Discount factor γ	0.99
Annotation-window length H_{ann}	3
Boundary threshold δ	0.5
Boundary-loss weight λ_{bnd}	1.0
Semantic-loss weight λ_{sem}	0.1
Dynamics-loss weight λ_{dyn}	1.0
Terminal-prediction weight β_{pf}	1.0
KL-distillation weight	0.1
Behavior-cloning weight λ_{BC}	1.0
Gradient-norm clipping	10
Weight decay	10^{-4}
Training steps	35,000

Table 12: Main hyperparameters of SGVD.

D.5 Computational Resources and LLM

All experiments were conducted on a server equipped with eight NVIDIA GeForce RTX 4090 GPUs. We used DeepSeek-V4 to generate the natural-language skill descriptions and to construct and iteratively refine the label functions. The LLM was not used during full-dataset annotation, policy training, or decentralized execution.

D.6 Baselines and Evaluation Protocol

We compare against BC-best, ODIS, HiSSD, and BiKT. BC-best reports the better result of a timestep-conditioned behavior-cloning policy (BC-t) and a return-conditioned behavior-cloning policy (BC-r), following the protocol used in prior offline multi-task studies. For ODIS, HiSSD, and BiKT, we use their official implementations and task-specific hyperparameter settings. Every method is trained on exactly the same source-task trajectories for a given task set and data quality. No method receives target-task trajectories or performs online adaptation on an unseen task.

At evaluation time, policies act greedily and are evaluated on both the three source tasks and all held-out tasks listed in Tables 8–10. For SGVD, skill transitions are determined only by the learned observation-conditioned boundary predictor; the LLM, LFs, semantic labels, and global states are unavailable. We run each method with five random seeds and report the mean and standard deviation of test win rates. Following prior work, source-task results measure offline policy learning, whereas held-out-task results measure zero-shot cross-task generalization.

E Detailed Ablation Experiments

While the main paper reports the ablation results averaged over the source and unseen tasks, Table 13 provides the complete results under each offline data quality.

Table 13: Detailed ablation results on *Marine-Hard* under different offline data qualities. Each entry reports the mean test win rate and standard deviation over five random seeds.

Tasks	w/o Semantic Grounding	Fixed Duration	w/o Terminal Prediction	w/o Dynamics Distillation	SGVD
Expert					
Source Tasks					
3m	99.4±0.6	99.2±0.8	99.2±0.6	99.3±0.5	**99.6±0.4**
5m6m	65.4±3.8	64.5±6.8	66.7±5.4	65.8±6.3	**68.2±5.3**
9m10m	**97.0±2.4**	96.1±1.8	96.4±2.0	96.5±2.1	96.9±2.2
Unseen Tasks					
4m	97.4±1.8	97.9±2.6	97.8±1.8	98.0±2.2	**99.4±1.4**
5m	99.0±1.4	99.6±1.4	98.6±1.2	97.9±1.8	**99.9±0.1**
10m	98.5±1.1	98.7±1.2	98.6±0.9	98.9±1.2	**99.6±0.2**
12m	84.2±9.5	82.8±12.7	86.4±8.9	86.6±11.2	**89.5±10.2**
7m8m	38.3±6.9	34.5±9.8	37.5±7.1	39.6±8.9	**42.4±8.5**
8m9m	54.3±5.6	48.7±6.8	55.3±3.9	56.2±3.4	**57.1±4.3**
10m11m	83.5±7.8	75.6±14.3	84.7±6.4	85.5±7.7	**91.1±4.2**
10m12m	14.5±7.6	11.8±6.5	15.2±5.6	15.1±4.7	**15.4±4.3**
13m15m	**5.6±2.4**	4.4±0.4	4.8±0.8	4.8±1.2	5.2±3.8
Medium					
Source Tasks					
3m	65.6±5.6	62.3±5.1	64.4±3.4	65.5±3.4	**66.4±6.2**
5m6m	34.6±2.3	31.5±3.5	36.6±2.5	36.5±2.2	**37.2±3.4**
9m10m	70.2±3.5	66.4±3.4	71.7±2.2	72.2±2.4	**74.7±2.6**
Unseen Tasks					
4m	78.2±5.3	77.6±4.8	78.1±4.9	79.4±5.4	**80.1±6.5**
5m	93.7±6.1	92.8±5.6	93.5±6.6	94.1±7.7	**95.8±4.2**
10m	88.5±6.7	82.3±5.7	89.7±5.6	89.3±4.7	**92.4±6.3**
12m	86.2±3.4	81.8±4.3	86.5±4.2	85.2±5.7	**88.0±4.1**
7m8m	13.7±4.8	8.7±7.5	13.8±6.8	13.4±7.3	**15.9±8.0**
8m9m	12.1±4.5	8.5±6.2	13.3±4.1	14.0±3.7	**17.6±3.7**
10m11m	36.9±7.9	28.0±11.5	36.4±8.5	37.1±6.9	**45.5±5.4**
10m12m	2.0±0.4	1.8±0.2	1.9±0.3	2.1±0.3	**2.4±1.8**
13m15m	2.1±0.5	1.6±0.4	**2.9±2.8**	2.6±2.4	2.8±1.6

Continued on next page

Table 13 (continued)

Tasks	w/o Semantic Grounding	Fixed Duration	w/o Terminal Prediction	w/o Dynamics Distillation	SGVD
Medium-Expert					
Source Tasks					
3m	88.7±3.7	87.7±6.5	88.1±4.3	87.9±5.6	**92.2±5.7**
5m6m	48.2±6.6	44.5±7.1	48.2±4.2	49.3±5.1	**50.2±8.4**
9m10m	82.8±8.7	83.3±7.6	83.6±6.9	84.5±7.1	**85.2±7.2**
Unseen Tasks					
4m	96.3±3.7	94.4±4.0	95.5±2.3	95.9±4.1	**96.4±4.2**
5m	97.6±2.0	94.7±3.4	96.5±2.3	96.7±2.6	**98.9±1.2**
10m	91.6±5.7	89.5±5.3	93.4±3.7	94.5±4.3	**97.7±3.4**
12m	88.7±11.9	90.5±9.1	93.5±8.9	93.1±10.1	**96.5±3.7**
7m8m	44.5±4.9	35.7±6.9	**44.6±5.6**	42.1±5.6	42.6±6.2
8m9m	30.7±12.9	23.3±15.8	35.5±14.5	33.7±15.6	**40.7±15.4**
10m11m	58.8±7.8	39.8±10.1	55.4±6.9	56.7±5.8	**62.4±9.3**
10m12m	4.1±1.8	1.8±1.8	3.8±2.0	3.9±1.9	**5.7±2.5**
13m15m	3.0±1.2	1.8±0.8	2.9±0.4	2.8±1.4	**5.5±2.4**
Medium-Replay					
Source Tasks					
3m	82.3±4.7	79.4±5.3	82.3±4.7	81.4±5.1	**84.5±6.2**
5m6m	23.3±2.3	17.2±5.6	26.6±4.7	24.3±3.5	**28.6±4.0**
9m10m	40.8±3.1	37.4±5.7	45.5±4.4	45.7±3.4	**48.6±4.5**
Unseen Tasks					
4m	94.4±3.6	92.3±5.6	92.3±4.1	92.4±3.1	**95.6±3.4**
5m	93.6±16.7	94.5±15.3	96.4±14.5	96.4±13.9	**98.1±5.4**
10m	94.2±3.7	95.2±2.3	**97.7±3.9**	94.8±5.4	97.6±4.1
12m	89.9±4.8	89.1±4.4	92.7±3.4	**93.9±3.7**	93.6±2.4
7m8m	16.7±4.9	13.2±6.7	18.3±5.1	18.7±5.9	**22.6±4.3**
8m9m	10.6±3.4	8.6±5.7	14.3±2.9	13.6±3.2	**15.6±4.3**
10m11m	44.4±4.7	32.7±8.9	45.6±6.1	44.9±5.9	**47.2±5.3**
10m12m	2.7±0.4	2.1±0.1	2.6±0.4	2.5±0.5	**3.2±2.6**
13m15m	2.4±1.4	1.0±1.0	2.3±1.2	2.2±2.2	**4.0±2.8**

Table 14: This table illustrative window-size ablation on *Marine-Hard* under different offline data qualities. $H = 3$ achieves the strongest performance and is therefore adopted in the main experiments. On a few MEDIUM-REPLAY tasks, $H = 5$ performs slightly better because the replay data contain relatively long behavioral phases. As H increases further, a window is more likely to span semantic transition points and mix distinct behavioral phases, reducing annotation coherence and consequently degrading performance.

Tasks	Expert				Medium			
	$H=3$	$H=5$	$H=7$	$H=10$	$H=3$	$H=5$	$H=7$	$H=10$
Source Tasks								
3m	**99.6±0.4**	98.7±0.4	82.7±0.5	64.7±0.6	**66.4±6.2**	64.8±6.5	55.1±7.1	43.2±8.0
5m6m	**68.2±5.3**	66.4±5.6	56.3±6.1	44.3±6.9	**37.2±3.4**	35.9±3.6	30.9±3.9	24.2±4.4
9m10m	**96.9±2.2**	95.2±2.3	80.4±2.5	62.9±2.9	**74.7±2.6**	72.8±2.7	62.0±3.0	48.5±3.4
Unseen Tasks								
4m	**99.4±1.4**	98.1±1.5	82.5±1.6	64.6±1.8	**80.1±6.5**	78.4±6.8	66.5±7.5	52.1±8.5
5m	**99.9±0.1**	98.8±0.1	83.0±0.1	65.0±0.2	**95.8±4.2**	94.2±4.4	79.5±4.8	62.3±5.5
10m	**99.6±0.2**	98.4±0.2	82.9±0.2	64.7±0.3	**92.4±6.3**	90.6±6.6	76.7±7.2	60.1±8.2
12m	**89.5±10.2**	87.8±10.7	74.3±11.7	58.2±13.3	**88.0±4.1**	86.1±4.3	73.0±4.7	57.2±5.3
7m8m	**42.4±8.5**	40.3±8.9	35.2±9.8	27.1±11.1	**15.9±8.0**	14.4±8.4	13.2±9.2	10.2±10.4
8m9m	**57.1±4.3**	58.3±4.5	47.4±4.9	36.5±5.6	**17.6±3.7**	15.1±3.9	14.6±4.3	11.3±4.8
10m11m	91.1±4.2	90.4±4.4	75.6±4.8	58.3±5.5	45.5±5.4	**46.4±5.7**	37.8±6.2	29.1±7.0
10m12m	**15.4±4.3**	14.8±4.5	12.8±4.9	9.8±5.6	**2.4±1.8**	2.3±1.9	2.0±2.1	1.5±2.3
13m15m	5.2±3.8	**5.4±4.0**	4.3±4.4	3.3±4.9	**2.8±1.6**	2.1±1.7	2.3±1.8	1.8±2.1

Tasks	Medium-Expert				Medium-Replay			
	$H=3$	$H=5$	$H=7$	$H=10$	$H=3$	$H=5$	$H=7$	$H=10$
Source Tasks								
3m	**92.2±5.7**	90.3±6.0	76.5±6.6	59.9±7.4	**84.5±6.2**	82.8±6.5	70.1±7.1	54.9±8.1
5m6m	**50.2±8.4**	48.6±8.8	41.7±9.7	32.6±10.9	**28.6±4.0**	27.5±4.2	23.7±4.6	18.3±5.2
9m10m	**85.2±7.2**	83.4±7.6	70.7±8.3	55.4±9.4	48.6±4.5	**50.1±4.7**	40.3±5.2	31.1±5.9
Unseen Tasks								
4m	**96.4±4.2**	94.7±4.4	80.0±4.8	62.7±5.5	**95.6±3.4**	93.9±3.6	79.3±3.9	62.1±4.4
5m	**98.9±1.2**	97.2±1.3	82.1±1.4	64.3±1.6	**98.1±5.4**	96.4±5.7	81.4±6.2	63.8±7.0
10m	**97.7±3.4**	96.0±3.6	81.1±3.9	63.5±4.4	**97.6±4.1**	95.8±4.3	81.0±4.7	63.4±5.3
12m	**96.5±3.7**	94.8±3.9	80.1±4.3	62.7±4.8	**93.6±2.4**	91.8±2.5	77.7±2.8	60.8±3.1
7m8m	**42.6±6.2**	39.5±6.5	35.4±7.1	27.3±8.1	22.6±4.3	**27.2±4.5**	18.8±4.9	14.5±5.6
8m9m	40.7±15.4	**41.6±16.2**	33.8±17.7	26.0±20.0	**15.6±4.3**	14.0±4.5	13.0±4.9	10.0±5.6
10m11m	**62.4±9.3**	60.6±9.8	51.8±10.7	39.9±12.1	47.2±5.3	**48.1±5.6**	39.2±6.1	30.2±6.9
10m12m	**5.7±2.5**	4.9±2.6	4.7±2.9	3.6±3.3	3.2±2.6	**4.3±2.7**	2.7±3.0	2.0±3.4
13m15m	5.5±2.4	**5.7±2.5**	4.6±2.8	3.5±3.1	4.0±2.8	**4.1±2.9**	3.3±3.2	2.6±3.6