Appendix

August 2, 2025

Table 1: Comparison of Algorithms in Terms of Time Complexity and Runtime on 25m Map

Algorithm	Time Complexity	Runtime on SMAC 25m Map
GraphSem	$\mathcal{O}(n^2)$	3 days, 10 hours, 57 minutes, 30 seconds; peak memory usage: 11 GB
QMIX	$\mathcal{O}(n)$	1 day, 48 minutes, 40 seconds
DFAC	$\mathcal{O}(n)$	2 days, 16 minutes, 41 seconds
SIDE	$\mathcal{O}(n)$	1 day, 3 hours, 52 minutes, 28 seconds
AERIAL	$\mathcal{O}(n^2)$	1 day, 7 hours, 29 minutes, 23 seconds
CAMA	$\mathcal{O}(n^2)$	8 hours, 38 minutes, 17 seconds (parallel training; excluded from direct comparison)
GACG	$\mathcal{O}(n^4)$	4 days, 53 minutes, 41 seconds

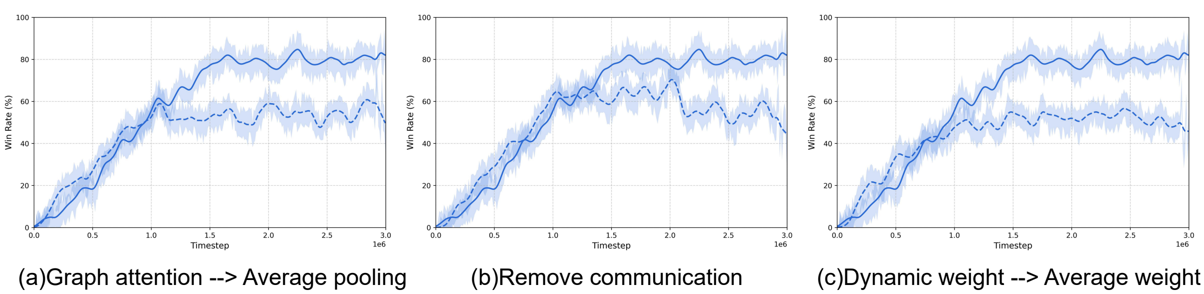
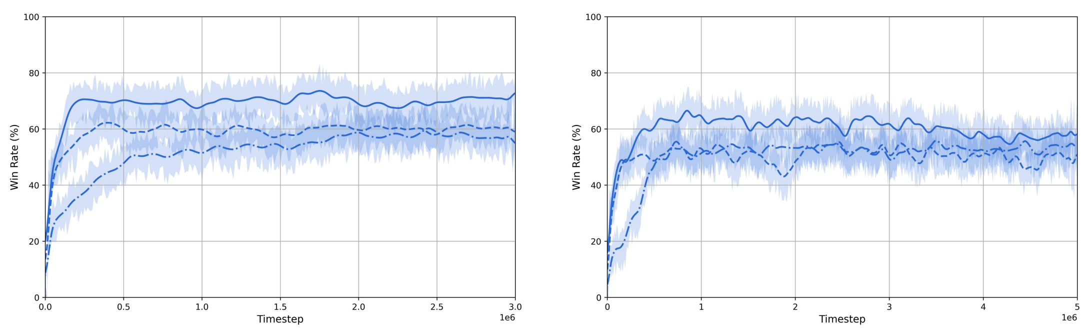


Figure 1: There are more detailed mechanism ablation experiments of the GraphSem algorithm on MMM maps, such as (a) converting the graph attention mechanism into average pooling, (b) removing the communication mechanism, and (c) converting the dynamic weight mechanism into average weights

Table 2: Supplementary Ablation Results of GraphSem Components on MMM of Gumble-softmax module ablation

Model Variant	Avg. Win Rate (%)	Convergence Time
w/o Gumbel	50.00	2.61M
w/o Enc	61.54	2.18M
w/o DCW	79.97	1.85M
w/o Dec	73.15	1.80M
w/o GACF	78.80	1.75M
GraphSem (Full)	**81.25**	**1.61M**



(a) Traffic_junction medium (b) Traffic_junction hard

Figure 2: The ablation experiments of the GraphSem algorithm were conducted on the hard and medium difficulty maps of Traffic. The dotted lines in the graphs represent the Encoder ablation experiments, and the dotted lines represent the graph attention ablation experiments

Table 3: Hyperparameters Used in GraphSem

Category	Parameter	Value
Network Architecture	hidden_dim	128
	coarsening_embed_dim	32
	mixing_embed_dim	32
	hypernet_layers	2
	hypernet_embed	64
Transformer	enc_att_heads	2
	dec_att_heads	4
	att_enc_dim	128
	att_dec_dim	32
	num_layers	2
	dropout	0
Training	lr	0.0005
	gamma	0.99
	batch_size	32
	buffer_size	5000
	target_update_interval_or_tau	200
	grad_norm_clip	10
Optimizer	optim_alpha	0.99
	optim_eps	0.00001
Exploration	epsilon_start	1.0
	epsilon_finish	0.05
	epsilon_anneal_time	50000
	evaluation_epsilon	0.0
Noise / Perturbation	obs_tamper	[0.1, 0.01]
	tamper	True
Gumbel Softmax	tau	0.01
Training Control	t_max	3005000
	test_interval	2000
	test_nepisode	20
	log_interval	2000
Other Important	double_q	True
	use_rnn	True
	standardise_rewards	True
	standardise_returns	False
	obs_agent_id	True
	obs_last_action	False

1 Message Clustering and Attention Analysis.

While GraphSem adaptively modulates communication via attention-based weighting and graph-based fusion, it remains an open question how messages are clustered and prioritized over time. A promising direction is to incorporate interpretability tools to probe the learned attention weights and investigate how semantic messages are grouped based on agent roles or task contexts. Such analysis could reveal latent communication patterns and help optimize message routing for improved efficiency and interpretability.

2 Heterogeneous Agents and Asymmetric Noise.

Our current evaluation includes several heterogeneous scenarios from SMAC, such as 1c3s5z,3s5z and so on, which involve agents with differing roles and unit types. These results indicate that GraphSem can generalize well to mild heterogeneity in observation and action spaces. However, future work could further explore more complex forms of heterogeneity, such as agents with distinct sensor modalities, task-specific capabilities, or asymmetric communication privileges. In addition, while we introduce controlled stochasticity into agent observations, the noise is currently sampled uniformly across agents. Extending the framework to model asymmetric noise distributions—for example, agents exposed to different sensor faults or environmental uncertainties—may offer new insights into robustness under uneven information reliability.

3 Defining "Semantic" in GraphSem.

In this work, we adopt a task-oriented and signal-level interpretation of semantics, rather than the traditional symbol-grounded or language-based notion. Specifically, we define a message as semantic if it selectively encodes task-relevant, decision-critical information while discarding irrelevant or redundant details from the original observation. Semantics is interpreted as the portion of information that affects the success of task execution (e.g., control, coordination, planning), regardless of whether it is grounded in symbolic structures.

In our context, the Transformer-based encoder abstracts observation histories into compact latent vectors that are optimized to retain only those features that improve joint decision-making performance under partial observability. These representations are not symbolic, but they are semantically meaningful in the sense that they carry high decision utility. Moreover, our use of structured attention and graph-guided message fusion ensures that these "semantic messages" contribute to accurate joint value estimation and robust coordination.

Therefore, our use of the term semantic follows a pragmatic, task-driven understanding commonly adopted in non-linguistic domains of representation learning and cooperative MARL.