EPISODIC MEMORY TEMPORAL CONSISTENCY FOR COOPERATIVE MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Cooperative Multi-Agent Reinforcement Learning (MARL) frequently suffers from severe reward sparsity and exploration bottlenecks. While episodic memory mechanisms mitigate these issues by reusing high-return trajectories, they often trap agents in local optima due to unconstrained incentive distribution and semantic representation collapse. To address this, we propose Episodic Memory Temporal Consistency (EMTC), a framework that robustly constructs and selectively leverages historical experiences. EMTC introduces two synergistic components: (1) a Temporally Consistent Semantic Embedder that integrates contrastive learning with time-conditioned state reconstruction, preventing representation collapse and enabling precise memory retrieval; and (2) a Temporal Consistency Gating Mechanism that dynamically modulates episodic incentives based on temporal consistency error. This adaptive gate filters misleading signals from pseudo-successful trajectories, effectively mitigating Q-value overestimation. We provide theoretical guarantees, establishing a strict error bound that directly links the observable temporal consistency error to the underlying trajectory optimality and representation quality. Extensive evaluations on the SMAC and GRF benchmarks demonstrate that EMTC consistently outperforms state-of-the-art baselines. Notably, compared to the strongest episodic baseline, EMTC achieves absolute win-rate improvements of up to 24% in super-hard SMAC scenarios and an average improvement of 28% across GRF tasks. Our code is provided in the supplementary materials.

1 INTRODUCTION

Recently, cooperative multi-agent systems have demonstrated remarkable success in tackling complex problems across a wide spectrum of domains, including robotics (Chen et al. (2025)), cooperative autonomous driving (Hao et al. (2025); Yu et al. (2025)), and communication networks (Hong et al. (2024)). Nevertheless, deploying these systems in real-world applications still faces formidable challenges, mainly stemming from inherent partial observability and highly complex coordination dynamics among interacting agents.

To tackle these challenges, Centralized Training with Decentralized Execution (CTDE) (Oliehoek et al. (2008; 2016); Gupta et al. (2017)) and value factorization algorithms (e.g., QMIX (Rashid et al. (2020)), DMIX (Sun et al. (2021)), and QPLEX (Wang et al. (2020a)) have emerged as the mainstream paradigm in cooperative MARL (Samvelyan et al. (2019)). Despite their success, these value-based methods fundamentally rely on dense step-wise rewards, suffering from profound exploration bottlenecks and suboptimal entrapment in environments with severe reward sparsity and vast joint action spaces. To mitigate this, episodic memory mechanisms have been integrated into MARL (Blundell et al. (2016); Lengyel & Dayan (2007); Lin et al. (2018); Pritzel et al. (2017)). By caching historical high-return trajectories to generate intrinsic incentives, recent frameworks like EMC (Zheng et al. (2021)) and EMU (Na et al. (2024)) break initial exploration deadlocks and significantly accelerate convergence. In a typical episodic memory workflow, a parametric encoder first maps visited states into a compact latent space, where high-return trajectories are stored as key-value pairs. During subsequent exploration, agents retrieve the maximum historical return associated with the current state's nearest neighbor in this latent space. This retrieved value is then

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.

used to construct an intrinsic incentive that guides the policy toward previously successful regions. A comprehensive discussion of related work is provided in Appendix B.

However, this initial benefit proves transient, as conventional memory workflows (Figure 1(A)) suffer from a coupled architectural flaw: **imbalanced feature prioritization during state embedding**. By aggressively modeling temporal sequences (Figure 1(B)), standard encoders inadvertently suppress discriminative spatial semantics[1], triggering a detrimental cascade of failures summarized as two coupled bottlenecks (Figure 1(D)): ***i)*** **Representation Collapse.** The erosion of spatial semantics destroys the topological structure of the latent space. As shown in Figure 1(C), critical detrimental states (e.g., agents being surrounded and losing formation) are erroneously mapped into dense clusters of successful experiences. This severe overlap between beneficial and harmful states fundamentally invalidates distance-based memory retrieval. ***ii)*** **Value Overestimation.** Retrieval ambiguity directly compromises incentive validity. When a pseudo-successful memory is retrieved (Figure 1(C), probe③), existing frameworks blindly propagate its stored return as an intrinsic reward. This uncalibrated reward injection artificially inflates Q-values, creating a deceptive illusion of progress that traps agents in inescapable local optima and ultimately causes episode failure.

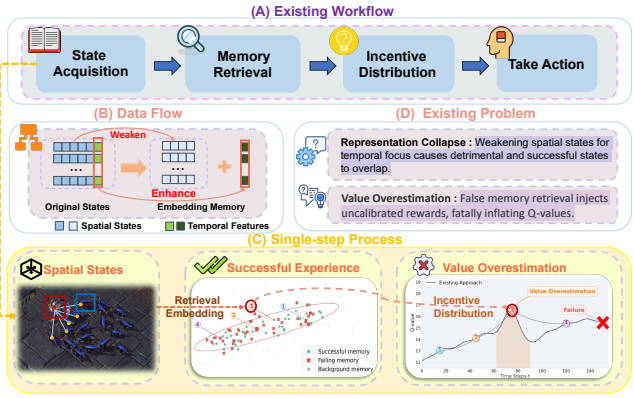


Figure 1: **The coupled bottlenecks of conventional Episodic Memory in MARL. (A)** The standard episodic workflow. **(B)** Biased data flow weakens spatial states in favor of temporal features. **(C)** A detrimental spatial state is falsely embedded into a successful cluster (probe ③), which subsequently triggers an uncalibrated reward injection and a fatal Q-value spike (Value Overestimation). **(D)** Summary of the existing problems.

To resolve these coupled limitations, we propose **E**pisodic **M**emory **T**emporal **C**onsistency (EMTC), a unified framework that aligns robust semantic memory construction with temporally constrained incentive distribution. EMTC directly addresses representation collapse via a contrastive encoder-decoder architecture for precise memory retrieval, and mitigates value overestimation through a novel Bellman-consistency gating mechanism that dynamically filters misleading episodic rewards. Our **main contributions** are summarized as follows:

- **Temporally Consistent Semantic Embedder:** By synergizing contrastive learning with time-conditioned reconstruction, our embedder extracts semantics-preserving state representations. This enforces a strictly clustered latent space, fundamentally resolving representation collapse and retrieval distortion in complex environments.
- **Temporal Consistency Gating Mechanism:** We introduce an adaptive incentive regulator grounded in the Bellman optimality criterion. By evaluating the temporal consistency error of retrieved memories, the gate dynamically attenuates rewards from pseudo-successful trajectories, effectively preventing Q-value overestimation.
- **Theoretical Guarantees & Empirical Superiority:** We mathematically establish a rigorous error bound for the episodic target, formally proving that our gating mechanism safely isolates genuine optimal transitions from representation mismatch and memory staleness. Extensive evaluations on the SMAC and GRF benchmarks demonstrate that EMTC consistently outperforms state-of-the-art baselines, achieving absolute win-rate improvements of up to 24% in super-hard SMAC scenarios and an average of 28% across highly sparse GRF tasks.

[1]Fundamentally, *spatial semantics* denotes an agent's structural awareness of its environment. While instantiated as explicit coordinate arrays in our tasks, this concept directly aligns with the visual/3D spatial representations in Embodied AI.

2 BACKGROUND

2.1 DECENTRALIZED PARTIALLY OBSERVABLE MARKOV DECISION PROCESS

A fully cooperative multi-agent task is formalized as a Dec-POMDP(Oliehoek et al. (2016)), defined by the tuple $\mathcal{G} = \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, P, R, \Omega, O, \gamma \rangle$. Each agent $i \in \mathcal{I}$ receives a local observation $o_i = O(s, i)$ and selects action a_i, forming joint action $\boldsymbol{a}$ that leads to next state $s' \sim P(\cdot|s, \boldsymbol{a})$ and shared reward $r = R(s, \boldsymbol{a})$. Due to partial observability, agents condition policies on action-observation histories τ_i, with the goal of maximizing $V^{\boldsymbol{\pi}}(s) = \mathbb{E}\left[\sum_t \gamma^t r_t \mid s_0 = s\right]$.

2.2 EPISODIC MEMORY IN MARL

Given the partial observability in Dec-POMDPs, agents must rely on historical trajectories for decision-making, motivating the integration of episodic memory mechanisms. The concept is inspired by the hippocampus's capacity to rapidly encode and recall salient past experiences, and in model-free RL it was formalized to circumvent slow parametric convergence by storing high-return trajectories(Lin et al. (2018); Pritzel et al. (2017)).

State Representation and Memory Storage. A parametric encoder $f_\phi : \mathcal{S} \to \mathbb{R}^k$ maps a global state s_t to a compact embedding $x_t = f_\phi(s_t)$. A buffer $\mathcal{D}_E$ stores key-value pairs $(x, H(x))$, where $H(x)$ is the maximum empirical return. The update follows a nearest-neighbor rule:

$$H(x_t) = \begin{cases} \max\{H(\hat{x}_t), \mathcal{R}_t\}, & \|x_t - \hat{x}_t\|_2 < \delta \\ \mathcal{R}_t, & \text{otherwise} \end{cases} \tag{1}$$

where $\hat{x}_t = \arg\min_{x \in \mathcal{D}_E} \|x_t - x\|_2$, δ is a similarity threshold, and $\mathcal{R}_t$ is the cumulative return from (s_t, a_t).

Episodic Integration in MARL. In cooperative MARL, episodic memory alleviates reward sparsity by recalling successful joint-action sequences. Early methods like EMC(Zheng et al. (2021)) directly inject stored returns into the TD target, while EMU(Na et al. (2024)) refines this by designing an explicit intrinsic reward r_p based on the gap between the historical optimum and the current critic:

$$r_p \triangleq \gamma \frac{N_\xi(s')}{N_{\text{call}}(s')} \left(H(f_\phi(s')) - \max_{a'} Q_{\theta^-}(s', a') \right) \tag{2}$$

where $N_{\text{call}}(s')$ and $N_\xi(s')$ count total visits and successful transitions near s'. The joint policy is optimized with a reshaped TD loss:

$$\mathcal{L}_p(\theta) = \left(r(s, a) + r_p + \gamma \max_{a'} Q_{\theta^-}(s', a') - Q_{tot}(s, a; \theta) \right)^2 \tag{3}$$

where Q_{tot} is the centralized joint Q-network and Q_{θ^-} is the target network.

Limitations of Unconstrained Episodic Incentives. Conventional episodic memory paradigms exhibit two critical coupled vulnerabilities in stochastic MARL environments. The primary bottleneck is *representation collapse*, where standard deterministic reconstruction fails to enforce semantic discriminability. This distorts distance-based retrieval in $\mathcal{D}_E$, causing agents to erroneously match distinct states and recall misleading returns. Compounding this retrieval failure is the subsequent risk of *value overestimation*. Derived solely from coarse, episode-level metrics, the episodic incentive r_p in Eq. 2 is temporally oblivious and blind to the sequential causality of individual transitions. Consequently, unregulated incentives from pseudo-successful trajectories systematically inflate Q-value estimates, trapping policies in suboptimal equilibria. Addressing this fundamental decoupling between retrieval accuracy and incentive validity necessitates a unified framework combining robust semantic representation with temporally consistent incentive gating.

3 METHOD

This section details the architecture of EMTC, a unified framework designed to rectify the systemic deficiencies of unconstrained episodic guidance, as illustrated in Figure 2. Rather than treating memory retrieval and incentive distribution as isolated modules, EMTC establishes a two-stage

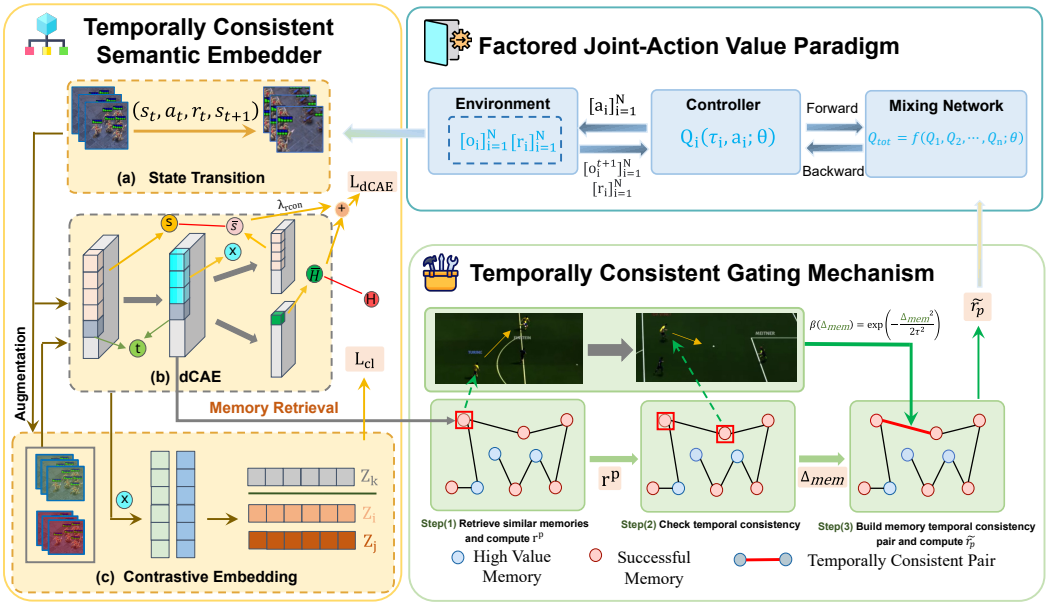


Figure 2: **Overview of the EMTC framework.** The architecture operates through a two-stage pipeline: (1) *TCSE* regularizes the latent space via contrastive learning and time-augmented reconstruction to ensure semantically precise memory retrieval; (2) *TCGM* evaluates the temporal coherence of retrieved memories using Bellman-consistency errors, dynamically scaling the episodic incentive $\tilde{r}_p$ before integration into the standard CTDE value update (denoted as the Factored Joint-Action Value Paradigm in the figure). This paradigm acts as a standard backbone (e.g., QPLEX, CDS), demonstrating that EMTC is a plug-and-play framework.

pipeline with periodic feedback, where latent representation quality directly governs incentive validity. The workflow operates sequentially within each training cycle: the *Temporally Consistent Semantic Embedder* (TCSE) first constructs a structured latent space that preserves semantic discriminability across temporal horizons, ensuring retrieved memories are intrinsically aligned with current states. Subsequently, the *Temporal Consistency Gating Mechanism* (TCGM) rigorously vets these trajectories against the Bellman optimality criterion before releasing any intrinsic reward. This design decouples representation learning from policy optimization while maintaining alignment through episodic buffer updates, enabling stable policy refinement even under severe reward sparsity.

3.1 Temporally Consistent Semantic Embedder

Structural Organization of Episodic Memory. To ensure retrieval stability and preserve temporal dependencies, we maintain a structured episodic buffer $\mathcal{D}_E$. Each stored transition is represented as a tuple $\langle s_i, x_i, H_i, \xi_i, \text{idx}_{\text{next}} \rangle$, where s_i is the raw global state, $x_i = f_\phi(s_i)$ is its latent embedding, H_i records the maximum empirical return, and $\xi_i \in \{0, 1\}$ denotes the trajectory's success indicator. To accommodate periodic updates to the embedding network f_ϕ, we retain s_i in the buffer, enabling on-the-fly re-projection of x_i and preventing representation drift from inducing stale memory matches. Crucially, idx_{next} denotes the index of the memory entry corresponding to the immediate successor state s_{t+1} within the same stored trajectory. This chain-based architecture preserves the sequential causality of high-return experiences. The specific update mechanism for the episodic memory buffer is detailed in Appendix F.2.

Multi-Objective Semantic Representation Learning. We propose a dual-objective learning framework to establish a latent space where temporal consistency emerges from the synergy between semantic discriminability and temporal anchoring. As identified in Section 1 , conventional embeddings often over-enhance trajectory-level temporal trends, inadvertently blurring instantaneous spatial topologies and causing *representation collapse*. To resolve this, we explicitly regularize the time-conditioned encoder f_ϕ and decoder f_ψ via a composite loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{dCAE}}(\phi, \psi) + \lambda_{\text{cl}} \mathcal{L}_{\text{cl}}(\phi), \tag{4}$$

where λ_{cl} balances the two objectives. The first component $\mathcal{L}_{\text{dCAE}}$ is a deterministic conditional autoencoder that reconstructs the state and associated highest return, conditioned on the timestep t:

$$\mathcal{L}_{\text{dCAE}} = \left(H_t - f_\psi^H\left(f_\phi(s_t,t),t\right)\right)^2 + \lambda_{\text{rcon}}\left\|s_t - f_\psi^s\left(f_\phi(s_t,t),t\right)\right\|_2^2, \tag{5}$$

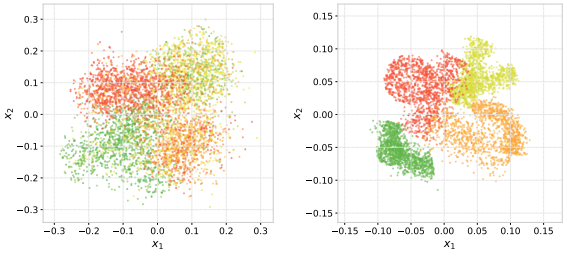
where t is the conditioning timestep and s_t is the global state. H_t denotes the target highest return. The encoder f_ϕ extracts time-anchored state features, while f_ψ^H and f_ψ^s represent the networks for predicting the highest return and reconstructing s_t, respectively. λ_{rcon} is the scale factor for the reconstruction loss. To prevent the aforementioned spatial blurring, we introduce a contrastive loss $\mathcal{L}_{\text{cl}}$ to enforce strict semantic discriminability across the temporal landscape. We adopt the InfoNCE objective Oord et al. (2018) to maximize mutual information between augmented views of the identical spatio-temporal state:

$$\mathcal{L}_{\text{cl}} = -\mathbb{E}_{(s_i,s_j)\sim\mathcal{P}}\left[\log\frac{\exp\left(\text{sim}(z_i,z_j)/\tau_{\text{cl}}\right)}{\sum_{k=1}^{N}\exp\left(\text{sim}(z_i,z_k)/\tau_{\text{cl}}\right)}\right], \tag{6}$$

where function sim denotes cosine similarity, and $z = f_\phi(s,t)$ represents the time-conditioned latent embedding. The temperature parameter τ_{cl} controls the sharpness of the distribution. Crucially, (z_i, z_j) forms a positive pair derived from two different augmented views of the same state at the same timestep (augmentation protocols detailed in Appendix E.8). Conversely, all other embeddings z_k $(k \neq j)$ originating from different timesteps or different states serve as negative samples. Algorithm 1 in Appendix F.1 presents the learning framework for f_ϕ and f_ψ.

Empirical Validation via Latent Space Visualization. Figure 3 presents a t-SNE visualization (Van der Maaten & Hinton (2008)) of state embeddings $x \in \mathcal{D}_E$ sampled from the super-hard *6h_vs_8z* scenario in SMAC. Embeddings are colored by their paired highest return H_t, ranging from red (low) to green (high). For visualization purposes, we divide the continuous return values into four percentile-based intervals with approximately equal numbers of samples, allowing different return levels to be visually compared. This visualization setting is only used to improve interpretability and does not reflect the actual distribution of these intervals in the 1M episodic memory buffer, where proportions are naturally imbalanced. We provide the actual, complete, and specific proportions in the appendix E.1.

As shown in Figure 3a, embeddings generated by dCAE exhibit partial semantic grouping but suffer from cluster overlap between states of differing returns, confirming the representation collapse phenomenon. In contrast, TCSE (Figure 3b) yields markedly more compact and well-separated semantic clusters. By synergizing contrastive learning with time-augmented reconstruction, our latent space effectively captures both instantaneous state features and long-term temporal progression. This enhanced clustering precision eliminates retrieval ambiguity, enabling agents to reliably leverage historical experiences for complex coordination tasks.



(a) dCAE: overlapping clusters.

(b) TCSE: compact, semantically separated clusters.

Figure 3: Latent space comparison in the super-hard SMAC *6h_vs_8z* scenario. The axes x_1 , x_2 denote two reduced abstract dimensions of the t-SNE embedding.

3.2 Temporally Consistent Gating Mechanism

While the semantic embedder ensures precise memory retrieval, blindly distributing episodic incentives remains vulnerable to local optima when recalled trajectories lack sequential coherence. To resolve this, we introduce a *Temporal Consistency Gating Mechanism* that dynamically modulates episodic rewards based on Bellman optimality.

We first establish why temporal consistency serves as a rigorous reliability metric.

Proposition 1 (Reliability Bound of the TCGM)**.** *Let $V^*(s)$ be the true optimal value function. Let $\hat{V}_E(s) \doteq \mathcal{H}(\hat{s})$ denote the episodic value retrieved from the δ-nearest neighbor $\hat{s} \in \mathcal{D}_E$. The empirically observed temporal consistency error $\Delta_{mem} \doteq |r + \gamma\mathcal{H}(\hat{s}') - \mathcal{H}(\hat{s})|$ is strictly bounded*

by:

$$\Delta_{mem} \leq \Delta_{opt} + \alpha, \quad \textit{where} \quad \alpha \doteq (1+\gamma)(\epsilon_{rep} + \epsilon_{store}).$$

Here, $\Delta_{opt} \doteq |r + \gamma V^*(s') - V^*(s)|$ is the true underlying optimality gap of the transition. The aggregated noise term α consists of the representation error $\epsilon_{rep} \doteq \max_s |V^*(s) - V^*(\hat{s})|$ and the memory staleness $\epsilon_{store} \doteq \max_{\hat{s}} |\mathcal{H}(\hat{s}) - V^*(\hat{s})|$. (Proof in Appendix C.1)

Practical Implication. Proposition 1 establishes the mathematical necessity of the synergy between our two proposed components. If the semantic representation were to collapse, the noise term α would explode, causing the gate to falsely penalize optimal trajectories ($\Delta_{\text{opt}} \approx 0$ but Δ_{mem} is huge). However, our TCSE module explicitly minimizes ϵ_{rep} by enforcing a structured latent space. By tightly bounding the noise α, EMTC guarantees that a large Δ_{mem} strictly implies a large Δ_{opt}. Thus, the gating mechanism acts as a safe and unbiased filter, effectively decoupling true exploration signals from representation artifacts.

Mechanism Workflow & Formulation. Building on this insight, the gating mechanism operates through a closed-loop pipeline that directly leverages the structured buffer from Section 3.1:

(i) **Consistency Error Computation:** For each transition (s, a), we retrieve the successor's stored value $\mathcal{H}(s')$ via the δ-nearest neighbor retrieval in $\mathcal{D}_E$. We compute the *Temporal Consistency Error*:

$$\Delta_{\text{mem}}(s, a) = |r(s, a) + \gamma \mathcal{H}(s') - \mathcal{H}(s)|, \tag{7}$$

which quantifies the discrepancy between the one-step Bellman update and the stored memory value.

(ii) **Adaptive Gating:** To translate this error into an incentive constraint, we define the *Consistency Coefficient* $\beta(\Delta_{\text{mem}})$ as a Gaussian modulation function:

$$\beta(\Delta_{\text{mem}}) = \exp\left(-\frac{\Delta_{\text{mem}}(s, a)^2}{2\tau^2}\right), \tag{8}$$

where τ is a temperature parameter (distinct from the contrastive learning temperature τ_{cl}). Our exhaustive hyperparameter experiments reveal that different values of τ exert distinct impacts on the learning dynamics across the early and late stages of training. Motivated by these findings, we adopt a linear annealing strategy for τ (see Appendix E.9 for details). Since $\beta \in (0, 1]$, highly inconsistent transitions ($\Delta_{\text{mem}} \gg 0$) are heavily attenuated ($\beta \to 0$), automatically suppressing misleading incentives.

(iii) **Gated Reward & Loss Integration:** The coefficient dynamically scales the baseline episodic incentive r_p (Eq. (2)) to produce the *Consistency-Aware Episodic Reward* $\tilde{r}_p$:

$$\tilde{r}_p(s, a) = \beta(\Delta_{\text{mem}}) \cdot r_p(s, a) = \beta(\Delta_{\text{mem}}) \cdot \gamma \frac{N_\xi(s')}{N_{\text{call}}(s')} \left(\mathcal{H}(s') - \max_{a'} Q_{\theta^-}(s', a')\right). \tag{9}$$

Integrating $\tilde{r}_p$ into the value factorization objective yields the gated TD loss:

$$\mathcal{L}_\theta^\beta = (r + \tilde{r}_p + \gamma \max_{a'} Q_{\theta^-}(s', a') - Q_{tot}(s, a; \theta))^2 \tag{10}$$

Gradient Analysis & Regulatory Behavior. The gradient of $\mathcal{L}_\theta^\beta$ with respect to θ reveals the mechanism's regulatory dynamics:

$$\begin{aligned} \nabla_\theta \mathcal{L}_\theta^\beta &= -2\nabla_\theta Q_{tot}(s, a; \theta)\,(\Delta_{\text{TD}} + \tilde{r}_p) \\ &= -2\nabla_\theta Q_{tot}(s, a; \theta) \left(\Delta_{\text{TD}} + \beta(\Delta_{\text{mem}}) \cdot \gamma \frac{N_\xi(s')}{N_{\text{call}}(s')} \eta_{\text{val}}(s')\right), \end{aligned} \tag{11}$$

where $\Delta_{\text{TD}} = r + \gamma \max_{a'} Q_{\theta^-}(s', a') - Q_{tot}(s, a; \theta)$ is the standard TD error, and $\eta_{\text{val}}(s') \doteq \mathcal{H}(s') - \max_{a'} Q_{\theta^-}(s', a')$ is the episodic value gap. Here, the gradient signal $\nabla_\theta \mathcal{L}_\theta^\beta$ with the proposed *Consistency-Aware Episodic Reward* $\tilde{r}_p$ can accurately estimate the optimal gradient signal as follows.

Proposition 2 (Asymptotic Unbiasedness). *Let $\nabla_\theta \mathcal{L}_\theta^\beta$ be the gradient induced by the gated reward. Assuming the policy π_θ approaches optimality and the episodic memory matures ($\pi_\theta \to \pi^*$, $\mathcal{H} \to V^*$), the gated gradient accurately recovers the true optimal gradient: $\lim_{\pi_\theta \to \pi^*,\, \mathcal{H} \to V^*} \nabla_\theta \mathcal{L}_\theta^\beta = \nabla_\theta \mathcal{L}_\theta^*$. (Proof in Appendix C.2)*

Proposition 2 confirms that our mechanism functions as a *safe regularizer* with a dual-phase property that directly addresses the limitations of unconstrained episodic MARL:

- **Early-Stage Lenient Exploration:** During initial training, Δ_{mem} is inherently large due to disjointed trajectories. To prevent exploration deadlocks, we utilize a higher initial temperature τ. This lenient gating allows moderately consistent episodic incentives to pass, actively encouraging rapid and directed early exploration.
- **Late-Stage Strict Filtering & Convergence:** As training advances, τ linearly decays to impose a stricter consistency bound, rigorously filtering out spurious pseudo-memories to prevent Q-value overestimation. Concurrently, as high-quality experiences accumulate and $\mathcal{H} \to V^*$, genuine optimal memories yield $\Delta_{\text{mem}} \to 0$ and $\beta \to 1$. The mechanism naturally anneals to an unbiased guidance signal, ensuring the optimization trajectory is never perturbed away from the true optimum.

This dual-phase regulation enables EMTC to maintain stable, directed exploration in super-hard scenarios where traditional episodic incentives typically collapse. The complete gating procedure and its integration into the training loop are detailed in Appendix F.3.

4 EXPERIMENTS

We design our experimental evaluation to systematically address three core research questions:

- **Q1 (Overall Efficacy):** How does EMTC compare to state-of-the-art MARL frameworks?
- **Q2 (Embedder Contribution):** To what extent does the Temporally Consistent Semantic Embedder (TCSE) improve retrieval precision and latent space discriminability?
- **Q3 (Gating Mechanism):** How does the Temporal Consistency Gating Mechanism (TCGM) regulate episodic incentives to ensure stable policy refinement?

Benchmarks. We evaluate EMTC on two widely-adopted MARL benchmarks that jointly cover diverse coordination challenges: (1) **StarCraft Multi-Agent Challenge (SMAC)**(Samvelyan et al. (2019)), featuring heterogeneous unit types, partial observability, and combinatorial action spaces; (2) **Google Research Football (GRF)**(Kurach et al. (2020)), characterized by continuous-state dynamics, sparse goal-based rewards, and long-horizon strategic planning. This dual-domain setup enables comprehensive assessment of generalization across discrete/continuous state spaces and dense/sparse reward regimes. The details of the benchmarks are presented in Appendix D.1.

Baselines and Implementations. Our primary comparison target is **EMU**(Na et al. (2024)), the current state-of-the-art episodic memory framework in MARL. To contextualize performance gains, we additionally include strong value-based baselines: **QMIX**(Rashid et al. (2020)), **QPLEX**(Wang et al. (2020a)), and **CDS**(Li et al. (2021)). Since EMTC is architecture-agnostic and plugs into any CTDE value-factorization backbone, we report results for two instantiations: **EMTC(QPLEX)** and **EMTC(CDS)**. Appendix D.2 and D.3 provides further details of experiment settings and computational overhead. And additional performance comparisons demonstrating EMTC's superiority over recent competitive baselines such as MAPPO (Yu et al., 2022), DMIX (Sun et al., 2021), ACORM (Hu et al., 2024), and TEE (Li & Zhu, 2025) are provided in Appendix E.12.

4.1 Q1. EVALUATION ON STARCRAFT II (SMAC)

Figure 4 presents win-rate curves across six SMAC scenarios of increasing difficulty. Both EMTC(QPLEX) and EMTC(CDS) consistently surpass their backbone architectures and the episodic baseline EMU. Especially in super-hard scenarios (corridor, 6h_vs_8z, 3s5z_vs_3s6z), while maintaining almost the same convergence speed, EMTC achieves an average win-rate gain of 24%.

4.2 Q1. EVALUATION ON GOOGLE RESEARCH FOOTBALL (GRF)

Here, we evaluate EMTC on GRF, which features continuous-state dynamics and extremely sparse goal-based rewards, i.e., a setting markedly different from SMAC. As shown in Figure 5, EMTC maintains its advantage over EMU and value-based baselines, delivering an average win-rate improvement of 28%. Notably, the same two mechanisms (TCSE + TCGM) that proved effective in SMAC here preserve spatial semantics under partial observability and filter inconsistent incentives in sparse-reward regimes. This cross-environment consistency confirms that temporally constrained

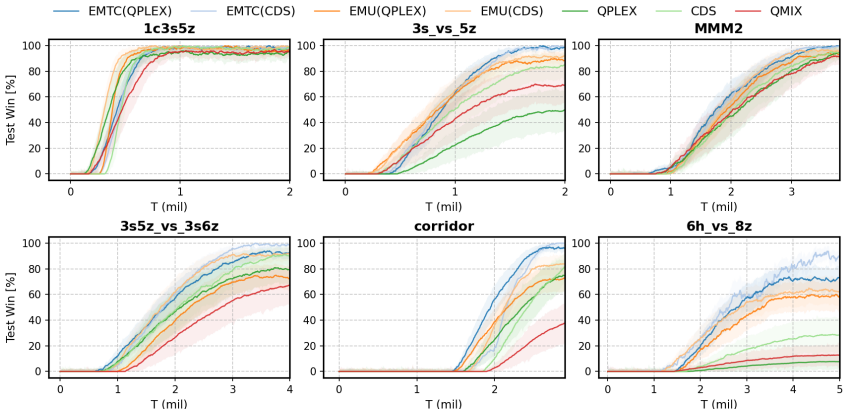


Figure 4: Performance comparison of EMTC against EMU and other state-of-the-art baselines on SMAC scenarios. The evaluation includes easy and hard maps (1c3s5z, 3s_vs_5z) and super-hard maps (MMM2, 3s5z_vs_3s6z, corridor, and 6h_vs_8z). Notably, T represents the training time in millions of steps in all figures.

episodic guidance generalizes beyond discrete-action benchmarks to complex, real-world-inspired coordination tasks. See Appendix E.3 for detailed quantitative results.

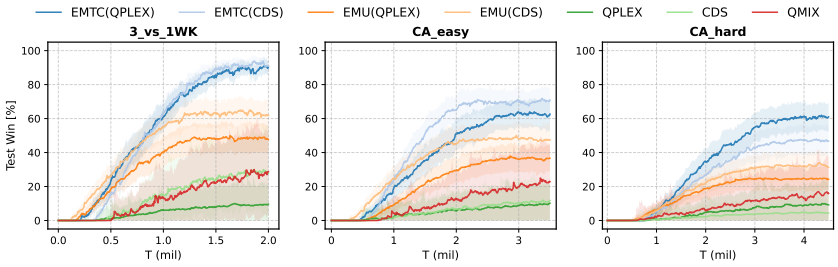
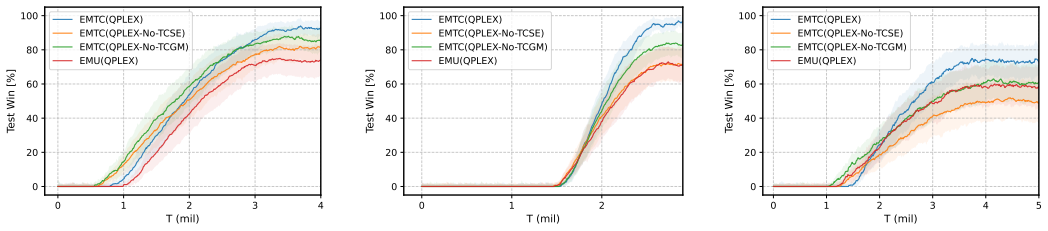


Figure 5: Performance comparison of EMTC against baseline algorithms on GRF.

4.3 Q2 & Q3. Further Ablation Study



(a) *3s5z_vs_3s6z* (b) *corridor* (c) *6h_vs_8z*

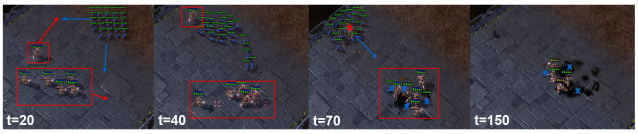
Figure 6: Further ablation studies on complex MARL tasks.

We conduct comprehensive ablation studies to evaluate the individual contributions of the core components. Starting from the full **EMTC (QPLEX)** model, we remove the gating mechanism to investigate the effect of unregulated incentives, denoted as **EMTC (QPLEX-No-TCGM)**. We additionally ablate the TCSE component, denoted as **EMTC (QPLEX-No-TCSE)**, to assess the impact of representation quality on memory retrieval efficiency. Furthermore, we include **EMU**

(QPLEX) as a prominent baseline to highlight the performance gains achieved by our temporal consistency constraints. The performance of each variant is evaluated on three super-hard SMAC scenarios: *3s5z_vs_3s6z*, *corridor*, and *6h_vs_8z*. As illustrated in Figure 6, the experimental results demonstrate that the removal of either component leads to a significant degradation in win rates, underscoring the synergy between robust embedding and rigorous incentive regulation. More ablation experiments on embedder retrieval accuracy and state-space coverage are provided in Appendix E.10 and E.11. The detailed ablation study on TCSE is provided in Appendix E.5.

4.4 Q3. Qualitative Analysis on TCGM

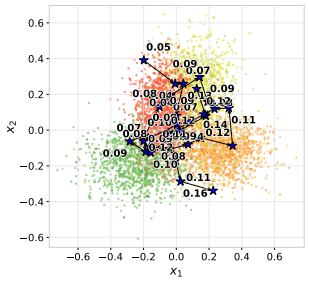
We qualitatively investigate how the Temporal Consistency Gating Mechanism (TCGM) regulates policy learning by visualizing the gameplay dynamics alongside the latent memory buffer $\mathcal{D}_E$. We conduct our analysis on the challenging *corridor* scenario, where agents must navigate complex spatial constraints. During exploration, two strategic patterns emerge: a successful asymmetrical 1-5 split, where one agent baits enemies while the others encircle and eventually eliminate all opponents (Figure 7a); and a sub-optimal even 3-3 split, which lacks sufficient concentrated firepower and consistently fails to defeat the enemies (Figure 7b).



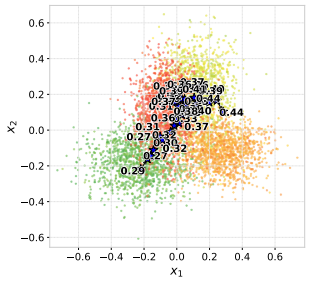
(a) Effective 1-5 asymmetrical split (Bait-and-encircle)



(b) Sub-optimal 3-3 even split



(c) Good Strategy (1-5 split)



(d) Bad Strategy (3-3 split)

Figure 7: Overall analysis in *corridor*. (a)&(b) Strategy divergence; (c)&(d) t-SNE projections of the corresponding memory retrieval trajectories. The axes x_1 and x_2 denote the two reduced abstract dimensions of the t-SNE embedding.

To understand TCGM's evaluation, we project $\mathcal{D}_E$ via t-SNE and trace memory retrievals during the first 200 steps (sampled every 10 steps), where the strategic divergence occurs. As shown in Figures 7c and 7d, the superior 1-5 split traverses a smooth latent region with consistently low Δ_{mem}, prompting TCGM to assign $\beta \to 1$ and preserve the episodic incentive. Conversely, the flawed 3-3 split retrieves disjointed memories with dramatically larger Δ_{mem} (annotated in Figure 7d), causing TCGM to aggressively attenuate the reward ($\beta \to 0$) and shield the policy from misleading guidance. This contrast validates that TCGM acts as a temporally-grounded safeguard, propagating episodic incentives only when they are reliable. Appendix E.6 details the training dynamics of Δ_{mem} and β.

5 Conclusion

In this paper, we address the severe reward sparsity and limitations of traditional episodic memory in cooperative Multi-Agent Reinforcement Learning (MARL). To prevent agents from being trapped in local optima caused by blind incentive distributions, we propose Episodic Memory Temporal Consistency (EMTC). EMTC utilizes a contrastive learning-enhanced architecture to construct a temporally consistent semantic embedder, and introduces a novel gating mechanism to selectively distribute incentives. This approach effectively filters out misleading signals and mitigates TD target overestimation in Q-learning. Supported by theoretical guarantees, extensive evaluations on the GRF and SMAC benchmarks demonstrate that EMTC significantly outperforms conventional paradigms. A detailed discussion of the limitations and future directions of this work is provided in Appendix A.

AI USAGE STATEMENT

We used AI-based language tools solely for proofreading and polishing the manuscript, including improving grammar, clarity, and overall readability. All research ideas, methodology, experiments, analyses, and conclusions were developed and verified by the authors. The authors take full responsibility for the content of this paper.

REFERENCES

Guangyan Bao, Lifeng Ma, and Xiaojian Yi. Recent advances on cooperative control of heterogeneous multi-agent systems subject to constraints: A survey. *Systems Science & Control Engineering*, 10 (1):539–551, 2022.

Andrew G Barto. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, pp. 17–47. Springer, 2012.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Charles Blundell, Benigno Uria, Alexander Pritzel, Yazhe Li, Avraham Ruderman, Joel Z Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. Model-free episodic control. *arXiv preprint arXiv:1606.04460*, 2016.

Junhong Chen, Ziqi Yang, Haoyuan G Xu, Dandan Zhang, and George Mylonas. Multi-agent systems for robotic autonomy with llms. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4194–4204. IEEE, 2025.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. Pmlr, 2018.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International conference on autonomous agents and multiagent systems*, pp. 66–83. Springer, 2017.

Ruiyang Hao, Haibao Yu, Jiaru Zhong, Chuanye Wang, Jiahao Wang, Yiming Kan, Wenxian Yang, Siqi Fan, Huilin Yin, Jianing Qiu, et al. Research challenges and progress in the end-to-end v2x cooperative autonomous driving competition. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 1849–1860. IEEE, 2025.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Shixin Hong, Yu Liu, Zhi Li, Shaohui Li, and You He. Multi-agent collaborative perception via motion-aware robust communication network. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15301–15310. IEEE, 2024.

Zican Hu, Zongzhang Zhang, Huaxiong Li, Chunlin Chen, Hongyu Ding, and Zhi Wang. Attention-guided contrastive role representations for multi-agent reinforcement learning. In *International conference on learning representations*, volume 2024, pp. 23627–23649, 2024.

Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4501–4510, 2020.

Máté Lengyel and Peter Dayan. Hippocampal contributions to control: the third way. *Advances in neural information processing systems*, 20, 2007.

Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3991–4002, 2021.

Tianxu Li and Kun Zhu. Toward efficient multi-agent exploration with trajectory entropy maximization. In *International Conference on Learning Representations*, volume 2025, pp. 21863–21882, 2025.

Zichuan Lin, Tianqi Zhao, Guangwen Yang, and Lintao Zhang. Episodic memory deep q-networks. *arXiv preprint arXiv:1805.07603*, 2018.

Jinzhu Luo, Dingyang Chen, and Qi Zhang. Reinforcement learning with euclidean data augmentation for state-based continuous control. *Advances in Neural Information Processing Systems*, 37:90253–90276, 2024.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in neural information processing systems*, 32, 2019.

Hyungho Na, Yunkyeong Seo, and Il-chul Moon. Efficient episodic memory utilization of cooperative multi-agent reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 31483–31515, 2024.

Michal Nauman, Michał Bortkiewicz, Piotr Miłoś, Tomasz Trzciński, Mateusz Ostaszewski, and Marek Cygan. Overestimation, overfitting, and plasticity in actor-critic: the bitter lesson of reinforcement learning. *arXiv preprint arXiv:2403.00514*, 2024.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *International conference on machine learning*, pp. 2827–2836. PMLR, 2017.

Yiran Qin, Li Kang, Xiufeng Song, Zhenfei Yin, Xiaohong Liu, Xihui Liu, Ruimao Zhang, and Lei Bai. Robofactory: Exploring embodied agent collaboration with compositional constraints. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10075–10085. IEEE, 2025.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21:1–51, 2020.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Wei-Fang Sun, Cheng-Kuang Lee, and Chun-Yi Lee. Dfac framework: Factorizing the value function via quantile mixture for multi-agent distributional q-learning. In *International conference on machine learning*, pp. 9945–9954. PMLR, 2021.

Richard S Sutton, Andrew G Barto, and Andrew Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Huaiqing Wang, Stephen Liao, and Lejian Liao. Modeling constraint-based negotiating agents. *Decision Support Systems*, 33(2):201–217, 2002.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020a.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523*, 2020b.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Taoyuan Yu, Kui Wang, Zongdian Li, Tao Yu, and Kei Sakaguchi. Multi-agent reinforcement learning-based cooperative autonomous driving in smart intersections. *arXiv preprint arXiv:2505.04231*, 2025.

Xin Yu, Rongye Shi, Pu Feng, Yongkai Tian, Jie Luo, and Wenjun Wu. Esp: Exploiting symmetry prior for multi-agent reinforcement learning. In *ECAI 2023: 26th European Conference on Artificial Intelligence, September 30–October 4, 2023, Kraków, Poland–Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*, pp. 2946–2953. SAGE Publications 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2023.

Xin Yu, Yongkai Tian, Li Wang, Pu Feng, Wenjun Wu, and Rongye Shi. Adaptaug: Adaptive data augmentation framework for multi-agent reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10814–10820. IEEE, 2024.

Cong Zhang, Xin Deik Goh, Dexun Li, Hao Zhang, and Yong Liu. Planning with multi-constraints via collaborative language agents. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10054–10082, 2025.

Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34:3757–3769, 2021.

A Limitations and Future Work

While the proposed Episodic Memory Temporal Consistency (EMTC) framework demonstrates significant improvements in sample efficiency and asymptotic performance across challenging cooperative MARL benchmarks, it is not without limitations. We explicitly acknowledge the following aspects, which present exciting avenues for future research:

Heuristic Annealing of Hyperparameters. In the Temporal Consistency Gating Mechanism (TCGM), the temperature parameter τ dictates the strictness of the consistency filter. Currently, EMTC adopts a linear annealing schedule for τ (as detailed in Appendix E.9) to balance early-stage exploration and late-stage exploitation. While empirically effective, this fixed-step annealing strategy ($\mathcal{T}_{decay}$) is largely heuristic and may require environment-specific tuning. Developing an adaptive scheduling mechanism, potentially based on the rolling variance of the temporal consistency error Δ_{mem}, would make the framework more robust and plug-and-play.

Theoretical Gap in Lipschitz Continuity. As established in Proposition 1 and Equation 17, our theoretical guarantees rely on the assumption that the true optimal value function V^* is L-Lipschitz continuous with respect to the learned latent space. While our TCSE module actively enforces a well-behaved and semantically clustered latent space to support this assumption, standard deep neural networks are not inherently Lipschitz bounded. In highly stochastic environments with sparse, discrete rewards, local discontinuities may still occur. Future work could explore incorporating explicit Lipschitz regularization techniques into the semantic embedder.

B Related Works

This section reviews related work on incentive generation for exploration, as well as on constraints and overestimation regulation in multi-agent reinforcement learning.

B.1 Incentive for Multi-Agent Exploration

Balancing exploration and exploitation in MARL is notoriously difficult due to the combinatorial explosion of the joint action space. To address this, various intrinsic motivation and exploration incentive mechanisms have been proposed. For instance, MAVEN (Mahajan et al. (2019)) and CDS (Li et al. (2021)) leverage mutual information to encourage diverse and individualized behaviors. EMC (Zheng et al. (2021)) introduces curiosity-driven exploration by utilizing prediction errors of individual Q-values as intrinsic rewards. Beyond parametric intrinsic rewards, episodic control (Blundell et al. (2016); Pritzel et al. (2017)) has been integrated into MARL to directly exploit high-return historical trajectories for sample-efficient learning. EMDQN (Lin et al. (2018)) utilized episodic memory to accelerate conventional Q-learning, an approach later extended to cooperative MARL frameworks. Recently, EMU (Na et al. (2024)) proposed explicit episodic incentives based on trajectory desirability to rapidly guide exploration toward successful states. However, these conventional frameworks typically inject episodic incentives indiscriminately. By relying on coarse, episode-level desirability metrics, they neglect the underlying temporal causality and semantic consistency of individual state transitions, frequently leaving agents vulnerable to suboptimal entrapment.

B.2 Constraints and Overestimation Regulation in MARL

Incorporating constraints into multi-agent reinforcement learning is essential for safe, robust, and practical coordination. A prominent line of work formulates the problem as a constrained cooperative control problem, where agents must optimize a shared objective while strictly satisfying operational or safety restrictions, as surveyed by (Bao et al. (2022)). Early approaches modelled constraints within negotiation-based agent interactions (Wang et al. (2002)), while more recent works have introduced compositional constraints for embodied collaboration (Qin et al. (2025)) and planning with multiple constraints via collaborative language agents (Zhang et al. (2025)).

Beyond environmental safety, another critical algorithmic constraint in MARL lies in regulating Q-value estimates. Due to decentralized execution and the max operator over vast joint action spaces, value-based MARL is inherently susceptible to severe overestimation bias. Recent analysis has

identified overestimation, together with overfitting and plasticity loss, as a principal failure mode in modern actor-critic methods (Nauman et al. (2024)), motivating explicit constraints on the learning dynamics.

C Mathematical Proof

C.1 Proof of Proposition 1

To formally establish the reliability of the temporal consistency error Δ_{mem} as a gating signal, we must evaluate its relationship with the true transition suboptimality while accounting for representation and storage inaccuracies.

Error Decomposition of the Episodic Target. During the learning process, for any encountered transition (s, a, r, s'), the algorithm retrieves the highest empirical return $\mathcal{H}$ from the δ-nearest neighbors in the latent space, defined as $\hat{V}_E(s) \doteq \mathcal{H}(\hat{s})$ and $\hat{V}_E(s') \doteq \mathcal{H}(\hat{s}')$. The total approximation error at state s can be decoupled using the triangle inequality:

$$|\hat{V}_E(s) - V^*(s)| = |\mathcal{H}(\hat{s}) - V^*(s)| \leq |\mathcal{H}(\hat{s}) - V^*(\hat{s})| + |V^*(\hat{s}) - V^*(s)|. \tag{12}$$

This error inherently consists of two components: the storage error $\epsilon_{\text{store}}(\hat{s}) \doteq |\mathcal{H}(\hat{s}) - V^*(\hat{s})|$ and the representation error $\epsilon_{\text{rep}}(s) \doteq |V^*(\hat{s}) - V^*(s)|$. Defining the global maximum over the state space as ϵ_{store} and ϵ_{rep}, the total approximation error is strictly bounded by $\epsilon_{\text{rep}} + \epsilon_{\text{store}}$.

Bounding the Temporal Consistency Error. The temporal consistency error utilized by our gating mechanism is calculated as $\Delta_{\text{mem}} = |r + \gamma \hat{V}_E(s') - \hat{V}_E(s)|$. To evaluate its reliability, we inject the optimal value function V^* by adding and subtracting $V^*(s)$ and $\gamma V^*(s')$:

$$\Delta_{\text{mem}} = \left| (r + \gamma V^*(s') - V^*(s)) + \gamma(\hat{V}_E(s') - V^*(s')) - (\hat{V}_E(s) - V^*(s)) \right|. \tag{13}$$

By applying the triangle inequality, we isolate the true underlying optimality gap of the executed transition, denoted as $\Delta_{\text{opt}} \doteq |r + \gamma V^*(s') - V^*(s)|$:

$$\Delta_{\text{mem}} \leq \Delta_{\text{opt}} + \gamma |\hat{V}_E(s') - V^*(s')| + |\hat{V}_E(s) - V^*(s)|. \tag{14}$$

Substituting the maximum approximation error bounds derived above, we obtain:

$$\Delta_{\text{mem}} \leq \Delta_{\text{opt}} + (1 + \gamma)(\epsilon_{\text{rep}} + \epsilon_{\text{store}}). \tag{15}$$

For simplicity, we define the aggregated noise induced by representation and storage as $\alpha \doteq (1 + \gamma)(\epsilon_{\text{rep}} + \epsilon_{\text{store}})$, allowing us to concisely express the upper bound as:

$$\Delta_{\text{mem}} \leq \Delta_{\text{opt}} + \alpha. \tag{16}$$

The Role of Temporally Consistent Semantic Embedding. Equation (16) reveals the precise operational condition required for a safe gating mechanism. If the representation collapses, the retrieved state $\hat{s}$ severely mismatches the true semantic state s, causing ϵ_{rep} and consequently α to explode. Under such representation failures, a genuinely optimal trajectory ($\Delta_{\text{opt}} \approx 0$) might still yield a massive Δ_{mem}, causing the gate to erroneously penalize high-quality experiences. Crucially, this failure mode is explicitly prevented by the Temporally Consistent Semantic Embedder (TCSE). Assuming V^* is L-Lipschitz continuous with respect to the latent space, the representation error is theoretically bounded by the retrieval radius δ:

$$\epsilon_{\text{rep}} \leq L \| f_\phi(\hat{s}) - f_\phi(s) \|_2 \leq L\delta. \tag{17}$$

The dual-objective optimization of TCSE actively prevents representation collapse and preserves semantic discriminability, ensuring that the Lipschitz constant L remains well-behaved and that the δ-radius retrieval successfully identifies true semantic equivalents. Consequently, TCSE strictly limits ϵ_{rep}, while periodic memory updates incrementally minimize ϵ_{store}.

Reliability of the Gating Mechanism. This algorithmic synergy guarantees that the joint noise α is systematically minimized and bounded at a low magnitude throughout training. When encountering a misleading pseudo-successful trajectory, the true suboptimality is large ($\Delta_{\text{opt}} \gg \alpha$), ensuring that Δ_{mem} accurately tracks Δ_{opt} and safely activates the penalty. Conversely, for a genuinely high-quality trajectory where $\Delta_{\text{opt}} \approx 0$, the bounded nature of α ensures that Δ_{mem} remains sufficiently small ($\Delta_{\text{mem}} \leq \alpha \approx 0$). This yields a gating coefficient $\beta \to 1$, preserving the optimal reward injection. Thus, our framework rigorously proves that effective incentive gating is strictly contingent upon robust semantic representation. ■

C.2 Proof of Proposition 2

Recall from Eq. (11) that the gradient signal derived from the consistency-aware episodic reward is given by:

$$\nabla_\theta \mathcal{L}_\theta^\beta = -2\nabla_\theta Q_{tot}(s,a;\theta)\left(\Delta_{\text{TD}} + \tilde{r}_p(s,a)\right), \tag{18}$$

where the standard one-step TD error is $\Delta_{\text{TD}} = r + \gamma \max_{a'} Q_{\theta^-}(s',a') - Q_{tot}(s,a;\theta)$, and the gated episodic incentive is defined as:

$$\tilde{r}_p(s,a) = \beta(\Delta_{\text{mem}}) \cdot \gamma \frac{N_\xi(s')}{N_{\text{call}}(s')} \left(\mathcal{H}(s') - \max_{a'} Q_{\theta^-}(s',a')\right). \tag{19}$$

Asymptotic Behaviour Setup. To analyze the asymptotic behavior, consider the training process as the joint policy converges to the optimal policy ($\pi_\theta \to \pi^*$) and the memory buffer matures.

Convergence of the Gating Coefficient. According to **Proposition 1**, as the episodic operator continuously updates under adequate exploration, the memory transitions inherently satisfy the Bellman optimality equation, causing the temporal consistency error to vanish ($\Delta_{\text{mem}} \to 0$). Based on the definition of our Gaussian-style gating function in Eq. (8), this ensures that the consistency coefficient asymptotically approaches exactly 1:

$$\lim_{\mathcal{H}\to V^*} \beta(\Delta_{\text{mem}}) = 1. \tag{20}$$

Convergence of the Desirability Ratio. When the actions follow the optimal policy $a \sim \pi^*$, every retrieved trajectory in $\mathcal{D}_E$ naturally leads to a desirable outcome. Thus, the desirability indicator $\xi(s') = 1$ consistently holds. As a result, the ratio of desirable transitions to total visits converges to 1:

$$\lim_{\pi_\theta \to \pi^*} \frac{N_\xi(s')}{N_{\text{call}}(s')} = 1. \tag{21}$$

Asymptotic Reward Recovery. By substituting the limits from Eq. (20) and Eq. (21) into Eq. (19), the asymptotic behavior of the gated episodic reward becomes:

$$\lim_{\pi_\theta\to\pi^*, \mathcal{H}\to V^*} \tilde{r}_p(s,a) = \gamma\left(V^*(s') - \max_{a'} Q_{\theta^-}(s',a')\right). \tag{22}$$

Gradient Consistency Proof. Finally, we substitute this converged incentive back into the overall gradient equation (Eq. (18)):

$$\begin{aligned}
\lim_{\pi_\theta\to\pi^*, \mathcal{H}\to V^*} \nabla_\theta \mathcal{L}_\theta^\beta &= -2\nabla_\theta Q_{tot}(s,a;\theta)\left[\Delta_{\text{TD}} + \lim_{\pi_\theta\to\pi^*, \mathcal{H}\to V^*} \tilde{r}_p(s,a)\right] \\
&= -2\nabla_\theta Q_{tot}(s,a;\theta)\Big[r + \gamma \max_{a'} Q_{\theta^-}(s',a') - Q_{tot}(s,a;\theta) \\
&\qquad + \gamma V^*(s') - \gamma \max_{a'} Q_{\theta^-}(s',a')\Big] \\
&= -2\nabla_\theta Q_{tot}(s,a;\theta)\left[r + \gamma V^*(s') - Q_{tot}(s,a;\theta)\right] \\
&= \nabla_\theta \mathcal{L}_\theta^*.
\end{aligned} \tag{23}$$

Unbiasedness Conclusion. Therefore, as the policy reaches optimality, the consistency-aware gradient $\nabla_\theta \mathcal{L}_\theta^\beta$ perfectly cancels out the estimation bias introduced by the target network $\max_{a'} Q_{\theta^-}(s',a')$, and gracefully degenerates to the unbiased optimal gradient signal $\nabla_\theta \mathcal{L}_\theta^*$ guided directly by $V^*(s')$. This proves that the gating mechanism preserves the original optimization fixed point without introducing any asymptotic bias. ■

D Implementation and Experiment Details

D.1 Supplementary Information on the Experimental Environment

In this section, we detail the environmental configurations, including the dimensions of the state and action spaces for both the StarCraft Multi-Agent Challenge (SMAC) Samvelyan et al. (2019) and Google Research Football (GRF) Kurach et al. (2020) benchmarks.

D.1.1 STARCRAFT MULTI-AGENT CHALLENGE (SMAC)

In SMAC, the global state encompasses the exact coordinates of all agents along with the specific features of both allied and enemy units. The discrete action space includes movement directions and targeted attacks, meaning its size naturally scales with the number of enemies in a given scenario. As summarized in Table 1, the exact dimensions of the state and action spaces vary across different maps. We utilize the default shaped reward structure for all scenarios, which grants positive signals for dealing damage to enemies and an additional bonus for winning the battle.

Table 1: Dimension of the state space and the action space of SMAC

Task	**Dimension of state space**	**Dimension of action space**
1c3s5z	270	15
3s5z	216	14
3s_vs_5z	68	11
5m_vs_6m	98	12
MMM2	322	18
6h_vs_8z	140	14
3s5z_vs_3s6z	230	15
corridor	282	30

D.1.2 GOOGLE RESEARCH FOOTBALL (GRF)

In the GRF environment, the global state provides comprehensive match information, including player coordinates, orientations, and ball possession. While the state dimension varies by scenario, the action space remains strictly uniform across all tasks. Each agent can execute a predefined set of actions, including moving, various kicking techniques, sprinting, intercepting, and dribbling. Table 2 summarizes these dimensionalities. Regarding the reward function, GRF offers both dense and sparse modes. In our experiments, we strictly adopt the challenging sparse reward setting, where agents receive a reward of $+1$ for scoring a goal and -1 for conceding one.

Table 2: Dimension of the state space and the action space of GRF

Task	**Dimension of state space**	**Dimension of action space**
3_vs_1WK	26	19
CA_easy	30	19
CA_hard	34	19

D.2 EXPERIMENT DETAILS

We utilize PyMARL Samvelyan et al. (2019) to execute all of the baseline algorithms with their open-source codes, and the same hyperparameters are used for experiments if they are presented either in uploaded codes or in their manuscripts. All SMAC experiments were conducted on StarCraft II version 4.10.0 in a Linux environment.

For Google Research Football task, we use the environmental code provided by Wang et al. (2020b). In the experiments, we consider three official scenarios such as academy_3_vs_1_with_keeper (3_vs_1WK), academy_counterattack_easy (CA_easy), and academy_counterattack_hard (CA_hard).

Win-rate is computed with 80 samples: 16 episodes for each training random seed, and 5 different random seeds unless denoted otherwise. Both the mean and the variance of the performance are presented for all the figures to show their overall performance according to different seeds. EMTC set task-dependent δ values as presented in Table 3. For other hyperparameters introduced by EMTC, the same values presented in Table 4 are used throughout all tasks. For EMTC (QPLEX) in *corridor*, $\delta = 1.8 \times 10^{-5}$ is used instead of $\delta = 1.8 \times 10^{-3}$. Appendix E.9 presents the discussion about τ.

Table 3: Task-dependent Hyperparameter of EMTC.

Category	δ
easy/hard SMAC maps	1.8×10^{-5}
super hard SMAC maps	1.8×10^{-3}
GRF	1.8×10^{-3}

Table 4: Other EMTC Hyperparameters

Configuration	Value
a scale factor of reconstruction loss, λ_{rcon}	0.1
a scale factor of contrastive loss, λ_{cl}	0.1
a scale factor of contrastive learning, τ_{cl}	0.1
update interval, t_{emb}	1K
training samples, N	102.4K
batch size of training, B	1024
episodic latent dimension, $\dim(x)$	4
episodic memory capacity	1M
a scale factor, λ (for conventional episodic control only)	0.1

D.3 Infrastructure and Computational Overhead

All experiments for the SMAC and GRF environments were conducted on a computing cluster equipped with NVIDIA A100 Tensor Core GPUs and high-performance CPUs.

In terms of computational overhead, EMTC is highly efficient and comparable to the state-of-the-art baseline, EMU. For the most computationally demanding super-hard scenario in SMAC, such as *corridor*, the total training process takes approximately 15 hours on a single A100 GPU.

Although EMTC introduces an additional contrastive learning objective ($\mathcal{L}_{cl}$) and a Temporal Consistency Gating Mechanism (β), they do not introduce significant computational bottlenecks. Specifically:

- **Periodic Embedder Update:** Similar to EMU, the semantic embedder (f_ϕ and f_ψ) is updated periodically rather than at every timestep. Benefiting from the high memory bandwidth and Tensor Cores of the A100 GPU, the forward-backward pass for the dual-objective learning (time-augmented reconstruction and contrastive InfoNCE) and the subsequent re-projection of the episodic buffer $\mathcal{D}_E$ are extremely fast, taking less than 2 seconds per update cycle.
- **Gating Mechanism Calculation:** The computation of the temporal consistency error Δ_{mem} and the gating coefficient $\beta(\Delta_{\text{mem}})$ is essentially an $\mathcal{O}(1)$ scalar operation executed during the standard one-step TD inference. It naturally integrates into the vectorized Bellman updates on the GPU, incurring virtually zero extra wall-clock time.

To further verify that our embedder TCSE introduces no additional inference latency compared to the conventional dCAE encoder (as used in EMU), we benchmarked the per-step forward time of both architectures across representative state dimensions and batch sizes. As summarized in Table 5, the TCSE encoder achieves virtually identical computational efficiency with an average speedup of $1.01\times$ over the dCAE baseline and a memory footprint ratio of $0.99\times$. This result confirms that the enhanced representation quality of TCSE comes at essentially zero extra cost at inference time.

Consequently, the extra computational time required for our temporally consistent memory module is strictly marginal, establishing EMTC as a highly practical algorithm for complex MARL tasks.

Table 5: Per-step Forward Time Comparison.

State Dim	Batch Size	Seq Len	dCAE Time (ms)	TCSE Time (ms)	Speedup
128	1	1	0.100	0.090	0.90×
128	8	10	0.110	0.110	1.00×
128	32	50	0.090	0.099	1.10×
128	64	100	0.110	0.080	0.73×
128	64	1	0.105	0.100	0.95×
512	1	10	0.135	0.110	0.81×
512	8	50	0.138	0.116	0.84×
512	32	10	0.106	0.090	0.85×
512	64	1	0.111	0.130	1.17×
512	64	50	0.070	0.081	1.16×
Overall Average			—	—	**1.01**×

E Further Experiment Result

E.1 Visualization Settings

To provide further clarity of specific proportions, we conducted a profiling experiment on the 1M episodic buffer in map 6h_vs_8z, calculating memory proportions and mean values for each interval.

Table 6: Distribution of return intervals in the 1M episodic memory buffer on `6h_vs_8z`.

Interval	Range	Mean Value	Ratio
Red	$[0, 4]$	2.3	9.4%
Orange	$(4, 10]$	7.5	33.0%
Yellow	$(10, 14]$	12.4	31.7%
Green	$(14, 18]$	16.8	25.9%

As shown in Table 6, medium and high-return experiences (returns > 4) constitute over 90.6% of the episodic buffer $\mathcal{D}_E$, while low-return transitions (≤ 4) occupy merely 9.4%. Notably, elite trajectories make up 25.9%, confirming that our buffer mechanism effectively filters out uninformative exploratory noise while actively preserving reliable high-value guidance.

E.2 Indicator Explanation

To comprehensively evaluate the performance of our proposed method, relying solely on the final test win-rate at the end of training is insufficient, as it fails to capture the learning efficiency. Therefore, following the evaluation protocol introduced in EMU Na et al. (2024), we adopt a composite evaluation metric denoted as the Normalized Overall Win-Rate, $\bar{\mu}_w(t)$.

Let $f_w^i(s)$ define the test win-rate achieved by the agent at training step s under the i-th random seed. As illustrated in Figure 8, the cumulative performance of a single training run up to time t can be geometrically interpreted as the area under its learning curve, calculated by the integral $\mu_w^i(t) = \int_0^t f_w^i(s)ds$. To aggregate this performance across n independent runs and bound the metric within the range $[0, 1]$, we average these integrals and divide by the theoretical maximum area $\mu_{max} = t$. The formal calculation is defined as:

$$\bar{\mu}_w(t) = \frac{1}{t \cdot n} \sum_{i=1}^{n} \int_0^t f_w^i(s)ds \tag{24}$$

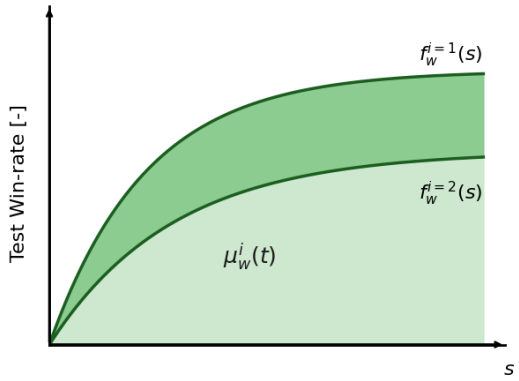


Figure 8: Visual illustration of the Normalized Overall Win-Rate $\bar{\mu}_w(t)$ and s is denoted as the training time. The shaded area corresponds to the cumulative test win-rate $\int_0^t f_w^i(s)ds$. By calculating this area under the curve, $\bar{\mu}_w(t)$ evaluates not only the final asymptotic win-rate but also the learning efficiency, rewarding algorithms that converge faster to optimal policies.

E.3 Detailed Quantitative Results

To complement the learning curves presented in Section 4.2, we provide the detailed quantitative results of all evaluated algorithms across the StarCraft Multi-Agent Challenge (SMAC) and Google Research Football (GRF) environments.

Table 7 and Table 8 summarize the median of the final test win rates across multiple independent runs with different random seeds. The results clearly demonstrate that EMTC consistently achieves the highest final performance.

Table 7: Final test win rates (%) on six SMAC scenarios. Results are shown as mean ± standard deviation across multiple independent runs. The best performances are highlighted in **bold**.

Algorithm	1c3s5z	3s_vs_5z	MMM2	3s5z_vs_3s6z	corridor	6h_vs_8z
EMTC (CDS)	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{98.4 \pm 1.5}$	$\mathbf{99.2 \pm 0.5}$	$\mathbf{89.4 \pm 3.2}$
EMTC (QPLEX)	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	99.3 ± 0.3	94.0 ± 2.1	97.4 ± 1.7	75.0 ± 4.8
EMU (CDS)	$\mathbf{100.0 \pm 0.0}$	92.1 ± 2.7	99.3 ± 0.5	92.5 ± 2.5	84.0 ± 3.8	65.2 ± 5.6
EMU (QPLEX)	$\mathbf{100.0 \pm 0.0}$	90.3 ± 2.9	98.6 ± 1.3	78.2 ± 5.2	73.4 ± 5.0	60.5 ± 6.0
CDS	$\mathbf{100.0 \pm 0.0}$	85.0 ± 4.8	95.1 ± 3.4	91.1 ± 3.0	81.5 ± 4.3	29.4 ± 9.8
QPLEX	96.8 ± 2.1	50.6 ± 8.4	96.9 ± 2.7	81.3 ± 4.5	75.0 ± 5.0	7.8 ± 2.8
QMIX	97.6 ± 2.0	70.4 ± 5.6	92.4 ± 3.7	67.2 ± 6.0	37.6 ± 9.8	12.8 ± 4.4

Table 8: Final test win rates (%) on three Google Research Football (GRF) scenarios. Results are shown as mean ± standard deviation across multiple independent runs.

Algorithm	3_vs_1WK	CA_easy	CA_hard
EMTC (CDS)	$\mathbf{94.5 \pm 1.4}$	$\mathbf{72.1 \pm 2.2}$	48.0 ± 2.8
EMTC (QPLEX)	91.3 ± 1.3	64.1 ± 2.3	$\mathbf{62.6 \pm 2.5}$
EMU (CDS)	65.1 ± 3.2	50.4 ± 3.5	34.8 ± 3.0
EMU (QPLEX)	50.2 ± 4.6	37.9 ± 4.0	25.0 ± 3.4
CDS	30.3 ± 7.1	12.7 ± 2.2	5.1 ± 4.0
QPLEX	10.9 ± 3.2	10.0 ± 3.1	10.2 ± 3.5
QMIX	30.2 ± 7.4	25.6 ± 5.7	17.0 ± 6.8

E.4 Ablation on TCSE without TCGM

In the experimental evaluation presented in Section 4, we utilized the composite metric $\bar{\mu}_w(t)$ to investigate the embedding efficacy on the `3s_vs_5z` and `6h_vs_8z` maps. In this section, we extend this analysis to further isolate and rigorously evaluate the representation quality of different semantic embedders. Specifically, we focus on two distinct scenarios: the relatively simple map `3s_vs_5z` and the super-hard map `corridor`.

To ensure that the performance variations are strictly attributable to the embedding structures themselves, we completely deactivate the Temporal Consistency Gating Mechanism (β) for these evaluations. We compare three distinct embedding designs: Random Projection, deterministic Conditional Autoencoder (dCAE), and our proposed Temporally Consistent Semantic Embedder (TCSE).

Table 9: $\bar{\mu}_w$ according to δ and design choice of embedding function on **easy** SMAC map, `3s_vs_5z`.

Steps[mil]	0.8			1.6			2.4		
δ	random	dCAE	TCSE	random	dCAE	TCSE	random	dCAE	TCSE
1.8e-7	0.087	0.102	**0.119**	0.211	0.283	**0.326**	0.401	0.501	**0.550**
1.8e-5	0.083	0.112	**0.120**	0.225	0.298	**0.315**	0.424	0.527	**0.541**
1.8e-3	0.090	0.091	**0.102**	0.237	0.244	**0.280**	0.446	0.442	**0.514**
1.8e-2	0.075	**0.102**	0.098	0.201	**0.265**	0.252	0.384	**0.470**	0.458

Table 10: $\bar{\mu}_w$ according to δ and design choice of embedding function on **super-hard** SMAC map, `corridor`.

Steps[mil]	0.8			1.6			2.4		
δ	random	dCAE	TCSE	random	dCAE	TCSE	random	dCAE	TCSE
1.8e-7	0.003	0.025	**0.033**	0.038	0.145	**0.171**	0.110	0.315	**0.350**
1.8e-5	0.002	0.021	**0.032**	0.024	0.136	**0.180**	0.089	0.321	**0.362**
1.8e-3	0.006	0.037	**0.044**	0.047	0.152	**0.187**	0.105	0.325	**0.377**
1.8e-2	0.001	0.018	**0.031**	0.019	0.125	**0.150**	0.065	0.302	**0.338**

Tables 9 and 10 present the $\bar{\mu}_w(t)$ values recorded at critical training milestones: 0.8M, 1.6M, and 2.4M timesteps, across various search radius thresholds δ. To ensure a fair and rigorous comparison, the candidate values for δ are directly adopted from the evaluation protocol established in the EMU baseline(Na et al. (2024)).

As demonstrated in the tables, by eliminating the regulatory effect of β, the inherent representation collapse issues of basic embedders become more pronounced. In contrast, the TCSE consistently maintains superior $\bar{\mu}_w(t)$ scores across both easy and super-hard scenarios. This confirms that the synergy of contrastive learning and time-augmented reconstruction intrinsically provides a much more robust and semantically meaningful memory retrieval space, even without downstream incentive regulation.

Result Analysis. In Table 9, we observe an intriguing phenomenon where the conventional dCAE architecture achieves competitive performance when δ is relatively large (e.g., δ = 1.8e-2). We can explain this behavior through the theoretical noise bounds established in Appendix C (Proof of Proposition 1). Recall that the total retrieval noise is bounded by $\alpha = (1+\gamma)(\epsilon_{\text{rep}} + \epsilon_{\text{store}})$, where the representation error satisfies $\epsilon_{\text{rep}} \leq L\delta$.

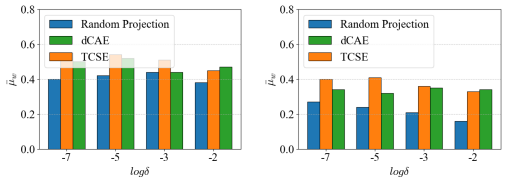
- **Theoretical Noise Bounds.** In low-dimensional, simple maps like `3s_vs_5z` (State Dim = 68), the state space is compact and the effective variation of V^* is relatively limited, making the overall noise α governed primarily by the trade-off between representation error (ϵ_{rep}) and storage coverage error (ϵ_{store}). Because dCAE's latent space lacks explicit temporal and contrastive discriminative

constraints, it suffers from larger retrieval uncertainty (higher ϵ_{store}) under very small search radii δ. Moderately increasing δ significantly reduces ϵ_{store} while $L\delta$ grows slowly in simple state spaces, allowing a larger δ to act as a beneficial smoothing heuristic that improves retrieval quality for dCAE.

- **Advantage of TCSE.** In contrast, TCSE leverages contrastive learning to construct tightly clustered, temporally consistent embeddings where ϵ_{store} is already minimal even under small δ. Consequently, further expanding δ yields negligible reduction in ϵ_{store}, but linearly increases the representation error bound $L\delta$. This explains why TCSE is substantially less sensitive to hyperparameter tuning and achieves peak performance even under strict, tight retrieval thresholds.

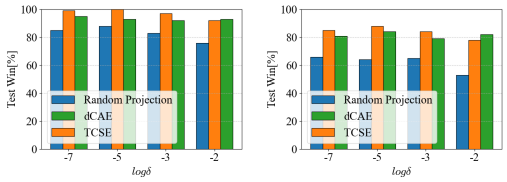
E.5 Ablation Study on TCSE

To isolate the contribution of the semantic embedder to retrieval precision, we evaluate three designs, including random projection (Dasgupta & Gupta (2003)), dCAE (Eq. 5), and TCSE (Eq. 4), using both final win-rate and the composite efficiency metric $\bar{\mu}_w$ (Appendix E.2). All evaluations disable the TCGM (β) to decouple embedding quality from incentive regulation, and vary the retrieval radius δ across logarithmic scales.



(a) 3s_vs_5z (b) 6h_vs_8z

Figure 9: $\bar{\mu}_w$ values according to different δ.



(a) 3s_vs_5z (b) 6h_vs_8z

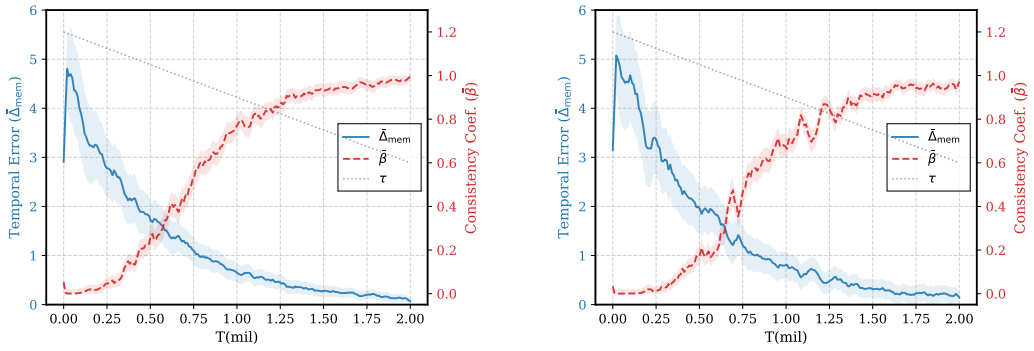
Figure 10: Test win-rate according to different δ.

Figures 9 and 10 report $\bar{\mu}_w$ and final test win-rates under varying δ on SMAC scenarios. More data of $\bar{\mu}_w$ is presented in Tables 9 and 10 in Appendix E.4. As shown, TCSE consistently achieves higher $\bar{\mu}_w$ and win-rates across all δ settings on both easy (3s_vs_5z) and super-hard (6h_vs_8z) maps. Crucially, TCSE exhibits markedly lower sensitivity to δ fluctuations compared to dCAE and random projection. We attribute this robustness to TCSE's dual-objective design, i.e., contrastive learning enforces semantic clustering to reduce retrieval ambiguity, while time-conditioned reconstruction preserves temporal anchors for coherent trajectory matching. More parametric study is in Appendix E.7

E.6 Dynamics of Temporal Consistency Error and Consistency Coefficient

To empirically validate the theoretical properties of our gating mechanism in Section 3.2, we track the evolution of the Temporal Consistency Error (Δ_{mem}) and the Consistency Coefficient (β) during training. Specifically, we sampled the average $\bar{\Delta}_{\text{mem}}$ and $\bar{\beta}$ from the episodic buffer $\mathcal{D}_E$ every 10,000 environment steps on the SMAC (`3s_vs_5z`) and GRF (`3_vs_1WK`) environments.

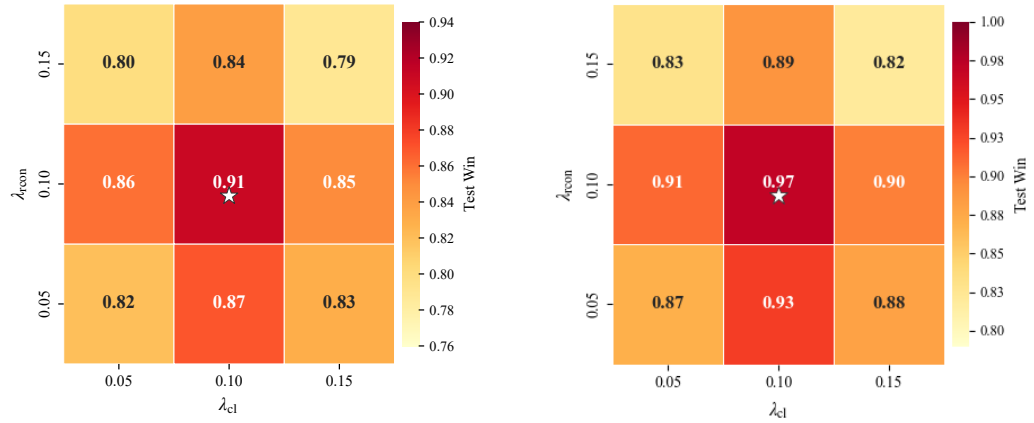
As illustrated in Figure 11, the empirical results demonstrate a highly adaptive dynamic interplay among the temporal error ($\bar{\Delta}_{\text{mem}}$), the consistency coefficient ($\bar{\beta}$), and the temperature parameter (τ). During the early stages of training, the episodic buffer predominantly contains disjointed exploratory trajectories, yielding a notably high $\bar{\Delta}_{\text{mem}}$. Driven by this large error, the gating mechanism aggressively suppresses $\bar{\beta}$ to near zero, effectively shielding the agent from misleading episodic incentives. As training progresses, high-quality and temporally coherent experiences accumulate. Consistent with our theoretical analysis, $\bar{\Delta}_{\text{mem}}$ drops rapidly and converges toward zero. Crucially, while τ is systematically annealed to enforce progressively stricter consistency bounds, the rapid decay of $\bar{\Delta}_{\text{mem}}$ outpaces this restriction, allowing $\bar{\beta}$ to naturally recover and approach 1.0. This cross-domain evaluation verifies that the EMTC gating mechanism robustly penalizes pseudo-memories during initial exploration, while smoothly transitioning to deliver unattenuated, unbiased guidance as the joint policy approaches convergence.



(a) 3s_vs_5z (b) 3_vs_1WK

Figure 11: Evolution of the average Temporal Consistency Error ($\bar{\Delta}_{\text{mem}}$) and Consistency Coefficient ($\bar{\beta}$) alongside the annealed temperature τ.

E.7 Parametric Study on TCSE



(a) 3_vs_1WK (b) corridor

Figure 12: Impact of λ_{rcon} and λ_{cl} on final test win-rates. The star ($\star$) indicates the optimal configuration.

In this section, we conduct a comprehensive parametric study to evaluate the impact of two crucial hyperparameters in the Temporally Consistent Semantic Embedder (TCSE): the reconstruction loss weight (λ_{rcon}) and the contrastive learning loss weight (λ_{cl}). The primary objective of this experiment is to identify the optimal configuration that maximizes the embedder's representation quality by effectively balancing temporal-aware state feature extraction and semantic discriminability.

To thoroughly investigate this, we performed a grid search over $\lambda_{rcon}, \lambda_{cl} \in \{0.05, 0.10, 0.15\}$. The evaluation was carried out on two representative and challenging scenarios: the 3_vs_1WK task from Google Research Football (GRF) and the super-hard corridor map from the StarCraft Multi-Agent Challenge (SMAC). As illustrated by the heatmaps in Figure 12, our EMTC framework consistently achieves peak performance across both domains when the parameters are set to $\lambda_{rcon} = 0.10$ and $\lambda_{cl} = 0.10$ (marked by the white star).

E.8 Data Augmentation Strategies for TCSE

The efficacy of contrastive learning relies heavily on the construction of meaningful positive pairs. In visual domains, standard image transformations (e.g., cropping, shifting) are widely applied. However, as recent studies highlight, directly applying spatial transformations to 1D multi-agent state vectors destroys fundamental intrinsic semantics. For instance, spatially shifting an agent's coordinate features into an enemy's feature slot fundamentally alters the underlying Markov Decision Process (MDP) dynamics(Yu et al. (2024); Luo et al. (2024)). While data augmentation has recently demonstrated significant potential to enhance sample efficiency and representational robustness in RL and MARL(Yu et al. (2023)), the transformations must be carefully restricted to strictly preserve state-based semantic alignments.

To generate robust positive pairs without violating these intrinsic state semantics, we designed a streamlined data augmentation pipeline tailored for 1D MARL state vectors. Our approach integrates two distinct, semantics-preserving operations:

- **Random Masking / Cutout:** Elements within the state vector are randomly masked out (set to zero) with a probability p_{mask}. This acts as a localized information dropout mechanism, forcing the semantic embedder to reconstruct and align features even under severe partial observability or simulated sensor occlusion(Yu et al. (2024)).
- **Gaussian Noise:** We continuously inject standard normal noise scaled by a factor of σ. This ubiquitous high-frequency noise simulates dynamic measurement perturbations, ensuring that the learned representations remain highly resilient to stochastic environmental dynamics(Yu et al. (2023)).

To provide a more intuitive understanding of our customized data augmentation pipeline for 1D state vectors, the specific operational formats are visually detailed in Figure 13. Unlike standard image transformations, our approach directly manipulates the numerical feature arrays to construct robust positive pairs while strictly preserving entity-level index alignment. Specifically, given an original state vector (Figure 13A), we employ *Random Masking* to simulate information loss by zeroing out specific indices (Figure 13B), and *Gaussian Noise* to perturb the continuous feature values (Figure 13C). By synergizing these operations, TCSE effectively learns representations that are robust to partial observability and invariant to environmental noise.

Furthermore, to systematically evaluate this combined data augmentation strategy, we conducted an extensive parametric study utilizing the **EMTC (QPLEX)** architecture as our fixed baseline. By keeping all core algorithmic components strictly unchanged and varying solely the data augmentation parameters, we detail the final test win-rates across various hyperparameter combinations on the `6h_vs_8z` and `corridor` maps in Tables 11 and 12. The empirical results consistently demonstrate that the framework achieves its peak performance at the optimal configuration of $p_{mask} = 0.2$ and $\sigma = 0.05$ across both scenarios. For instance, insufficient masking ($p_{mask} = 0.1$) limits the regularization effect, making the semantic embedder prone to overfitting local features, while excessive noise ($\sigma = 0.10$) heavily disrupts the inherent state structure. This rigorous evaluation verifies that our selected configuration provides the ideal balance between representation robustness and semantic feature retention in highly complex environments.

Table 11: Parameter sensitivity of the combined augmentation strategy on the `6h_vs_8z` map. The optimal configuration is highlighted in bold.

Mask (p_{mask})	**Noise** (σ)	**Test Win**
0.1	0.05	0.721
0.2	**0.05**	**0.748**
0.4	0.05	0.742
0.2	0.10	0.701

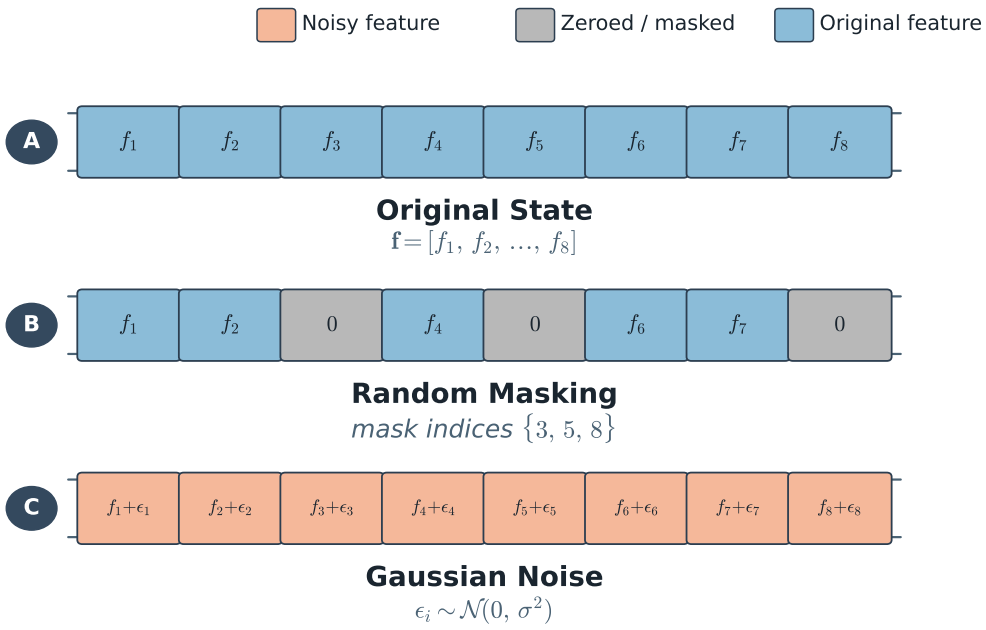


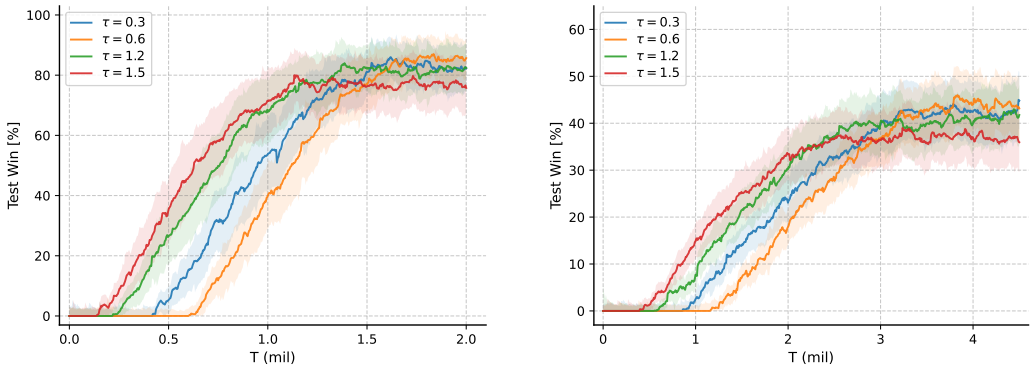
Figure 13: Conceptual illustration of the tailored data augmentation operations for 1D MARL state vectors. **(A)** The original state feature vector. **(B)** Random Masking drops out specific elements (e.g., indices 3, 5, 8) to simulate sensor occlusion without disrupting the alignment of other entities. **(C)** Gaussian Noise perturbs each individual feature with continuous noise ($\epsilon_i \sim \mathcal{N}(0, \sigma^2)$) to enhance measurement robustness.

Table 12: Parameter sensitivity of the combined augmentation strategy on the `corridor` map. The table mirrors the experimental setup of `6h_vs_8z` to verify generalizability in a super-hard scenario.

Mask (p_{mask})	**Noise** (σ)	**Test Win**
0.1	0.05	0.922
0.2	**0.05**	**0.935**
0.4	0.05	0.928
0.2	0.10	0.892

E.9 PARAMETRIC STUDY ON TCGM

The temperature parameter τ in the Gaussian gating function (Eq. (8)) dictates the strictness of the temporal consistency constraint. A larger τ produces a wider Gaussian curve, allowing episodic incentives (Barto (2012); Ng et al. (1999)) to pass through even when the temporal error Δ_{mem} is relatively large. Conversely, a smaller τ tightly restricts the distribution of incentives to only highly consistent trajectories.



(a) 3_vs_1WK with EMTC (b) CA_hard with EMTC(CDS)

Figure 14: Ablation study of the temporal consistency parameter τ using fixed values. The evaluation is conducted on two distinct scenarios: (a) 3_vs_1WK using the EMTC architecture, and (b) the challenging CA_hard using EMTC(CDS).

We conduct an ablation study using fixed values of τ in $\{0.3, 0.6, 1.2, 1.5\}$. We evaluate these values across two distinct scenarios: the 3_vs_1WK map using the standard EMTC architecture and the CA_hard map using EMTC(CDS). Figure 14 illustrates a consistent trend across both environments. An extremely large temperature like $\tau = 1.5$ achieves rapid performance growth early in training. A lenient gate allows many episodic rewards to pass. However, as noted in general reinforcement learning (Sutton et al. (1998)), without rigorous incentive constraints (Van Hasselt et al. (2016)), the unconstrained accumulation of auxiliary incentives can introduce significant estimation bias and systematic value overestimation (Fujimoto et al. (2018); Henderson et al. (2018)). Consequently, this leniency eventually traps the policy in suboptimal behaviors and local optima (Bellemare et al. (2016)), where the agent prioritizes the maximization of unconstrained auxiliary incentives over the global task objective. Thus, the agent achieves a very low final win rate, making this value unsuitable for final adoption.

Conversely, strict temperatures rigorously filter out unreliable memories. They ensure robust model refinement and a higher performance plateau. Still, the initial learning speed is extremely slow. Notably, the experimental results show that $\tau = 0.6$ yields better overall performance than $\tau = 0.3$. This indicates that an overly strict gate blocks too many valid signals and harms the learning process. These contrasting observations clearly highlight the classic exploration and exploitation dilemma.

To resolve this dilemma, we introduce a dynamic simulated annealing strategy. Based on the ablation results, we initialize the temperature at 1.2 to encourage safe and rapid early exploration. We then gradually decay it to 0.6 to ensure stable final convergence. The dynamic value of τ at training step t follows a linear decay formulation:

$$\tau(t) = \max\left(\tau_{end}, \tau_{start} - \frac{\tau_{start} - \tau_{end}}{T_{decay}} \cdot t\right) \tag{25}$$

where we set $\tau_{start} = 1.2$ and $\tau_{end} = 0.6$. The parameter T_{decay} represents the designated duration for the annealing process. Our final EMTC framework exclusively adopts this formula to regulate the temporal consistency gate throughout the entire training phase.

E.10 Retrieval Error Validation

To rigorously evaluate the retrieval reliability of episodic memory embeddings and verify our motivation regarding representation collapse, we conduct a perturbation-based consistency analysis on the `3s_vs_5z` scenario.

Evaluation Setup. We apply minimal random spatial perturbations within the range $[0, 0.5\%]$ to a single agent while preserving the overall tactical configuration. For each sampled state s, we perform $N = 1000$ perturbation trials to generate perturbed states $\tilde{s}$. We then measure the **Retrieval Inconsistency Rate** (ERR_{ret}), defined as:

$$\text{ERR}_{\text{ret}} = \frac{1}{N} \sum_{k=1}^{N} \mathbb{I}\left(m(s) \neq m(\tilde{s}_k)\right), \tag{26}$$

where $m(s)$ and $m(\tilde{s}_k)$ denote the retrieved episodic memory entries for the original and perturbed states, respectively, and $\mathbb{I}(\cdot)$ is the indicator function.

Table 13: Retrieval inconsistency rates under spatial perturbations on `3s_vs_5z`.

Embedder	State Dimension	Retrieval Inconsistency Rate (ERR_{ret})
dCAE	68	16.3%
TCSE (Ours)	68	**7.2%**

Analysis and Causal Link to Q-Value Overestimation. As shown in Table 13, TCSE substantially reduces the retrieval inconsistency rate from 16.3% to 7.2%. Traditional reconstruction-based autoencoders (such as dCAE) tend to over-smooth spatial fine-grained details in complex environments, resulting in fuzzy, unstable retrieval decision boundaries under minor state variations. In contrast, TCSE maintains a geometrically stable and temporally discriminative semantic topology via contrastive regularization.

E.11 State Space Exploration Capability and Coverage Evaluation

A common concern in episodic-memory-guided reinforcement learning is whether aggressively steering policy updates toward historical high-return trajectories compromises exploratory diversity across the global state space. We address this from both theoretical safeguards and empirical coverage dynamics.

Theoretical Safeguards Against Premature Exploitation. EMTC prevents policy collapse into narrow sub-spaces via two integrated design mechanisms:

1. **Dual-Phase Dynamic Temperature Scheduling (τ).** As defined in Eq. 25, the gating temperature starts at a high value ($\tau_{\text{start}} = 1.2$). During early training, this produces a broad, lenient Gaussian gating coefficient, explicitly preserving extensive exploration.
2. **Fallback Protection for Novel States.** When agents explore completely unvisited state regions, the temporal consistency error Δ_{mem} is naturally elevated. By design, the gating coefficient decays ($\beta(\Delta_{\text{mem}}) \to 0$), causing the target to automatically fall back to standard one-step TD updates. This guarantees that the underlying backbone (e.g., QPLEX or CDS) conducts unhindered baseline exploration without being misled by sparse or stale memories.

Quantitative Coverage Evaluation. In high-dimensional MARL domains, calculating the exact continuous state coverage is intractable. Following the memory insertion protocol, a transition is stored as a new memory node if and only if its latent embedding x_t satisfies $\|x_t - \hat{x}_t\|_2 \geq \delta$ with respect to all existing nodes $\hat{x}_t$. Therefore, the **average environment steps required per 1M novel memory insertions** directly reflects the discovery rate of semantically distinct state regions per unit time. We evaluate this metric across two learning phases on the super-hard map `6h_vs_8z`:

- **Phase 1 (Early Exploration):** From training initialization until the policy achieves its first winning episode (Win Rate $= 0$).
- **Phase 2 (Convergence Phase):** From the first win until final policy convergence (Win Rate > 0).

Table 14: Environment steps required for discovering 1M novel memory states on `6h_vs_8z` (fewer steps indicate faster state exploration and higher coverage efficiency).

Algorithm	Backbone	Phase 1 (Win Rate $= 0$)	Phase 2 (Win Rate > 0)
EMU	QPLEX	33K	47K
EMU	CDS	28K	39K
EMTC (Ours)	QPLEX	**30K**	**44K**
EMTC (Ours)	CDS	**26K**	**38K**

Empirical Results. As presented in Table 14, EMTC consistently requires fewer environment steps per 1M novel memory insertions across both training stages compared to the EMU baseline under identical backbones. During Phase 1, the faster insertion rate confirms that EMTC actively discovers novel state regions rather than getting trapped in early suboptimal loops. In Phase 2, EMTC continues to sustain higher state refresh rates. While conventional frameworks suffer from representation collapse that falsely merges distinct states into existing clusters, EMTC effectively preserves discriminative state boundaries, ensuring steady and healthy state-space coverage throughout policy convergence.

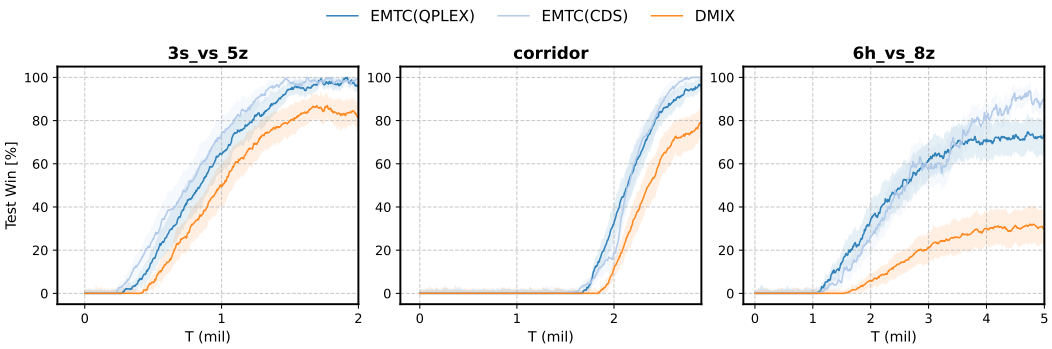
E.12 Comparison with Other Baselines

In this subsection, we evaluate our framework against more prominent state of the art baselines. We select DMIX (Sun et al. (2021)) as a representative advanced value based method and we select MAPPO (Yu et al. (2022)) as a highly popular policy gradient method. We conduct these comparisons across three diverse SMAC maps. These maps include `3s_vs_5z`, `corridor`, and `6h_vs_8z`. The figures illustrate the experimental learning curves for our two variants, EMTC(QPLEX) and EMTC(CDS). Our method demonstrates a clear performance gap over both baselines.

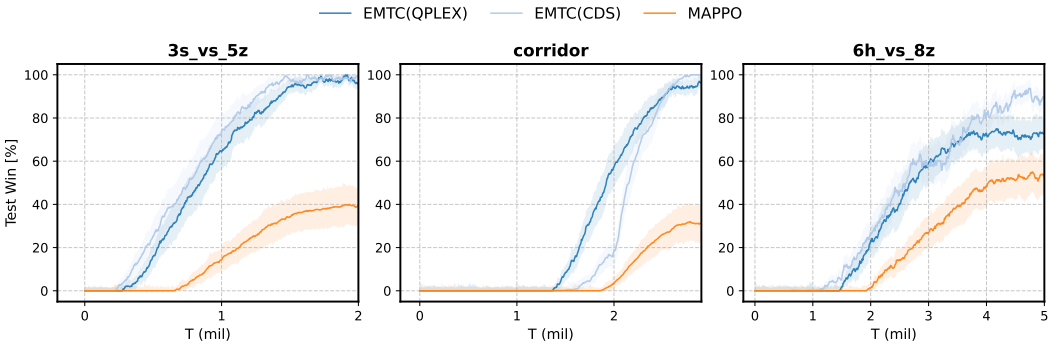
First, we compare our method with DMIX. DMIX uses distributional reinforcement learning to handle environmental randomness. However, it still relies on stepwise rewards. In super hard scenarios with severe reward sparsity, DMIX struggles to find successful joint actions. As Figure 15a shows, DMIX learns slowly in `3s_vs_5z` and `corridor`. Furthermore, it suffers from a severe exploration deadlock in `6h_vs_8z`. It only achieves a very low final win rate. In contrast, our EMTC framework reuses historical high return trajectories. This memory mechanism directly guides agents toward rewarding states. It completely bypasses the exploration bottleneck and achieves superior performance.

Second, we compare our method with MAPPO. MAPPO is a highly popular on policy algorithm. The original paper (Yu et al. (2022)) notes that MAPPO can match off policy algorithms after extensive training. However, it exhibits significant limitations within our restricted training timesteps. We can see that EMTC performs better than MAPPO with an evident gap in Figure 15b. Specifically, MAPPO suffers from severe sample inefficiency. The massive joint action space makes on policy exploration extremely inefficient here. MAPPO shows very slow progress and high performance variance across all three maps. Our EMTC framework completely avoids these issues. It maximizes sample efficiency through a robust semantic episodic buffer. As a result, our method drastically accelerates training and achieves a much higher final win rate.

Comparison with Recent Representation-Learning and Exploration Baselines. To further benchmark EMTC against the latest MARL advancements from representation learning and exploration perspectives, we evaluate against two recent competitive baselines: **ACORM** (ICLR 2024, Hu et al. (2024)) and **TEE** (ICLR 2025, Li & Zhu (2025)). We conduct these comparisons across three challenging SMAC maps: `6h_vs_8z`, `corridor`, and the super-hard map `3s5z_vs_3s6z`.



(a) EMTC vs. DMIX



(b) EMTC vs. MAPPO

Figure 15: Performance comparison of EMTC variants against state of the art baselines. The evaluations span three diverse SMAC scenarios. The top row displays the comparison with DMIX, while The bottom row displays the comparison with MAPPO.

Table 15: Comparison with recent MARL methods on challenging SMAC maps (final test win rates in %).

Algorithm	6h_vs_8z (%)	corridor (%)	3s5z_vs_3s6z (%)
ACORM	45.5	92.1	89.8
TEE	82.1	90.6	91.5
EMTC (CDS)	**89.4**	**99.2**	**98.4**
EMTC (QPLEX)	75.0	97.4	94.0

As reported in Table 15, EMTC remains remarkably competitive against these recent methods, with EMTC(CDS) achieving the highest win rates across all three scenarios (89.4%, 99.2%, and 98.4%, respectively). These results confirm that our temporally consistent episodic guidance effectively resolves sparse-reward exploration bottlenecks and provides more targeted, reliable bootstrapping than generic representation or exploration heuristics alone.

F ALGORITHM

F.1 TRAINING ALGORITHM FOR THE TCSE EMBEDDER

Algorithm 1 outlines the optimization process for our semantic embedder. We periodically sample a dataset of historical transitions from the episodic buffer. Then, we divide this dataset into multiple mini batches. For each raw state, we apply random masking and Gaussian noise. These operations generate a robust augmented view. Next, we feed both the raw state and its augmented view into the encoder network and this encoding step produces a positive pair of latent embeddings.

The decoder network processes the raw state embedding and it reconstructs the original state and predicts the expected historical return. We compute the total objective loss using two main components. The first component is the contrastive loss. It aligns the latent embeddings of the positive pairs to extract robust semantic features. The second component is the autoencoder loss and it minimizes the reconstruction errors for both the state and the return. We use the weight parameter λ_{rcon} to balance these two reconstruction targets, while using another parameter λ_{cl} to scale the overall contrastive loss.

We update the encoder and decoder networks via gradient descent and completely refresh the episodic buffer. Then, we use the updated encoder to project all historical raw states into a new latent space. This vital updating step effectively prevents stale representations and ensures highly accurate memory retrieval during the main policy training.

Algorithm 1 Optimization for TCSE Embedder

Parameter: learning rate α, number of training dataset N, batch size B, mask p_{mask}, noise σ, loss weights $\lambda_{rcon}, \lambda_{cl}$
Sample Training dataset $(s^{(i)}, H^{(i)}, t^{(i)})_{i=1}^{N} \sim \mathcal{D}_E$
Initialize weights $\phi, \psi \leftarrow \mathbf{0}$
for $i = 1$ **to** $\lfloor N/B \rfloor$ **do**
    # 1. Mini-batch fetching and Data Augmentation
    Get a mini-batch of raw transitions $(s^{(j)}, H^{(j)}, t^{(j)})_{j=(i-1)B+1}^{iB}$
    Generate augmented states $s'^{(j)}$ via Mask (p_{mask}), and Noise (σ)
    # 2. Latent Embedding and Contrastive Pairs
    Compute latent embeddings for raw states $x^{(j)} = f_\phi(s^{(j)}|t^{(j)})$
    Compute latent embeddings for augmented states $x'^{(j)} = f_\phi(s'^{(j)}|t^{(j)})$
    # 3. Decoder Reconstruction
    Predict expected return $\bar{H}^{(j)} = f_\psi^H(x^{(j)}|t^{(j)})$
    Reconstruct original state $\bar{s}^{(j)} = f_\psi^s(x^{(j)}|t^{(j)})$
    # 4. Explicit Loss Computation
    Compute InfoNCE loss $\mathcal{L}_{cl}$ using positive pairs $(x^{(j)}, x'^{(j)})$
    Compute reconstruction loss $\mathcal{L}_{dCAE} = \frac{1}{B}\sum_j \left(\|H^{(j)} - \bar{H}^{(j)}\|_2^2 + \lambda_{rcon}\|s^{(j)} - \bar{s}^{(j)}\|_2^2\right)$
    Compute total loss $\mathcal{L}(\phi, \psi) = \mathcal{L}_{dCAE} + \lambda_{cl}\mathcal{L}_{cl}$
    # 5. Network Update
    Update $\phi \leftarrow \phi - \alpha\frac{\partial \mathcal{L}}{\partial \phi}, \psi \leftarrow \psi - \alpha\frac{\partial \mathcal{L}}{\partial \psi}$
end for
Re-project and update all memory embeddings $x \in \mathcal{D}_E$ with the updated f_ϕ

F.2 Construction of the Episodic Memory Buffer

During centralized training, we continually evaluate the final episodic return. We check if the return reaches the maximum possible value R_{max} or a specific success threshold R_{thr}. This success condition corresponds to defeating all enemies in SMAC or scoring a goal in GRF. When storing a successful trajectory $\mathcal{T}_\xi$ into the episodic buffer $\mathcal{D}_E$, we assign a positive desirability score. Specifically, we set $\xi(s) = 1$ for all states $s \in \mathcal{T}_\xi$.

To ensure efficient memory construction, we implement a desirability propagation mechanism. We propagate the positive desirability of a new state to any similar stored state within the distance threshold δ. This explicit propagation provides a strong incentive for the agent to visit these similar states. Furthermore, stored memories generally remain in $\mathcal{D}_E$ until they become obsolete and get replaced. However, we actively update suboptimal records to maintain optimal buffer quality. The algorithm might detect a new desirable state near an existing suboptimal memory within the threshold δ. In this case, we replace the old suboptimal observation with the new desirable one. This replacement essentially creates a memory shift toward highly rewarding states. Algorithm 2 presents the complete memory construction procedure alongside these propagation and shift mechanisms.

Algorithm 2 Episodic Memory Construction

Input: Episodic trajectory $\mathcal{T} = \{s_0, a_0, r_0, ..., s_T\}$, Optimality $\xi_\mathcal{T}$, Episodic buffer $\mathcal{D}_E$
Initialize accumulated return $R_t = 0$
Initialize successor index $idx_{next} \leftarrow$ Null
for $t = T$ **to** 0 **do**
  Compute latent embedding $x_t = f_\phi(s_t)$
  Update accumulated return $R_t \leftarrow r_t + \gamma R_t$
  Find the nearest neighbor $\hat{x}_t$ in $\mathcal{D}_E$ and retrieve its index $\hat{idx}_t$
  if $\|x_t - \hat{x}_t\|_2 < \delta$ **then**
    # Update frequency and desirability counts
    $N_{call}(\hat{x}_t) \leftarrow N_{call}(\hat{x}_t) + 1$
    if $\xi_\mathcal{T} == 1$ **then**
      $N_\xi(\hat{x}_t) \leftarrow N_\xi(\hat{x}_t) + 1$
    end if
    # Update stored returns and representation
    if $\hat{\xi}_t == 0$ **and** $\xi_\mathcal{T} == 1$ **then**
      $\hat{\xi}_t \leftarrow \xi_\mathcal{T}$ # Desirability propagation
      $\hat{x}_t \leftarrow x_t, \hat{s}_t \leftarrow s_t$ # Memory shift
      $\hat{H}_t \leftarrow R_t$
    else if $\hat{H}_t < R_t$ **then**
      $\hat{H}_t \leftarrow R_t$
    end if
    Update successor link: store idx_{next} into the node at $\hat{idx}_t$
    $idx_{curr} \leftarrow \hat{idx}_t$
  else
    Add new memory tuple $\mathcal{D}_E \leftarrow (x_t, R_t, s_t, \xi_\mathcal{T}, idx_{next})$
    $idx_{curr} \leftarrow$ index of this newly added tuple
  end if
  # Carry the current index to the previous timestep
  $idx_{next} \leftarrow idx_{curr}$
end for

F.3 Main Training Framework

Algorithm 3 presents the main reinforcement learning loop of our framework. The agents continually interact with the environment to collect new trajectories. During this exploration process, the framework linearly decays the dynamic temperature parameter τ. It then calls the memory construction subroutine to update the episodic buffer.

Next, the algorithm evaluates the temporal consistency for the newly collected trajectory. For each transition (s_t, a_t, r_t, s_{t+1}), it retrieves the optimal historical return for the current state s_t by finding its nearest neighbor in the episodic buffer $\mathcal{D}_E$. Crucially, to accurately capture the true suboptimality of the executed action, the algorithm also retrieves the expected optimal return for the successor state by querying $\mathcal{D}_E$ using the actual next state s_{t+1} observed in the environment, rather than blindly following historical sequence links. The algorithm then calculates the exact temporal consistency error Δ_{mem}. It uses the currently annealed temperature τ to compute the Gaussian gating coefficient $\beta(\Delta_{\text{mem}})$. The framework stores this coefficient alongside the regular transition data in the standard replay buffer. During the policy optimization phase, the algorithm samples a minibatch of transitions and calculates the standard unconstrained episodic reward. It strictly modulates this reward using the precomputed gating coefficient. The agent updates the joint action value networks using this gated loss function. Finally, the framework periodically triggers the embedder optimization subroutine to maintain semantic accuracy.

Algorithm 3 Main Training Framework for EMTC

Initialize: Replay buffer $\mathcal{D}$, Episodic buffer $\mathcal{D}_E$
Initialize: Joint Q networks θ, TCSE parameters ϕ, ψ
Initialize: Annealing parameters τ_{start}, τ_{end}, T_{decay}
while $t_{env} \leq t_{max}$ **do**
  Interact with environment via ϵ-greedy policy to collect trajectory $\mathcal{T}$
  # 1. Anneal Temperature and Construct Memory
  $\tau \leftarrow \max\left(\tau_{end}, \tau_{start} - \frac{\tau_{start}-\tau_{end}}{T_{decay}} \cdot t_{env}\right)$
  Run **Algorithm 2** to update $\mathcal{D}_E$ with trajectory $\mathcal{T}$
  # 2. Evaluate Temporal Consistency Gating
  for each transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{T}$ **do**
    Find nearest neighbor index for current state: $k_t \leftarrow \arg\min_k \|f_\phi(s_t) - x_k\|_2$ in $\mathcal{D}_E$
    Retrieve current stored value $\mathcal{H}_t \leftarrow \mathcal{D}_E[k_t].H$
    if s_{t+1} is terminal **then**
      $\mathcal{H}_{t+1} \leftarrow 0$
    else
      Find nearest neighbor index for successor: $k_{t+1} \leftarrow \arg\min_k \|f_\phi(s_{t+1}) - x_k\|_2$ in $\mathcal{D}_E$
      Retrieve successor stored value $\mathcal{H}_{t+1} \leftarrow \mathcal{D}_E[k_{t+1}].H$
    end if
    Calculate consistency error $\Delta_{\text{mem}} \leftarrow |r_t + \gamma\mathcal{H}_{t+1} - \mathcal{H}_t|$
    Compute gating coefficient $\beta(\Delta_{\text{mem}}) \leftarrow \exp\left(-\frac{\Delta_{\text{mem}}^2}{2\tau^2}\right)$
    Append transition $\langle s_t, a_t, r_t, s_{t+1}, \beta(\Delta_{\text{mem}})\rangle$ to replay buffer $\mathcal{D}$
  end for
  # 3. Main Policy Optimization
  for $k = 1$ **to** n_{circle} **do**
    Sample a minibatch of M transitions from $\mathcal{D}$
    Calculate the unconstrained episodic reward r_p
    Calculate the gated episodic reward $\tilde{r}_p = \beta(\Delta_{\text{mem}}) \cdot r_p$
    Update policy parameters θ using the gated TD loss $\mathcal{L}_\theta^\beta$
  end for
  # 4. Periodic TCSE Update
  if $t_{env} \bmod t_{emb} == 0$ **then**
    Run **Algorithm 1** to optimize parameters ϕ, ψ
  end if
end while