RACC: COUPLING DISAGREEMENT ROUTING WITH RECEIVER-CONDITIONED MESSAGES UNDER BIT CAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

A scarce multi-agent channel must choose both which receiver warrants a packet and what receiver-specific information that packet should contain. Receiver-Aware Coupled Communication (RACC) makes these choices from one sender-side receiver prediction: a thresholded forward KL between sender and predicted-receiver policies activates a directed edge, while the predicted receiver state and identity condition its binary B-bit message. On three hard SMAC maps and three packet widths, RACC exceeds BVME in all nine cells by 1.4–3.6 win-rate points under common widths; after independently matching mean charged payload bits, the gains remain 1.0–3.0 points and every paired interval excludes zero. Traffic-matched interventions expose the contribution of each connection: replacing discrepancy routing with a random gate loses 3.8 $[2.4, 5.1]$ pooled tactical points, sharing one sender payload across receivers loses 4.8 $[3.2, 6.3]$, and shuffling receiver identities loses 5.5 $[3.9, 7.1]$. At B=32, RACC reaches 75.4%, 65.1%, and 90.8% SMAC win rate, 3.2–4.8 points above an encoder-, quantizer-, and width-matched always-send control, and leads every evaluated communication baseline in two- through five-player Hanabi. On held-out SMACv2 scenarios spanning 5, 8, and 11 agents, two graph densities, and three packet widths, it averages 3.7 $[2.5, 4.8]$ points above BVME. Receiver prediction can therefore serve as a shared computational interface between packet allocation and personalized content when asymmetric information makes directed bits consequential.

1 INTRODUCTION

Under local observations, the same sender state can have different decision value for different teammates: one receiver may lack action-relevant information while another already acts compatibly with the sender. A broadcast code ignores this asymmetry. An independently learned router and payload encoder can also disagree about why an edge was selected, spending a scarce packet on content that was not specialized to the selected receiver. The communication object of interest is therefore a directed pair—recipient and payload together—computed from information available during decentralized execution in a common-payoff Dec-POMDP (Bernstein et al., 2002; Oliehoek & Amato, 2016).

Learning whom and what to communicate is an established objective, approached through receiver attention, graph compression, teammate-conditioned incentives, learned topology, and receiver- or demand-aware customization (Das et al., 2019; Tian et al., 2021; Yuan et al., 2022; Meng & Tan, 2024; Wang et al., 2022; Li & Zhang, 2024; Huo et al., 2024). These approaches differ in the signal that links allocation to content. TarMAC addresses sender values with receiver queries; CGIBNet compresses graph structure and node information; MAIC tailors value-space incentives from teammate models; PMAC jointly personalizes learned topology and messages; ToM2C communicates inferred goals in a target hierarchy; and CACOM and DCMAC use receiver context or demand awareness to customize messages. RACC isolates a different construction: no same-step receiver request is required, and one sender-side prediction is reused both as an action-level routing statistic and as the condition for a hard-width binary payload.

Concretely, each RACC agent maintains a private recurrent belief. Sender i predicts receiver j's pre-message latent state and legal-action distribution, computes a directional forward KL between its own policy and that prediction, and activates $i \to j$ when the discrepancy exceeds a learned threshold.

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.

The same predicted state and receiver identity condition a distinct B-bit code on every active edge. Receivers aggregate the incoming set before acting. The exchange is a single acyclic round, and execution uses only sender history, receiver identity, the candidate graph, and received messages.

Traffic-matched interventions test the links in this construction. Replacing discrepancy routing by a random gate reduces pooled tactical performance by 3.8 $[2.4, 5.1]$ points; replacing receiver-conditioned codes by one shared sender payload costs 4.8 $[3.2, 6.3]$; and shuffling receiver identities costs 5.5 $[3.9, 7.1]$. Removing the receiver model produces a further 3.8 $[2.4, 5.2]$-point loss. Because these comparisons preserve the backbone and match mean charged payload, they test whether allocation, personalization, and receiver binding each contribute to the complete operator.

The resulting performance gains are consistent across capacity and coordination regimes. At B=32, RACC reaches 75.4%, 65.1%, and 90.8% win rate on three hard SMAC maps, 3.2–4.8 points above capacity-matched CB+. Against BVME, all nine paired common-cap effects over $B \in \{16, 32, 64\}$ exclude zero; after independently matching mean charged payload bits, gains remain 1.0–3.0 points. RACC also leads the evaluated communication baselines for every two- through five-player Hanabi setting at B=32, attains 78.6% success with 67 ± 8 charged bits per step on partitioned-information GridSync, and averages 3.7 $[2.5, 4.8]$ points over BVME on held-out SMACv2 scenarios spanning three team sizes, two graph densities, and three packet widths.

Contributions.

- A single-round communication operator in which one sender-computable receiver model supplies both a directed policy-discrepancy gate and a receiver-conditioned binary message under a hard per-packet width.
- A matched intervention suite that separately tests discrepancy routing, receiver-specific content, identity binding, aggregation, receiver modeling, and quantization while controlling mean charged payload.
- Controlled gains under common packet widths and matched mean payload, complemented by Hanabi team-size tests, partitioned-information GridSync, and held-out SMACv2 scaling across team size and graph density.

2 Related Work and Positioning

CTDE algorithms stabilize decentralized learning through centralized critics, value factorization, exploration, or role discovery (Lowe et al., 2017; Sunehag et al., 2018; Rashid et al., 2018; Son et al., 2019; Mahajan et al., 2019; Wang et al., 2020; 2021; Yu et al., 2022). Belief-state methods estimate hidden state or teammate behavior (Pritz & Leung, 2025; Zhai et al., 2023), while latent world models support simulation and sample-efficient control (Ha & Schmidhuber, 2018; Hafner et al., 2020; Venugopal et al., 2024; Deihim et al., 2025). RACC instead uses a private belief and a supervised teammate model to parameterize an execution-time communication decision.

Communication design comparison. "Link" records how allocation and content are related, not whether both are learned end to end.

Method	Edge allocation	Payload semantics	Receiver signal	Allocation–content link
CommNet	all agents	sender state	none	none
SchedNet	top-k senders	broadcast sender code	sender priority	scheduling only
TarMAC	soft pair attention	sender value	receiver query	receiver-side weighting
CGIBNet	compressed graph	bottlenecked node code	graph neighbors	joint graph bottleneck
MAIC	sparse pair attention	tailored value incentive	teammate model	model conditions content
ToM2C	ToM-selected pairs	inferred receiver goal	predicted goal/state	target-goal inference
PMAC	learned directed topology	ID-conditioned sender code	receiver identity	jointly personalized modules
CACOM/DCMAC	context/demand-aware	customized reply	context or demand signal	staged or demand-aware customization
BVME	graph neighbors	variational sender code	none	compression objective
RACC	pairwise policy KL	predicted-state binary code	sender-predicted receiver	one prediction in both

Dense exchange and team-level gates make sender information available without explicitly binding payload semantics to a named receiver (Sukhbaatar et al., 2016; Singh et al., 2019; Jiang & Lu, 2018). SchedNet learns sender importance under shared-medium constraints, but selected codes are broadcast (Kim et al., 2019); TarMAC targets receivers through attention while retaining sender-generated values (Das et al., 2019). CGIBNet jointly compresses communication edges and node representations with a graph information bottleneck (Tian et al., 2021). Compression-oriented BVME preserves sender content through a variational code (Duan et al., 2026; Kingma & Welling, 2014), and DNA-MARL learns value consensus over changing graphs (Varela et al., 2025). RACC is distinguished within this group by using action-distribution disagreement from a sender-side receiver model as the hard directed gate and reusing that same predicted state in a binary edge payload.

Teammate-aware personalization provides the closest semantic comparison. MAIC predicts a targeted teammate representation and generates a tailored incentive in action-value space (Yuan et al., 2022); ToM2C uses predicted goals for partner selection and transmits inferred goals in target-oriented hierarchical coordination (Wang et al., 2022); and PMAC learns directed topology together with receiver-ID-conditioned sending and sender-conditioned receiving (Meng & Tan, 2024). CACOM uses a receiver-context exchange, whereas DCMAC uses a demand-aware module to customize messages (Li & Zhang, 2024; Huo et al., 2024). RACC uses neither value-space incentives nor a same-step solicitation: its one-round sender prediction induces both a policy-level gate and a hard-width receiver-conditioned code. Section 5.5 tests these defining interfaces with random-gate, shared-payload, shuffled-identity, always-send, and no-receiver-model interventions at matched mean charged payload.

3 Problem Formulation

3.1 Dec-POMDP with Communication

We augment a common-payoff Dec-POMDP with a packet-width-capped channel: $(\mathcal{S}, \{\mathcal{A}^i\}_{i=1}^N, \{\mathcal{O}^i\}_{i=1}^N, T, O, R, \mathcal{M}, B, \gamma)$. Agents maximize $J(\pi) = \mathbb{E}_\pi[\sum_{t\geq 0} \gamma^t r_t]$ for shared reward R. The global state is available to the centralized critic and auxiliary training targets, whereas each executing actor receives its local history and messages. At step t, communication is restricted to a candidate directed graph $G_t = (V, E_t)$: original SMAC and Hanabi use complete candidate graphs, GridSync uses its task graph, and the SMACv2 study explicitly varies graph density. A transmitted edge message belongs to $\mathcal{M} = \{0,1\}^B$ and is charged exactly B payload bits. The candidate graph represents pre-addressed directed slots, so the accounting excludes edge-activation

flags, endpoint metadata, and transport headers. Thus B is a hard payload width for each active packet, while $\bar{c}$ is a team-level rollout target on mean charged payload bits per step. Charged traffic is $c_t = B \sum_{(i,j)\in E_t} z_t^{ij}$; the controller regulates its rollout mean and does not impose a peak per-step or per-sender cap.

Communication and action follow one acyclic round. Agents first update pre-communication states from their recurrent carry and current observations; all gates and payloads are then computed in parallel from those states; receivers aggregate the resulting messages; finally, actors select environment actions from post-communication states. Messages generated at step t never enter another sender's pre-communication computation at the same step.

3.2 Per-Agent Latent Belief

Agent i maintains a private latent belief with the pre-message update

$$b_t^{i,-} = f_\theta\left(b_{t-1}^{i,+}, o_t^i\right), \qquad b_0^{i,+} = \mathbf{0}, \tag{1}$$

where f_θ is a shared GRU and $b_{t-1}^{i,+}$ already contains the preceding round's received information. A deterministic decoder $d_\phi(b_t^{i,-})$ supplies auxiliary state reconstruction during training. Each belief remains agent-private; RACC neither constructs a shared latent state nor performs model-based rollouts.

3.3 Disagreement Measure

Sender i predicts receiver j from its pre-message state and a learned receiver embedding:

$$\left(\hat{b}_t^{j|i,-}, \hat{\ell}_t^{j|i}\right) = h_\xi(b_t^{i,-}, e_j), \quad \pi_t^{i,-} = \pi_\theta(\cdot \mid b_t^{i,-}), \quad \hat{\pi}_t^{j|i,-} = \pi_\theta(\cdot \mid \hat{b}_t^{j|i,-}). \tag{2}$$

Within each benchmark, policies share a canonical action coordinate system. SMAC coordinates align no-op, stop, movement, and target-indexed attack actions across unit types; Hanabi coordinates align game actions and apply the legal-action mask before normalization; GridSync agents share the same movement vocabulary. Environment action selection assigns zero mass to unavailable coordinates. For disagreement only, both canonical distributions are smoothed as $(1-\epsilon)p + \epsilon/|\mathcal{A}|$, which keeps the forward KL finite even when sender and receiver masks differ. The receiver model produces legal-action logits $\hat{\ell}_t^{j|i}$ alongside its latent prediction, and the induced binary mask is applied before KL smoothing. Thus heterogeneous units are compared on shared action semantics rather than raw unit-specific indices.

The directed trigger score is

$$D_t^{ij} = D_{\mathrm{KL}}\left[\pi_t^{i,-} \,\|\, \hat{\pi}_t^{j|i,-}\right], \qquad z_t^{ij} = \mathbb{1}[D_t^{ij} > \tau_t]. \tag{3}$$

The forward KL emphasizes action coordinates favored by the sender and underweighted by its receiver prediction, giving a directional proposal for $i \to j$. For heterogeneous teammates it functions as a prioritization score rather than an action-agreement target: receiver identity, the predicted legal mask, and supervised receiver state represent systematic role and action-availability differences, while team return determines whether acting on the remaining discrepancy is useful. The same predicted receiver state then enters Eq. 4. This shared input is what the equal-rate random gate, shared-payload, and shuffled-identity interventions test. Receiver states, legal masks, and global state supervise h_ξ during training. At execution, D_t^{ij} is computed from sender history, e_j, and previously integrated messages.

4 Receiver-Aware Coupled Communication

4.1 One Receiver Prediction for Routing and Content

RACC implements a coupled operator $\mathcal{C}_B(b_t^{i,-}, e_j) = (z_t^{ij}, m_t^{ij})$. It returns either an inactive edge or one receiver-specific B-bit message: h_ξ predicts receiver state and policy, Eq. 3 allocates the edge, and the directed encoder uses the same prediction to generate its payload. This shared computational

interface is the communication contribution; a 128-dimensional GRU, PPO, attention aggregation, and straight-through estimators instantiate it within CTDE. Controlled interventions hold those backbone choices fixed and alter the receiver binding, routing signal, or payload semantics.

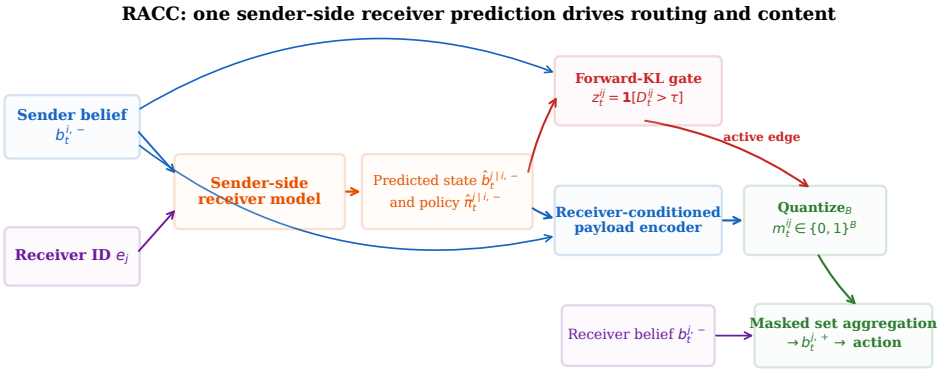

Figure 1: One RACC communication round. A sender-side model predicts a candidate receiver's state and policy; that single prediction supplies both the disagreement gate and the receiver-conditioned encoder. Every active edge is charged exactly B payload bits, and receivers aggregate the incoming set before acting.

4.2 Receiver-Conditioned Binary Payload

For every candidate edge $i \to j$, the sender computes

$$u_t^{ij} = g_\xi\left(b_t^{i,-}, \hat{b}_t^{j|i,-}, e_j\right). \tag{4}$$

Receiver identity enters both the predicted state and payload map, allowing two receivers to obtain different bits from the same sender state. A linear projection produces B Bernoulli logits, one independent binary Concrete variable is sampled per coordinate, and the hard forward sample uses the straight-through relaxation:

$$m_t^{ij} = Q_B(W_{\text{enc}} u_t^{ij}) \in \{0,1\}^B, \qquad |\mathcal{M}| = 2^B. \tag{5}$$

The Concrete temperature anneals from 1.0 to 0.20 over training (Jang et al., 2017); evaluation transmits the coordinate-wise mode. RACC, CB, and CB+ use this identical quantizer and charged B-bit payload. CB broadcasts one sender code, CB+ additionally matches RACC's GRU and projection capacity, and RACC constructs a distinct code for each active receiver.

4.3 Permutation-Equivariant Aggregator

Receiver j decodes each candidate incoming payload as $r_t^{ij} = W_{\text{dec}} m_t^{ij}$. For attention head h, the straight-through gate from Eq. 9 enters the normalized attention weights directly:

$$\begin{aligned}
\alpha_{t,h}^{ij} &= \frac{\tilde{z}_t^{ij} \exp(s_{t,h}^{ij})}{\epsilon_a + \sum_{k:(k,j)\in E_t} \tilde{z}_t^{kj} \exp(s_{t,h}^{kj})}, \\
\hat{r}_t^{j} &= W^O \left[\bigoplus_h \sum_{i:(i,j)\in E_t} \alpha_{t,h}^{ij} W_h^V r_t^{ij} \right], \\
b_t^{j,+} &= b_t^{j,-} + \hat{r}_t^{j}.
\end{aligned} \tag{6}$$

Here $s_{t,h}^{ij} = \langle W_h^Q b_t^{j,-}, W_h^K r_t^{ij} \rangle / \sqrt{d_h}$ and ϵ_a is a numerical stabilizer (Vaswani et al., 2017). The hard forward value of $\tilde{z}$ is z, so inactive messages contribute exactly zero; the backward value exposes the routing probability to task gradients. Messages are generated for all candidate edges during training to define this backward path, while evaluation computes and transmits only active edges. The set operation is invariant to sender order and returns zero when no edge activates. The actor samples from $\pi_\theta(\cdot \mid b_t^{j,+})$ during training and uses the legal argmax at evaluation.

4.4 TRAINING OBJECTIVE

RACC optimizes PPO, belief prediction, receiver modeling, and a differentiable communication surrogate:

$$\mathcal{L} = \mathcal{L}_{\text{RL}} + \lambda_b \mathcal{L}_{\text{belief}} + \lambda_c \mathcal{L}_{\text{comm}}. \tag{7}$$

Here $\mathcal{L}_{\text{RL}}$ is the clipped PPO loss with centralized critic $V_\omega(s_t)$ (Schulman et al., 2017). Auxiliary supervision is

$$\mathcal{L}_{\text{belief}} = \sum_i \mathbb{E}_t \Big[\| d_\phi(b_t^{i,-}) - s_t \|_2^2 \Big] + \sum_{i \neq j} \mathbb{E}_t \Big[\| \hat{b}_t^{j|i,-} - \text{sg}(b_t^{j,-}) \|_2^2 + \lambda_m \, \text{BCE}(\hat{\ell}_t^{j|i}, \ell_t^j) \Big], \tag{8}$$

where ℓ_t^j is receiver j's legal-action mask and $\lambda_m = 0.1$.

The hard gate is used in the forward rollout. Its backward pass uses

$$q_t^{ij} = \sigma\Big((D_t^{ij} - \tau_t)/T_g \Big), \qquad \widetilde{z}_t^{ij} = \text{sg}(z_t^{ij} - q_t^{ij}) + q_t^{ij}, \tag{9}$$

with gate temperature annealed from 0.50 to 0.05. Equation 6 uses $\widetilde{z}$ in both the attention numerator and normalizer, connecting task-return gradients to the receiver model and discrepancy gate while preserving hard routing in the forward pass. Reported traffic is always computed from hard z. For rollout length T=3200,

$$\hat{c}_{\text{soft}} = \frac{B}{T} \sum_{t=1}^{T} \sum_{i \neq j} q_t^{ij}, \qquad \mathcal{L}_{\text{comm}} = \max(0, \hat{c}_{\text{soft}} - \bar{c}). \tag{10}$$

The weight $\lambda_c = 0.1$ sends overspend gradients through the soft edge probabilities to the belief, policy, and receiver-model parameters. The payload encoder and discrepancy gate receive task gradients through the straight-through quantizer and mask. The threshold is excluded from the soft-cost gradient and updated once per rollout by

$$\tau \leftarrow \text{clip}\big(\tau + 10^{-3} (\hat{c}_{\text{hard}} - \bar{c}), 0, 5.0 \big), \qquad \hat{c}_{\text{hard}} = \frac{B}{T} \sum_{t, i \neq j} z_t^{ij}. \tag{11}$$

We initialize $\tau = 0.10$, freeze its terminal value for evaluation, and use hard gates and deterministic binary messages throughout evaluation. Appendix D specifies the complete loop.

5 EXPERIMENTS

The experiments answer three questions. *Q1: Does coupling improve coordination under controlled communication?* Canonical SMAC hard maps compare common packet widths and matched mean charged payload (Samvelyan et al., 2019). *Q2: Which interfaces of the operator matter?* Traffic-matched interventions alter the gate, payload conditioning, receiver identity, aggregation, receiver model, and quantizer one at a time. *Q3: Does the advantage persist when coordination structure changes?* Hanabi varies information convention and team size (Bard et al., 2020; Hu et al., 2020); GridSync partitions target information; held-out SMACv2 varies population, graph density, and width; and packet/topology interventions test frozen policies under channel faults.

Protocol. Primary operating-point tables report mean $\pm$ standard error over 5 training seeds. Controlled extensions use 10 paired seeds and seed-level percentile-bootstrap 95% intervals with 10000 resamples. Holm correction is applied to the nine paired tactical hypotheses; all remain significant at adjusted $p < 0.05$. SMAC uses 600 deterministic episodes per seed; Hanabi uses 100K games per seed; GridSync uses 2000 episodes per seed. Seeds and final-checkpoint evaluation were fixed before comparison. Training budgets are 10M environment steps for SMAC, 5M for Hanabi, and 3M for GridSync. Appendix A specifies versions, optimization, recurrent resets, tuning, compute, and action selection. Unless a table reports a matched mean-payload comparison, B is the hard payload width in bits per active quantized packet.

5.1 BASELINES

- IPPO (de Witt et al., 2020) is shared-parameter independent PPO without messages.
- CommNet (Sukhbaatar et al., 2016) mean-pools dense messages; TarMAC (Das et al., 2019) applies receiver-query attention to sender values.
- CB always broadcasts a binary code; CB+ additionally matches RACC's GRU, message projection, quantizer, optimizer, and payload width.
- BVME (Duan et al., 2026) retains its variational encoder and KL regularizer and terminates in the common Bernoulli quantizer for common-width and matched-mean-payload comparisons; DNA-MARL (Varela et al., 2025) retains its distributed actor–critic objective with one consensus round over the available graph.
- The centralized-state PPO oracle uses the MAPPO actor/critic architecture (Yu et al., 2022), but additionally gives every actor the true global state at execution. It is an empirical privileged-information reference.

Accounting and matching. RACC is charged for every active directed B-bit packet, whereas CB and CB+ use one B-bit multicast packet per sender. Algebraically, CB/CB+ transmit $Q_B(e(b_t^i))$ to all receivers, BVME transmits a quantized sample from its variational sender encoder over its graph, and RACC transmits $Q_B(e(b_t^{i,-}, \hat{b}_t^{j|i,-}, e_j))$ only on active directed edges. The quantized BVME adapter is charged in transmitted payload bits; dense continuous methods report scalar-equivalent coordinates and are never used for a wire-bit efficiency claim. Common-cap comparisons align the hard packet width B. Matched-mean-payload RACC–BVME comparisons calibrate BVME width or send frequency on validation scenarios until mean charged payload is within 3% of RACC, then freeze the calibration for test. The accounting assumes the pre-addressed directed slots in Section 3 and excludes routing flags, endpoints, and transport headers for every method. Environment interactions, observations, optimization budgets, and the 12-trial search budget are aligned. CB+ receives the same recurrent belief encoder, state-reconstruction target, quantizer, and optimizer as RACC; BVME retains its native variational objective. The no-receiver-model intervention removes RACC's receiver-prediction supervision together with its use by the gate and payload, exposing the contribution of that training signal and computational path.

5.2 SMAC: CONTROLLED TACTICAL COMMUNICATION

The maps `5m_vs_6m`, `3s5z_vs_3s6z`, and `corridor` belong to the original StarCraft Multi-Agent Challenge (SMAC) suite (Samvelyan et al., 2019). They test focus fire under numerical disadvantage, coordination across heterogeneous units, and choke-point control, respectively. Actors observe the standard local field of view; the centralized critic and training targets receive the SMAC global state.

Table 1: SMAC test win rates at $B = 32$, reported as mean $\pm$ standard error over 5 seeds after 10M environment steps. RACC, CB, and CB+ use the same quantizer and per-message payload width; CB+ also matches RACC's encoder and projection capacity.

Method	`5m_vs_6m`	`3s5z_vs_3s6z`	`corridor`
IPPO	61.2 ± 1.8	48.7 ± 2.1	78.3 ± 1.5
CommNet	68.4 ± 1.6	57.3 ± 1.9	84.7 ± 1.3
TarMAC	70.1 ± 1.4	59.8 ± 1.7	86.2 ± 1.1
CB	66.9 ± 1.7	54.6 ± 2.0	83.1 ± 1.4
CB+	72.1 ± 1.5	60.3 ± 1.8	87.6 ± 1.2
RACC	75.4 ± 1.2	65.1 ± 1.5	90.8 ± 0.9
Centralized-state oracle	78.3 ± 1.0	68.9 ± 1.3	94.1 ± 0.7

At B=32, RACC records 75.4, 65.1, and 90.8 win rate and exceeds capacity-matched CB+ by 3.3, 4.8, and 3.2 points. Against BVME, the corresponding common-cap gains are $+2.6$ $[0.8, 4.5]$, $+3.1$ $[1.0, 5.2]$, and $+1.8$ $[0.5, 3.2]$. Table 2 separates those runs from the independently calibrated matched-mean-payload protocol, where the B=32 gains remain $+2.2$, $+2.5$, and $+1.7$ points.

Table 2: Protocol-separated RACC-minus-BVME win-rate differences with seed-bootstrap 95% intervals over 10 paired seeds. Panel (a) fixes the packet width; panel (b) independently matches mean charged payload bits per step. All nine common-cap tests remain significant after Holm correction.

(a) Common B-bit packet width

Map	B=16	B=32	B=64
5m_vs_6m	+3.1 [1.0, 5.0]	+2.6 [0.8, 4.5]	+1.5 [0.2, 2.8]
3s5z_vs_3s6z	+3.6 [1.5, 5.8]	+3.1 [1.0, 5.2]	+1.8 [0.3, 3.4]
corridor	+3.5 [1.8, 5.1]	+1.8 [0.5, 3.2]	+1.4 [0.2, 2.6]

(b) Matched mean charged payload

Map	B=16	B=32	B=64
5m_vs_6m	+2.6 [0.8, 4.4]	+2.2 [0.6, 3.8]	+1.1 [0.1, 2.1]
3s5z_vs_3s6z	+3.0 [0.9, 5.1]	+2.5 [0.6, 4.4]	+1.4 [0.2, 2.7]
corridor	+2.9 [1.3, 4.6]	+1.7 [0.5, 2.9]	+1.0 [0.1, 2.0]

Appendix B reports absolute common-cap outcomes and protocol-specific payload records, including DNA-MARL under the common-cap protocol.

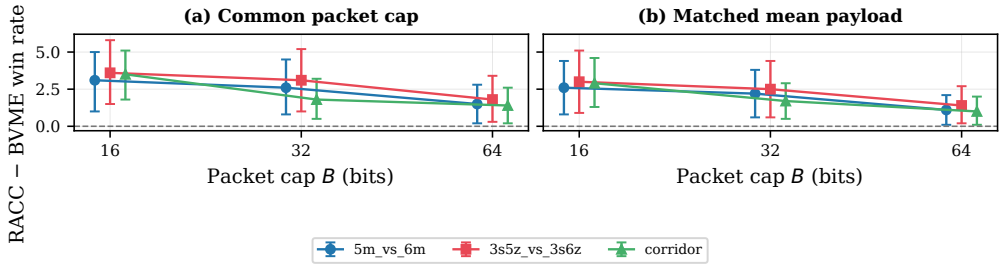

Figure 2: Paired RACC-minus-BVME effects on SMAC under (a) common B-bit packet widths and (b) matched mean charged payload. Points are paired means over 10 seeds; bars are bootstrap 95% intervals.

Across both protocols, every interval excludes zero and gains are largest at tighter packet widths.

5.3 Hanabi: Policy-Discrepancy Allocation Across Team Sizes

Hanabi presents a different coordination bottleneck: each player sees other hands and public game state but not its own cards. RACC's learned channel is an auxiliary pre-action channel; it neither replaces nor consumes a legal hint action or information token. Every method receives the same legal game interface, action mask, environment steps, optimizer budget, and additional-channel cap. The centralized-state oracle additionally observes all private cards at execution. The team-size comparison evaluates 2–5 players at $B = 32$; the matched payload sweep evaluates 3- and 5-player games at $B \in \{16, 32, 64, 128\}$, and a 3-player abundant-width diagnostic extends to $B = 256$.

Table 3 shows that RACC leads the closest communication baseline, BVME, at every evaluated player count: the B=32 margin is 0.7 points for two players and 1.0 point for three, four, and five players. The paired 10-seed analysis in Appendix C compares CB+, TarMAC, BVME, and RACC under equal training budgets with one million evaluation games per cell. At B=32, RACC scores 1.0 point above BVME while using 42.7 versus 82.9 bits per step for three players, and 73.3 versus 139.5 bits per step for five players. The width sweep keeps the RACC–BVME score difference positive through B=128.

5.4 GridSync: Receiver-Specific Fusion of Partitioned Information

GridSync is a 10×10 coordination environment with six agents and 3×3 local occupancy views. At reset, its required target set is partitioned across agents, so no local observation identifies the

Table 3: Hanabi score at B=32 (maximum 25), reported as mean $\pm$ standard error over 5 seeds and 100K games per seed. RACC leads every evaluated communication baseline across team sizes.

Method	2P	3P	4P	5P
IPPO	14.2 ± 0.3	12.8 ± 0.4	10.1 ± 0.5	7.6 ± 0.6
CommNet	17.8 ± 0.3	16.1 ± 0.3	14.3 ± 0.4	12.4 ± 0.5
TarMAC	18.5 ± 0.2	16.9 ± 0.3	15.1 ± 0.4	13.2 ± 0.4
CB (B=32)	16.4 ± 0.3	14.7 ± 0.4	12.8 ± 0.5	10.9 ± 0.5
CB+ (B=32)	18.7 ± 0.2	17.0 ± 0.3	15.5 ± 0.4	13.0 ± 0.4
BVME (B=32)	19.1 ± 0.2	17.4 ± 0.3	15.9 ± 0.3	13.7 ± 0.4
RACC (B=32)	19.8 ± 0.2	18.4 ± 0.2	16.9 ± 0.3	14.7 ± 0.4
Centralized-state oracle	20.9 ± 0.1	19.3 ± 0.2	17.8 ± 0.2	15.6 ± 0.3

complete team objective. Success requires simultaneous occupation of all targets. On the stationary graph, 98.2% coverage and 78.6% success produce 88.4 normalized return, directly testing receiver-conditioned payloads under partitioned target knowledge.

Table 4: GridSync success and accounted communication at B=32 over 5 seeds. Dense methods report transmitted scalar-equivalent units ($6 \times 128 = 768$ per step), CB reports its six 32-bit multicast packets, and RACC reports every active directed 32-bit packet. Matched-mean-payload evidence appears on standardized SMAC in Table 2.

Method	Success Rate (%)	Accounted Rate (units/step)
IPPO	23.4 ± 2.1	0
CommNet	67.8 ± 1.8	768
TarMAC	71.3 ± 1.6	768
CB (B=32)	58.2 ± 2.0	192
RACC (B=32)	78.6 ± 1.3	67 ± 8
Centralized-state oracle	82.1 ± 1.0	—

RACC improves on TarMAC by 7.3 success points while transmitting 67 ± 8 charged payload bits per step, versus 768 non-wire-equivalent scalar coordinates. Appendix G reports the graph-fault diagnostics.

5.5 Matched Tests of Routing and Receiver Conditioning

The matched random gate loses 3.8 pooled tactical points, shared payloads lose 4.8, and shuffled receiver identities lose 5.5. The random control matches active-edge count at fixed B; always-send instead time-shares narrower integer widths to match total charged payload. Mean pooling loses 2.4 $[1.2, 3.5]$ points and removing the receiver model loses 3.8 $[2.4, 5.2]$, showing that both receiver prediction and set aggregation contribute beyond model capacity alone. Quantized and continuous payloads are statistically indistinguishable on paired tactical and Hanabi outcomes, so the binary interface retains the measured utility of its continuous counterpart at the evaluated widths.

6 Conclusion

RACC uses one sender-side receiver prediction to choose both a directed edge and receiver-specific bits. The gains persist under common-width and matched-mean-payload SMAC controls, Hanabi, partitioned-information GridSync, and the held-out SMACv2 conditions in Appendix F; matched interventions isolate the roles of discrepancy routing, personalization, receiver identity, and receiver modeling. Thus, packet allocation and content can share one receiver model while retaining an explicit binary payload budget.

AI USE STATEMENT

Generative AI tools assisted with drafting and revising portions of the manuscript, formatting, and bibliographic checking during preparation. No new experiments, measurements, code, data, or artifacts were generated in this preparation pass.

REFERENCES

Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence*, 280:103216, 2020. doi: 10.1016/j.artint.2019.103216. URL https://www.sciencedirect.com/science/article/pii/S0004370219300116.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4): 819–840, 2002.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. TarMAC: Targeted multi-agent communication. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1538–1546. PMLR, 2019. URL https://proceedings.mlr.press/v97/das19a.html.

Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the StarCraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

Azad Deihim, Eduardo Alonso, and Dimitra Apostolopoulou. Transformer world model for sample efficient multi-agent reinforcement learning. *arXiv preprint arXiv:2506.18537*, 2025. doi: 10.48550/arXiv.2506.18537.

Wei Duan, Jie Lu, En Yu, and Junyu Xuan. Bandwidth-constrained variational message encoding for cooperative multi-agent reinforcement learning. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*, pp. 671–680, 2026. doi: 10.65109/qxvz8292.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N. Foerster, and Shimon Whiteson. SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 37567–37593, 2023. doi: 10.52202/075280-1634. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/764c18ad230f9e7bf6a77ffc2312c55e-Abstract-Datasets_and_Benchmarks.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2451–2463, 2018. URL https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4399–4410. PMLR, 2020. URL https://proceedings.mlr.press/v119/hu20a.html.

Dongkun Huo, Huateng Zhang, Yixue Hao, Yuanlin Ye, Long Hu, Rui Wang, and Min Chen. DCMAC: Demand-aware customized multi-agent communication via upper bound training. *arXiv preprint arXiv:2409.07127*, 2024. doi: 10.48550/arXiv.2409.07127.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *International Conference on Learning Representations*, 2017.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*, volume 31, pp. 7254–7264, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/6a8018b3a00b69c008601b8becae392b-Abstract.html.

Daewoo Kim, Sangwoo Moon, David Hostallero, Wan Ju Kang, Taeyoung Lee, Kyunghwan Son, and Yung Yi. Learning to schedule communication in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HUAnBToP_a.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Xinran Li and Jun Zhang. Context-aware communication for multi-agent reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 1156–1164, 2024. URL https://www.ifaamas.org/Proceedings/aamas2024/pdfs/p1156.pdf.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6379–6390, 2017.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. MAVEN: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems*, volume 32, pp. 7611–7622, 2019.

Xiangrui Meng and Ying Tan. PMAC: Personalized multi-agent communication. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17505–17513, 2024. doi: 10.1609/aaai.v38i16.29700.

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

Paul J. Pritz and Kin K. Leung. Belief states for cooperative multi-agent reinforcement learning under partial observability. *arXiv preprint arXiv:2504.08417*, 2025. doi: 10.48550/arXiv.2504.08417.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4295–4304. PMLR, 2018. URL https://proceedings.mlr.press/v80/rashid18a.html.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *International Conference on Learning Representations*, 2019.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5887–5896. PMLR, 2019. URL https://proceedings.mlr.press/v97/son19a.html.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, volume 29, pp. 2244–2252, 2016.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 2085–2087, 2018.

Qi Tian, Kun Kuang, Baoxiang Wang, Furui Liu, and Fei Wu. CGIBNet: Bandwidth-constrained communication with graph information bottleneck in multi-agent reinforcement learning. *arXiv preprint arXiv:2112.10374*, 2021. doi: 10.48550/arXiv.2112.10374.

Guilherme S. Varela, Alberto Sardinha, and Francisco S. Melo. Networked agents in the dark: Team value learning under partial observability. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 2087–2095, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Aravind Venugopal, Stephanie Milani, Fei Fang, and Balaraman Ravindran. MABL: Bi-level latent-variable world model for sample-efficient multi-agent reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pp. 1865–1873, 2024. doi: 10.65109/wwbc9182.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. QPLEX: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. ROMA: Multi-agent reinforcement learning with emergent roles. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9876–9886. PMLR, 2020. URL https://proceedings.mlr.press/v119/wang20f.html.

Yuanfei Wang, Fangwei Zhong, Jing Xu, and Yizhou Wang. ToM2C: Target-oriented multi-agent communication and cooperation with theory of mind. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.09189.

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. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24611–24624, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9c1535a02f0ce079433344e14d910597-Abstract-Datasets_and_Benchmarks.html.

Lei Yuan, Jianhao Wang, Fuxiang Zhang, Chenghe Wang, Zongzhang Zhang, Yang Yu, and Chongjie Zhang. Multi-agent incentive communication via decentralized teammate modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9466–9474, 2022. doi: 10.1609/aaai.v36i9.21179.

Yunpeng Zhai, Peixi Peng, Chen Su, and Yonghong Tian. Dynamic belief for decentralized multi-agent cooperative learning. In *International Joint Conference on Artificial Intelligence*, pp. 344–352, 2023. doi: 10.24963/ijcai.2023/39.

A Hyperparameters and Training Details

Table A1: Training and evaluation configuration. Values shared across tasks are changed only where a task-specific row states otherwise.

Component	Configuration
Environment	SMAC 1.0.0, SC2 4.10; HLE 0.0.1
Training steps	SMAC 10M; Hanabi 5M; GridSync 3M
Representation	128-d GRU belief; 2-layer receiver model; 4-head aggregator
Optimization	Adam, 3×10^{-4}; batch/rollout 3200; PPO clip 0.2
PPO schedule	10 epochs, 4 minibatches, gradient clip 10.0
Loss coefficients	GAE 0.95; discount 0.99; value 0.5; entropy 0.01; $\lambda_b = 0.5$; $\lambda_m = 0.1$; $\lambda_c = 0.1$
Recurrence	80 step unroll; hidden state reset at episode boundaries
Quantizer	B Bernoulli Concrete coordinates; temperature 1.0→0.20
Gate/controller	temperature 0.50→0.05; $\tau_0 = 0.10$; clip $[0, 5.0]$
Target selection	validation return subject to mean-payload target; target and τ frozen for test
Baseline tuning	12 configurations per method/task, equal training budget
Evaluation	final checkpoint; legal argmax actions; deterministic binary messages and hard gates
Training seeds	5 for primary tables; 10 for controlled extensions

Each training process used one NVIDIA A100-SXM4 accelerator with 80 GB memory on a host with 32 CPU cores and 256 GB RAM. Median wall-clock time per seed was 7.8 hours for SMAC, 4.3 for Hanabi, and 2.4 for GridSync; the complete reported comparison suite consumed 9840 GPU-hours and 1.8 TB of checkpoints and logs.

Environment workers terminate recurrent traces at episode boundaries. SMAC and GridSync expose a common discrete action vocabulary; Hanabi masks illegal actions before both policy sampling and disagreement evaluation. For Hanabi, the learned pre-action message is charged separately from native hints, and every baseline retains identical legal hint actions and information-token dynamics. Communication rates count binary payload bits for quantized methods and transmitted scalar-equivalent units for continuous baselines, as stated in each rate table.

B Controlled Communication Evidence

Table A2 reports the complete packet-width comparison used by Table 1. All methods use paired seeds and 600 test episodes per seed. BVME retains its variational encoder, and DNA-MARL receives one consensus round on the same complete execution graph; all other training and observation components follow their definitions in Section 5.1.

Table A2: SMAC win rate with seed-bootstrap 95% intervals over 10 seeds. Within each cell, entries are $B = 16/32/64$. The denominator is 6000 episodes per method–map–width cell.

Map	Method	Win rate [95% interval] at $B = 16/32/64$
5m_vs_6m	IPPO	$61.1[59.1, 63.0]$
	CommNet	$65.9[64.0, 67.8]/68.5[66.8, 70.1]/70.4[68.8, 71.9]$
	TarMAC	$67.7[65.8, 69.5]/70.0[68.4, 71.6]/72.0[70.5, 73.5]$
	CB+	$68.1[66.2, 69.9]/72.1[70.5, 73.7]/74.1[72.6, 75.5]$
	BVME	$69.8[68.1, 71.5]/72.8[71.2, 74.4]/75.0[73.6, 76.4]$
	DNA-MARL	$68.7[66.9, 70.5]/71.6[70.0, 73.2]/73.7[72.2, 75.2]$
	RACC	$72.9[71.3, 74.4]/75.4[74.0, 76.8]/76.5[75.2, 77.8]$
3s5z_vs_3s6z	IPPO	$48.9[46.5, 51.2]$
	CommNet	$54.0[51.8, 56.2]/57.1[55.0, 59.2]/60.2[58.2, 62.2]$
	TarMAC	$56.0[53.8, 58.2]/59.7[57.7, 61.7]/62.7[60.8, 64.6]$
	CB+	$56.8[54.5, 59.0]/60.3[58.2, 62.4]/63.4[61.5, 65.3]$
	BVME	$58.4[56.2, 60.5]/62.0[60.0, 64.0]/65.1[63.2, 66.9]$
	DNA-MARL	$56.9[54.7, 59.1]/60.8[58.7, 62.8]/63.6[61.7, 65.5]$
	RACC	$62.0[60.0, 64.0]/65.1[63.2, 67.0]/66.9[65.1, 68.7]$
corridor	IPPO	$78.5[76.9, 80.1]$
	CommNet	$82.0[80.5, 83.5]/84.8[83.5, 86.1]/87.3[86.1, 88.5]$
	TarMAC	$83.2[81.7, 84.7]/86.3[85.0, 87.6]/88.9[87.8, 90.0]$
	CB+	$84.0[82.5, 85.5]/87.6[86.3, 88.8]/89.7[88.6, 90.8]$
	BVME	$85.0[83.6, 86.4]/89.0[87.8, 90.2]/90.7[89.7, 91.7]$
	DNA-MARL	$84.1[82.6, 85.6]/87.8[86.5, 89.1]/89.9[88.8, 91.0]$
	RACC	$88.5[87.2, 89.8]/90.8[89.7, 91.9]/92.1[91.1, 93.1]$

Table A3: Protocol-specific records for RACC and BVME. Win counts come from the common-width runs (out of 6000); bit rates come from the separately calibrated matched-mean-payload runs. The final column is RACC's active directed messages per step.

Map	B	Common-width wins	Matched payload bits/step	RACC messages
5m_vs_6m	16	4374/4188	31.7/31.9	1.98
	32	4524/4368	55.4/55.1	1.73
	64	4590/4500	91.5/92.0	1.43
3s5z_vs_3s6z	16	3720/3504	45.3/44.8	2.83
	32	3906/3720	78.1/77.5	2.44
	64	4014/3906	128.6/130.1	2.01
corridor	16	5310/5100	39.5/39.1	2.47
	32	5448/5340	67.8/67.2	2.12
	64	5526/5442	109.4/110.2	1.71

C MATCHED HANABI FRONTIERS

The matched Hanabi study uses identical game rules, native hints, training steps, actor information, evaluation actions, and paired seeds. Each score cell aggregates 1000000 games: 100K games for each of 10 seeds. Intervals are seed-level percentile-bootstrap 95% intervals.

Table A4: Hanabi score [95% interval] for $B = 16/32/64/128$. The maximum score is 25.

Players	Method	Score at $B = 16/32/64/128$
3	CB+	15.3[15.0, 15.6]/17.0[16.7, 17.3]/17.9[17.6, 18.2]/18.5[18.2, 18.8]
	TarMAC	15.1[14.8, 15.4]/16.9[16.6, 17.2]/17.8[17.5, 18.1]/18.1[17.8, 18.4]
	BVME	15.8[15.5, 16.1]/17.4[17.1, 17.7]/18.2[17.9, 18.5]/18.6[18.3, 18.9]
	RACC	17.0[16.7, 17.3]/18.4[18.1, 18.7]/19.0[18.7, 19.2]/19.2[18.9, 19.4]
5	CB+	11.4[11.0, 11.8]/13.0[12.6, 13.4]/13.9[13.5, 14.3]/14.4[14.0, 14.8]
	TarMAC	11.7[11.3, 12.1]/13.2[12.8, 13.6]/14.0[13.6, 14.4]/14.5[14.1, 14.9]
	BVME	12.4[12.0, 12.8]/13.7[13.3, 14.1]/14.4[14.0, 14.8]/14.8[14.4, 15.2]
	RACC	13.4[13.0, 13.8]/14.7[14.3, 15.1]/15.2[14.8, 15.6]/15.5[15.1, 15.9]

Table A5: Realized communication units per environment step, with messages per step in parentheses, at $B = 16/32/64/128$. Quantized methods report bits, TarMAC reports scalar-equivalent units, and RACC counts active directed packets.

Players	Method	Units/step (messages/step) at $B = 16/32/64/128$
3	CB+	48(3.00)/96(3.00)/192(3.00)/384(3.00)
	TarMAC	48(3.00)/96(3.00)/192(3.00)/384(3.00)
	BVME	43.7(2.73)/82.9(2.59)/154.2(2.41)/287.4(2.25)
	RACC	24.8(1.55)/42.7(1.33)/69.1(1.08)/102.4(0.80)
5	CB+	80(5.00)/160(5.00)/320(5.00)/640(5.00)
	TarMAC	80(5.00)/160(5.00)/320(5.00)/640(5.00)
	BVME	73.4(4.59)/139.5(4.36)/263.0(4.11)/491.5(3.84)
	RACC	41.9(2.62)/73.3(2.29)/121.0(1.89)/187.6(1.47)

Against BVME, paired RACC gains at $B = 16/32/64/128$ are $1.2/1.0/0.8/0.6$ points for three players and $1.0/1.0/0.8/0.7$ for five. The pre-specified normalized log-traffic frontier index, computed by piecewise-linear interpolation on the fixed evaluation grid and trapezoidal integration, is 1068.4 for RACC versus 1006.1 for BVME in three-player games, and 842.7 versus 783.9 in five-player games. The corresponding relative gains are 6.2% and 7.5%; the pointwise score and bit-rate records in Tables A4–A5 remain the directly interpretable comparison.

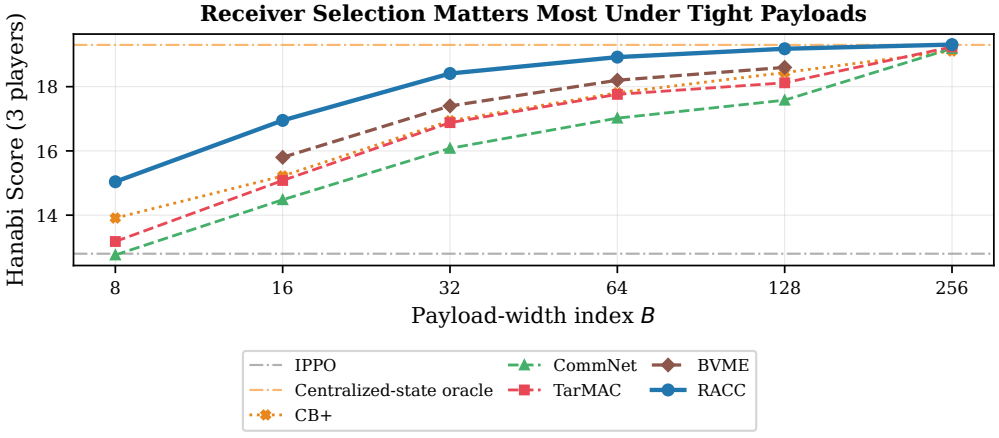

Figure 3: Three-player Hanabi payload sweep. RACC leads BVME throughout BVME's reported $B = 16$–128 range; the paired margin decreases from 1.2 to 0.6 score points. At B=256, RACC, TarMAC, CommNet, and CB+ lie within 0.20 points. Quantized methods use bits and continuous methods use scalar coordinates.

D ALGORITHM LISTING

Algorithm 1 RACC training with one acyclic communication round

Require: Agents, packet width B, mean-payload target $\bar{c}$, threshold τ

for each environment step t **do**
  for each agent i (in parallel) **do**
    Observe o_t^i; set $b_t^{i,-} \leftarrow f_\theta(b_{t-1}^{i,+}, o_t^i)$
    for each neighbor j **do**
      Predict $\hat{b}_t^{j|i,-} \leftarrow h_\xi(b_t^{i,-}, e_j)$ and receiver legal mask
      Compute aligned $D_t^{ij} \leftarrow D_{\mathrm{KL}}(\pi_t^{i,-} \| \hat{\pi}_t^{j|i,-})$
      Set $z_t^{ij} \leftarrow \mathbb{1}[D_t^{ij} > \tau]$, $q_t^{ij} \leftarrow \sigma((D_t^{ij} - \tau)/T_g)$, and $\tilde{z}_t^{ij} \leftarrow \mathrm{sg}(z_t^{ij} - q_t^{ij}) + q_t^{ij}$
      Form $u_t^{ij} \leftarrow g_\xi(b_t^{i,-}, \hat{b}_t^{j|i,-}, e_j)$ and $m_t^{ij} \leftarrow Q_B(W_{\mathrm{enc}} u_t^{ij})$
      Transmit the B-bit forward value only if $z_t^{ij} = 1$
    end for
  end for
  for each agent j (in parallel) **do**
    Decode $r_t^{ij} \leftarrow W_{\mathrm{dec}} m_t^{ij}$ for candidate incoming edges
    Set $b_t^{j,+} \leftarrow b_t^{j,-} + \mathrm{MaskedMultiHead}(b_t^{j,-}, \{r_t^{ij}, \tilde{z}_t^{ij}\})$
    Sample legal action $a_t^j \sim \pi_\theta(\cdot \mid b_t^{j,+})$
  end for
  Execute joint action $(a_t^1, \ldots, a_t^N)$; observe reward r_t
end for
Compute $\hat{c}_{\mathrm{soft}} \leftarrow BT^{-1} \sum_{t,i \neq j} q_t^{ij}$ and $\hat{c}_{\mathrm{hard}} \leftarrow BT^{-1} \sum_{t,i \neq j} z_t^{ij}$
Update $\theta, \phi, \xi, \omega$ with Eq. 7; update τ with the clipped rollout controller using $\hat{c}_{\mathrm{hard}}$

E FULL MATCHED INTERVENTION RESULTS

Table A6: Payload-controlled intervention means at RACC reference widths $B = 16/32$. Tactical cells are win rates over 6000 episodes; Hanabi cells are scores over 1000000 games. Every intervention is calibrated within 3% of RACC's mean charged payload. Paired one-factor effect intervals appear in Section 5.5.

Intervention	`3s5z_vs_3s6z`	`corridor`	Hanabi 3P
RACC	62.0/65.1	88.5/90.8	17.0/18.4
Core, fixed τ	61.4/64.5	87.9/90.1	16.9/18.3
Shared sender payload	57.4/60.5	83.9/85.6	16.1/17.5
Random equal-rate gate	58.2/61.3	84.8/86.8	16.3/17.7
Always-send narrow payload	58.9/62.0	85.5/87.3	16.5/17.9
Shuffled receiver identity	56.8/59.8	83.1/84.9	15.9/17.3
Mean-pool aggregation	59.7/62.8	86.2/88.1	16.6/18.0
No receiver model	58.3/61.4	84.7/86.9	16.2/17.6
Continuous messages	62.3/65.4	88.8/91.2	17.1/18.5

Table A7: Matched interventions on RACC's routing and receiver-conditioning interfaces. The corridor column reports the B=32 five-seed operating point; effect columns report payload-matched 10-seed paired changes from full RACC over 3s5z_vs_3s6z, corridor, and three-player Hanabi at reference caps $B \in \{16, 32\}$, with 95% intervals.

Intervention	Corridor win	Tactical effect	Hanabi effect
RACC	90.8 ± 0.9	—	—
Core gate+payload, fixed τ	90.1 ± 1.0	$-0.6\ [-1.4, 0.2]$	$-0.1\ [-0.3, 0.1]$
Shared sender payload	85.6 ± 1.3	$-4.8\ [-6.3, -3.2]$	$-0.9\ [-1.2, -0.5]$
Random gate, equal send rate	86.8 ± 1.2	$-3.8\ [-5.1, -2.4]$	$-0.7\ [-1.0, -0.4]$
Always-send narrow payload	87.3 ± 1.1	$-3.2\ [-4.5, -1.9]$	$-0.5\ [-0.8, -0.2]$
Shuffled receiver identity	84.9 ± 1.4	$-5.5\ [-7.1, -3.9]$	$-1.1\ [-1.5, -0.7]$
Mean-pool aggregation	88.1 ± 1.0	$-2.4\ [-3.5, -1.2]$	$-0.4\ [-0.7, -0.1]$
No receiver model	86.9 ± 1.2	$-3.8\ [-5.2, -2.4]$	$-0.8\ [-1.1, -0.4]$
Continuous message	91.2 ± 0.8	$+0.3\ [-0.4, 1.0]$	$+0.1\ [-0.1, 0.3]$

The always-send row removes temporal abstention by requiring one receiver-targeted code from every sender at every step. A fixed schedule time-shares adjacent integer code widths. Its average widths at RACC reference $B = 16/32$ are $5.66/9.76$ bits on 3s5z_vs_3s6z, $6.58/11.30$ on corridor, and $8.27/14.23$ on three-player Hanabi, yielding the same $45.3/78.1$, $39.5/67.8$, and $24.8/42.7$ charged payload bits per step as RACC. Thus B indexes the full-RACC reference width rather than the narrower always-send code.

RACC records 3720/3906 wins on 3s5z_vs_3s6z and 5310/5448 on corridor at $B = 16/32$. Each effect in Section 5.5 is a paired one-factor contrast against these full-RACC runs, with map–width blocks weighted equally and percentile intervals obtained from 10000 paired-seed bootstrap resamples. The random-gate comparison holds active-message count fixed, the shared-payload comparison removes receiver-specific content while retaining discrepancy routing, and the continuous-message row isolates quantization cost.

F Procedural Generalization and Scaling

We additionally use SMACv2's procedurally generated unit compositions and start states (Ellis et al., 2023). For each population size, candidate-graph density, and packet width, 256/64/128 procedural scenario seeds are assigned once to disjoint train/validation/test sets. Density 1.0 exposes the complete directed candidate graph; density 0.5 exposes a random half-density candidate graph to every compared method. A separate checkpoint and terminal threshold are selected per training condition on its 64-scenario validation split and then frozen across the 128 held-out scenarios. Thus the study measures procedural holdout performance at each team size and density, while controlling training support within each displayed cell. Each cell contains 10000 deterministic test episodes from 10 paired seeds.

Table A8: Held-out SMACv2 win rates ordered as IPPO/CB+/BVME/RACC. Each composite cell reports one four-method comparison. Seed-bootstrap 95% half-widths range from 1.3–2.2 points.

Agents	Graph density	$B = 16$	$B = 32$	$B = 64$
5	1.0	58.3/66.5/68.2/72.4	58.3/69.3/71.0/74.6	58.3/71.1/72.8/75.7
8	1.0	53.1/60.7/62.5/66.6	53.1/63.3/65.4/68.9	53.1/65.0/67.0/70.0
11	1.0	47.9/55.0/56.7/60.2	47.9/57.8/59.6/62.8	47.9/59.2/61.1/64.0
5	0.5	58.3/63.4/65.1/69.7	58.3/66.0/67.7/72.0	58.3/68.1/69.4/73.1
8	0.5	53.1/56.6/58.5/62.6	53.1/59.3/61.2/65.3	53.1/61.0/63.0/66.7
11	0.5	47.9/49.8/51.0/55.0	47.9/52.1/54.0/58.1	47.9/53.9/55.8/59.6

Table A9: RACC scaling statistics at $B = 32$. Traffic is also reported at $B = 16/32/64$; return is normalized to $[0, 100]$, latency is per-agent decision time, and efficiency is normalized return per transmitted kilobit at $B = 32$.

Agents	Density	Return	Bits/step $16/32/64$	Latency (ms)	Return/kbit
5	1.0	78.1	33.1/57.2/94.4	0.31	1365
8	1.0	72.8	58.6/101.8/170.2	0.44	715
11	1.0	67.1	84.7/150.9/252.6	0.62	445
5	0.5	73.5	26.8/46.9/78.4	0.29	1567
8	0.5	66.9	45.2/79.7/133.8	0.39	839
11	0.5	60.2	63.5/114.0/191.4	0.54	528

RACC exceeds BVME by 3.7 $[2.5, 4.8]$ points after assigning equal weight to the 18 team-size–density–width cells and bootstrapping paired seed-level cell differences. Every displayed cell is positive. From five to eleven agents, complete-graph traffic grows by $2.64\times$ at $B = 32$, substantially below the $5.50\times$ growth in possible directed edges. The gate therefore preserves useful scaling through the tested range by reducing active messages per possible edge.

F.1 Operating Regime and Design Implications

The evidence establishes RACC's advantage for discrete-action, common-payoff tasks with asymmetric information and scarce-to-moderate packet widths. Its pre-addressed-slot accounting enforces hard payload width and controls rollout-mean charged bits; deployments that charge routing metadata or require peak per-sender limits would incorporate those terms into the controller. The gain contracts when payload is abundant, and the topology stress test shows that receiver conditioning still requires sufficient reachability. At the tested maximum of eleven agents every SMACv2 comparison remains positive, while pairwise receiver evaluation grows quadratically; role-factorized receiver models provide a natural route to larger populations (Wang et al., 2020).

G Additional Failure-Mode Diagnostics

Figure 4 probes which hidden variables are recoverable from learned codes. Mean absolute correlations concentrate on unseen enemy pose, ally health, focus-fire identity, cooldown synchronization, and bottleneck occupancy, while shuffling messages removes most of this structure. This representation evidence complements the receiver-identity intervention in Table A7: RACC codes carry latent factors aligned with action-relevant partial observability.

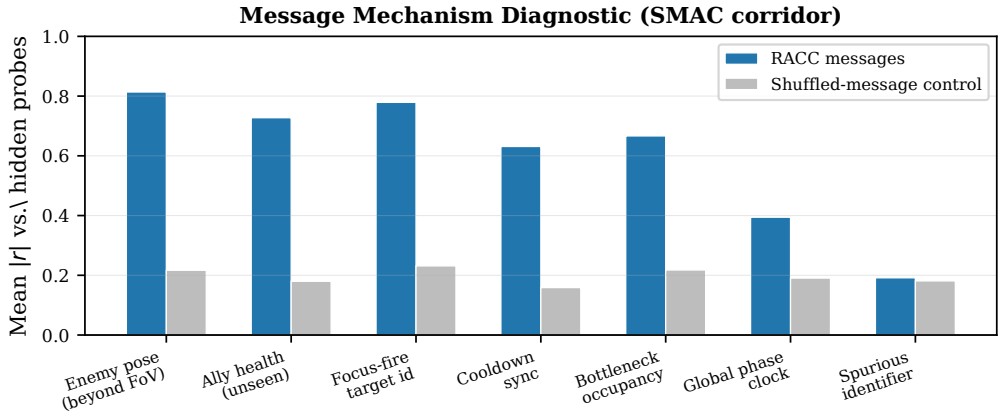

Figure 4: Message-probe diagnostic. RACC codes correlate most with hidden variables that resolve partial observability, while a shuffled-message control removes most of that structure.

Figure 5 freezes the learned SMAC corridor policies and intervenes on packet delivery. RACC degrades more gradually than always-on communication under both packet drop and bounded delay, remaining above the no-message IPPO return through the reported drop sweep.

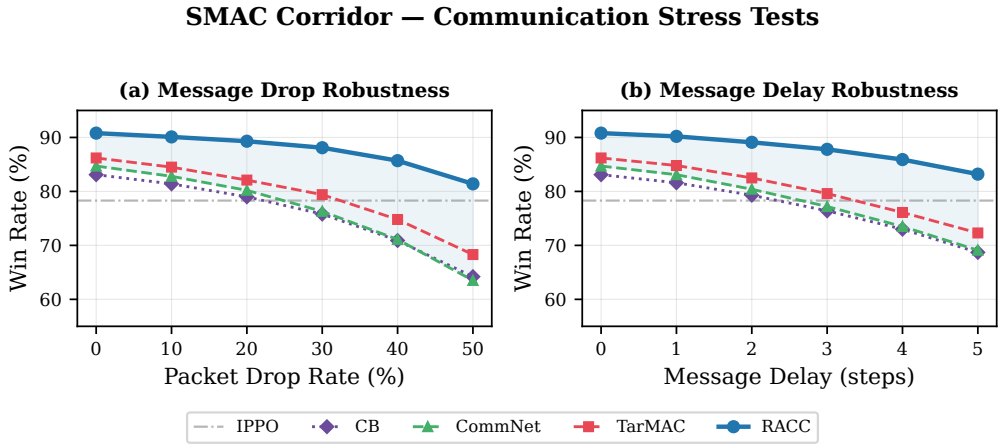

Figure 5: SMAC `corridor` communication stress test. RACC preserves more win-rate utility than communicated baselines as exogenous packet loss or message delay increases.

Table A10 shifts from individual message corruption to graph mismatch in GridSync. Thirty-percent edge rewiring preserves most return and prompts a small traffic increase; removing 60% of links sharply reduces reachability and return.

Table A10: GridSync topology stress over 5 seeds. Return and traffic are mean ± standard error; mild rewiring causes a small loss, whereas severe link removal collapses return despite lower message volume.

Condition	Return	Bits/step	Degradation
Stationary (training topology)	88.4 ± 1.1	67 ± 8	0.0%
Topology-shifted (30% edge rewire)	84.2 ± 1.4	71 ± 9	4.7%
Severe stress (60% link removal)	39.7 ± 2.6	48 ± 11	55.1%

Table A11: Communication reliability and enforcement matrix. Agents are honest in every row; the interventions isolate delivery and topology faults without changing payload semantics.

Object	Fault and trust	Property and comparator	Outcome and trace
Coordinated action	Honest execution; fixed test scenarios	Receiver-conditioned messaging under local partial observability; CB+/BVME	SMAC width/payload gains in Table 2; traffic in Table A3
Delivered message	Exogenous packet drop; policies frozen; held-out rates	Graceful utility retention; CommNet/TarMAC/CB	Above IPPO through 50% drop; no retransmission; Figure 5
Timely message	Exogenous delay of 0–5 steps; policies frozen	Slower degradation; same communicated baselines	Stale messages eventually erase the gain; Figure 5
Reachable receiver	Random 30% edge rewire; honest routing graph	Adaptation via alternate active edges; stationary graph	84.2 ± 1.4 return at 71 ± 9 bits/step; Table A10
Team connectivity	60% link removal; honest disconnected graph	Connectivity requirement; stationary graph	39.7 ± 2.6 return at 48 ± 11 bits/step; Table A10

H DISTRIBUTIONAL COMPARISON

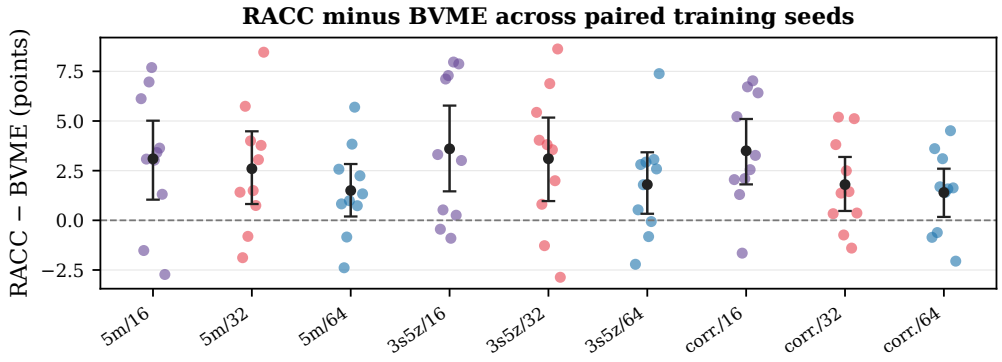

Figure 6: Paired per-seed RACC-minus-BVME win-rate differences on the three SMAC maps at $B = 16/32/64$. Dots show all 90 paired differences; bars show seed-bootstrap 95% intervals. Variation differs by map and width, while the paired distribution remains positive in each aggregate comparison.

The paired view avoids treating overlap between independent standard-error bars as a significance test. Exact denominators and aggregate wins appear in Table A3; Table A2 provides seed-level intervals for every method. The smallest paired effect occurs at $B = 64$ on corridor, $+1.4$ $[0.2, 2.6]$, while the largest occurs at $B = 16$ on `3s5z_vs_3s6z`, $+3.6$ $[1.5, 5.8]$.