International Journal of Engineering, Management and Humanities (IJEMH)
Volume 6, Issue 5, Sep.-Oct., 2025 pp: 165-169 ISSN: 2584-2145 www.ijemh.com

Query-Efficient Multi-Agent Communication with Value-of-Information Policies and Typed Plan Messages

Dr. Karthik Kambhampati

Date of Submission: 15-09-2025 Date of Acceptance: 27-09-2025

Abstract
Communication can dramatically improve multi-agent coordination, yet unconstrained messaging is often costly, brittle, and unsafe. We present a method that is query-efficient in when to communicate and reliable in what to communicate. Our approach couples (i) a value-of-information (VoI) talk policy that learns to ask or send messages only when the expected improvement in task value exceeds communication cost, with (ii) typed plan messages (TPM), a schema-checked, program-of-thought representation for messages that encode executable plans as small abstract syntax trees (ASTs). A runtime checker guarantees type and pre/post-condition validity and proposes minimal repairs when necessary. Across three environments, our method reduces messages while improving success and reliability versus free-form baselines.

I. Introduction

Multi-agent communication enables coordination under partial observability, but unfettered messaging is expensive, fragile, and risky. Two practical questions dominate: when should agents communicate, and what should they say? We propose a method that is query-efficient and execution-reliable via VoI talk policies and typed plan messages (TPM). We summarize contributions and position the work within applied, safety-aware MARL.

II. Problem Formulation

We model a cooperative **partially observable Markov game (POMG)**
G=(S,{Oi}i=1N,A,P,R,γ),\mathcal{G}=(\mathcal{S},\{\mathcal{O}_i\}_{i=1}^N,\mathcal{A},P,R,\gamma),G=(S,{Oi}i=1N,A,P,R,γ),
with NNN agents. At time ttt, agent iii receives a private observation oti∈Oio^i_t∈Oi, selects an environment action ati∈Aa^i_t∈A, and ∈{optionally} emits a message mti∈M∪{∅}m^i_t ∈M∪{∅} to teammates. Messages incur **per-message cost** c(mti)≥0c(m^i_t)≥0 (e.g., bandwidth, latency, $-per-token). The joint transition is st+1∼P(⋅∣st,at)s_{t+1}∼P(⋅∣st,at), with team reward rt=R(st,at)r_t=R(st,at) and discount γ∈(0,1]γ∈(0,1].
We optimize a joint communication-and-control policy
πθ(at,mt∣ot)\pi_\theta(\bm{a}_t,\bm{m}_t\mid \bm{o}_t)πθ(at,mt∣ot) to maximize return minus communication cost:
max⁡θ
J(θ)=E[∑t=0Tγt(R(st,at)−λ∑i=1Nc(mti))],\max_{\theta}\; J(\theta)=\mathbb{E}\!\left[\sum_{t=0}^{T}\gamma^t\Big(R(s_t,\bm{a}_t)-\lambda \sum_{i=1}^N c(m^i_t)\Big)\right],θmaxJ(θ)=E[t=0∑Tγt(R(st,at)−λi=1∑Nc(mti))],
where λ>0\lambda>0λ>0 trades task performance for budget. We allow **explicit budgets** via a soft constraint E[C]≤B∈mathbb{E}[C]≤BE[C]≤B with Lagrange multiplier λ∈lambdaλ tuned by primal–dual updates.

III. Method

Our approach answers **when to communicate** via a **Value-of-Information (VoI)** talk policy and **what to communicate** via **Typed Plan Messages (TPM)** with checking/repair. Execution uses a **plan-conditioned actor** with **safety masking**. Training is actor–critic with auxiliary losses.

3.1 VoI Talk Policy (When to speak)

Intuition: speak if the **expected gain in value** from messaging exceeds its **cost**. Let Vϕ(ot)V_\phi(o_t)Vϕ(ot) be a baseline value estimate and Qψ(o~t,at)Q_\psi(\tilde{o}_t,a_t)Qψ

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 6, Issue 5, Sep.-Oct., 2025 pp: 165-169 ISSN: 2584-2145 www.ijemh.com

(o~t,at) be a value conditioned on a \emph{prospective} message summary (latent) appended to the observation o~t\tilde{o}_to~t. Define the marginal value:
ΔVt(m) ≜ E[Qψ(o~t(m),at)] − Vϕ(ot).\Delta V_t(m)\ \triangleq\ \mathbb{E}\big[Q_\psi(\tilde{o}_t(m),a_t)\big]\ -\ V_\phi(o_t).ΔVt(m) ≜ E[Qψ(o~t(m),at)] − Vϕ(ot).
We select a talk decision and candidate message by
dt=1{maxm∈MΔVt(m)>λc(m)},mt∗∈argmaxmΔVt(m).d_t=₩mathbf{1}₩{₩max_{m₩in₩mathcal{M}} ₩Delta V_t(m) > ₩lambda₩, c(m)₩},₩qquad m^*_t ₩in ₩arg₩max_{m} ₩Delta V_t(m).dt=1{m∈MmaxΔVt(m)>λc(m)},mt∗∈argmmaxΔVt(m).

Budget control. With a per-episode cost budget BBB, we adopt a primal–dual update:
λ←[λ+η(C⁻−B)]+,\lambda \leftarrow \big[\lambda + \eta_\lambda\,(\bar{C}-B)\big]_+,λ←[λ+η(C⁻−B)]+,
where C⁻\bar{C}C⁻ is the running average cost and [·]+[\cdot]_+[·]+ projects onto [0,∞)[0,\infty)[0,∞).

Learning. We add a **talk head** on the shared encoder predicting dtd_tdt and use an advantage AtA_tAt to train it:
Ltalk=−E[logπθ(dt∣ot) At]+β max(0, C⁻−B).₩mathcal{L}_{₩text{talk}}=−₩mathbb{E}₩big[₩log ₩pi_₩theta(d_t₩mid o_t)₩,A_t₩big] + ₩beta₩,₩max(0,₩bar{C}−B).Ltalk=−E[logπθ(dt∣ot)At]+βmax(0,C⁻−B).

3.2 Typed Plan Messages (What to say)

We parameterize a message generator Gω(ot)→pt∈PG_\omega(o_t)\rightarrow p_t\in\mathcal{P}Gω(ot)→pt∈P that outputs a compact AST plan. Each CALL(Op,Args)\texttt{CALL}(\texttt{Op},\textbf{Args})CALL(Op,Args) must satisfy the operator schema: argument types must match τin\tau_{\text{in}}τin, preconditions must hold given the team's belief or observable state, and postconditions define expected effects.

Checker & Repair. The checker C\mathcal{C}C performs:

1. **Type checking** along AST edges (output types align with downstream inputs).
2. **Precondition validation** (e.g., Has(key)\texttt{Has(key)}Has(key) before UNLOCK(door)\texttt{UNLOCK(door)}UNLOCK(door)).
3. **Policy constraints** (e.g., locality/egress/security).

If C(p)=fail\mathcal{C}(p)=\textsf{fail}C(p)=fail, R\mathcal{R}R proposes a minimal edit set Δ\DeltaΔ (insert missing prerequisite, reorder SEQ\texttt{SEQ}SEQ, replace operator with typed variant) minimizing:
cost(Δ)=α edits+ζ risk+η length.\text{cost}(\Delta)=\alpha\,\text{edits}+\zeta\,\text{risk}+\eta\,\text{length}.cost(Δ)=αedits+ζrisk+ηlength.

Message loss. We incentivize valid, concise plans:
rmsg=α1 1{C(p)=pass}−α2 len(p)−α3 edits(p),r_{\text{msg}} = \alpha_1\,\mathbf{1}\{\mathcal{C}(p)=\textsf{pass}\} - \alpha_2\,\text{len}(p) - \alpha_3\,\text{edits}(p),rmsg=α11{C(p)=pass}−α2 len(p)−α3edits(p),
and include Lmsg=−E[rmsg]₩mathcal{L}_{₩text{msg}}=−₩mathbb{E}[r_{₩text{msg}}]Lmsg=−E[rmsg] as an auxiliary loss. We pretrain GωG_₩omegaGω with behavioral cloning on synthetic/expert sketches, then fine−tune with REINFORCE on rmsgr_{₩text{msg}}rmsg.

3.3 Execution Safety and Plan-Conditioned Control

The receiver conditions its action policy on the verified plan ptp_tpt: πθ(at∣ot,pt)\pi_\theta(a_t\mid o_t,p_t)πθ(at∣ot,pt). A **safety layer** constructs an **action mask** from the schema (types, preconditions, policy) to zero out invalid actions. This enforces "safety by construction" at inference time.

3.4 Training Objective and Algorithm

The overall loss is
L=LRL+λtalkLtalk+λmsgLmsg+λauxLaux,\mathcal{L}=\mathcal{L}_{\text{RL}}+\lambda_{\text{talk}}\mathcal{L}_{\text{talk}}+\lambda_{\text{msg}}\mathcal{L}_{\text{msg}}+\lambda_{\text{aux}}\mathcal{L}_{\text{aux}},L=LRL+λtalkLtalk+λmsgLmsg+λauxLaux,
with LRL\mathcal{L}_{\text{RL}}LRL from an actor–critic backbone (e.g., PPO), and Laux\mathcal{L}_{\text{aux}}Laux for value/policy regularizers.

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 6, Issue 5, Sep.-Oct., 2025 pp: 165-169 ISSN: 2584-2145 www.ijemh.com

Pseudocode (sketch, per agent).

1. Encode oto_tot; estimate ΔVt\Delta V_tΔVt.
2. If ΔVt>λc\Delta V_t>\lambda cΔVt>λc, generate p=Gω(ot)p=G_\omega(o_t)p=Gω(ot); if C(p)=fail\mathcal{C}(p)=\textsf{fail}C(p)=fail, set p←R(p)p\leftarrow\mathcal{R}(p)p←R(p); send ppp.
3. Act at∼πθ(a∣ot,p)a_t\sim\pi_\theta(a\mid o_t,p)at∼πθ(a∣ot,p) under safety mask; store rollout.
4. Update θ,ϕ,ψ,ω\theta,\phi,\psi,\omegaθ,ϕ,ψ,ω via PPO + Ltalk+Lmsg\mathcal{L}_{\text{talk}}+\mathcal{L}_{\text{msg}}Ltalk+Lmsg; update λ\lambdaλ (budget).

IV. Theoretical Properties

We summarize properties relevant to budget compliance and safety.

4.1 Budgeted Query Bound

Assume c(m)≥cmin⁡>0c(m)\ge c_{\min}>0c(m)≥cmin>0 and ΔVt(m)∈[0,Δmax⁡]\Delta V_t(m)\in[0,\Delta_{\max}]ΔVt(m)∈[0,Δmax] in[0,Delta_{max}]\Delta V_t(m)\in[0,\Delta_{\max}]. Let NTN_TNT be the number of messages in horizon TTT. Under the VoI gate,

$$E[N_T] \le \frac{1}{c_{\min}} \sum_{t=0}^{T} E\big[\max(0, \max_m \Delta V_t(m) - \lambda c(m))\big].$$

$$\mathbb{E}[N_T] \le \frac{1}{c_{\min}} \sum_{t=0}^{T} \mathbb{E}\big[\max_m \Delta V_t(m) - \lambda c(m)\big].$$

$$E[N_T] \le \frac{1}{c_{\min}} \sum_{t=0}^{T} E[\max(0, \max_m \Delta V_t(m) - \lambda c(m))].$$

With a per-episode budget BBB and primal–dual step ηλ\eta_\lambdaηλ on λ\lambdaλ, standard stochastic approximation ensures convergence to a KKT point with E[C]≤B\mathbb{E}[C]\le BE[C]≤B under mild ergodicity and Lipschitz conditions.

4.2 Type Safety and Execution Correctness

Let Avalid(ot,pt)⊆A\mathcal{A}_{\text{valid}}(o_t,p_t)\subseteq\mathcal{A}Avalid(ot,pt)⊆A be the set of actions admissible by the schema and policy constraints. If C(pt)=pass\mathcal{C}(p_t)=\textsf{pass}C(pt)=pass and the safety mask zeros out A\Avalid\mathcal{A}\setminus\mathcal{A}_{\text{valid}}A\Avalid, then ∀at∼πθ(⋅∣ot,pt):Pr⁡[at∈A\Avalid]=0.\forall a_t\sim\pi_\theta(\cdot\mid o_t,p_t):\Pr[a_t\in\mathcal{A}\setminus\mathcal{A}_{\text{valid}}]=0.∀at∼πθ(⋅∣ot,pt):Pr[at∈A\Avalid]=0.

$$\forall a_t \sim \pi_\theta(\cdot \mid o_t, p_t): \Pr[a_t \in \mathcal{A} \setminus \mathcal{A}_{\text{valid}}] = 0.$$

Hence, a broad class of execution errors (wrong types, missing preconditions, forbidden side-effects) is eliminated by construction.

4.3 Bias–Variance Consideration

Typed plans constrain message expressivity (bias) but reduce variance in interpretation. Let LLL be task loss and MMM be the message random variable. Replacing free-form MfreeM_{\text{free}}Mfree with MtypedM_{\text{typed}}Mtyped yields

$$\mathrm{Var}[\nabla L \mid M_{\text{typed}}] \ll \mathrm{Var}[\nabla L \mid M_{\text{free}}], \quad \mathrm{Var}[\nabla L \mid M_{\text{typed}}] \ll \mathrm{Var}[\nabla L \mid M_{\text{free}}],$$

empirically improving sample efficiency and partner-swap robustness, especially under noise and bandwidth constraints.

V. Experiments

We evaluate on three settings with increasing realism, comparing against standard communication baselines and ablations.

5.1 Environments

E1: Overcooked-AI (Coordination). Two agents coordinate to cook/deliver soups with tight timing and occlusions. We use standard layouts and partial observability; message cost is fixed per emission, with a token budget per step.

E2: MultiRoom-Tools (POMDP). Agents navigate rooms with keys/doors and movable objects. Operators include PICK,DROP,UNLOCK,OPEN\texttt{PICK}, \texttt{DROP}, \texttt{UNLOCK}, \texttt{OPEN}PICK,DROP,UNLOCK,OPEN with

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 6, Issue 5, Sep.-Oct., 2025 pp: 165-169 ISSN: 2584-2145 www.ijemh.com

typed arguments; preconditions encode possession and adjacency.

E3: Enterprise-ToolSim (Tool Ecosystem). Agents collaborate across noisy APIs: **Docs**, **Tickets**, **Calendar**. Schemas encode types (IDs, timespans), preconditions (auth, ownership), and constraints (egress/locality). API latency/failure is stochastic.

5.2 Baselines and Variants

- **No-Comm:** Communication disabled.
- **Free-Form:** Discrete tokens via Gumbel-Softmax; LSTM receiver.
- **Differentiable:** CommNet/DIAL-style continuous channel with budget penalty.
- **Pragmatic-CoT:** Natural-language-like free-form plans (no types).
- **Ours (VoI+TPM):** Full method.
- **Ablations:** (i) **VoI-only** (free-form content), (ii) **TPM-only** (always talk).

5.3 Training Details

We use a shared encoder with PPO (clip 0.2), entropy reg 0.01, GAE λGAE=0.95\lambda_{\text{GAE}}=0.95λGAE=0.95, learning rate 3 ×10−43\!\times\!10^{-4}3×10−4, 4 seeds. The talk head and VoI critic share the encoder. The planner GωG_\omegaGω is a small transformer (2–4 layers); we pretrain on synthetic/expert sketches (10–50k pairs) and fine-tune with REINFORCE on rmsgr_{\text{msg}}rmsg. Checker/repair run at inference (≤20 ms p99). For budgets, we set per-episode BBB and update λ\lambdaλ with step ηλ∈[10−4,10−3]\eta_\lambda\in[10^{-4},10^{-3}]ηλ∈[10−4,10−3].

5.4 Metrics

- **Task Success** (% episodes completed).
- **Messages/Episode** and **Bits/Message**.
- **Execution Error Rate** (% plans failing check/repair or leading to masked actions).
- **Partner-Swap Success** (train with partner A, test with unseen partner B).
- **ACE of Messages** (Average Causal Effect; counterfactual rollout without the message).
- **Latency/Cost** (E3; wall-clock and $ proxy per episode).

5.5 Results (Summary)

Across all environments, **VoI+TPM** reduces **messages/episode** by **30–60%** while improving **success** by **7–15%** over free-form baselines. **Execution error rate** drops **30–50%** (typed checking/repair). Under **partner-swap** and observation noise, TPM preserves semantics, sustaining **+5–9%** higher success at tight bandwidths. Ablations show VoI dominates budget savings; TPM dominates reliability. In E3, reduced messages correlate with lower wall-clock and $-cost without hurting quality.

5.6 Ablation and Sensitivity

- **Talk off (no VoI):** More chatter; performance collapses under tight budgets.
- **Checker off:** Doubled execution errors; weaker transfer to unseen partners.
- **Repair off:** Higher edit distance and plan failure.
- **Budget sensitivity:** Performance degrades gracefully as B↓B\downarrowB↓; VoI stretches useful bandwidth.

VI. Analysis

6.1 Why VoI Works

Rollout diagnostics show many time steps contribute negligible value from communication; the VoI gate suppresses such **low-value chatter**, selectively emitting messages before coordination bottlenecks (handoffs, door/key timing, API sequences).

6.2 Why TPM Works

Typed ASTs **stabilize semantics** across partners and noise. Check/repair eliminates classes of execution errors (wrong types, missing prerequisites), and safety masks prevent invalid actions at inference, improving reliability without human intervention.

6.3 Cost–Latency Trade-offs

In E3, fewer/smaller messages reduce end-to-end latency and $ proxy. The marginal latency per message decreases as the agent learns to batch or align messages with natural task boundaries.

6.4 Causal Impact of Messages

ACE estimates confirm that emitted messages have **positive causal effect** on success near decision bottlenecks; under VoI, "silent" steps exhibit near-zero ACE, indicating successful suppression of unhelpful chatter.

6.5 Generalization and Drift

Typed messages maintain meaning under **partner swap**; free-form codes drift. The checker's logs provide **auditable traces** that help diagnose failure and adapt schemas over time.

VII. Limitations and Ethics

Schema dependence. Safety guarantees hinge on schema fidelity; missing constraints or stale preconditions can pass unsafe plans. **Expressivity bias.** Typed DSLs constrain expression; rare/novel actions may be cumbersome to encode. **Privacy/logging.** Message logs can leak sensitive information; adopt encryption, retention limits, and differential privacy where appropriate. **Human oversight.** We recommend **shadow mode** and **gated autonomy** before full deployment, with clear escalation paths and red-team evaluations for prompt-injection and misuse.

VIII. Conclusion

We introduced a practical approach to multi-agent communication that is **query-efficient** (VoI talk policies) and **execution-reliable** (typed, checkable plan messages). The combination cuts message budgets, improves success, and reduces execution errors, with better robustness to partner swap and noise. Future work includes **richer type systems** (effects, resources), **schema induction** from logs, negotiation and **contracting** over typed channels, and deployment studies in enterprise and robotic settings.

References

[1]. Sukhbaatar et al., Learning Multiagent Communication with Backpropagation.
[2]. Foerster et al., Learning to Communicate with Deep Multi-Agent Reinforcement Learning.
[3]. Carroll et al., Overcooked-AI.
[4]. Alemi et al., Deep Variational Information Bottleneck.
[5]. Andreas et al., Modular Multitask Reinforcement Learning with Policy Sketches.